

المهارات الرقمية

الصف الحادي عشر - دليل المعلم
الفصل الدراسي الأول

11

لجنة الإشراف على التأليف

أ.د. باسل علي محافظة

أ.د. وليد خالد سلامة

ليلى محمد العطوي

أ.د. خالد إبراهيم العجلوني

هذا الكتاب جزء من مشروع الشباب والتكنولوجيا والوظائف
لدى وزارة الإقتصاد الرقمي والريادة.

الناشر: المركز الوطني لتطوير المناهج

يسر المركز الوطني لتطوير المناهج استقبال آرائكم وملحوظاتكم على هذا الكتاب عن طريق العناوين الآتية:

☎ 06-5376262 / 237 📠 06-5376266 ✉ P.O.Box: 2088 Amman 11941

📱 @nccdjr 📧 feedback@nccd.gov.jo 🌐 www.nccd.gov.jo

قررت وزارة التربية والتعليم استخدام هذا الدليل في مدارس المملكة الأردنية الهاشمية جميعها، بناء على قرار المجلس الأعلى للمركز الوطني لتطوير المناهج في جلسته رقم (2024/0)، تاريخ 2024/0/00 م، وقرار مجلس التربية والتعليم رقم (2024/00) تاريخ 2024/0/00 م بدءاً من العام الدراسي 2024 م.

ISBN: 978 - 9923 - 41 - 660 - 0

المملكة الأردنية الهاشمية
رقم الإيداع لدى دائرة المكتبة الوطنية
(2024/07/3870)

الأردن. المركز الوطني لتطوير المناهج
المهارات الرقمية: الصف الحادي عشر: دليل المعلم (الفصل الدراسي الأول)
المركز الوطني لتطوير المناهج. - عمان: المركز، 2024
رقم التصنيف: 373.19
ر.إ.: 0000/00/0000
الواصفات: / المهارات الرقمية / علم الحاسوب / المناهج / التعليم الثانوي
يتحمل المؤلف كامل المسؤولية القانونية عن محتوى مصنفه ولا يعبر هذا المصنف عن رأي دائرة المكتبة الوطنية.

فريق التأليف المكلف من شركة عالم الاستثمار للتنمية والتكنولوجيا

د. اسماء حسن حمدان د. ابراهيم سلامة البلوي د. محمد جمال عبدالرحمن
أنوار يعقوب حامدة



1445 هـ / 2024 م

الطبعة الأولى

المُقدِّمة

انسجامًا مع الرؤية الملكية السامية، يستمر المركز الوطني في أداء رسالته؛ بتطوير المناهج الدراسية لتحقيق التعليم النوعي المتميز، ورفد الطلبة بالعلم والمعرفة. ويُقدِّم المركز الوطني هذه النسخة من دليل المعلم للمعلمين والمعلمات؛ لتكون عونًا لهم، ودليلاً لتعليم الطلبة، وتحقيق الأهداف المرجوة في تدريس المهارات الرقمية.

يوفر دليل المعلم نظرة شاملة على كل وحدة في كتاب الطالب والدروس المشتملة عليها، مقدِّمًا كل درس وفق نموذج تعليمي متكامل، يشمل التهيئة، والتعليم والتعلم، والإثراء والتقييم.

يحتوي دليل المعلم على توضيح لاستراتيجيات التدريس وطرائق التعليم والتعلم المعتمدة في كتاب الطالب، إلى جانب الاستراتيجيات والأدوات التقييمية المتوافقة مع هذه الطرائق، سواء أكانت كمية أو نوعية، ويُلخص المهارات الحياتية التي يمكن تطويرها وتعزيزها عن طريق الأنشطة والمهام التي سيشترك فيها الطلبة في أثناء تعلُّمهم للمناهج.

يقدم دليل المعلم نظرة عامة على ترقية كتاب الطالب والعناصر الأساسية في الوحدات والدروس، مُبينًا هذه العناصر وما تحتويه من مكونات، ودورها في دعم عملية التعليم والتعلم.

يُفصّل دليل المعلم محتوى الدروس في كل وحدة من كتاب الطالب، ويُناقشها بعمق؛ لتحسين التعليم والتعلم، وتسهيل استيعاب الطلبة للمفاهيم والمعلومات والأفكار المُقدَّمة في كل درس.

تبدأ كل وحدة بجدول نتائج التعلم الذي يُظهر النتائج المتوقعة للوحدة، والنتائج المرتبطة بها مُسبقًا ولاحقًا؛ ليساعد المعلمين على الربط العمودي للمفاهيم، وإدارة التعليم والتعلم بكفاءة.

يُقدِّم دليل المعلم أيضًا سياقات تعلم ومهام تعليمية متميزة، تتمثل في مشروعات يُنفّذها الطلبة بإشراف المعلمين؛ لتقوية مهارات التفكير النقدي، والتعاون، وحل المشكلات.

ويُعرض دليل المعلم كذلك استراتيجيات تعليم وتعلم ملائمة للسياق، وخططًا دراسية مفصلة لكل قسم من الوحدة؛ لمساعدة المعلمين على تقديم تجارب تعليمية غنية ومحفزة.

وتمكّن الهيكل المنظم لدليل المعلمين من إعداد دروس تفاعلية، تلبي احتياجات الطلبة وتطلعاتهم التعليمية.

ونحن إذ نُقدِّم هذا الكتاب، فإننا نأمل أن يسهم في مساعدة المعلمين/ والمعلمات، وأن يكون دليلاً إضافيًا لمهاراتهم التدريسية داخل الغرفة الصفية، ولتكون مادة المهارات الرقمية مادة ممتعة ومفضلة لطلبتنا.

المركز الوطني لتطوير المناهج

قائمة المحتويات

i	المقدمة.....
A	نظرة عامة إلى دليل المُعلِّم
C	نموذج التعليم والتعلُّم:.....
E	نظرة عامة إلى كتاب الطالب
F	بنية وحدات التعلُّم:
R	الاستراتيجيات والطرائق الداعمة لعملية التعليم والتعلُّم
V	استراتيجيات التقويم وأدواته:
X	أدوات التقييم في دليل المُعلِّم :
Y	أدوات التقويم في كتاب الطالب

AA	الخوارزميات والبرمجة (Algorithms and Programing)
10	الدرس الأول: مُقدمة في لغات البرمجة (Introduction to Programming Languages).....
20	الدرس الثاني: أساسيات لغة البرمجة بايثون (Basics of Python Programming)
48	الدرس الثالث: الجمل الشرطية (Conditional Statements)
62	الدرس الرابع: الحلقات (Loops).....
78	الدرس الخامس: القوائم (Lists).....
110	الدرس السادس: الدوال البرمجية (Functions)
130J	أدوات التقويم الخاصة بالوحدة.
130L	ملاحق خاصة بالوحدة.
130AK	قائمة المراجع

AM	الحوسبة والحياة (Computing and Life)
144	الدرس الأول: الحوسبة الخضراء (Green Computing)
158	الدرس الثاني: النفايات الإلكترونية (Electronic Waste)
172	الدرس الثالث: تطبيقات الحاسوب في الحياة اليومية (Computer Application in our Daily Life)
189J	ملاحق خاصة بالوحدة.....
189T	قائمة المراجع



نظرة عامة إلى دليل المُعلِّم

يُقدِّم دليل المُعلِّم لمحةً عن كل وَحدة في كتاب الطالب والدروس التي فيها، ويعرض كل درس وفق نموذج للتعليم والتعلُّم من ثلاث مراحل، هي: التهيئة، والتعليم والتعلُّم، والإثراء والتأمُّل والتقويم.

يتضمَّن دليل المُعلِّم بيانًا لاستراتيجيات التدريس وطرائق التعليم والتعلُّم المُستخدمة في كتاب الطالب، والاستراتيجيات والأدوات الخاصة بالتقويم التي تُؤمِّن تلك الطرائق والاستراتيجيات بنوعها الكَمِّي والنوعي، ونبذة عن المهارات الحياتية التي يُمكن صقلها وتطويرها عن طريق الأنشطة والمهام التي سيشارك فيها الطلبة على مدار تعلُّمهم منهاج المهارات الرقمية؛ مما يساعد المُعلِّم / المُعلِّمة على تكوين فهم شامل لمحتوى المنهاج وأهدافه الكبرى.



يُقدِّم دليل المُعلِّم نظرة عامة عن بنية كتاب الطالب ومُكوّناته الرئيسة على مستوى الوحدة ومستوى الدروس فيها، مُبينًا تلك المُكوّنات، وما تحويه من عناصر، ودور كل منها في إثراء عملية التعليم والتعلُّم.

يعرض دليل المُعلِّم محتوى الدروس في كل وحدة من كتاب الطالب، ويتناولها بالشرح والتحليل والمناقشة؛ مما يُعزِّز عملية التعليم والتعلُّم، ويُسهِّل على الطلبة تعلُّم المفاهيم والمعارف والأفكار الواردة في كل درس.

تبدأ كل وحدة بمصفوفة نتائج تتضمن نتائج التعلُّم الحالية المُتوقَّعة للوحدة، ونتائج التعلُّم السابقة، ونتائج التعلُّم اللاحقة المُرتبطة بها؛ مما يساعد المُعلِّم/ المُعلِّمة على الترابط الرأسي للمفاهيم والأفكار، وإدارة عملية التعليم والتعلُّم في الوحدة على نحوٍ فعال.



يُقدِّم دليل المُعلِّم سياقات تعلُّم ومهام أصيلة تتبلور في صورة مُنتجات تعلُّم، يعمل الطلبة على أدائها وتنفيذها بتوجيه وتيسير وإرشاد من المُعلِّم/ المُعلِّمة؛ مما يُعزِّز مهارات التفكير الناقد والتعاون وحلُّ المشكلات.

يعرض دليل المُعلِّم استراتيجيات التعليم والتعلُّم التي تُناسب السياق المطروح، وخُطط تدريس مُفصَّلة لكل جزء من الوحدة؛ مما يساعد المُعلِّم/ المُعلِّمة على تقديم تجارب تعليمية ثريَّة ومُلهمة للطلبة. كذلك تُمكن بنية الدليل المُنظَّمة المُعلِّم والمُعلِّمة من إعداد دروس تفاعلية ومُحفِّزة، تفي بحاجات الطلبة وتطلُّعاتهم في مختلف المراحل التعليمية.



نموذج التعليم والتعلم:

1- مرحلة التهيئة:

تشمل هذه المرحلة تهيئة الطلبة للدرس نفسياً وذهنياً. ويُقدّم دليل المُعلِّم مُقترحات يُمكن بها إثارة تفكير الطلبة، وتحفيزهم على التعلُّم، وتذكُّر ما تعلَّموه مسبقاً عن موضوع الدرس، ويشمل ذلك إشراك الطلبة في مواقف تعلُّم واقعية أو مهام حقيقية أصيلة؛ مما يساعد المُعلِّم / المُعلِّمة على لفت انتباه الطلبة، وإثارة دافعيتهم للتعلُّم، وتفعيل دورهم في الصف؛ ليكونوا المحور الرئيس لعملية التعلُّم، ويزيد استعدادهم للمشاركة في هذه العملية.

كذلك يُقدِّم الدليل بياناً لكيفية عرض فكرة الدرس الرئيسة، وآلية مناقشتها، وطريقة ربطها بمعرفة الطلبة السابقة؛ تحقيقاً للتكامل المعرفي في عملية التعلُّم. يضاف إلى ذلك تقديم الدليل مُقترحات تختصُّ بكيفية مناقشة النشاط التمهيدي، وما يلزمه من تغذية راجعة، فضلاً عن استخدام هذا النشاط مُقدِّمة لبدء تنفيذ المرحلة الثانية من الدرس.

2- مرحلة التعليم والتعلم:

تتضمَّن هذه المرحلة عرض المحتوى العلمي بطرائق مُتنوّعة تتيح للطلبة فهم المادة واستيعابها، ويشمل ذلك شرح الدروس، والعروض العملية، والنقاشات، واستخدام الوسائل التعليمية المختلفة، فضلاً عن تحفيز الطلبة على المشاركة والتفاعل في ما بينهم عن طريق طرح الأسئلة، والعمل الجماعي، والأنشطة التفاعلية، والأنشطة العملية، والتدريبات، والواجبات.

يُقدِّم دليل المُعلِّم مُقترحات وإرشادات تختصُّ بمناقشة موضوع الدرس، وبناء المفاهيم التي تنوِّع تبعاً لاختلاف طبيعة الدرس ومحتواه العلمي؛ فقد أورد الدليل مُقترحات لأساليب إدارة الأنشطة المُدرّجة في كتاب الطالب على اختلاف أنواعها؛ مما يجعل الطلبة مشاركين حقيقيين وفاعلين في تعلُّمهم، ويساعدهم على تحقيق أهدافهم.

أورد الدليل أيضاً إجابات نموذجية و/ أو إجابات مُحتملة لما يردُّ في تلك الأنشطة من أسئلة، إضافةً إلى ما يلزم من أدوات تقويم تبعاً لنوع النشاط ومُتطلّباته.



ويُقدِّم أيضًا مُقترحات لكيفية استخدام الأشكال والصور الواردة في كتاب الطالب في عملية التعليم والتعلُّم، وكيف يُمكن توظيفها في تحفيز قدرات الطلبة على التأمل، والتحليل، والاستنتاج، والمناقشة.



يعرض الدليل أيضًا إضاءات تتعلّق ببعض القضايا المُعيّنة، ويورد تفاصيل عن موضوع ما في دروس الوحدة.



يلفت الدليل الانتباه إلى الأخطاء الشائعة التي يقع فيها الطلبة؛ فقد يكون البناء المعرفي لدى بعضهم غير صحيح؛ مما يُحتمّ التنبيه على هذه الأخطاء، وتقديم إرشادات لكيفية معالجتها.

3- مرحلة الإثراء والتأمل والتقويم:

يُعنى دليل المُعلِّم بإثراء فهم الطلبة لبعض دروس الوحدة، وذلك بالربط بين ما تعلّموه والموضوعات الأخرى أو الحياة العملية، وتشجيع التفكير الناقد، والتفكير الإبداعي، والبحث، والتجربة. يشمل التقويم في هذا الدليل ما يأتي:

1. إجابات نموذجية لأسئلة الدرس الواردة في كتاب الطالب (أقيم تعلّمي).
2. إجابات نموذجية لأسئلة الوحدة الواردة في كتاب الطالب.
3. إرشادات عن كيفية توجيه الطلبة في أثناء اشتغالهم بالمنتجات (مُنتجات التعلُّم) على مستوى الدروس، والأدوات المُقترحة للتقويم وتقييم الأداء.
4. إرشادات عن كيفية توجيه الطلبة في أثناء اشتغالهم بمشروعات الوحدة، والأدوات المُقترحة للتقويم وتقييم الأداء.



نظرة عامة إلى كتاب الطالب

يتألف كتاب المهارات الرقمية للصف الحادي عشر من وحدات تعليمية، أُعدت على نحوٍ يزوّد الطلبة بالمعارف والمهارات اللازمة لفهم جوانب التكنولوجيا الحديثة واستخدام وسائلها. وقد نُظّم الكتاب بطريقة تتيح للطلبة الانتقال السلس من المفاهيم الأساسية إلى أكثر الموضوعات تعقيداً.

تشمل بنية كتاب المهارات الرقمية ما يأتي:

المُقَدِّمة:

يبدأ الكتاب بمُقَدِّمة عامة تُبيّن أهميته ودوره في تنمية مهارات الطلبة الرقمية.

المقدمة

انطلاقاً من إيمان المملكة الأردنية الهاشمية بأهمية تنمية قدرات الإنسان الأردني، وتسليحه بالعلم والمعرفة؛ سعى المركز الوطني لتطوير المناهج، بالتعاون مع وزارة التربية والتعليم، إلى تحديث المناهج الدراسية وتطويرها، لتكون مُعِيناً للطلبة على الارتقاء بمستواهم المعرفي والمهاري، ومجارة أقرانهم في الدول المُتقدّمة.

ونظراً إلى أهمية مبحث المهارات الرقمية ودوره في تنمية مهارات التفكير لدى الطلبة، وفتح آفاق جديدة لهم تُواكب مُتطلّبات سوق العمل؛ فقد أوّلى المركز مناهجه عناية فائقة، وأعدّها وفق أفضل الأساليب والطرائق المُتّبعة عالمياً وأشرف عليها خبراء أردنيين؛ لضمان توافقها مع القِيَم الوطنية الأصيلة، ووفائها بحاجات الطلبة.

المقدمة

كرمية الهاشمية بأهمية تنمية قدرات الإنسان الأردني، وتسليحه بالعلم والمعرفة، على الارتقاء بمستواهم المعرفي والمهاري، ومجارة أقرانهم في الدول المُتقدّمة.

ونظراً إلى أهمية مبحث المهارات الرقمية ودوره في تنمية مهارات التفكير لدى الطلبة، وفتح آفاق جديدة لهم تُواكب مُتطلّبات سوق العمل؛ فقد أوّلى المركز مناهجه عناية فائقة، وأعدّها وفق أفضل الأساليب والطرائق المُتّبعة عالمياً وأشرف عليها خبراء أردنيين؛ لضمان توافقها مع القِيَم الوطنية الأصيلة، ووفائها بحاجات الطلبة.

ونظراً إلى ما تشكّله الألسنة من حصة كبرى في فهم الموضوعات وتعزيز الفعالة الإيجابية لدى الطلبة، فقد اشتمل الكتاب على أنشطة تشريعية تشاركية، وأجّلتها ودمجتها مع موضوعات تعليمية، وتكرّرت عبر جميع الصفحات من الوحدة الثانية وتكرّرت بشكلٍ متكرر في الأجزاء المتبقية من هذا الكتاب المعرفية، ومن شأن هذه المهارات

وتسعى إلى تلمّذ هذا الكتاب، وتلّزّز بالأنشطة التي تُسهم في بناء جيل واعٍ، ومبتكر، وقادر على التعامل مع التكنولوجيا بسهولة، وفاعلة، وأن يكون لديه أسس في فهم المملكة الأردنية الهاشمية وازدهارها.

المركز الوطني لتطوير المناهج

الفهرس:

يُقدّم الفهرس نظرة شاملة عن محتوى الكتاب، وفيه تتوزّع كل وحدة تعليمية إلى دروس رئيسة وعناوين فرعية، تليها أسئلة الوحدة والمراجع؛ مما يساعد المُعلِّمين/ المُعلِّمات والطلبة على متابعة تقدّمهم في دراسة المهارات الرقمية بصورة مُنظّمة وفاعلة.

الفهرس

المحتويات والفهرس (Algorithms and Programming)

10	مقدمة في لغات البرمجة (Introduction to Programming Language)
11	1.1 لغات البرمجة عالية المستوى (High-Level Language)
12	1.2 لغات البرمجة منخفضة المستوى (Low-Level Language)
20	أساسيات لغة البرمجة بايثون (Basic of Python Programming)
21	2.1 بيئات العمل (IDEs)
22	2.2 Text Editors and Integrated Development Environments
23	2.3 بيئات العمل (IDEs)
24	2.4 بيئات العمل (IDEs)
25	2.5 بيئات العمل (IDEs)
26	2.6 بيئات العمل (IDEs)
27	2.7 بيئات العمل (IDEs)
28	2.8 بيئات العمل (IDEs)
29	2.9 بيئات العمل (IDEs)
30	2.10 بيئات العمل (IDEs)
31	2.11 بيئات العمل (IDEs)
32	2.12 بيئات العمل (IDEs)
33	2.13 بيئات العمل (IDEs)
34	2.14 بيئات العمل (IDEs)
35	2.15 بيئات العمل (IDEs)
36	2.16 بيئات العمل (IDEs)
37	2.17 بيئات العمل (IDEs)
38	2.18 بيئات العمل (IDEs)
39	2.19 بيئات العمل (IDEs)
40	2.20 بيئات العمل (IDEs)
41	2.21 بيئات العمل (IDEs)
42	2.22 بيئات العمل (IDEs)
43	2.23 بيئات العمل (IDEs)
44	2.24 بيئات العمل (IDEs)
45	2.25 بيئات العمل (IDEs)
46	2.26 بيئات العمل (IDEs)
47	2.27 بيئات العمل (IDEs)
48	2.28 بيئات العمل (IDEs)
49	2.29 بيئات العمل (IDEs)
50	2.30 بيئات العمل (IDEs)
51	2.31 بيئات العمل (IDEs)
52	2.32 بيئات العمل (IDEs)
53	2.33 بيئات العمل (IDEs)
54	2.34 بيئات العمل (IDEs)
55	2.35 بيئات العمل (IDEs)
56	2.36 بيئات العمل (IDEs)
57	2.37 بيئات العمل (IDEs)
58	2.38 بيئات العمل (IDEs)
59	2.39 بيئات العمل (IDEs)
60	2.40 بيئات العمل (IDEs)
61	2.41 بيئات العمل (IDEs)
62	2.42 بيئات العمل (IDEs)
63	2.43 بيئات العمل (IDEs)
64	2.44 بيئات العمل (IDEs)
65	2.45 بيئات العمل (IDEs)
66	2.46 بيئات العمل (IDEs)
67	2.47 بيئات العمل (IDEs)
68	2.48 بيئات العمل (IDEs)
69	2.49 بيئات العمل (IDEs)
70	2.50 بيئات العمل (IDEs)

دلالات أيقونات الكتاب:

تحتوي الصفحات الأولى من كتاب الطالب على مفتاح توضيحي للأيقونات المُستخدمة في الكتاب، ودلالة كلّ منها؛ ما يتيح للطلبة التمييز بين مختلف مُكوّنات الدروس، مثل: النشاط الفردي، والنشاط الجماعي، والنشاط التمهيدي، والنشاط العملي، وغير ذلك.

دلالات أيقونات الكتاب

نشاط تمهيدي	نشاط استطلاحي يربط التعلم السابق بالتعلم الحالي	توسيع في المعلومات مرتبطة بمحتوى الدرس	إثراء
نشاط عملي	نشاط تطبيقي يربط مهارات الدرس	عرض الأفكار وتبادلها مع الزملاء والمعلم	إثراء
نشاط جماعي	نشاط مرتبط بمحتوى الدرس المعرفي أو المهاري	معلومة إضافية	إثراء
نشاط فردي	نشاط يُلَبّز بشكل فردي	عرض محتوى فيديو مرتبط بالمحتوى	إثراء
نشاط جماعي	نشاط يُلَبّز في مجموعات	نشاط تكميلي توظف فيه معارف ومهارات الوحدة	مفهوم
ابحث	استخدام شبكة الإنترنت للبحث عن المعلومات	الإجراءات الواجب اتباعها لتحقيق مبادئ الوحدة الرقمية	مواظبة (الواجب)
		المهارات التكنولوجية التي سألها في الوحدة	المهارات (الواجب)

بنية وحدات التعلُّم:

يتألّف كتاب الطالب من وحدات تعليمية عديدة، تشمل كلُّ منها ما يأتي:

نظرة عامة على الوحدة:

مُقدِّمة تتناول موضوع الوحدة، وأهميته، ونبذة مختصرة عمّا سيتعلّمه الطلبة في الوحدة.



نتائج التعلُّم:

توقّع ما سيُحقّقه الطلبة في نهاية الوحدة، بما في ذلك تحديد المهارات والمعارف التي سيكتسبها الطلبة في أثناء عملية التعليم والتعلُّم.

تستند نتائج التعلُّم إلى إطار المنهاج الخاص بمبحث الحاسوب والمهارات الرّقمية، ومعايير، ومؤشّرات أدائه. وهي تهدف إلى تطوير قدرات الطلبة على استخدام التكنولوجيا بفاعلية، وتعزيز فهمهم للمفاهيم الرّقمية الأساسية، وتطبيقها في سياقات عملية مختلفة.



مُنتَجَات التعلُّم (Learning Products):

منتجات يُتوقَّع من الطلبة إنجازها في أثناء رحلتهم التعليمية على مدار الوحدة؛ إذ يشارك الطلبة في سياق تعلُّم يهدف إلى بلوغ المُنتَج المُحدَّد الذي يُعدُّ تنويجًا للأنشطة التعليمية والتطبيقية التي يشارك فيها الطلبة في أثناء تعلُّم الوحدة الدراسية.

تشمل مُنتَجَات التعلُّم مجموعة مُتنوّعة من الأعمال والأنشطة، (مثل: التقارير المكتوبة، والعروض التقديمية باستخدام أدوات تكنولوجيا مُتنوّعة)، والمشروعات التطبيقية الفردية والجماعية (مثل: تصميم تطبيق بسيط، وإنشاء موقع إلكتروني، وبرمجة لعبة ما)، والوسائط المُتعدّدة (مثل: مقاطع الفيديو التعليمية، والتسجيلات الصوتية، والرسوم المُتحرّكة)، والتصاميم والرسوم البيانية (مثل: التصميم الجرافيكي، والخرائط الذهنية، والرسوم البيانية التوضيحية التي تشرح موضوعًا مُحدَّدًا، وتُبيِّن نتائج دراسة ما)، والنمذجة والمحاكاة، والمُدوّنات، والمواقع الإلكترونية، وغير ذلك ممّا يتناسب مع سياق التعلُّم وطبيعة المحتوى العلمي في الوحدة.

تهدف مُنتَجَات التعلُّم إلى تعزيز قدرة الطلبة على التعبير عن أفكارهم بطرائق إبداعية، وتطبيق ما تعلَّموه في سياقات عملية، وتطوير مهارات البحث والتحليل والتواصل لديهم، وهي تساعد المُعلِّمين/ المُعلِّمات على تقييم مدى استيعاب الطلبة للمواد التعليمية، وتحديد المجالات التي تحتاج إلى تحسين أو دعم إضافي.

تتضمَّن كل وحدة مُنتَجًا رئيسًا واحدًا يَرِدُ في بداية الوحدة، ويُقسَّم إلى مُنتَجَات صغيرة على شكل مهام مُوزَّعة في كل درس. وهذه المهام يُبنى بعضها على بعض؛ لكي يتمكن الطلبة من إنجاز المُنتَج الرئيس عند انتهاء رحلة تعلُّمهم في الوحدة.

مُنتَجَات التعلُّم (Learning Products):

تصميم برنامج وإعداده لإنشاء لعبة تخمين الأرقام باستخدام لغة بايثون (Python).



مُنتَجَات التعلُّم (Learning Products):

تصميم مشروع ريادي رقمي، يقوم على استخدام أحد تطبيقات الحاسوب، ويتناول القضايا البيئية والاجتماعية المُتعلِّقة بالحوسبة وأثرها في الفرد



مشروعات الوحدة:

مشروعات مُقترحة يختار الطلبة أحدها لتنفيذه بعد إنهاء رحلة تعلُّمهم لموضوعات الوحدة. تتيح هذه المشروعات للطلبة تطبيق المعارف والمهارات التي اكتسبوها في أثناء تعلُّم الدروس بطرائق عملية وإبداعية؛ مما يُعزِّز لديهم الفهم العميق للمواد التعليمية. كذلك تساعد مشروعات الوحدة الطلبة على تطبيق المفاهيم في سياقات عملية حقيقية، وتطوير مهاراتهم العملية، مثل: التصميم، والبرمجة، والتواصل الرقمي، وهي تُحفِّزهم على التفكير الإبداعي والتفكير الابتكاري في حلّ المشكلات وإنجاز المشروعات، وتُعزِّز لديهم مهارات البحث عن المعلومات وتحليلها واستخدامها بفاعلية، فضلاً عن توفيرها فرصاً للعمل الجماعي والتعاون بين الطلبة.





مشروع

أختار مع أفراد مجموعتي أحد المشروعات الآتية لتنفيذه في نهاية الوحدة:

المشروع الأول: تصميم برنامج وإعداده لإنشاء لعبة الألغاز باستخدام لغة بايثون (Python).

المشروع الثاني: تصميم نموذج أولي وإعداده لإنشاء برنامج، بناءً على أفكار تخدم المجتمع والبيئة المحيطة.

المشروع الثالث: تصميم نموذج أولي وإعداده لإنشاء برنامج يساعد الطالب على إدارة المهام المنوطة به.



مشروع

أختار مع أفراد مجموعتي أحد المشروعين الآتيين لتنفيذه في نهاية الوحدة:

المشروع الأول: تنظيم حملة تثقيفية عن الحوسبة الخضراء؛ لتعزيز وعي المجتمع المدرسي بأهمية الحوسبة الخضراء.

المشروع الثاني: تنظيم مسابقة تحمل عنوان (البرمجة الخضراء)، وتشترط كتابة تعليمات برمجية - باستخدام لغة البرمجة بايثون (Python) - تهدف إلى الحد من استهلاك الطاقة، والتركيز على تحسين كفاءة المقطع البرمجي، والتقليل من استخدام الموارد الحاسوبية.



الأدوات والبرامج:

مجموعة مُتنوّعة من المصادر الإلكترونية والبرمجيات والأدوات التكنولوجية التي تدعم الأنشطة التعليمية والتطبيقية التي تلزم الطلبة والمُعَلِّمين/ المُعَلِّمات في أثناء عملية التعلّم في الوحدة، وتساعدهم على تحقيق نتائج التعلّم، وتعزيز تجربة التعلّم الرّقمي لديهم.

الأدوات والبرامج (Digital Tools and Programs):

Python, IDLE, Command Line, (draw.io), google slides



الأدوات والبرامج: (Digital Tools and Programs):

Canva, Paython, Google Forms, MS Words, Gantt Charts



المهارات الرّقمية:

مجموعة من المهارات الرّقمية الأساسية التي تلزم الطلبة في العصر الحديث، وتساعدهم على تحقيق أهدافهم التعلّمية، وتُعِدُّهم لمواجهة تحديات العالم الرقمي المتنامي. وهي تستند إلى معايير الجمعية الدولية للتكنولوجيا في مجال التعليم للطلبة (ISTE).

المهارات الرّقمية (Digital Skills): تصميم الخوارزميات، التفكير الحاسوبي، البرمجة، حلّ المشكلات البرمجية.

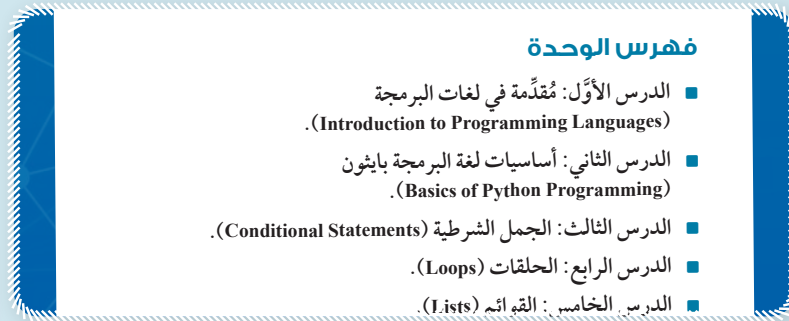


التفكير الحاسوبي، التعاون الرقمي، الابتكار العالمي، التصميم الرقمي، التعليم المستمر، التواصل الرقمي.



فهرس الوحدة:

قائمة تحوي دروس الوحدة التعليمية وعناوينها باللغة العربية واللغة الإنجليزية.



بنية الدرس:

يتكوّن كل درس في كتاب الطالب من العناصر الآتية:

الفكرة الرئيسية:

توضح الهدف الأساسي من الدرس والمفاهيم التي سيتناولها.

المفاهيم والمصطلحات:

مفاهيم ومصطلحات أساسية تتعلق بموضوع الدرس، وتُكتب باللغة العربية واللغة الإنجليزية.

نتائج التعلّم:

تحديد ما يُتوقع من الطلبة تحقيقه من معارف ومهارات في نهاية الدرس.

مُنتجات التعلّم:

مهمة مُرتبطة بالمُخرجات والنتائج الملموسة التي يُتوقع من الطلبة تنفيذها في أثناء عملية التعليم والتعلّم المُتعلّقة بالدرس.

1

الفكرة الرئيسية:

تعرف كيف يُمكن التعامل مع القوائم وسلاسل الحروف في برمجة بايثون (Python)، واستخدامها في تخزين مجموعات البيانات المُترابطة وإدارتها وتنفيذ عمليات فيها.

2

المفاهيم والمصطلحات:

القائمة (List)، سلسلة الحروف (String)، موقع العنصر في القائمة (أو الحرف في السلسلة) (Index)، إلصاق القوائم أو السلاسل (Concatenation)، القابلية للتغيير (Mutability)، القائمة المُركّبة (Nested List)، المصفوفة ثنائية الأبعاد

4

مُنتجات التعلّم (Learning Products)

تعديل البرنامج باستخدام قوائم اللعبة في توليد الأرقام العشوائية، ضمن سياق تصميم لعبة تخمين الأرقام باستخدام برمجة بايثون (Python).

3

نتائج التعلّم (Learning Objectives):

- أُعرّف مفهوم المُتغيّر (قائمة)، وأُبين استخداماته في البرمجة.
- أنشئ قوائم على اختلاف أنواعها (مُتسلسلة، ومُتغيّرة، ومُركّبة)، وأستخدمها في تخزين مجموعة مُتنوعة من القيم.
- أحدّد كيف يُمكن تجميع مجموعة القيم في قائمة واحدة.
- أوضح أنواع القوائم (المُتسلسلة، والمُتغيّرة، والمُركّبة)، وأنفذ عمليات مختلفة فيها، مثل: الإضافة، والحذف.



نشاط تمهيدي:

نشاط تفاعلي يربط بين معرفة الطلبة السابقة والتمهيد لتعلم موضوع جديد.

المحتوى التعليمي:

شرح مُفصّل لمفاهيم الدرس وموضوعاته الرئيسة، مُدعّمًا بالأمثلة، والصور، والأشكال التوضيحية، والتمثيلات.

5



نشاط
تمهيدي

أفترض أنني أريد إنشاء برنامج يتولّى قراءة علامات (100) طالب في أحد الامتحانات، ثمّ يعرض هذه العلامات ترتيباً بصوريّة تصاعديّة. أناقش أفراد مجموعتي في الأسئلة الآتية بناءً على ما تعلّمته عن لغة البرمجة بايثون (Python):

1. هل يجب عليّ تخزين علامات هؤلاء الطلبة أم يمكنني كتابة هذا البرنامج دون الحاجة

6

القوائم (Lists)

تُمثّل القائمة في لغة البرمجة بايثون (Python) عدداً من القيم التي يُخزّن بعضها مع بعض، وترتبط معاً بمعنى وظيفي مشترك. فمثلاً، يُمكن للمُستخدم تخزين أسماء الشوارع داخل إحدى المدن في قائمة، وكذلك تخزين أسعار البضائع التي اشتراها أحد العملاء، أو تخزين علامات الطلبة في الصف الحادي عشر. يتطلب استخدام القوائم في لغة البرمجة بايثون (Python) تعريف القائمة أولاً، ثمّ تخزين العناصر



الأنشطة:

تنوّع الأنشطة الواردة في كتاب الطالب، وهي تهدف إلى تفعيل دور الطلبة في عملية تعلّمهم؛ مما يُعزّز لديهم نهج التعلّم النشط، والتعلّم التفاعلي، والتعلّم التعاوني، إضافةً إلى تطوير مهاراتهم في التفكير الناقد والبحث والتحليل، وتحفيزهم على تطبيق المفاهيم في سياقات عملية.

تتضمّن الأنشطة مجموعة مُتنوّعة من الأشكال والأنواع، يستفاد منها في تمثيل المواقف التعليمية وسياق التعلّم. وهي تُصنّف إلى الأنواع الآتية:

1. الأنشطة الجماعية والأنشطة الفردية: أنشطة تُعزّز العمل الجماعي والمشاركة الفردية.

أُجرب وأستنتج:



نشاط
جماعي

في المثال السابق، إذا وُضعت جملة print() بعد جملة (if)، فماذا سيحدث؟
أتوقع ناتج البرنامج، ثمّ أنفذه باستخدام برمجة بايثون (Python)، وألاحظ الناتج.



نشاط
فردى

- أكتب المقطع البرمجي (الكود) السابق بطريقة أخرى، ثمّ أتحقّق من النتيجة عن طريق تنفيذ البرنامج.
- أعدّل قيمة المُتغيّر (Avg) لتصبح (70)، ثمّ أنفذه البرنامج. ما الناتج الظاهر على شاشة جهاز الحاسوب؟

2. أنشطة الاستقصاء والاستكشاف:: أنشطة تُعزّز الفضول والرغبة في المعرفة.

 **نشاط
عملي**

أَجْرَبُ وَأَسْتَكْشِفُ:

أعمل - بالتعاون مع أفراد مجموعتي - على استخدام لغة البرمجة بايثون (Python)، لإنشاء قائمة تضم أسماء طلبة المجموعة، وقائمة أخرى تُبين هواياتهم المُفضلة. بعد ذلك أكتب الأوامر البعديّة، وأُنتِجُها على طابعة لاسلكيّة. طابعتهما على طابعة لاسلكيّة، طابعتهما على طابعة لاسلكيّة.

3. الأنشطة العملية التطبيقية: أنشطة تُمكن الطلبة من تطبيق ما تعلّموه بصورة عملية.

4. الضغط على زر الإدخال (Enter)، أنظر الشكل (2-11).



4. أنشطة التحليل والتفسير: أنشطة تطوّر مهارات التفكير الناقد والتفكير التحليلي.

 **نشاط
عملي**

أفكر في روتيني اليومي، وأحدد الأعمال والأنشطة التي أقوم بها كل يوم بأنظام، مثل: الاستيقاظ، وتناول طعام الإفطار، والذهاب إلى المدرسة، والتعلم، وتناول طعام الغداء، واللعب، وتناول طعام العشاء، والنوم. بعد ذلك أدون هذه الأعمال والأنشطة في دفتر ملاحظاتي، ثم أناقشها مع زملائي/ زميلاتي ومعلمي/ مُعلّمتي.

5. أنشطة الملاحظة والتأمل: أنشطة تعزز الفهم العميق والتفكير المستقل.

أَبْحَثْ وَأَقَارِنْ: أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن أمثلة على كل من المترجم والمفسر، ثم أقارن بينهما من حيث آلية التنفيذ، والسرعة في التنفيذ، وسهولة اكتشاف الأخطاء.

نشاط

6. أنشطة المناقشة والحوار والمناظرة: أنشطة تعزز التواصل وتبادل الأفكار.

أُفَارِنْ وَأُنَاقِشْ: أُفَارِنْ يبين لغات البرمجة المختلفة، ثُمَّ أَنَاقِشْ زملائي/ زميلاتي في الكيفية التي تتغير فيها طبيعة البرمجة تبعاً لتغير المزايا في كل لغة برمجة.

 نشاط

7. أنشطة البحث والاستقصاء الرقمي: أنشطة تطوّر مهارات البحث باستخدام الأدوات الرقمية والمصادر الإلكترونية الموثوقة.

أُبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن استخدامات العوامل الحسابية والعوامل المنطقية في التطبيقات العملية الحياتية للغة البرمجة بايثون (Python)، ثم أُدوّن ما أتوصّل إليه من نتائج، وأشاركها مع زملائي / زميلاتي في الصف.

 أبحث

إثراء:

تقديم معلومات إضافية تساعد على تعزيز فهم الطلبة للموضوعات التي يدرسونها، وتهدف إلى تقديم منظور أعمق وأوسع للمواد الدراسية؛ مما يتيح للطلبة استكشاف الموضوعات بصورة أكثر شمولاً وإبداعاً. وهي تشمل عادةً تفاصيل إضافية، وأمثلة موسَّعة، وقصصاً تطبيقيةً مُرتبطةً بالموضوع.

إضاعة:

معلومات إضافية تتناول جانباً مُرتبطاً بمعلومة واردة في الدرس، وتُقدِّم تفاصيل أكبر أو توسَّعاً في المعلومة.

المواطنة الرقمية:

مفهوم يشير إلى المسؤوليات والسلوكيات المُتوقَّعة من الطلبة عند استخدامهم وسائل التكنولوجيا وشبكة الإنترنت. وهو يُبيِّن كيف يُمكن التفاعل مع العالم الرقمي بصورة آمنة وأخلاقية، وقانونية. تتناول المواطنة الرقمية جوانب عديدة، تتعلَّق باستخدام الطلبة وسائل التكنولوجيا وشبكة الإنترنت، مثل:

1. البصمة الرقمية (Digital Footprint): تنمية الطلبة هُويَّاتهم الرقمية وإدارتها، ووعيهم بديمومة أفعالهم في العالم الرقمي.

2. السلوك في شبكة الإنترنت (Online Behaviour): مشاركة الطلبة في سلوك إيجابي وآمن وقانوني وأخلاقي عند استخدام التكنولوجيا، وداخل المجتمع الرقمي، بما في ذلك التفاعلات الاجتماعية عبر شبكة الإنترنت، واستخدام الأجهزة المتصلة بالشبكة.

3. المُلْكِيَّة الفكرية (Intellectual Property): إظهار الطلبة فهمًا واحترامًا للحقوق والالتزامات المُتعلِّقة باستخدام المُلْكِيَّة الفكرية ومشاركتها.

4. الخصوصية الرقمية (Digital Privacy): إدارة الطلبة بياناتهم الشخصية للحفاظ على الخصوصية الرقمية والأمان، وإحاطتهم بتقنيَّات جمع البيانات المُستخدمة في تتبُّع تنقُّلاتهم وتحركاتهم عبر شبكة الإنترنت.

أناقش زملائي/ زميلاتي ومُعَلِّمي/ مُعَلِّماتي في إجاباتي.

إضاعة

الخطأ المعروف باسم "off-by-one error"، أو باسم الخطأ بخطوة واحدة، هو من أكثر الأخطاء البرمجية شيوعاً؛ قد يُخطئ المُبرمج عند تحديد شرط نهاية الدوران في حلقة التكرار (while)، أو عند تحديد المدى في حلقة التكرار (for)؛ ما يدفع البرنامج إلى تنفيذ دورة واحدة أقل من المطلوب أو أكثر منه.



المواطنة الرقمية:

- الأخلاقيات الرقمية: احترام آراء الآخرين وأفكارهم عند مناقشة التعليمات البرمجية أو مناقشة المشروعات، وأدِّم النقد البناء والمساعدة للآخرين عند مراجعة تعليمات البرمجية.
- الوعي بالأمن السيبراني: أدرك أهمية استخدام برامج مكافحة الفيروسات وتحديث أنظمة التشغيل بانتظام أثناء العمل في بيئة بايثون (Python).

تصميم برنامج وإعداداته لإنشاء لعبة تخمين الأرقام باستخدام لغة بايثون (Python) / المهمة (5).

أُكمل - بالتعاون مع أفراد مجموعتي - تنفيذ مشروع التصميم والتطوير للعبة تخمين الأرقام. وكُنَّا قد انتهينا في الخطوات السابقة من عرض القائمة الرئيسية في اللعبة، بالاستجابة لِمَا يُدخِله اللاعب عن طريق عرض تعليمات اللعبة، أو كتابة رسالة بحسب الخيار المُدخَّل، وتعديل البرنامج مُتَمِّلاً بتكرار عرض القائمة الرئيسية والخيارات. والآن سأعمل - ضمن المجموعة - على إضافة العنصر الرئيس في اللعبة، وهو إدخال البرنامج للرقم العشوائي.

سأستعمل - مع أفراد مجموعتي - دالة (random) لتوليد رقم مخفي في بداية اللعبة، يتألَّف من سلسلة تحوي (4) أعداد مختلفة، ثم أُخزِّن الأرقام المُمكنة (المُحتملة) في قائمة، وأعمل على تغيير ترتيبها باستخدام الدالة random.shuffle.

أتحقَّق - مع أفراد مجموعتي - من استخدام القوائم بصورة صحيحة، ومن تفعيل عملية توليد الأرقام، ثمَّ أحتفظ بما كُتِب - ضمن المجموعة - في ملف حتى يُمكن لأفراد المجموعة التعديل عليه في الخطوات القادمة.



مشروع

أَقِيْمْ تَعَلَّمِي:

عنصر يقع في نهاية الدرس، ويأخذ شكل مجموعة من الأسئلة، تشمل جميع معارف الطلبة، ومهاراتهم، وقيمتهم، واتجاهاتهم. وفيه تتوزع الأسئلة إلى الجزئيات الثلاث الآتية، التي يُركّز كلٌّ منها على جانب مُعيّن:

1. المعارف: تحفيز الطلبة على استخدام المعارف المُكتسبة خلال الدرس في الإجابة عن الأسئلة؛ مما يُعزّز فهمهم للمحتوى التعليمي.
2. المهارات: تحفيز الطلبة على استخدام مهارات التفكير الناقد والتواصل الرقمي والبحث الرقمي وغيرها في الإجابة عن الأسئلة؛ مما يُطوّر قدراتهم التحليلية والتفاعلية في البيئات الرقمية.
3. القيم والاتجاهات: تحفيز الطلبة على توظيف القيم المُكتسبة والاتجاهات الإيجابية في أداء المهام ومناقشة القضايا؛ مما يُعزّز الوعي الأخلاقي لديهم، ويوجّه سلوكياتهم على نحوٍ إيجابي ومسؤول.

المعرفة: أوظّف في هذا الدرس ما تعلّمتُه من معارف في الإجابة عن الأسئلة الآتية:
السؤال الأول: ما أبرز أوجه التشابه وأوجه الاختلاف بين القوائم والسلاسل؟

السؤال الثاني: أكتب جملة يُمكن استخدامها في حذف الحرف الأخير من سلسلة تُسمّى

السؤال الثالث: أذكر طريقتين مختلفتين (أو أكثر) لتنفيذ كل مهمة من المهام الآتية، ثمّ هذه الطرائق أسهل أو أفضل:

1. حذف الحرف الأخير من سلسلة تُسمّى "abc".

المهارات: أوظّف مهارة التفكير الناقد ومهارة حلّ المشكلات والمهارات البرمجية في الإجابة عن الأسئلة الآتية:

السؤال الأول: اتّبِع البرنامج الآتي من دون تشغيله (اتّبِع كل جزء من البرنامج بصورة مُنفصلة)، ثمّ أذكر ناتج تشغيله.

```
a = [1, 3, 5, 2, 0, 4]
# 1
for i in range(len(a)):
    print(max(a[i:]))
# 2
for i in range(1, len(a)):
    print(max(a[i:]))
# 3
```

القيم والاتجاهات:

أعدّ - بالتعاون مع أفراد مجموعتي - كتيباً إرشادياً يبيّن أكثر الأخطاء البرمجية التي تعرضنا لها أثناء تطبيق المقاطع البرمجية وتوثيق الحلول بهدف مساعدة الطلبة الآخرين على مواجهة المشكلات البرمجية. وأسّعتين بأحد برامج التصميم لتنفيذه، ثمّ أنشره بين طلبة المدرسة بعد مراجعته وتدقيقه مع معلمي/ معلمي وفي الموقع الإلكتروني الخاص بها.

أَقِيْمْ تَعَلَّمِي

الدرس: أوظّف في هذا الدرس ما تعلّمتُه من معارف في الإجابة عن الأسئلة الآتية:

السؤال الأول: ما أبرز أوجه التشابه وأوجه الاختلاف بين القوائم والسلاسل؟

السؤال الثاني: أكتب جملة يُمكن استخدامها في حذف الحرف الأخير من سلسلة تُسمّى

السؤال الثالث: أذكر طريقتين مختلفتين (أو أكثر) لتنفيذ كل مهمة من المهام الآتية، ثمّ هذه الطرائق أسهل أو أفضل:

1. حذف الحرف الأخير من سلسلة تُسمّى "abc".

2. حذف الحرف الأخير من سلسلة تُسمّى "abc".

3. حذف الحرف الأخير من سلسلة تُسمّى "abc".

4. حذف الحرف الأخير من سلسلة تُسمّى "abc".

5. حذف الحرف الأخير من سلسلة تُسمّى "abc".

6. حذف الحرف الأخير من سلسلة تُسمّى "abc".

7. حذف الحرف الأخير من سلسلة تُسمّى "abc".

8. حذف الحرف الأخير من سلسلة تُسمّى "abc".

9. حذف الحرف الأخير من سلسلة تُسمّى "abc".

10. حذف الحرف الأخير من سلسلة تُسمّى "abc".

11. حذف الحرف الأخير من سلسلة تُسمّى "abc".

12. حذف الحرف الأخير من سلسلة تُسمّى "abc".

13. حذف الحرف الأخير من سلسلة تُسمّى "abc".

14. حذف الحرف الأخير من سلسلة تُسمّى "abc".

15. حذف الحرف الأخير من سلسلة تُسمّى "abc".

16. حذف الحرف الأخير من سلسلة تُسمّى "abc".

17. حذف الحرف الأخير من سلسلة تُسمّى "abc".

18. حذف الحرف الأخير من سلسلة تُسمّى "abc".

19. حذف الحرف الأخير من سلسلة تُسمّى "abc".

20. حذف الحرف الأخير من سلسلة تُسمّى "abc".

21. حذف الحرف الأخير من سلسلة تُسمّى "abc".

22. حذف الحرف الأخير من سلسلة تُسمّى "abc".

23. حذف الحرف الأخير من سلسلة تُسمّى "abc".

24. حذف الحرف الأخير من سلسلة تُسمّى "abc".

25. حذف الحرف الأخير من سلسلة تُسمّى "abc".

مُلخَص الوحدة:

نظرة موجزة لأهم البنود التي تعلّمها الطلبة في دروس الوحدة؛ مما يساعد على تعزيز فهمهم، ومراجعة المعلومات والمعارف المكتسبة.



أسئلة الوحدة:

أسئلة تقع في نهاية الوحدة، وتهدف إلى تقييم ما تعلّمه الطلبة من معارف، ومهارات، وقِيَم، واتجاهات. وهي تأخذ أشكالاً مختلفة من حيث النوع (تشمل أسئلة اختيار من مُتعدّد، وأسئلة إجاباتها نعم أو لا، وأسئلة إجاباتها قصيرة، وأسئلة مقالية، وغير ذلك)، والطبيعة (تنوّع بين نظرية وتطبيقية وتحليلية وتأملية)، والمستوى (تدرّج من أسئلة بسيطة لتقييم المعرفة الأساسية إلى أسئلة مُتوسّطة لتقييم التطبيق والتحليل، فأسئلة مُعقّدة لتقييم التركيب والتقييم والإبداع)، بحيث تشمل جميع مؤشّرات الأداء المُتوقّع تحقيقها.



تقويم ذاتي:

أداة تهدف إلى تمكين الطلبة من تقييم تقدّمهم وأدائهم بصورة مستقلة في أثناء عملية التعلّم؛ مما يُعزّز الوعي الذاتي لديهم، ويساعدهم على تعرّف مواطن قوّتهم ومجالات التحسين التي تُلزمهم. ومن ثمّ يُمكن للطلبة وضع خُطط لتحسين أدائهم استناداً إلى نتائج التقييم الذاتي، مثل: تحديد استراتيجيات جديدة للتعلّم، وطلب المساعدة من المُعلّم/ المُعلّمة. يستخدم الطلبة أنموذج التقويم الذاتي في تحديد مدى تحقيقهم الأهداف التعليمية، وتطوير مهاراتهم بصورة مستمرة.

تقويم ذاتي (Self Evaluation)

بعدّ دراستي هذه الوحدة، أقرأ الفقرات الواردة في الجدول الآتي، ثمّ أضع إشارة (✓) في العمود المناسب:

مؤشرات الأداء	نعم	لا	لم أستطع
أُعرف مفهوم الحوسبة الخضراء.	☐	☐	☐
أُبين أهمية الحوسبة الخضراء وفوائدها ومزاياها.	☐	☐	☐
أُسهم في تطبيق الحوسبة الخضراء في حياتي اليومية.	☐	☐	☐
أُعرف مفهوم الفجوات الإلكترونية.	☐	☐	☐
أُوضح طرائق التخلّص الآمنة من الفجوات الإلكترونية.	☐	☐	☐
أذكر أدوات حاسوبية صديقة للبيئة.	☐	☐	☐
أذكر تطبيقات حاسوبية في مجال التعلّم الإلكتروني.	☐	☐	☐
أستخدم تطبيقات حاسوبية في مجال التعلّم عن بُعد.	☐	☐	☐
أذكر تطبيقات حاسوبية في مجال الصحة، وأبين أهميتها.	☐	☐	☐
أستخدم تطبيقات حاسوبية في مجال الشوق والتسويق الإلكتروني.	☐	☐	☐
أبين أهمية تطبيقات الحكومة الإلكترونية في تسهيل المعاملات.	☐	☐	☐
أوضح أهمية بعض تطبيقات الحاسوب في الحياة، مثل: صناعة الأفلام، والتصميم ثلاثي الأبعاد، والطباعة ثلاثية.	☐	☐	☐

195

تأملات ذاتية:

أداة تهدف إلى تحفيز الطلبة على التفكير بعمق في تجاربهم التعليمية، وتمكينهم من تقييم عملية التعلّم، والتعبير عمّا تعلّموه، وبيان كيفية تطبيقه في المواقف العملية، واستخلاص الدروس المستفادة منه، وفهم التحديات التي واجهوها، وتطوير استراتيجيات لتحسين عملية التعلّم مستقبلاً. يُعد أنموذج التأمل الذاتي أداة فاعلة لتعزيز التفكير النقدي والتعلّم الذاتي لدى الطلبة، والإسهام في تحقيق نموّ شخصي وأكاديمي مستدام.

تأملات ذاتية

عزيزي الطالب/ عزيزتي الطالبة:

التأملات الذاتية هي فرصة لتقييم عملية التعلّم، وفهم التحديات، وتطوير استراتيجيات لتحسين عملية التعلّم مستقبلاً. أملأ الفراغ في ما يأتي بالأفكار والتأملات الشخصية التي يُمكن بها تحقيق أفضل استفادة من التجربة التعليمية:

تعلّمت في هذه الوحدة:

يُمكنني أن أطق ما تعلّمتُه في:

الصعوبات التي واجهتها أثناء عملية التعلّم:

دلّلت هذه الصعوبات عن طريق:

يُمكنني مستقبلاً تحسين:

197

الاستراتيجيات والطرائق الداعمة لعملية التعليم والتعلم

استند كتاب الطالب ودليل المعلم على العديد من استراتيجيات التعليم والتعلم المُتداخلة؛ إذ اعتُمدت أكثر من استراتيجية، ودُمجت معاً؛ لضمان تجربة تعلم أفضل، وتحقيق الأهداف المنشودة بأفضل المنهجيات، وصولاً إلى تعلم مُعمّق ومستدام للطلبة.

يعتمد اختيار الاستراتيجية على عوامل عدّة، أبرزها: سياق التعلم، والتّاجات، وخصائص الطلبة النمائية والمعرفية، والإمكانات المتوافرة، والزمن المتاح. وهذه بعض المنهجيات المُقترحة:

التعلم القائم على المشروع (Project Based Learning: PBL):

نهج تعليمي تعاوني يُركّز على الطالب/ الطالبة بدلاً من المنهج. وفيه يعمل الطلبة في مجموعات، ويستخدمون مجموعة مُتنوّعة من المهارات (مثل: التفكير الناقد، والتواصل، والإبداع) في بناء معرفتهم. إنّه نهج ديناميكي ومرن للتعليم والتعلم، يتضمّن مشاركة الطلبة في نشاط مستوحى من العالم الحقيقي، ومشروعات ذات مغزى شخصي؛ مما يُعزّز لديهم التعلم المنشود، ويزيد من مشاركتهم الفاعلة، ويُحفّزهم على العمل بكفاءة أكثر.

يكتسب الطلبة في هذا النهج المعرفة والمهارات عن طريق معالجة تحدّيات حقيقية أو مشكلات واقعية. وهذا يُحتمّ عليهم أداء مهام مُحدّدة لإثبات إتقانهم المحتوى. وتحقيقاً لذلك، يجب أن يكون التعلم ضمن سياق مُمنهج ومُحدّد، يقوم على مشاركة الطلبة في عملية التعلم، فتتحقّق الأهداف والمُخرجات بالتفاعل بين الطلبة أنفسهم، وبين الطلبة والمُعَلِّمين/ المُعلِّمات، وبين الطلبة وأفراد المجتمع، ويشمل ذلك تشارك المعرفة، والتغذية الراجعة البناءة.

أمّا أهم مُخرجات التعلم القائم على المشروع فهي: المسؤولية، والاستقلالية، والنشر، والتوسعة؛ إذ يُمنح الطلبة حرية الاختيار، والعمل ضمن مساحة خاصة بكلّ منهم عند اختيار المُنتج أو التحكّم في خصائصه؛ مما يجعلهم أكثر استقلالية ومسؤولية عن تعلّمهم، ويزيد من دافعيتهم للبحث والتقصّي والعمل، ويُفضي إلى تعلّم أكثر عمقاً.

يشمل هذا النهج أنشطة تعليمية يتطلّب تنفيذها عملاً جماعياً وآخر فردياً؛ مما يُعزّز مهارات التعاون والتواصل بين الطلبة. كذلك تُستخدم في هذا النوع من التعلم أدوات رقمية وبرامج تعليمية تعمل على دعم عملية التعلم وتطوير المشروعات.

في نهاية الوحدة التعليمية، يعرض الطلبة مُنتجاتهم، ويشاركونها مع الزملاء/ الزميلات والمُعَلِّمين/ المُعلِّمات، ثم تُقيّم هذه المُنتجات وفق معايير مُحدّدة. يلي ذلك تنظيم جلسات تأمل وتقييم ذاتي، يُوظّف فيها الطلبة ما تعلّموه من معارف ومهارات، مُبيّنين كيف يُمكنهم تحسين الأداء مستقبلاً.

السقالات التعليمية (Instructional Scaffolding):¹

تدعم السقالات التعليمية تطوير مهارات حلّ المشكلات، والتعلّم الذاتي، والعمل الجماعي. وهي تكون على شكل تعليمات، أو مصادر سمعية وبصرية، أو أدوات رقمية، أو نماذج، أو مُخطّطات، أو روابط إلكترونية، أو غير ذلك. يُقدّم الدعم للطلبة مُوقّتاً، ثم يُطلّب إليهم الاستمرار في أداء المهام وحدهم؛ مما يُعزّز لديهم الثقة بأنفسهم. وتساعد

1 McLeod, S. (2020). Zone of Proximal Development. Simply Psychology. Retrieved August 10, 2024 from <https://www.simplypsychology.org/zone-of-proximal-development.html>.

التعليمات الواضحة والمُحدّدة في توجيه الطلبة؛ لكيلا يَحيدوا عن المسار الصحيح للتعلّم، فضلاً عن استخدامها مرجعيةً للتقييم الذاتي والتقييم النهائي. تُعدُّ الأخطاء فرصةً للتعلّم؛ لذا يجب التفكير فيها، وتأملها، وتوضيح أسبابها، وتعرّف آليّة تجاوزها.

التعلّم التعاوني (Collaborative Learning)²

نهج تعليمي يشمل مجموعة مُتنوّعة من الأساليب التي تتطلّب جهداً فكرياً مشتركاً بين الطلبة، أو بينهم وبين المُعلّمين/ المُعلّّات.

يهدف هذا النوع من التعلّم إلى بناء المعرفة عن طريق الأنشطة التي تتطلّب عملاً جماعياً؛ إذ يعمل الطلبة في مجموعات ثنائية أو أكثر، ويبحثون معاً عن المعرفة والحلول والمعاني، أو يُنشئون مُنتجاً مُعيّناً.

يقوم هذا النهج على تصميم تجارب تعليمية تفاعلية، فيصبح الطلبة شركاء في عملية التعلّم بدلاً من بقائهم مُتلّقين للمعلومات فقط. تختلف أنشطة التعلّم التعاوني بعضها عن بعض بصورة كبيرة، لكنّ معظمها يُركّز على استكشاف الطلبة المادة الدراسية أو تطبيقها، بدلاً من تقديمها أو شرحها من طرف المُعلّم/ المُعلّّة.

تمتاز الفصول التعاونية بأنّ عملية الاستماع للدروس وتدوين الملاحظات قد لا تختفي بصورة كاملة، وإنّما توجد إلى جانب عمليات أُخرى، تعتمد على مناقشات الطلبة والعمل النشط في المادة الدراسية؛ لذا يميل المُعلّمون/ المُعلّّات في أثناء استخدام أساليب التعلّم التعاوني إلى العمل بوصفهم مُدرّبين أو مُوجّهين في عملية تعلّم أكثر تفاعلاً، ويُسهّمون في تصميم تجارب فكرية ثريّة للطلبة.

تتعدّد مزايا التعلّم التعاوني؛ فهو يُحفّز الطلبة على المشاركة الفاعلة في ما بينهم، ويُعزّز التعاون والعمل الجماعي، ويُطوّر مهارات القيادة والمسؤولية المدنية.

يُطبّق هذا النوع من التعلّم عن طريق الأنشطة المُتنوّعة، مثل: المناقشات الجماعية، وجلسات العمل التشاركية.

التعلّم النشط (Active Learning)³

نهج تعليمي يتضمّن إشراك الطلبة في أنشطة تعليمية تتطلّب تفكيراً وبحثاً واستقصاءً لما يتعيّن عليهم تعلّمه. يُركّز هذا النهج على تطوير مهارات الطلبة، ولا يكفي فقط بعرض المعلومات ونقلها؛ إذ يتطلّب تحقيقه أداء الطلبة العديد من الأنشطة، مثل: العروض التقديمية، ولعب الأدوار، والتدريس التبادلي، والتعلّم التعاوني، والتعلّم بالاكشاف، وغير ذلك. يعمل التعلّم النشط على تحسين نتائج تعلّم الطلبة، خلافاً للنهج التقليدي القائم على المحاضرة؛ فقد أظهرت نتائج الدراسات أنّ الطلبة الذين يشاركون في التعلّم النشط يُحقّقون أداءً أفضل في الاختبارات، مقارنةً بمن يتلقّون تعليمًا تقليدياً. يضاف إلى ذلك أنّ التعلّم النشط يُسهّم في تعزيز بيئة تعليمية شاملة تدعم الطلبة من خلفيات مُتنوّعة؛ ما يُقلّل من فجوة التحصيل بين الطلبة.

2 Smith, B. L., & MacGregor, J. T. (1992). What is collaborative learning? In A. Goodsell, M. Maher, V. Tinto, B. L. Smith, & J. T. MacGregor (Eds.), Collaborative learning: A sourcebook for higher education (pp. 9–22). University Park, PA: National Center on Postsecondary Teaching, Learning, and Assessment. Retrieved August 10, 2024 from <https://teach.ufl.edu/wp-content/uploads/2016/07/WhatisCollaborativeLearning.pdf>

3 Brame, C. (2016). Active learning. Vanderbilt University Center for Teaching. Retrieved June 28, 2024, from <https://cft.vanderbilt.edu/active-learning/>

كذلك اعتمد في دليل المعلم عدد من استراتيجيات التعلم، مثل:

التعلم بالاستقصاء والاستقصاء الرقمي (Inquiry Based Learning & Digital Inquiry):

يُعرّف التعلم بالاستقصاء بأنه نهج تعليمي يركّز على إثارة فضول الطلبة، وتحفيزهم على طرح الأسئلة والبحث عن الإجابات عن طريق استقصاء الموضوعات بصورة مستقلة. أمّا الاستقصاء الرقمي فهو تطبيق هذا النهج باستخدام الأدوات والتقنيات الرقمية.

يمتاز هذا النهج بتحفيز الطلبة على استكشاف الموضوعات وحدهم عن طريق إجراء البحوث وتنفيذ المشروعات التي تتطلب جمعًا وتحليلًا للبيانات؛ مما يساعدهم على تطوير مهارة البحث ومهارة التفكير الناقد، ويُعزّز لديهم الاستقلالية في التعلم.

الصف المقلوب (Flipped Classroom):

استراتيجية تعليمية تُستخدم في تحفيز التعلم الذاتي والتفاعل النشط داخل الغرفة الصفية. وفيها يؤدّي الطلبة عددًا من المهام التي تتضمن بحثًا واستكشافًا لقضايا مُحدّدة اتّفق عليها مُسبقًا، ثمّ يتشاركون معًا النتائج باستخدام بعض الأدوات الرقمية، فيحدث التفاعل والنقاش في الفضاء الرقمي، حيث يناقشون ما تعلّموه وما توصّلوا إليه من نتائج، ويعملون على تطبيقه بتنفيذ أنشطة تفاعلية داخل الغرفة الصفية؛ مما يساعد الطلبة على تحسين عملية فهمهم للمواد التعليمية، ويُمكن المعلمين/ المُعلّمات من تركيز جهودهم على توجيه الطلبة، ومساعدتهم على تطبيق المفاهيم بدلًا من الاكتفاء بنقل المعلومات.

عباءة الخبير⁴ (Mantle of the Expert):

نهج تعليمي تفاعلي يستخدم الدراما وسيلةً لتعليم الطلبة عن طريق تقمُّص أدوار الخبراء في مواقف مُحدّدة. يهدف هذا النهج إلى إشراك الطلبة في عملية التعلم عن طريق تحفيز التفكير الناقد والتفكير الإبداعي والتعاون لديهم. وفيه يتعلّم الطلبة بوصفهم مجموعة مُتخيّلة من الخبراء؛ مما يجعل عملية التعلم مُفعّمة بالحيوية والإثارة، ويُحفّز الطلبة على استكشاف وجهات النظر المختلفة. تُسهّم هذه المقاربة التعليمية القائمة على الاستقصاء الدرامي في النموّ المعرفي والاجتماعي للطلبة، وتطوير مهاراتهم الحياتية.

التعلم الذاتي (Self Learning):

حصول الطلبة على المعرفة المنشودة وتنمية مهاراتهم بصورة مستقلة اعتمادًا على المصادر المتوافرة (المقروءة، أو المسموعة، أو المرئية) غير الكتاب المدرسي والمُعلّم؛ بُغية تحسين مهاراتهم وقدراتهم. يساعد التعلم الذاتي على تنمية مهارات البحث والربط والتحليل والاستنتاج والتعلّم مدى الحياة. وهو يُعدّ مُكمّلًا للتعلم الاعتيادي، ويساعد الطلبة على تحقيق النجاح في الحياة والعمل؛ إذ يُمكنهم من تطوير المهارات والمعرفة اللازمة لتحقيق أهدافهم.

4 تايلور، ت. (2018). دليل المعلمين إلى دراما عباءة الخبير- نهج في التعلم التحويلي. برنامج البحث والتطوير التربوي، مؤسسة عبد المحسن القطان.

التعلُّم الإلكتروني (E-Learning):

يُعرَّف التعلُّم الإلكتروني بأنه عملية تعليم وتعلُّم تقوم على استخدام الوسائط الإلكترونية (مثل: جهاز الحاسوب، وبرمجياته المُتعدِّدة، والشبكات، والإنترنت، والمكتبات الإلكترونية) في نقل المعلومات بين المُعلِّم/ المُعلِّمة والطالب/ الطالبة؛ لتحقيق نتائج تعليمية مُحدَّدة وواضحة. يُمكن التعلُّم الإلكتروني الطلبة من التفاعل مع المحتوى عن طريق الشبكات أو المصادر الرقمية الأخرى، ويمتاز بخصيصة المرونة في الزمان والمكان.

المهارات

يُعنى منهاج المهارات الرقمية بتطوير المهارات الحياتية (مهارات القرن الحادي والعشرين). استنادًا إلى أبعاد التعلُّم الأربعة التي حددها ديلور*، والإطار العام للمهارات الحياتية الذي أصدرته اليونسف لمنطقة الشرق الأوسط وشمال إفريقيا، فإنَّ المهارات الحياتية التي يجب العمل على تطويرها لدى الطلبة لتحقيق أهداف التعلُّم الكبرى هي:



UNICEF MENA. (2017). Reimagining Life Skills and Citizenship Education in the Middle East and North Africa: Conceptual and Programmatic Framework - Executive Summary. Retrieved June 30, 2024, from <https://www.unicef.org/mena/reports/reimagining-life-skills-and-citizenship-education-middle-east-and-north-africa>

- **التعلُّم لأجل المعرفة:** يشمل ذلك عددًا من المهارات، مثل: مهارة حلّ المشكلات، ومهارة التفكير الناقد، ومهارة الإبداع. وهذه المهارات ضرورية لتطوير قدرة الطلبة على اكتساب المعرفة وتطبيقها بفاعلية في حياتهم اليومية والمهنية.
- **التعلُّم لأجل العيش مع الآخرين:** يشمل ذلك عددًا من المهارات، مثل: المشاركة، والتعاطف، واحترام التنوع. وهذه المهارات تُعزِّز قدرة الطلبة على التعامل الفاعل مع الآخرين في بيئات مُتنوعة؛ مما يُسهم في بناء مجتمع مُتماسك يحترم التنوع الثقافي والتنوع الاجتماعي.
- **التعلُّم لأجل العمل:** يشمل ذلك عددًا من المهارات، مثل: التعاون، والتفاوض، وصنع القرارات. وهذه المهارات لا بُدَّ من تطويرها؛ لكي يتمكن الطلبة من دخول سوق العمل بفاعلية وكفاءة؛ مما يدعم الاقتصاد، ويُعزِّز قدرته التنافسية.
- **التعلُّم لنكون:** يشمل ذلك عددًا من المهارات، مثل: التواصل، والصمود، وإدارة الذات، والكفاءة الذاتية. وهذه المهارات لا بُدَّ من تطويرها؛ لكي يتمكن الطلبة من بناء شخصيات قوية ومستقلة وقادرة على مواجهة التحديات، وتحقيق النجاح على المستوى الشخصي والمستوى المهني.

* Delors, J. (1996). Learning: The Treasure Within: Report to UNESCO of the International Commission on Education for the Twenty-first Century. UNESCO Publishing.

استراتيجيات التقويم وأدواته:

التقويم التشخيصي (Diagnostic Assessment):

تقويم خاص ببدء عملية التعلم، أو بدء محطة جديدة من محطات التعلم. وفيه يهدف المعلم / المعلمة إلى استكشاف المعرفة السابقة لدى الطلبة في موضوع مُعَيَّن، أو تقييم مستوى معرفتهم ومهاراتهم؛ للكشف عن درجة استعدادهم لعملية التعلم.

يساعد هذا النوع من التقويم المعلم / المعلمة على تخطيط الخطوات القادمة بناءً على النتائج، وقد يتخذ أشكالاً متعددة، مثل: النشاط التمهيدي، والأنشطة من نوع (KWL: Know, Want-to-know, and Learned)، والاختبار التشخيصي القصير، وذلك تبعاً لماهية الموضوع، والمهارة التي يراد إكسابها للطلبة.

التقويم التكويني (Formative Assessment):

تقويم يُستخدم في أثناء عملية التعليم والتعلم، ويساعد كلاً من المعلم / المعلمة والطلبة على متابعة التقدم في العملية التعليمية التعلمية، وتقديم تغذية راجعة مستمرة، تتعلق بنتائج التعلم، وأهدافه، ومواطن القوة، والمواطن التي تتطلب تحسيناً وتعزيزاً؛ مما يتيح اتخاذ الإجراءات اللازمة في الوقت المناسب، قبل الانتقال إلى مراحل أخرى. في ما يأتي بعض المواقف التكوينية التي تخدم عملية التقويم لأجل التعلم:

1. الأسئلة والأجوبة المباشرة: تبادل المعلم / المعلمة والطلبة الأسئلة والأجوبة؛ مما يعطي مؤشرات عن مجريات عملية التعليم والتعلم، ومن ثمَّ يُمكن المعلم / المعلمة والطلبة من الكشف عن درجة فهمهم موضوعاً ما، وتصحيح الأخطاء حالاً، أو اتخاذ الإجراءات اللازمة في وقتها قبل الانتقال إلى محطة أخرى من محطات التعلم.

2. الملاحظة المنظمة: تخطيط المعلم / المعلمة ملاحظة الطلبة وفق معايير ومؤشرات واضحة قبل البدء بنشاط ما أو موقف تعليمي مُحدَّد. وفيها يعمل المعلم / المعلمة على مراقبة الطلبة خلال عملية التعلم (في أثناء النقاش، أو العمل الجماعي، أو العمل الفردي)؛ لتقييم مدى فهمهم ومشاركتهم في الأنشطة. تساعد هذه الملاحظة المعلم / المعلمة على الكشف عن الثغرات والمواطن التي تحتاج إلى تركيز، أو توضيح أكثر، أو تغيير في الأسلوب.

3. العروض التوضيحية والنقاشات الجماعية: موقف يحدث عادةً بعد نشاط جماعي أو نشاط فردي. وفيه تُمثل التغذية الراجعة اللحظية من المعلم / المعلمة، والنقاش بين زملاء، والتغذية الراجعة بين الأقران فرصة لتبادل الأفكار والمعلومات؛ مما يُعزِّز التفكير الناقد والتعلم التعاوني لدى الطلبة، ويساعد المعلم / المعلمة على تقييم درجة تمكُّن الطلبة، وفهمهم للمحتوى.

4. أوراق العمل والأنشطة العملية: أوراق تُسهم في تقييم تعلُّم الطلبة، وتساعد المعلم / المعلمة على متابعة تقدُّم الطلبة وتحديد ما يعانون من مواطن ضعف بحاجة إلى معالجة واهتمام. أمَّا الأنشطة العملية، فهي أنشطة واضحة الأهداف والنتائج، ولها ارتباط بعملية التعليم والتعلم.

5. الاختبارات القصيرة والمهام القصيرة: بطاقات أو أوراق يكتب فيها الطلبة مُلخصًا سريعًا لما تعلّموه في نهاية الموقف التعليمي، أو هي اختبارات قصيرة يجيب عنها الطلبة، وتساعد المُعلّم / المُعلّمة على تقييم فهم الطلبة لحظيًا، وتحديد النقاط التي تحتاج إلى مراجعة في الدروس القادمة.
6. مُنتجات التعلّم / المشروعات القصيرة: مُنتجات ومشروعات تكون على مستوى الدرس، أو ترتبط بجزء مُحدّد من أهداف التعلّم المنشودة؛ مما يتيح للطلبة تطبيق ما تعلّموه مباشرة في سياق عملي تطبيقي، يليه عرض للمنتجات، وتأمل في التجربة، ونقاش وتحديد للتحديات، وبحث في سُبُل التغلّب عليها في حينها قبل الانتقال إلى مرحلة مختلفة من عملية التعلّم.
7. المحاكاة وتمثيل الأدوار: أنشطة تتيح للطلبة تمثيل مواقف واقعية، وتطبيق ما تعلّموه في سياقات حيّة.
8. المناظرة: مناقشة الطلبة قضايا ذات صلة بموضوعات التعلّم وأهدافه. وهي تتطلّب من الطلبة تنظيم أفكارهم وخبراتهم وما تعلّموه عن موضوع مُعيّن، وتقديم الحُجج والأدلة التي تدعم مواقفهم، وتساعدتهم على تحسين مهاراتهم في الاتصال والتواصل عن طريق التحدّث أمام الجمهور والدفاع عن مواقفهم، وتُسهم في تعزيز ثقتهم بأنفسهم وكفاءتهم الذاتية. كذلك تُمكن المناظرة المُعلّمين / المُعلّمات والطلبة من الكشف عن مدى الفهم العميق للموضوع؛ إذ يتطلّب تحضير الحُجج والأدلة فهمًا شاملاً للمادة.
9. خرائط المفاهيم: أدوات بصرية يستخدمها الطلبة في تنظيم الأفكار والمفاهيم، والربط بينها.
10. التقييم الذاتي: أدوات تُمكن الطلبة من تقييم أدائهم، وتحديد مواطن القوّة والضعف لديهم.

التقويم الختامي (Summative Assessment):

تقويم يُستخدم في نهاية عملية التعليم والتعلّم (أي في نهاية مرحلة دراسية أو وحدة تعليمية)؛ لتحديد درجة تحقيق الأهداف التعليمية المُحدّدة. يهدف هذا النوع من التقويم إلى قياس مستوى تعلّم الطلبة ومعارفهم المُكتسبة، وتقييم أدائهم بصورة شاملة ونهائية. يأخذ التقويم الختامي أشكالًا وأنواعًا عديدة، يتلاءم كلّ منها مع استراتيجيات التعلّم المُتبّعة. وفي ما يأتي بعض الأمثلة عليها:

1. الاختبارات النهائية: اختبارات تشمل جميع الموضوعات التي درسها الطلبة خلال مرحلة دراسية. وهي تُستخدم في تقييم درجة تحقّق الأهداف التعليمية، وتحديد مستوى فهم الطلبة للمواد الدراسية. تشمل الاختبارات النهائية عددًا من الأسئلة المُتنوّعة (مثل: أسئلة اختيار من مُتعدّد، وصح وخطأ، وأسئلة مقالية، وأسئلة قصيرة)، فضلًا عن أسئلة تتناول الجوانب المعرفية، والجوانب المهارية التطبيقية، والقيّم والاتجاهات.
2. المشروعات الكبرى (Unit Projects): مهام طويلة الأمد، تتطلّب من الطلبة تطبيق المعرفة والمهارات المُكتسبة في سياق عملي. وهي تُستخدم في تقييم قدرة الطلبة على التخطيط، والتنفيذ، والتحليل، وتقديم الحلول، وتطبيق

ما تعلّموه في سياقات عملية وحقيقية وواقعية، مثل: المشروعات البحثية، ومشروعات تصميم الوسائط المتعددة، ومشروعات تصميم البرمجيات، ومشروعات دراسة الحالة.

3. **التقويم العملي (Practical Assessments)**: تقويم يشمل المهارات العملية، ويُنفَّذ عن طريق الأنشطة أو التجارب العملية التطبيقية، ويُستخدم في تقييم قدرة الطلبة على تطبيق المعرفة والمهارات في بيئة عملية.

أدوات التقييم في دليل المُعلِّم :

أدوات التقييم النوعية :

1. **سلالم التقدير الوصفية (Descriptive Rubrics)**.
2. **سلالم التقدير العددية (Numerical Rubrics)**.
3. **سلالم التقدير اللفظية (Verbal Rubrics)**.
4. **قوائم الرصد (Checklists)**.

أدوات التقييم الكمية:

1. **الاختبارات القصيرة (Quizzes)**.
2. **اختبارات الوحدة (Unit Exam)**.
3. **الاختبارات النهائية (Final Exams)**.

أدوات التقويم في كتاب الطالب

1. نشاط تمهيدي.

نُقدِّمُ القوائم إحدى المزايا الهُمةُ للغات البرمجة، إذ تُسهِّمُ في إدارة تنظيم البيانات، وتتيح للمستخدم التعامل معها واسترجاعها بسهولة. ولكن، كيف يُمكن توليف القوائم في لغة البرمجة بايثون (Python)؟ وهل يختلف ذلك عن توليفها في لغات البرمجة الأخرى؟

أفترض أنني أريد إنشاء برنامج يتولى قراءة علامات (100) طالب في أحد الامتحانات، ثم يعرض هذه العلامات مرتبة بصورة تصاعديّة. أناقش أفراد مجموعتي في الأسئلة الآتية بناءً على ما تعلّمته عن لغة البرمجة بايثون (Python):

1. هل يجب عليّ تخزين علامات هؤلاء الطلبة أم يمكنني كتابة هذا البرنامج دون الحاجة لتخزين العلامات؟
- إذا عليّ عليّ تخزين علامات هؤلاء الطلبة، فإلى كم متغير أحسّاج؟ وما التّعليقات التي سأواجهها عند محاولة تخزين العلامات؟
- كيف يمكنني طباعة العلامات وترتيبها تصاعديًا؟ وما حلّ تعلّمه عن لغة البرمجة بايثون (Python) سيساعدني على ذلك؟

القوائم (Lists)

تمثل القائمة في لغة البرمجة بايثون (Python) عدداً من القيم التي يُخزّن بعضها مع بعض، وترتبط معاً بمعنى وظيفي مشترك. فمثلاً، يمكن للمستخدم تخزين أسماء الشوارع داخل إحدى المدن في قائمته، وكذلك تخزين أسعار البضائع التي اشتراها أحد العملاء، أو تخزين علامات الطلبة في الصف الحادي عشر.

تتطلب استخدام القوائم في لغة البرمجة بايثون (Python) تعريف القائمة أولاً، ثم تخزين العناصر داخلها.

على النحو الآتي:

المقطع البرمجي التالي يعرف قائمة باسم List name تحتوي على أربعة عناصر، وكل عنصر في القائمة يحتوي على قيمة ثابتة واحدة تتكرر، وهي 'value 1'

```
mylist = ['value1', 'value1', 'value1', 'value1']
```

79

2. أَقِمْ تَعْلَمِي.

أَقِيْمُ تَعْلُمِي

المعرفة: أوظف في هذا الدرس ما تعلمته من معارف في الإجابة عن السؤالين الآتيين:
السؤال الأول: أوضّح المقصود بكلّ مما يأتي:
1. لغة البرمجة.

2. لغة الآلة.
3. البرنامج.

السؤال الثاني: أقرن بين لغات البرمجة عالية المستوى ولغات البرمجة منخفضة المستوى من حيث سهولة القراءة، والتطوير، والأداء، والكفاءة.

الآتية:

السؤال الأول: أُمِّزْ بين لغات البرمجة الكتلية ولغات البرمجة النصية من حيث آلية تمثيل الأوامر، ثمَّ اذكر مثالاً على كلٍّ منهما.

السؤال الثاني: لماذا يُعَدُّ المُتَرَجِّمُ أسرعَ تنفيذًا من المُفَسِّرِ؟

18

3. أنشطة مُتنوّعة خلال الدرس.

٤. الذكاء الاصطناعي (Artificial Intelligence): يُعرف الذكاء الاصطناعي بأنه نظام حاسوبي قادر على محاكاة الذكاء البشري، مثل: التفكير، والتمميم، والتعلم من التجارب السابقة، وحل المشكلات، واتخاذ القرارات.

قد أمكن لبعض برامج الذكاء الاصطناعي الوصول إلى

أَتَأْتِلُ الصُّورَةَ الْمُشَبَّهَةَ فِي الشَّكْلِ
(3-1)، ثُمَّ أَفَكِّرُ كَيْفَ أَتَصَوَّرُ
إِذَا ظَهَرَتْ بَيْنَ رَسَائِلِ بَرِيدِي
الْإِلِكْتُرُونِيِّ، أَوْ فِي حِسَابِي ضَمَّنَ
وَسَائِلِ التَّوَاصُلِ الْاِحْتِمَالِيَّةِ

يَبْتَغِ وَأَشَارِكْ: اخْتَارَ خَيْرًا أَشْكَلَ فِي صَحِيحِهِ مِنْ بَيْنِ الْأَخْبَارِ الْمَشْهُورَةِ فِي سَائِلِ التَّوَالِفِ الْأَجْنَمِيِّ، ثُمَّ اتَّخَذَ مِنْهُ بِاسْتِخْدَامِ أَحَدِ مَوَاقِعِ التَّحْقِيقِ مِنَ الْأَخْبَارِ، ثُمَّ أَشَارَكَ زَمَلَانِي/ زَمِيلَانِي فِي مَا اتَّصَلَ إِلَيْهِ مِنْ نَتَائِجِ، وَأَحْرَصَ عَلَى تَوْعِيهِمْ بِوُجُوبِ التَّيَبُّتِ مِنْ صَحَّةِ الْأَخْبَارِ قَبْلَ نَرْسَاهَا أَوْ تَدَاوُلِهَا.

أنتفض زملائي/ زميلاتي في ما تبته بعض وسائل التواصل الاجتماعي من أخبار كاذبة ومضلّلة،
وأثر ذلك في أشاع رغبة انتشارها، وكيف يمكن للأفراد والمجتمعات الحد من انتشار هذه
الأخبار في وسائل التواصل الاجتماعي. بعد ذلك أشارك زملائي/ زميلاتي في أفكارتي
ومقتزحاتي.

المواطنة الرقمية:

يتعينُ عليَّ مراعاةُ الجوانبِ الآتيةِ بعدَ دراسةِ موضوعِ (الإعلامُ الرقميُّ):

- **التحقّق من صحة المعلومات:** تأبّد أن المعلومات صحيحة وأن مصداقها موثوقة قبل نشرها في وسائل التواصل الاجتماعي.
- **التفكير النقّاد:** العمل مع المعلومات الرقمية بحذر، واعتمد مبدأ الشك والتحقّق قبل التسليم بكل ما زار أو شاهده.
- **مراعاة القانون:** أراّن في استخدامي للإعلام الرقمي وبحاتي الواقعية، واتجنّب الإفراط في الاعتماد على الوسائل الرقمية.
- **العمل المسؤول:** أبلغ الجهات المتخصصة عن المحتوى الرقمي المشبّهي أو المحتوى الرقمي الكاذب، يتّمة بإيجاد بيئة رقمية أكثر أمنا وموضوعية.

الدرس الخامس

القوائم (Lists)

الفكرة الرئيسة:

نعرف كيف يمكن التعامل مع القوائم وسلاسل الحروف في برمجة بايثون (Python)، واستخدامها في تخزين مجموعات البيانات المترابطة وإدارتها وتنفيذ عمليات فيها.

المفاهيم والمصطلحات:

قائمة (List)، سلسلة الحروف (String)، موقع العنصر في القائمة (أو الحرف في السلسلة) (Index)، إصااق القوائم أو سلاسل (Concatenation)، القابلية للتغيير (Mutability)، قائمة المُركَّبة (Nested List)، المصفوفة ثنائية الأبعاد (2D Matrix).

مُنْتَجَاتُ التَّعْلِيمِ
(Learning Products)

تعديل البرنامج باستخدام
قوائم اللعبة في توليد
الأرقام العشوائية، ضمن
سياق تصميم لعبة تخمين
الأرقام باستخدام برمجة
بايثون (Python).

صميم برنامج وإعداده لإنشاء لعبة تخمين الأرقام باستخدام لغة بايثون (Python) / المهمة (5).

تُنفَّذُ أهدافُها في إطار مجموعة من تنفيذ مشروع التصميم والتطوير للعبة تخمين الأرقام. ختمه الرابع من طرِيق عرض تعليمات اللعبة، أو كتابة وصف اللعبة بحسب الشدّخل، ويبدل البرنامج مُثْمَلًا بتكرار عرض القائمة الرئيسية والرسائل. والأما سَاحِبُ الأرقام - ضمن استعمال - مع أضافة مجموعة من دالة (random) لتوليد رقم مخفي في بداية اللعبة، وتُنفَّذُ سلسلة تحوي (4) أعداد مختلفة، تُعْزَرُ الأرقام المُمكنة (المُختلِفة) في قائمة، وأعمل تغيير ترتيبها باستخدام الدالة random.shuffle.

ل - مع أفراد مجموعتي - من استخدام القوائم بصورة صحيحة، ومن تفعيل عملية توليد م، ثم أحفظ بما يُجب - ضمن المجموعة - في ملصف حتى يُمكن لأفراد المجموعة ل عليه في الخطوات القادمة.

5. أسئلة الوحدة.

7. تأملات ذاتية.

أسئلة الوحدة

السؤال الأول: أكتب المصطلح المناسب بجانب كل عبارة من العبارات الآتية:

العبارة

تصميم أجهزة الحاسوب والأجهزة الإلكترونية الأخرى، أو تصنيعها، أو استخدامها بصورة تُشكّل من آثارها الضارة بالبيئة، مثل: انبعاثات الكربون، واستهلاك الطاقة.

أدوات تحتوي على مقابس وأسلاك ومكونات إلكترونية، مثل: أجهزة التلفاز، وأجهزة الحاسوب، والهواتف المحمولة، ومُكيّفات الهواء، وألعاب الأطفال الإلكترونية.

إنشاء كائن ثلاثي الأبعاد من ملف رقمي.

برنامج وطني مُهمّ لحوسبة قطاع الصحة فسي الأردن، أُطلق عام 2009م.

جهاز حاسوب لا يحوي شاشة أو لوحة مفاتيح، ويُشبه جهاز العرض (Projector) في مبدأ عمله.

السؤال الثاني: ما التحديثات التي تتف حائلًا دون تطبيق مبدأ الحوسبة الخضراء؟

السؤال الثالث: أعلّل ما يأتي:

1. استبدال مادة الخيزران (Bamboo) بمادة البلاستيك التي تدخل في صناعة ملحقات أجهزة الحاسوب.

تأملات ذاتية

عزيزي الطالب/ عزيزتي الطالبة:

التأملات الذاتية هي فرصة لتقييم عملية التعلم، وفهم التحديات، وتطوير استراتيجيات لتحسين عملية التعلم مستقبلاً. أملاً الفراع في ما يأتي بالأفكار والتأملات الشخصية التي يُمكنُ بها تحقيق أفضل استفادة من التجربة التعليمية:

تعلّمت في هذه الوحدة:

يُمكنني أن أعلّق ما تعلّمتُه في:

الصعوبات التي واجهتها أثناء عملية التعلم:

دَلَّت هذه الصعوبات عن طريق:

يُمكنني مستقبلاً تحسين:

6. تقويم ذاتي.

8. مشروع الوحدة.

تقويم ذاتي (Self Evaluation)

بعد دراستي هذه الوحدة، أقرأ الفقرات الواردة في الجدول الآتي، ثمّ أضع إشارة (✓) في العمود المناسب:

مؤشرات الأداء	نعم	لا	لنشت متّخذًا
أُعرّف مفهوم الحوسبة الخضراء.	☐	☐	☐
أُبين أهمية الحوسبة الخضراء وفوائدها ومزاياها.	☐	☐	☐
أُسميهم في تطبيق الحوسبة الخضراء في حياتي اليومية.	☐	☐	☐
أُعرّف مفهوم النفايات الإلكترونية.	☐	☐	☐
أُوضّح طرق التخلص الآمن من النفايات الإلكترونية.	☐	☐	☐
أذكر أدوات حاسوبية صديقة للبيئة.	☐	☐	☐
أذكر تطبيقات حاسوبية في مجال التعلّم الإلكتروني.	☐	☐	☐
أستخدم تطبيقات حاسوبية في مجال التعلّم عن بُعد.	☐	☐	☐
أذكر تطبيقات حاسوبية في مجال الصحة، وأبين أهميتها.	☐	☐	☐
أستخدم تطبيقات حاسوبية في مجال التوثّق والتسويق الإلكتروني.	☐	☐	☐
أبين أهمية تطبيقات الحكومة الإلكترونية في تسهيل المعاملات.	☐	☐	☐
أوضح أهمية بعض تطبيقات الحاسوب في الحياة، مثل: صناعة الأفلام، والتصميم ثلاثي الأبعاد، والطباعة ثلاثية.	☐	☐	☐

المواطنة الرقمية.

الأخلاقيات الرقمية: أحترم آراء الآخرين وأفكارهم عند مناقشة التعليمات البرمجية أو مناقشة المشروعات، وأقدم النقد البناء والمساعدة للآخرين عند مراجعة تعليمات البرمجية.

الوعي بالأمن السيبراني: أدرك أهمية استخدام برامج مكافحة الفيروسات وتحديث أنظمة التشغيل بانتظام أثناء العمل في بيئة بايثون (Python).

تصميم برنامج وإعداده لإنشاء لعبة تخمين الأرقام باستخدام لغة بايثون (Python) المهمة (5).

أُكمل - بالتعاون مع أفراد مجموعتي - تنفيذ مشروع التصميم والتطوير للعبة تخمين الأرقام. وكنا قد انتهينا في الخطوات السابقة من عرض القائمة الرئيسية في اللعبة، بالاستجابة لما يُدخله اللاعب عن طريق عرض تعليمات اللعبة، أو كتابة رسالة بحسب الخيار المُدخّل، وتعديل البرنامج مُتملًا بتكرار عرض القائمة الرئيسية والخيارات. والآن سأعمل - ضمن المجموعة - على إضافة العنصر الرئيس في اللعبة، وهو إدخال البرنامج للرقم العشوائي. سأستخدم - مع أفراد مجموعتي - دالة (random) لتوليد رقم مخفي في بداية اللعبة، يتألف من سلسلة تحوي (4) أعداد مختلفة، ثمّ أُخرّن الأرقام المُمكنة (المُختلفة) في قائمة، وأعمل على تغيير ترتيبها باستخدام الدالة random.shuffle.

أنتحق - مع أفراد مجموعتي - من استخدام القوائم بصورة صحيحة، ومن تفعيل عملية توليد الأرقام، ثمّ أحتفظ بما نحبّ - ضمن المجموعة - في ملف حتى يُمكن لأفراد المجموعة التعديل عليه في الخطوات القادمة.

الخوارزميات والبرمجة (Algorithms and Programing)

الوحدة 1

```
import "../index.css";  
import { ReactComponent as ArrowIcon } from "../assets/icons/bolt.svg";  
import { ReactComponent as BoltIcon } from "../assets/icons/right-arrow.svg";  
import { ReactComponent as RightArrowIcon } from "../assets/icons/right-arrow.svg";  
  
import React, { useState, useEffect, useRef } from "react";  
import { CSSTransition } from "react-transition-group";
```

مُقدِّمة

في هذه الوحدة، يستكمل الطلبة ما بدؤوه في صفوف سابقة حول لغات البرمجة عالية ومنخفضة المستوى. سيتعرفون إلى المترجمات، والمفسرات، وبيئات التطوير المتكاملة، مع التركيز على لغة بايثون كمثال على اللغات المُفسرة عالية المستوى. وسيكتسب الطلبة معرفة أساسيات لغة بايثون، بما في ذلك المتغيرات، وأنواع البيانات، والعمليات الحسابية، وجمل الإدخال والطباعة. وسيعمل الطلبة على مشروعات صغيرة تشمل إنشاء برامج قصيرة، وألعاب بسيطة تتضمن استخدام متغيرات وعمليات حسابية؛ مما يعزز فهمهم للغة البرمجة وتطبيقاتها العملية.

مصفوفة نتائج التعلم

نتائج تعلم الصف العاشر	نتائج تعلم الصف الحادي عشر	نتائج تعلم الصف الثاني عشر
■ يتعامل مع برنامج HTML.	■ يتعرف لغات البرمجة.	■ يستخدم المخططات (Flow charts) شبيهة الكود (Pseudocode) لتمثيل حلول مشكلات مركبة لتمثيل حلول مشكلات مركبة باستخدام الخوارزميات
■ ينشئ مواقع باستخدام لغة HTML.	■ يتعامل مع البرنامج.	■ يقيم الخوارزميات بناء على وضوحها ودقتها وفعاليتها في حل المشكلة
■ يجزئ المشكلة إلى مشكلات أصغر؛ ليتمكن من برمجتها وحلها.	■ يتعرف قواعد كتابة الجملة البرمجية.	■ يقارن هياكل تراكيب البيانات المناسبة لعمل خوارزمية معينة في حل مشكلة محدده
■ يستخدم أوامر لتنسيق صفحة الويب HTML.	■ يتعامل مع المتغيرات بلغة بايثون.	■ يختار هياكل تراكيب البيانات المناسبة لعمل خوارزمية معينة في حل مشكلة محدده
■ يعدل، ويعيد تشكيل أجزاء من صفحات معدة مسبقاً ضمن صفحة الويب المراد تصميمها لتوفير الوقت والجهد.	■ يتعرف التعبيرات الحسابية.	■ يوضح كيفية عمل الخوارزميات الراجعة
■ يتواصل مع الزملاء لتطوير البرنامج باستخدام أسلوب التطوير المتكرر.	■ ينشئ متغيرات مسميات واضحة وملائمة لطبيعة البرنامج لتمثيل أنواع مختلفة من البيانات، وتنفيذ العمليات عليها.	■ يبني حلولاً برمجية باستخدام جمل برمجية معدة مسبقاً (روتين فرعي Modules) أو وحدات أو كائنات (Objects).

نتائج تعلّم الصف العاشر	نتائج تعلّم الصف الحادي عشر	نتائج تعلّم الصف الثاني عشر
يراعي حقوق الملكية الفكرية عند تطوير البرنامج.	يستخدم القوائم Lists لتخزين مجموعة من القيم في متغير واحد.	ينشئ أدوات حاسوبية باستخدام مجموعة من الدّوالّ والروتين الفرعي والمكتبات البرمجية.
يفحص تصميم صفحة الويب ويتبع الأخطاء.	يفسر سبب استخدام جملة تحكم برمجية محددة من حيث تحسين مقروئية البرنامج، وتحسين أداء البرنامج.	يحدد الأنماط القابلة للتعميم والتي يمكن تطبيقها على الحل.
يصحح الأخطاء في البرنامج، ويتأكد من صحة النتائج.	يكتب العلاقات الحسابية والمنطقية باستخدام لغة البرمجة بايثون.	يعيد استخدام برامج معدة مسبقاً في برنامج آخر لحل مشكلة محددة.
يتقمص مختلف الأدوار والعمل ضمن فريق وتحت إشراف المعلم لتصميم البرنامج وتطويره ومراجعته.	يكتب العبارات الحسابية والعبارات المنطقية باستخدام لغة البرمجة بايثون.	يصمم الأنظمة باستخدام طرائق مختلفة لدورة حياة تطوير الأنظمة.
يتواصل مع الزملاء لاتخاذ القرارات الخاصة بالبرنامج.	يجزئ المشكلة إلى أجزاء عن طريق التحليل المنظم وكتابتها على شكل جمل برمجية أو روتين فرعي أو وحدات (Modules) أو كائنات (Objects).	يطور الأنظمة باستخدام طرائق مختلفة لدورة حياة تطوير الأنظمة.
	يوثق البرنامج خلال مراحل التصميم والتطوير.	يطور برامج تعمل على مختلف الأجهزة.
	يستخدم أدوات التوثيق مثل المخططات (Flowcharts) وأدوات العرض مثل (MS Powerpoint).	يستخدم مجموعة من حالات الفحص للتأكد من أن البرنامج يحقق النتائج المرجوة من بناء على مواصفات البرنامج في مرحلة التصميم.
		يعدل برنامجاً موجوداً لإضافة وظائف إضافية.
		يناقش التأثيرات المقصودة وغير المقصودة للتعديل على البرنامج.
		يقيم الصفات الرئيسة للبرنامج.
		يقيم التعليمات البرمجية.

مصفوفة نتائج التعلم ومؤشرات الأداء حسب الوحدة

المحور	المعيار	نتائج التعلم	مؤشرات الأداء
■ مقدمة إلى لغات البرمجة عالية المستوى	■ التعرف إلى مفهوم لغات البرمجة. ■ التعامل مع البرنامج (جديد، فتح، حفظ، تنفيذ، نشر). ■ التعرف إلى قواعد كتابة الجملة البرمجية.	■ يتعرف لغات البرمجة.	■ يعرف لغة البرمجة. ■ يعدد لغات برمجة عالية المستوى. ■ يعدد لغات برمجة مختلفة ذات ميزات مختلفة. ■ يقارن بين لغة البرمجة المبنية على الكتل، ولغة البرمجة النصية (Blocks versus Text). ■ يقارن لغات البرمجة ذات المستوى العالي بلغات البرمجة ذات المستوى المنخفض (High-level versus Low-level). ■ يقارن بين لغات البرمجة المتعددة، ويناقش كيف تتغير لطبيعة تغير مميزات اللغة البرمجية.
	■ يتعامل مع البرنامج.		■ يعرف النموذج الأولي للبرنامج. ■ ينشئ نموذجاً أولياً لبرنامج؛ بناء على خوارزمية موضوعة مسبقاً لحل مشكلة معينة. ■ ينشئ نموذجاً أولياً لبرنامج؛ بناء على أفكار تخدم المجتمع وبيئته المحيطة.
	■ يتعرف قواعد كتابة الجملة البرمجية.		■ يبين قواعد كتابة الجملة البرمجية بلغة بايثون.

المحور	المعيار	نتائج التعلم	مؤشرات الأداء
■ المتغيرات	■ التعامل مع المتغيرات.	■ يتعامل مع المتغيرات بلغة بايثون.	■ يتعرف مفهوم المتغيرات.
	■ التعامل مع التعابير الحسابية.		■ يتعرف عناصر لغة بايثون (الثوابت، والمتغيرات، والرموز، والتعابير والعلاقات).
	■ تعريف المتغيرات واستخدامها لتمثيل البيانات ودعم العمليات الحاسوبية.		■ يميز الثوابت العددية والثوابت الرمزية.
	■ استخدام القوائم.		■ يميز مجموعات الرموز (ارقام، حروف، رموز خاصة).
		■ يتعرف التعابير الحسابية.	■ يميز المتغيرات العددية والمتغيرات.
			■ يتعرف التعبير الحسابي.
			■ يتعرف أولويات العمليات الحسابية.
			■ يجري العمليات الحسابية على التعابير الحسابية.
		■ ينشئ متغيرات مسميات واضحة وملائمة لطبيعة البرنامج لتمثيل أنواع مختلفة من البيانات وتنفيذ العمليات عليها.	■ ينشئ البرامج التي تستخدم متغيرين أو أكثر لتمثيل البيانات وتخزينها وتعديلها.
		■ يستخدم القوائم Lists لتخزين مجموعة من القيم في متغير واحد.	■ يسمي المتغيرات بأسماء مناسبة.
			■ يعرف المتغير "قائمه".
			■ يحدد كيفية تجميع مجموعة القيم في قائمة.
			■ يشرح أنواع القوائم المختلفة مثل المتسلسلة والمتغيرة.
			■ يستخدم لغة البرمجة بايثون لتمثيل مختلف أنواع القوائم؛ لتخزين مختلف أنواع البيانات.

المحور	المعيار	نتائج التعلم	مؤشرات الأداء
■ التحكم	■ تحسين مقروئية البرنامج ومدى وضوحه ليسهل على الآخرين تتبعه وفهمه والتعديل عليه. ■ كتابة التعابير الحسابية والتعابير المنطقية.	■ يفسر سبب استخدام جملة تحكم برمجية محددة من حيث تحسين مقروئية البرنامج، وتحسين أدائه.	■ يعرف مفهوم المقروئية للبرنامج. ■ يعرف مفهوم أداء البرنامج. ■ يحدد التركيب البرمجي الملائم لحل مشكلة محددة. ■ يستخدم لغة البرمجة بايثون لكتابة جمل التحكم. ■ يفسر سبب استخدام تركيب تحكم.
		■ يكتب العلاقات الحسابية والمنطقية باستخدام لغة البرمجة بايثون.	■ يميز التعابير الحسابية والتعابير المنطقية. ■ يميز العلاقات الحسابية (+, -, *, /, ^) والعمليات المنطقية (<, >, =, <=, >=, <>)
		■ يكتب العبارات الحسابية والعبارات المنطقية باستخدام لغة البرمجة بايثون.	■ يكتب عبارات حسابية باستخدام لغة بايثون. ■ يكتب عبارات منطقية باستخدام لغة بايثون. ■ يكتب جملاً شرطية مركبة و مترابطة من خلال المعاملات المنطقية مثل (AND, OR, and NOT) باستخدام لغة البرمجة بايثون. ■ يستخدم لغة البرمجة بايثون لينفذ مجموعة أوامر. ■ يجد ناتج تعابير حسابية. ■ يجد ناتج تعابير منطقية. ■ يكتب جملاً شرطية مركبة و مترابطة عن طريق المعاملات المنطقية مثل (AND, OR, and NOT) باستخدام لغة البرمجة بايثون.

المحور	المعيار	نتائج التعلم	مؤشرات الأداء
			■ يستخدم لغة البرمجة بايثون لينفذ مجموعة أوامر. ■ يجد ناتج تعابير حسابية. ■ يجد ناتج تعابير منطقية.
■ النمطية	■ تطوير مهارة التفكير التحليلي عن طريق تجزئة المشكلة إلى مشكلات أصغر لتصميم وتطوير برامج لها بالتعاون مع الزملاء.	■ يجزئ المشكلة إلى أجزاء عن طريق التحليل المنظم، ويكتبها على شكل جمل برمجية أو روتين فرعي.	■ يجزئ المشكلة إلى مشكلة أصغر لتصميمها وبرمجتها كل واحدة على حدة. ■ يحدد الطريقة المثلى لتمثيل أجزاء المشكلة على شكل جمل برمجية أو روتين فرعي أو وحدات أو كائنات. ■ يستخدم لغة البرمجة بايثون؛ ليستدعي روتيناً فرعياً جاهزاً عن طريق وقوع حدث محدد.
■ تطوير البرنامج	■ توثيق البرنامج وعرض مكوناته	■ يوثق البرنامج خلال مراحل التصميم والتطوير.	■ يوثق كتابة مكونات البرنامج وآلية عمله. ■ يكتب ملاحظات توضيحية داخل البرنامج. ■ يحدد الإطار العام للمشروع ومهامه. ■ يتواصل مع مستخدمي البرنامج والزملاء في فريق التطوير عن طريق المنتجات الحاسوبية الخاصة بالمشروع.
		■ يستخدم أدوات التوثيق مثل المخططات (Flowcharts) وأدوات العرض مثل (MS Powerpoint).	■ ينشئ عروضاً تقديمية، ويستخدم المخططات (Flowcharts) لوصف مراحل تصميم البرنامج وتطويره، وأهم مكوناته. ■ يعرض لمستخدمي البرنامج والزملاء في فريق التطوير التطورات على برنامج باستخدام المخططات وتطبيق العروض التقديمية (MS powerpoint).

مصفوفة نتائج التعلم ومؤشرات الأداء بحسب دروس الوحدة

الدرس	مؤشرات الأداء لكل درس	عدد الحصص المتوقعة	المفاهيم والمصطلحات
الدرس الأول: مقدمة في لغات البرمجة (Introduction to Programming Languages).	- يعرف لغة البرمجة. - يعدد لغات برمجة عالية المستوى. - يعدد لغات برمجة مختلفة ذات ميزات مختلفة. - يقارن بين لغة البرمجة المبنية على الكتل ولغة البرمجة النصية (Blocks versus Text). - يقارن لغات البرمجة ذات المستوى العالي بلغات البرمجة ذات المستوى المنخفض (High-Level versus Low-level). - يوضح العلاقة بين الخوارزميات والبرمجة. - يمثل البرامج بالخوارزميات ومخططات سير العمليات.	2	الخوارزميات (Algorithms). الخوارزمية شبه الرمزية (Pseudocode). مخططات سير العمل (Flowcharts)، لغة الآلة (Machine Language)، لغة التجميع (Assembly Language)، الكتل الرسومية (Graphical Blocks)، المترجم (Compiler)، المُفسر (Interpreter).
الدرس الثاني: أساسيات لغة البرمجة بايثون (Basics of Python Programming).	- يعرف النموذج الأولي للبرنامج. - ينشئ نموذجًا أوليًا لبرنامج بناء على خوارزمية موضوعة مسبقًا لحل مشكلة معينة. - ينشئ نموذجًا أوليًا لبرنامج بناء على أفكار تخدم المجتمع وبيئته المحيطة. - يعرف مفهوم المتغيرات. - يتعرف عناصر لغة بايثون (الثوابت، المتغيرات، الرموز، التعبيرات والعلاقات).	3	البرمجة بالكائنات (Object-Oriented Programming)، تعدد المهام (Multitasking)، موجه الأوامر (Command Prompt)، محررات النصوص (Text Editors)، تمييز الصيغة (Syntax Highlighting)، التعليقات (Comments)، الكلمات المحجوزة (Reserved Words)، المُعرِّفات (Identifiers)، الرموز (Literals)، التراكيب البرمجية (Suites).

الدرس	مؤشرات الأداء لكل درس	عدد الحصص المتوقعة	المفاهيم والمصطلحات
	■ يميز الثوابت العددية والثوابت الرمزية. ■ يميز مجموعات الرموز (أرقام، حروف، رموز خاصة). ■ يميز المتغيرات العددية والمتغيرات. ■ يتعرف التعبير الحسابي. ■ يتعرف أولويات العمليات الحسابية. ■ يبين قواعد كتابة الجملة البرمجية بلغة بايثون. ■ يجري عمليات حسابية. ■ يكتب العلاقات والعبارات الحسابية والمنطقية باستخدام لغة البرمجة بايثون. ■ ينشئ البرامج التي تستخدم متغيرين أو أكثر لتمثيل البيانات وتخزينها وتعديلها. ■ يسمي المتغيرات بأسماء مناسبة.		المسافات الفارغة (Indentations)، تمييز حالة الحرف (Case Sensitivity)، أسبقية العوامل (Operators Precedence)، الترابط (Associativity).
الدرس الثالث: الجمل الشرطية (Conditional Statements).	■ يكتب جملاً شرطية مركبة ومترابطة باستخدام المُعاملات المنطقية (مثل: And، Or، Not) في لغة البرمجة بايثون (Python). ■ يستخدم لغة البرمجة بايثون (Python) في إنشاء برامج تتضمن جملاً شرطية، ويتتبع ناتجها ويُنفذها.	4	■ الجمل الشرطية (Conditional Statements).

الدرس	مُؤشرات الأداء لكل درس	عدد الحصص المُتوقع	المفاهيم والمصطلحات
الدرس الرابع: الحلقات (Loops).	- يكتب جمل التحكم باستخدام الحلقات (مثل: For، و While) في برمجة بايثون (Python). - يستخدم أكثر الهياكل البرمجية (مثل: الحلقات، والجمل الشرطية) مناسبة لحلّ مشكلات مُعيّنة بكفاءة. - يكتب برامج مُتكاملة على نحوٍ يُسهّل على الآخرين قراءتها وفهمها.	4	الحلقات (Loops)، جمل التحكم (Control Statements)، التهيئة (Initialization)، التعليمات (Statements)، الزيادة (Increment)، النقصان (Decrement)، العنصر / المُتغيّر (Element)، المصفوفة (Sequence)، النطاق (Range).
الدرس الخامس: القوائم (Lists).	- يعرف مفهوم المُتغيّر (قائمة) ويبيّن استخداماته في البرمجة. - يوضح أنواع القوائم (المُتسلسلة، والمُتغيّرة، والمُركّبة)، ويُنفّذ عمليات مختلفة فيها، مثل: الإضافة، والحذف. - يستخدم لغة البرمجة بايثون لتمثيل مختلف أنواع القوائم لتخزين أنواع البيانات المختلفة. - يُحدّد كيف يُمكن تجميع مجموعة القيم في قائمة واحدة. - يستخدم الدّوال البرمجية الجاهزة في لغة البرمجة بايثون (Python) لمعالجة القوائم وإجراء بعض العمليات الأساسية فيها.	4	القائمة (List)، سلسلة الحروف (String)، موقع العنصر في القائمة (أو الحرف في السلسلة) (Index)، ضمّ القوائم أو السلاسل (Concatenation)، القابلية للتغيير (Mutability)، القائمة المُركّبة (Nested List)، المصفوفة ثنائية الأبعاد (2D Matrix).

الدرس	مؤشرات الأداء لكل درس	عدد الحصص المتوقعة	المفاهيم والمصطلحات
الدرس السادس: الدوال البرمجية (Functions).	- يعرّف المقصود بالوحدات. - يجزئ المشكلة إلى أجزاء صغيرة، ويعمل على تصميم كل جزء منها وبرمجته. - يحدد الطريقة الفضلى لتمثيل أجزاء المشكلة في صورة جمل برمجية، أو روتين فرعي، أو وحدات، أو كائنات. - يستعمل لغة البرمجة بايثون (Python) لاستدعاء روتين فرعي جاهز بناءً على وقوع حدث مُحدد.	3	الدالة البرمجية (Function)، الوحدة البرمجية (Module)، استيراد الوحدات (Importing Modules)، مدى المتغير (Variables Scope)، المدخلات أو مُعاملات الدالة (Function Parameters)، التوثيق (Documentation)، سلاسل التوثيق (Docstrings).
		20 حصة	

الوحدة 1

نظرة عامة على الوحدة

أناقش الطلبة في الفكرة العامة للوحدة، ثم أوضح لهم الأهداف التي يُتوقع منهم تحقيقها بعد الانتهاء من دراسة الوحدة. وأستعرض مع الطلبة الأدوات الرقمية والبرامج التي يُتوقع منهم استخدامها وتطوير مهاراتهم فيها في أثناء دراسة الوحدة، وأبين لهم كيف يمكن استخدام هذه الأدوات في أنشطة ومهام متعددة تُسهم في صقل مهاراتهم وزيادة خبراتهم.

سياق التعلّم والمهام الأصيلة

سياق التعلّم:

في هذه الوحدة، سيشارك الطلبة في عملية تعلّم مبنية على المشروعات، وسينخرطون بدور الناشئين في عالم البرمجة؛ إذ سيطلب منهم بناء لعبة تخمين باستخدام لغة البرمجة بايثون. وسينخرط الطلبة في رحلة تعلّم تشمل تثبيت البرنامج على أجهزتهم، ثم تعلّم أساسيات البرمجة باستخدام بايثون، والجمل الشرطية، والقوائم، والحلقات، والدوال البرمجية، والروتينات الفرعية. وفي أثناء رحلة تعلّمهم، سيعمل الطلبة على العديد من الأمثلة لكتابة مقاطع برمجية، وتتبع مقاطع برمجية جاهزة، وتنفيذها باستخدام بايثون، واكتشاف الأخطاء إن وجدت واقتراح التعديلات. وسيقوم الطلبة بتوظيف كل ما سبق لتطوير اللعبة على نحو متكرر، مع تحسينها تبعًا لتطور معارفهم ومهاراتهم في البرمجة.

المشروع:

تصميم برنامج وإعداده لإنشاء لعبة تخمين الأرقام باستخدام لغة بايثون (Python).

مُنْتَج التعلّم:

لعبة لتخمين الأرقام باستخدام لغة بايثون (Python).

الوحدة 1

الخوارزميات والبرمجة (Algorithms and Programing)

نظرة عامة على الوحدة

سأتعرّف في هذه الوحدة لغات البرمجة عالية المستوى، وأقارنها بلغات البرمجة منخفضة المستوى. كذلك سأتعرف كلاً من المترجمات، والمُفسّرات، وبيئات التطوير المتكاملة، بما في ذلك لغة بايثون (Python) بوصفها مثالاً على اللغات المُفسّرة عالية المستوى، التي تُبَيّن كيفية استخدام بيئة العمل البرمجية البسيطة في إنشاء البرامج وتشغيلها. سأتعرف في هذه الوحدة أيضًا أساسيات اللغة من متغيرات، وأنواع البيانات، والعمليات الحسابية، وبعض جمل الإدخال، والطباعة؛ ما يُمكنني - في نهاية الوحدة - من إنشاء برامج قصيرة، تحوي أكثر من مُتغيّر وعمليات حسابية بسيطة، وذلك باستخدام لغة بايثون (Python).

يُتَوَقَّع مِنِّي في نهاية الوحدة أن أكون قادرًا على:

- شرح ماهية لغات البرمجة، وبيان أهميتها في تطوير البرمجيات.
- تطوير برنامج بسيط باستخدام لغة بايثون (Python)؛ لحلّ مشكلة مُعيّنة، بناءً على الخوارزميات، أو الأفكار التي تخدم المجتمع.
- توضيح قواعد الكتابة الصحيحة للشيفرة البرمجية في لغة بايثون (Python).
- تعريف المُتغيّرات في لغة بايثون (Python)، واستعمالها لتخزين البيانات وإجراء العمليات عليها.
- كتابة التعابير والعلاقات الحسابية والمنطقية واستخدامها في لغة بايثون (Python).
- كتابة الجمل الشرطية والحلقات، واستعمالها لتنفيذ عمليات مُتكرّرة واتخاذ قرارات منطقية.
- إنشاء القوائم واستخدامها في لغة بايثون (Python)؛ لإدارة مجموعات البيانات.
- تحليل المشكلة وتقسيمها إلى أجزاء صغيرة؛ ما يتيح التعامل معها بفاعلية.
- توثيق الشيفرة البرمجية باستخدام المُخطّطات وأدوات العرض.

استراتيجية التعليم والتعلم

في سياق التعلم لهذه الوحدة، ستطبق استراتيجية تعليم وتعلم تشمل جوانب عدة: التعلم القائم على المشروع: سيقوم الطلبة بتصميم برنامج وإعداده لإنشاء لعبة تخمين الأرقام باستخدام لغة بايثون (Python) كجزء من مشروع تعليمي، ما يمكنهم من اكتساب مهارات عملية من خلال التعلم بالممارسة. التعلم التعاوني النشط: سيعمل الطلبة على نحو جماعي في فرق؛ مما يعزز التعاون والمشاركة الفعالة بينهم، ويمكنهم من تبادل الأفكار، والإسهام في إنجاز المشروع على نحو جماعي.

أساليب مفيدة لتعليم البرمجة

البرمجة الحية (Live Coding)

الهدف من وحدة البرمجة بلغة البايثون هو تعليم الطلبة "كيفية" البرمجة وليس تعليمهم "برامج" جاهزة؛ لذلك فإن عرض الكود وشرحه (إما بكتابته على

اللوحة ثم شرحه، أو باستخدام شرائح العرض) يضيع كثيراً من الفوائد التعليمية المتعلقة بـ"كيفية" البرمجة.

البرمجة الحية هي أن يقوم المعلم بكتابة البرنامج وتشغيله أمام الطلبة؛ مما يسمح بتحقيق الآتي:

- رؤية الطلبة للطريقة التي يفكر فيها المعلم عند كتابة البرامج، وللطريقة التي يتعامل فيها المعلم مع الأخطاء البرمجية عند حدوثها (يحصل تعليم ضمني وأحياناً غير مقصود للعديد من المهارات).
- سؤال الطلبة أسئلة "ماذا لو؟"؛ إذ يستطيع المعلم تعديل البرنامج وتشغيله مباشرة للإجابة عن مثل هذه الأسئلة.
- شرح كل سطر يقوم المعلم بكتابته والتأكد من فهم الطلبة له (بدلاً من عرض كامل الكود والحديث عنه بشكل عام).
- استخدام بعض أساليب التعليم الفعالة (بناءً على الأدلة البحثية) التي سنذكرها بعضها بعد قليل (مثل استخدام التوقعات، والتعليم بالأخطاء، وتوسيم الأهداف الفرعية).



مُنتجات التعلم (Learning Products)

تصميم برنامج وإعداده لإنشاء لعبة تخمين الأرقام باستخدام لغة بايثون (Python).

أختار مع أفراد مجموعتي أحد المشروعات الآتية لتنفيذه في نهاية الوحدة:
المشروع الأول: تصميم برنامج وإعداده لإنشاء لعبة الألغاز باستخدام لغة بايثون (Python).
المشروع الثاني: تصميم نموذج أولي وإعداده لإنشاء برنامج، بناءً على أفكار تخدم المجتمع والبيئة المحيطة.
المشروع الثالث: تصميم نموذج أولي وإعداده لإنشاء برنامج يساعد الطالب على إدارة المهام المنوطة به.

الأدوات والبرامج (Digital Tools and Programs)

Python, IDLE, Command Line, (draw.io), google slides

المهارات الرقمية (Digital Skills): تصميم الخوارزميات، التفكير الحاسوبي، البرمجة، حل المشكلات البرمجية.

فهرس الوحدة

- الدرس الأول: مُقدمة في لغات البرمجة (Introduction to Programming Languages).
- الدرس الثاني: أساسيات لغة البرمجة بايثون (Basics of Python Programming).
- الدرس الثالث: الجمل الشرطية (Conditional Statements).
- الدرس الرابع: الحلقات (Loops).
- الدرس الخامس: القوائم (Lists).
- الدرس السادس: الدوال البرمجية (Functions).

■ ولكن من المهم أن يتتبع المعلم للآتي:

■ من المفترض أن تكون سرعة الطباعة متوافقة مع سرعة حديث المعلم وشرحه، فالطباعة البطيئة جداً قد تسبب مللاً للطلبة وضياعاً لوقت الحصة؛ لذلك من المهم للمعلم التدرب على الطباعة السريعة إن كان بطيئاً في الطباعة.

■ يجب على المعلم تجريب كتابة البرنامج وتشغيله قبل الحصة؛ حتى لا تحصل مفاجآت غير متوقعة قد تربكه في أثناء الشرح، أو تضيع الوقت في محاولة التعامل معها.

■ يجب على المعلم عدم إضاعة الوقت في كتابة أجزاء من البرنامج غير مهمة للمفهوم الذي يُشرح (ولكنها مهمة لتشغيل البرنامج)؛ إذ يمكن للمعلم كتابة هذه الأجزاء قبل الحصة، واستكمال الأجزاء التي تخص موضوع الحصة أمام الطلبة على نحو حي.

■ من المفترض أن يشرك المعلم الطلبة في كتابة الكود على نحو ما، مثلاً عن طريق أخذ اقتراحات منهم لكيفية كتابة سطر ما، أو طلب توقع نتيجة ما، أو اكتشاف خطأ في البرنامج، إلخ.

■ من المهم ألا يفرض المعلم في إشراك الطلبة في كتابة الكود؛ حتى لا تشتت الفكرة، ولا يضيع وقت أكثر من اللازم في شرح المفهوم.

استخدام التوقعات (Using Predictions)

يمكن للمعلم تنفيذ إحدى الطرائق الآتية:

■ الطريقة الأولى: تشغيل البرنامج ثم شرح النتيجة أو نقاش الطلبة حولها.

■ الطريقة الثانية: أخذ توقعات من الطلبة حول نتيجة البرنامج، ثم تشغيله، ثم نقاش الطلبة حول صحة أو خطأ توقعاتهم.

وقد تكون الطريقة الثانية أكثر فاعلية في تثبيت المعلومة، خاصة إذا اقترنت بأساليب تعليمية أخرى مثل أن يطلب المعلم إلى الطلبة بعد تشغيل البرنامج أن يقوموا بمناقشة بعضهم بعضاً لدقائق معدودة حول سبب خطأ، أو صحة توقعهم، وذلك قبل شرح السبب لهم.

ملحوظة: يمكن استخدام هذا الأسلوب خلال البرمجة الحية أو كتمرين مستقل.

ومن المهم أن يصرّ المعلم على أن يقوم جميع الطلبة بالتوقع؛ لأن الفائدة تحصل عند مرور الطلبة بالمراحل الآتية:

1. التفكير في معنى البرنامج على نحو ناقد.
 2. الشعور بالفرح (إذا كان التوقع صحيحاً) أو بالاستغراب (إذا كان التوقع غير صحيح).
 3. التفكير الناقد في سبب إخفاق التوقع، أو لتكوين حجة لشرح سبب صحة التوقع.
- وهناك أكثر من طريقة لإشراك جميع طلبة الصف بالتوقع:

- أن يطلب المعلم إلى الطلبة كتابة توقعهم على ورقة (والاحتفاظ بها).
- أن يطلب المعلم سماع بعض التوقعات، ويكتبها على اللوح، ثم يجري تصويتاً؛ (برفع الأيدي) لمعرفة كم من الطلبة يتفق مع كل من هذه التوقعات.
- إذا كانت نتيجة البرنامج محدودة بخيارات معينة، يمكن للمعلم أن يقوم مباشرة بالتصويت على هذه الخيارات برفع الأيدي (مثلاً: من يتوقع أن يعمل البرنامج من دون مشكلات؟ من يتوقع أن يتحطم البرنامج؟)

البرمجة مع قرين (Pair Programming)

يمكن استخدام "البرمجة مع قرين" في الحصص العملية (التي يكون الجزء الأكبر منها مخصصاً لتطبيق الطلبة لتمارين برمجية على جهاز الحاسوب)، وتتم كما يأتي:

- يُقسّم الطلبة لفرق ثنائية.
- يستخدم كل فريق جهاز حاسوب واحد فقط لحل التمارين.
- يتناوب أعضاء الفريق بأخذ دور "القائد" ودور "المراقب" كل 10 دقائق أو 20 دقيقة تقريباً؛ إذ يقوم القائد بطباعة البرنامج على الحاسوب؛ في حين يقوم المراقب بالتأكد من صحة ما يكتبه القائد.
- وحتى يضمن المعلم تناوب الأدوار بالشكل الصحيح، يستخدم مؤقتاً أو ساعة لمراقبة الوقت وإعلان "وقت تبديل الأدوار" بصوت مرتفع عند مرور الوقت المحدد (10 دقائق مثلاً)، ويقوم بالمرور على الطلبة للتأكد

ملحوظة: يضمن تناوب الأدوار بهذا الشكل أن يحصل كل عضو من أعضاء الفريق على فرصة لممارسة مهارة الكتابة (بدور القائد) ومهارة التتبع (بدور المراقب) من دون أن يسيطر أحد الطلبة على الآخر (بحسب الشخصية أو القدرة الأكاديمية).

هناك ثلاث فوائد مهمة لاستخدام هذه الطريقة مقارنة بتطبيق كل طالب على جهاز على نحو منفصل:

- زيادة فرص التعلم عن طريق مناقشة الأقران للوصول لحل (Social Constructivism) أو عن طريق تعليمهم؛ إذ إن الطالب الأقوى علمياً في الفريق، قد يجد نفسه متقمصاً دور المعلم (Learning by Teaching).
- زيادة فرصة إتمام الطلبة للمهمة البرمجية بنجاح؛ مما يزيد من رغبتهم في تعلم البرمجة بشكل عام (أحد ركائز Expectancy-Value Theory of Motivation هي الثقة بالقدرة على إتمام المهام).
- مساعدة المعلم بتقليل عدد الأسئلة التي يجب أن يجيب عنها، والمشكلات البرمجية التي يجب أن يتعامل معها في الصف.

توسيم الأهداف الفرعية (Subgoal Labeling)

لكل برنامج هدف رئيس، قد تتفرع منه أهداف فرعية، ومن المفيد عند تعليم البرمجة توسيم هذه الأهداف الفرعية بأي من الطرائق الآتية:

- أن يُعطى الطلبة البرنامج ويطلب إليهم (كتمرين) إضافة تعليقات (وسوم) قبل كل جزء من أجزاء الكود؛ لشرح الهدف الفرعي لهذا الجزء من الكود (لاحظ بأن المطلوب ليس شرح كل "سطر" وإنما شرح الأجزاء التي ربما تكون مكونة من مجموعة من الأسطر).
- أن تكون الأمثلة المعطاة من المعلم للطلبة موسومة أصلاً بتعليقات توضح الهدف الفرعي لكل جزء من البرنامج، بحيث يستخدم الطلبة هذه التعليقات لفهم الكود عند الدراسة والمراجعة.
- أن يبدأ المعلم في البرمجة الحية؛ بتحديد الأهداف الفرعية وكتابتها على شكل تعليقات مرقمة في البرنامج، ثم يرمج كل جزء (تحت التعليق المكتوب سابقاً) بحيث يكون الهدف الفرعي الذي يرمجه المعلم واضحاً للطلبة في ذلك الحين.

خُطّة تدريس موضوعات الوحدة بحسب منهجية التعلّم القائم على المشروع

البرنامج / المهارة	المنتج	الدرس
Google Slides Draw.io Python	- إعداد عرض تقديمي عن مشكلة ما (لعبة ما) باستخدام تطبيق العروض التقديمية (Google Slides)، وتضمين العرض شرحاً لهذه المشكلة (اللعبة)، وأسباب اختيارها، والحلّ المُقترح لها، والهدف منها، ووصف سيناريو اللعبة وقواعدها من البداية إلى النهاية. - إعداد مخطط سير العمليات (Flowcharts) لتصميم لعبة تخمين الأرقام باستخدام إحدى الأدوات الرقمية، مثل Draw.io، وذلك ضمن سياق تصميم لعبة تخمين الأرقام باستخدام برمجة بايثون (Python).	الدرس الأول: مُقدّمة في لغات البرمجة (Introduction to) Programming (Languages).
Python	فتح برنامج خاص بلعبة تخمين الأرقام في برمجة بايثون (Python)، والبَدْء بطباعة رسالة الترحيب وخيارات القائمة الرئيسة، ضمن سياق تصميم لعبة تخمين الأرقام باستخدام برمجة بايثون (Python).	الدرس الثاني: أساسيات لغة البرمجة بايثون (Basics of Python) (Programming).
Python	تعديل البرنامج بإضافة حلقات تكرارية تُمكن من عرض القائمة بصورة مُتكرّرة؛ حتّى يتمّ اختيار الخيار الصحيح، ضمن سياق تصميم لعبة تخمين الأرقام باستخدام برمجة بايثون (Python).	الدرس الثالث: الجمل الشرطية (Conditional) (Statements)
Python	تعديل البرنامج بإضافة حلقات تكرارية تُمكن من عرض القائمة بصورة مُتكرّرة؛ حتّى يتمّ اختيار الخيار الصحيح، ضمن سياق تصميم لعبة تخمين الأرقام باستخدام برمجة بايثون (Python).	الدرس الرابع: الحلقات (Loops).
Python	تعديل البرنامج باستخدام قوائم اللعبة في توليد الأرقام العشوائية، ضمن سياق تصميم لعبة تخمين الأرقام باستخدام برمجة بايثون (Python).	الدرس الخامس: القوائم (Lists).
Python	تنظيم المقطع البرمجي باستخدام الدّوالّ؛ لتسهيل القراءة وإعادة الاستخدام، ضمن سياق تصميم لعبة تخمين الأرقام باستخدام برمجة بايثون (Python).	الدرس السادس: الدّوالّ البرمجية (Functions).

مقدمة في لغات البرمجة (Introduction to Programming Languages)

عدد الحصص المقترحة:
حصتان

المصادر والمراجع:
كتاب الطالب، لغة البرمجة بايثون.

الأدوات والتجهيزات:
أجهزة حاسوب، جهاز للعرض.

أولاً: التهيئة

سياق التعلم:

أوضح للطلبة الفكرة الرئيسة من الدرس، وأضعه في قالب سياق التعلم، وأخبرهم أنهم سيكملون رحلة التعلم بوصفهم مبرمجين ناشئين كالتالي بدؤوا بها في الصفوف السابقة؛ إذ ستسنى لهم هذه المرة فرصة تصميم ألعاب، وكتابة برامج متنوعة، وسيبدؤون في هذا الدرس التعريف بالمشكلة (فكرة اللعبة) المختارة، وبيان هدفها، وتحديد الجمهور المستهدف فيها، ووصف سيناريو اللعبة، ممثلاً برسم مخطط لسير العمليات فيها، وصولاً إلى بناء تصور شامل عن العمل، ثم البدء بتجزئته للتمكن من إنجازه على مراحل عدة.

أوضح لهم منتج التعلم المتوقع إنجازه خلال هذا الدرس، وهو إعداد عرض تقديمي عن مشكلة ما (لعبة ما) باستخدام تطبيق العروض التقديمية (Google Slides)، وتضمين العرض شرحاً لهذه المشكلة (اللعبة)، وأسباب اختيارها، والحل المقترح لها، والهدف منها، ووصف سيناريو اللعبة وقواعدها من البداية إلى النهاية. وكذلك إعداد مخطط سير العمليات (Flowcharts) لتصميم لعبة تخمين الأرقام باستخدام إحدى الأدوات الرقمية، مثل Draw.io، وذلك ضمن سياق تصميم لعبة تخمين الأرقام باستخدام برمجة بايثون (Python).

الدرس الأول

مقدمة في لغات البرمجة (Introduction to Programming Languages)

الفكرة الرئيسة:

تعرف لغات البرمجة، وبيان تصنيفاتها (لغات البرمجة عالية المستوى، لغات البرمجة منخفضة المستوى، لغات البرمجة الكتلية، لغات البرمجة النصية)، والمقارنة بينها، فضلاً عن تعرف العلاقة بين الخوارزميات والبرمجة، وتمثيل البرامج بالخوارزميات ومخططات سير العمليات.

المفاهيم والمصطلحات:

الخوارزميات (Algorithms)، الخوارزمية شبه الرمزية (Pseudocode)، مخططات سير العمل (Flowcharts)، لغة الآلة (Machine Language)، لغة التجميع (Assembly Language)، الكتل الرسومية (Graphical Blocks)، المترجم (Compiler)، المُفسر (Interpreter).

نتائج التعلم (Learning Outcomes):

- أعرف المقصود بلغة البرمجة.
- أعد بعض لغات البرمجة التي تختلف في مزاياها ووظائفها.
- أفرق بين لغة البرمجة الكتلية ولغة البرمجة النصية.
- أفرق بين لغات البرمجة عالية المستوى ولغات البرمجة منخفضة المستوى.
- أوضح العلاقة بين الخوارزميات والبرمجة.
- أمثل البرامج بالخوارزميات ومخططات سير العمليات.

منتجات التعلم (Learning Products)

إعداد عرض تقديمي باستخدام google slides للمشكلة وأسباب اختيارها والحل المقترح/ اللعبة والهدف منها، ووصف سيناريو اللعبة وقواعد اللعبة منذ بدايتها حتى انتهائها، وإعداد مخطط سير عمل (flowcharts) لتصميم لعبة التخمين باستخدام إحدى الأدوات الرقمية مثل draw.io، ضمن سياق تصميم لعبة تخمين الأرقام باستخدام برمجية بايثون

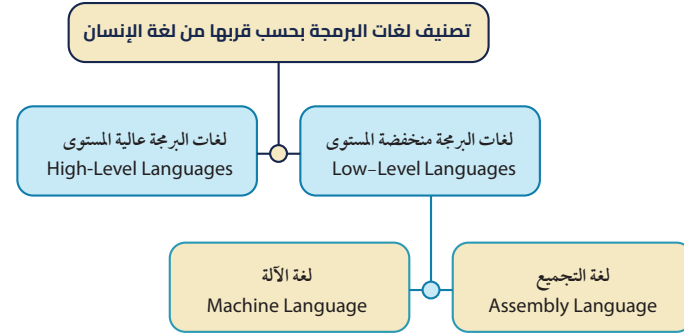
نشاط
تمهيدي

تعلّمتُ في صفوف سابقة كيف أُطوّر برامج باستخدام برمجة سكراتش (Scratch) كما تعلّمت تطوير مواقع الويب باستخدام لغة توصيف النص (HTML). - بالتعاون مع أفراد مجموعتي - أراجع ما تعلّمتُه عن هاتين البرمجتين، ثم أقارن بينهما من حيث الهدف، والاستخدام، والواجهة، وبيئة التطوير، وسهولة التعلّم، ثم أدوّن ما أتوصّل إليه في ملف مُعالج النصوص (Word).

لغة البرمجة (Programming Language)

تُعرف لغة البرمجة بأنّها مجموعة من الأوامر والتعليمات التي تُستخدم في كتابة البرامج والتطبيقات وفق قواعد مُحددة. وهي تُعدّ الأداة الرئيسة التي يستخدمها المبرمجون في التفاعل مع جهاز الحاسوب، وتوجيهه لتنفيذ مهام مُعيّنة. تُصنّف لغات البرمجة إلى أنواع عدّة، بناءً على وظائف كلّ منها، وتطبيقاتها، وطرائق معالجتها، وغير ذلك من المعايير والضوابط.

توجد جملة من المعايير والضوابط الرئيسة التي تُحكّم تصنيف لغات البرمجة، ويأتي في مقدّمة هذه المعايير والضوابط درجة قرب لغات البرمجة من اللغات الإنسانية، أنظر الشكل (1-1). وتأسيساً على ذلك، يُمكن تصنيف لغات البرمجة إلى نوعين، هما: لغات البرمجة عالية المستوى، ولغات البرمجة مُنخفضة المستوى.



الشكل (1-1): تصنيف لغات البرمجة تبعاً لقربها من لغات الإنسان.

نشاط تمهيدي

إجابة نموذجية

لغة توصيف النص (HTML)	برمجة سكراتش (Scratch)	الهدف
تعدّ لغة HTML أساساً لتطوير مواقع الويب. تستخدم لإنشاء هيكل صفحات الويب وتوصيفه، وهي جزء من مجموعة تقنيات الويب الأساسية إلى جانب CSS وJavaScript.	تهدف سكراتش إلى تعليم الأطفال والمبتدئين أساسيات البرمجة عن طريق بيئة مرئية وسهلة الاستخدام. ويمكن استخدامها لتطوير برامج تفاعلية وألعاب؛ لكنها ليست مخصصة لتطوير مواقع الويب	
تُستخدم HTML لإنشاء صفحات الويب عن طريق كتابة التعليمات البرمجية النصية. وتتطلب معرفة أساسيات بناء الصفحات والعناصر المختلفة مثل العناوين، والفقرات، والروابط، والصور.	يُستخدم سكراتش لإنشاء مشروعات برمجية تفاعلية عن طريق تجميع الكتل البرمجية. ولا يتطلب معرفة عميقة بالبرمجة، وهو مصمم ليكون بديهيًا وسهل الفهم.	الاستخدام

أولاً: لغات البرمجة منخفضة المستوى (Low-Level Languages):

تمتاز لغات البرمجة منخفضة المستوى بقربها من لغة الآلة، وبعدها عن لغة الإنسان ولذلك سميت بلغات البرمجة منخفضة المستوى. وهي تنقسم إلى قسمين، هما:

1. لغة الآلة (Machine Language): لغة برمجة تحتوي على أوامر وتعليمات يمكن لجهاز الحاسوب فهمها مباشرة ومعالجتها، خلافاً للإنسان الذي يصعب عليه فهمها. تمتاز هذه اللغة بأنها سريعة مقارنة بلغات البرمجة عالية المستوى.

2. لغة التجميع (Assembly Language): لغة تقوم على استخدام برنامج خاص يُسمى المجمع (Assembler)، ويعمل على تحويل الأوامر المكتوبة إلى لغة الآلة التي يفهمها جهاز الحاسوب. تمتاز هذه اللغة بأنها أسهل من لغة الآلة؛ نظراً إلى احتوائها على بعض مفردات اللغة الإنجليزية؛ ما يُسهّل قراءة برامجه وفهمها. غير أن تنفيذ البرامج المكتوبة بلغة التجميع يكون أبطأ مقارنة بلغة الآلة.

ثانياً: لغات البرمجة عالية المستوى (High-Level Languages):

تمتاز لغات البرمجة عالية المستوى بمواءمتها للغة التي يفهمها الإنسان؛ إذ تُستخدم في كتابة البرامج رموز ومفردات قريبة من تلك المتداولة في اللغة الإنجليزية. وقد سُميت هذه اللغات بهذا الاسم لقربها من لغة الإنسان وبعدها عن اللغة التي يفهمها جهاز الحاسوب؛ أي لغة الآلة. ومن ثم، فهي لا تعتمد على أنواع أجهزة الحاسوب في أداء وظائفها، وإنما صُممت على نحو يجعلها موائمة لجميع أجهزة الحاسوب، يَعضُ النظر عن نوع هذه الأجهزة وأنظمة تشغيلها.

من الأمثلة على لغات البرمجة عالية المستوى: لغة بايثون (Python)، ولغة جافا (Java)، ولغة سي ++ (C++)، ولغة سي شارب (C#).

كما يمكن تصنيف لغات البرمجة وفقاً لطريقة البرمجة إلى نوعين آخرين، هما:

1. لغات البرمجة الكتلية (Block-Based Programming Languages): لغات تُستخدم فيها الكتل الرسومية (Graphical Blocks) لتمثيل أجزاء البرامج بدلاً من النصوص، مثل لغة البرمجة سكراتش (Scratch).

2. لغات البرمجة النصية (Text-Based Programming Languages): لغات تُستخدم فيها النصوص لتمثيل أجزاء البرامج بدلاً من الكتل الرسومية، مثل لغة جافا سكريبت (JavaScript).

أقارن وأناقش: أقارن بين لغات البرمجة المختلفة، ثم أناقش زملائي / زميلاتي في الكيفية التي تتغير فيها طبيعة البرمجة تبعاً لتغير المزايا في كل لغة برمجة.



نشاط

لغة توصيف النص (HTML)	برمجة سكراتش (Scratch)	
لا تحتوي على واجهة مرئية، بل تُكتب التعليمات البرمجية في محرر نصوص. وتعتمد على المتصفح لعرض النتائج.	توافر واجهة مستخدم مرئية، تتضمن كتلاً برمجية يمكن سحبها وإفلاتها لتكوين البرامج. والواجهة مصممة لتكون جذابة وسهلة الاستخدام للمبتدئين.	الواجهة
يمكن استخدام أي محرر نصوص لكتابة تعليمات HTML، مثل Notepad أو Visual Studio Code. ويتطلب الأمر أحياناً إعدادات إضافية مثل خادم ويب لعرض الصفحات على نحو محلي.	تتضمن بيئة تطوير متكاملة وسهلة الاستخدام، وتتضمن أيضاً مكتبة من الكائنات والأصوات والخلفيات التي يمكن استخدامها في المشروعات. لا يحتاج المستخدم إلى أي إعدادات خاصة.	بيئة التطوير
تعدّ HTML سهلة التعلم نسبياً، لكن يتطلب الأمر معرفة ببعض المفاهيم الأساسية في بناء صفحات الويب واستخدام العلامات المختلفة.	تعدّ سكراتش من أسهل أدوات البرمجة للمبتدئين؛ إذ لا تتطلب معرفة مسبقة بالبرمجة. ويتم تعليم المفاهيم البرمجية الأساسية عن طريق التجربة العملية والمرئية.	سهولة التعلم
مناسبة للمستخدمين الذين يرغبون في تطوير مواقع الويب وفهم كيفية بناء هيكل الصفحات الإلكترونية.	مناسبة جداً للمبتدئين والأطفال الذين يرغبون في تعلّم أساسيات البرمجة بطريقة ممتعة وتفاعلية.	نتائج وتوصيات

وجه المقارنة	لغة البرمجة عالية المستوى	لغة البرمجة منخفضة المستوى
تعريف	هي لغات برمجة تستخدم جملاً وأوامر قريبة من اللغة البشرية؛ لتسهيل عملية البرمجة. ومن أمثلتها بايثون (Python)، جافا (Java)، وسي (C++).	هي لغات برمجة أقرب إلى لغة الآلة، وتتطلب معرفة تفصيلية بعتاد الحاسوب. من أمثلتها لغة التجميع (Assembly) واللغة الثنائية (Binary).
الهدف	تسهيل عملية البرمجة وتقليل تعقيد كتابة الشيفرة البرمجية.	تقديم تحكم دقيق بعتاد الحاسوب وتحسين الأداء.
الاستخدام	تطوير التطبيقات المكتبية، وتطبيقات الويب، والألعاب، والذكاء الاصطناعي، وتحليل البيانات، وغير ذلك.	تطوير أنظمة التشغيل، وبرامج التحكم بالأجهزة، والتطبيقات التي تتطلب أداءً عاليًا مثل الألعاب وأنظمة الزمن الحقيقي.
سهولة التعلم	تتطلب معرفة أقل بعتاد الحاسوب، وتكون أقرب للغة البشرية؛ مما يجعلها أسهل للتعلم.	تتطلب معرفة دقيقة بعتاد الحاسوب وتفاصيل تنفيذ الأوامر؛ مما يجعلها أصعب للتعلم.
الواجهة	تعتمد على محررات النصوص أو بيئات التطوير المتكاملة (IDEs) مثل Visual Studio، PyCharm، Eclipse.	تعتمد على محررات النصوص التي تدعم كتابة الشيفرة البرمجية بلغة الآلة أو التجميع.
بيئة التطوير	توفر بيئات تطوير متكاملة تحتوي على أدوات تسهل كتابة، وتصحيح، وتنفيذ الشيفرة البرمجية.	غالبًا ما تتطلب أدوات خاصة لتجميع الشيفرة البرمجية وربطها وتنفيذها على العتاد الفعلي.
المزايا	سهولة الكتابة والفهم. سرعة التطوير بسبب بساطة الشيفرة. توافر مكتبات وأطر عمل جاهزة، تسهل العديد من المهام.	أداء عالٍ جدًا وتحكم دقيق بموارد النظام. تتيح الاستفادة القصوى من قدرات العتاد.
العيوب	قد تكون أبطأ في التنفيذ مقارنة باللغات منخفضة المستوى. لا توفر مستوى التحكم نفسه في موارد النظام، كما في اللغات منخفضة المستوى.	صعوبة الكتابة والفهم؛ مما يزيد من احتمال وقوع الأخطاء. تتطلب وقتًا أطول لتطوير البرامج وصيانتها.
الأداء	عادة ما تكون أبطأ من اللغات منخفضة المستوى بسبب مستوى التجريد العالي.	تتميز بأداء عالٍ جدًا نظرًا لقربها من لغة الآلة.
التحكم في الموارد	تقدم مستوى أقل من التحكم في موارد النظام مقارنة باللغات منخفضة المستوى.	توفر تحكمًا دقيقًا بموارد النظام مثل الذاكرة والمعالج.

أبحث وأُفَارِن: أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن أمثلة على كلٍّ من المُترجم والمُفسّر، ثمَّ أُقَارِن بينهما من حيث آليّة التنفيذ، والسرعة في التنفيذ، وسهولة اكتشاف الأخطاء.

المواطنة الرقمية 🌐

- أحرص دائماً على استخدام البرمجيات المرخصة قانونياً، واحترم حقوق المُلكية الفكرية للمُطوِّرين والشركات.
- أتجنّب استخدام البرمجيات المُقرّصنة أو البرمجيات غير المرخصة؛ لأنها قد تكون غير آمنة، وتُعرّض جهازي ومعلوماتي الشخصية للخطر.

تصميم برنامج وإعداده لإنشاء لعبة تخمين الأرقام باستخدام لغة بايثون (Python) / المهمة (1).

أتعاون مع مجموعتي على البدء بالتحضيرات اللازمة لتصميم لعبة تخمين للأرقام ذات واجهة نصية عبر برمجية بايثون، حيث سيكون التركيز في هذا الدرس على مرحلة التخطيط والتحليل باتباع الخطوات الآتية:

- التعريف للمشكلة/ فكرة اللعبة التي تم اختيارها وتوضيح الهدف منها، وتحديد الجمهور المستهدف.

تقديم فكرة المشروع: برمجة لعبة التخمين "نجوم وأقمار".
سنعمل على إعداد عرض تقديمي باستخدام google slides يتضمن شرحاً للمشكلة وأسباب اختيارها والحل المقترح / اللعبة والهدف منها، ووصف لسيناريو اللعبة وقواعد اللعبة منذ بدايتها حتى انتهائها، وإعداد الخوارزمية ومخطط سير العمليات (flowcharts) لتصميم لعبة التخمين باستخدام إحدى الأدوات الرقمية مثل draw.io. كما تعلمنا في السنوات السابقة.

وجه المقارنة	المترجمات	المفسرات
مبدأ العمل	تحليل كامل للشفيرة المصدرية، وتحسينها وتوليد البرنامج التنفيذي.	تنفيذ الشيفرة المصدرية على نحو مباشر سطرًا تلو الآخر..
مدخلات	الشفيرة المصدرية كاملة.	سطر واحد من الشيفرة المصدرية.
مخرجات	برنامج تنفيذي بلغة الآلة.	تنفيذ مباشر للشفيرة المصدرية.
سهولة التعلم	تتطلب معرفة أقل بعتاد الحاسوب، وتكون أقرب للغة البشرية؛ مما يجعلها أسهل للتعلم.	تتطلب معرفة دقيقة بعتاد الحاسوب وتفاصيل تنفيذ الأوامر؛ مما يجعلها أصعب للتعلم.
شفيرة الآلة	تتطلب حفظ البرنامج التنفيذي على الذاكرة الذي يتضمن شيفرة الآلة.	لا تقوم بتوليد شيفرة تنفيذية، وتتطلب الحفظ على الذاكرة.

وجه المقارنة	المترجمات	المفسرات
سرعة التنفيذ	تسريع بالنسبة للبرنامج بعد ترجمته؛ ولكنه يتطلب وقتاً إضافياً للترجمة والبناء.	الشفيرة المُفسرة تتطلب وقتاً أطول للتنفيذ.
زمن التطوير	بشكل عام، تتطلب اللغات المترجمة وقتاً في أثناء عملية التطوير لتوليد البرنامج التنفيذي.	بشكل عام، تتطلب المُفسرات وقتاً أقصر في أثناء عملية التطوير لتنفيذ البرنامج.
آلية التنفيذ	تنفيذ البرنامج منفصل عن مرحلة الترجمة، ولا يمكن إجراء التنفيذ إلا بعد انتهاء كل عملية الترجمة بنجاح.	تنفيذ البرنامج هو جزء من عملية التفسير.
ما بعد التنفيذ	أي تعديل أو تطوير سيحتاج لتعديل الشيفرة المصدرية، وتوليد برنامج تنفيذي جديد.	تعديل الشيفرة المصدرية وتطويرها أبسط بسبب عدم وجود برنامج تنفيذي.
متطلبات الذاكرة	يتطلب البرنامج التنفيذي حجز جزء من الذاكرة، إلا أن تنفيذه لا يتطلب وجود مترجم الذاكرة.	لا يُحجز أي جزء من الذاكرة بسبب عدم توليد برنامج تنفيذي؛ ولكن المفسر نفسه يتطلب جزءاً من الذاكرة في أثناء التنفيذ.
تحسين الشيفرة المصدرية	عادةً ما يكون البرنامج الناتج أفضل بسبب إجراء المترجم لتحليل الشيفرة المصدرية كاملة.	عادةً ما يكون ناتج التفسير أقل كفاءة ومتانة بسبب تنفيذ البرنامج سطرًا تلو الآخر.
سهولة اكتشاف الأخطاء	يكتشف المترجم الأخطاء والتنبيهات كافة في أثناء زمن الترجمة، ومن ثم تُعرض. لن يتم توليد برنامج تنفيذي في حال وجود أخطاء	يكتشف المفسر الأخطاء على مستوى السطر في أثناء زمن التنفيذ، ويتوقف تنفيذ البرنامج حصراً في حال وجود خطأ في سطرٍ ما.
أشهر اللغات	يكتشف المترجم الأخطاء والتنبيهات كافة في أثناء زمن الترجمة، ومن ثم تُعرض. لن يتم توليد برنامج تنفيذي في حال وجود أخطاء.	يكتشف المفسر الأخطاء على مستوى السطر في أثناء زمن التنفيذ، ويتوقف تنفيذ البرنامج حصراً في حال وجود خطأ في سطرٍ ما.
أشهر اللغات	.C, C++, C#, Scala	.Python, Ruby, PH

تعليمات ذات صلة بمنتج

التعلم

المرحلة الأولى: التخطيط

والتحليل

■ أوزع الطلبة في مجموعات عمل غير متجانسة، وأكلفهم البدء بالعمل على المنتج الخاص بالوحدة.

■ أوضح للطلبة الفكرة من اللعبة وتعليمات المهمة مستعيناً بالتوضيحات الواردة.

■ أطلب إلى الطلبة مناقشة فكرة اللعبة "إنشاء لعبة تخمين أرقام بواجهة نصية".

■ أعطي الطلبة تعليمات تطوير اللعبة وتتضمن:

- تحديد هدف اللعبة.
- تحديد الجمهور المستهدف.
- وصف سيناريو اللعبة، وكتابة سيناريو اللعبة على نحو واضح.

● رسم مخططٍ لسير العمليات، (Flowchart) يوضح جميع الخطوات التي سيمر بها اللاعب من بداية اللعبة حتى نهايتها.

● توزيع الأدوار والمهام بين أفراد المجموعة: التأكد من وضوح المهام لكل عضو، والتعاون فيما بينهم لتحقيق الهدف المشترك.

أقيّم تعلّمي

المعرفة: أوظّف في هذا الدرس ما تعلّمته من معارف في الإجابة عن السؤالين الآتيين:

السؤال الأول: أوضّح المقصود بكلّ ممّا يأتي:
1. لغة البرمجة.

2. لغة الآلة.

3. البرنامج.

السؤال الثاني: أفرّن بين لغات البرمجة عالية المستوى ولغات البرمجة مُنخفضة المستوى من حيث سهولة القراءة، والتطوير، والأداء، والكفاءة.

المهارات: أوظّف مهارات التفكير الناقد والتواصل الرقمي والبحث الرقمي في الإجابة عن الأسئلة الآتية:

السؤال الأول: أُميّز بين لغات البرمجة الكتلية ولغات البرمجة النصية من حيث آلية تمثيل الأوامر، ثمّ أذكر مثالاً على كلّ منهما.

السؤال الثاني: لماذا يُعدّ المترجم أسرع تنفيذًا من المُفسّر؟

■ أوجه الطلبة لطرح الأسئلة الرئيسة الآتية، ومناقشتها:

• كيف سيبدأ العمل في تصميم اللعبة؟

• ما عدد اللاعبين اللازم للعبة؟

• ما القواعد والإجراءات الخاصة باللعبة؟

• متى يُعدُّ اللاعب فائزاً؟

• متى تنتهي اللعبة؟

■ التأكد من وضوح خطة العمل.

المرحلة الثانية: التصميم والتطوير

■ إعداد العرض التقديمي: استخدام Google Slides لإعداد عرض

تقديمي يتضمن:

• شرح فكرة اللعبة وأسباب اختيارها.

• الحل المقترح والهدف من اللعبة.

• وصف سيناريو اللعبة وقواعدها من البداية إلى النهاية.

■ مشاركة العرض التقديمي مع الزملاء والمعلم / المعلمة للحصول على

الملاحظات والتعديلات.

■ تصميم مخطط سير العمليات للعبة (Flowchart) باستخدام أداة مثل

.Draw.io

إجابات أسئلة أقيّم تعلّمي

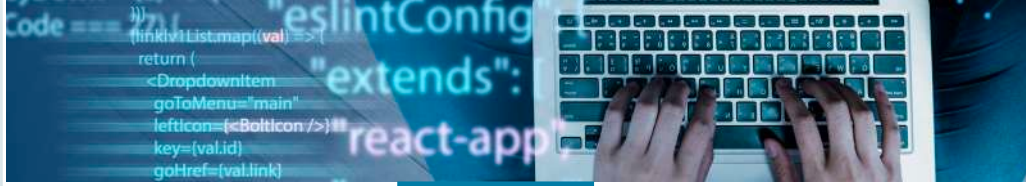
المعرفة: أوظف في هذا الدرس ما تعلّمته من معارف للإجابة عن السؤالين الآتيين: السؤال الأول: أوضح المقصود في ما يأتي:

■ لغة البرمجة: هي مجموعة من الأوامر والتعليمات التي تستخدم في كتابة البرامج والتطبيقات وفق قواعد محددة. وتعدّ الأداة الرئيسة التي يستخدمها المبرمجون للتفاعل مع جهاز الحاسوب وتوجيهه لتنفيذ مهام معينة. وتُصنّف لغات البرمجة إلى أنواع عدة بناءً على وظائف كل لغة وتطبيقاتها وطريقة معالجتها، وغيرها من المقاييس.

■ لغة الآلة: هي لغة برمجة تحتوي على أوامر وتعليمات، يمكن لجهاز الحاسوب فهمها مباشرة ومعالجتها. وعلى الرغم من أن لغة الآلة يصعب

فهمها على الإنسان، فإنّ لها ميزات عدة، فالحاسوب يفهمها وينفذها مباشرة من دون الحاجة إلى مترجم، وتتميز بالسرعة مقارنة مع اللغات عالية المستوى.

■ البرنامج: هو مجموعة من الأوامر المكتوبة باستخدام إحدى لغات البرمجة لحل مشكلة معينة، أو للقيام بمهمة محددة بوساطة جهاز الحاسوب.



الدرس الثاني

أساسيات لغة البرمجة بايثون (Basics of Python Programming) **أنظر الهامش**

الفكرة الرئيسة:

تعرّف لغة البرمجة بايثون (Python)، وتعلّم كيف يُمكن تحميل البرنامج، والتعامل مع الشاشة الرئيسة، وإنشاء برنامج بسيط وتنفيذه وحفظه واسترجاعه، إضافةً إلى تعرّف الأنواع المختلفة من البيانات في لغة البرمجة بايثون (Python)، واستخدام التعابير الحسابية والتعابير المنطقية وتمثيلها في لغة البرمجة بايثون (Python)، وفهم قواعد كتابة الجمل البرمجية عن طريق التطبيقات العملية.

المفاهيم والمصطلحات:

البرمجة بالكائنات (Object-Oriented Programming)، تعدّد المهام (Multitasking)، مُوجّه الأوامر (Command Prompt)، مُحرّرات النصوص (Text Editors)، تمييز الصيغة (Syntax Highlighting)، التعليقات (Comments)، الكلمات المحجوزة (Reserved Words)، المُعرّفات (Identifiers)، الرموز (Literals)، المسافات الفارغة (Indentations)، تمييز حالة الحرف (Case Sensitivity)، أولويات العوامل (Operators Precedence)، الترابط (Associativity).

أنظر الهامش (Learning Products)

فتح برنامج طابعة الرسائل
على برمجة بايثون، والبدء
بطباعة رسالة الترحيب وخيارات
القائمة الرئيسة، ضمن سياق
تصميم لعبة تخمين الأرقام
بإستخدام برمجة بايثون.

أنظر الهامش

أنظر الهامش

السؤال الثاني: أقرن بين لغات البرمجة عالية المستوى ولغات البرمجة ذات المستوى المنخفض من حيث سهولة القراءة، والتطوير، والأداء، والكفاءة.

وجه المقارنة	لغة البرمجة عالية المستوى	لغة البرمجة منخفضة المستوى
سهولة القراءة	تتميز لغات البرمجة عالية المستوى بسهولة قراءة الشيفرة البرمجية؛ لأنها تستخدم جملاً وأوامر قريبة من اللغة البشرية. وتوفر هذه اللغات بناء جملة واضح ومنظم، يجعل الشيفرة أكثر فهماً وقراءة.	تكون الشيفرة البرمجية في لغات البرمجة منخفضة المستوى أصعب قراءة؛ لأنها أقرب إلى لغة الآلة، وتتطلب معرفة دقيقة بعتاد الحاسوب. وتكون الأوامر قصيرة ومشفرة على نحو يجعل فهمها معقداً.
بيئة التطوير	توفر بيئات تطوير متكاملة وأدوات متقدمة تساعد في تسهيل عملية كتابة الشيفرة وتصحيح الأخطاء وتنفيذها. وتدعم هذه اللغات مكتبات وأطر عمل متعددة، تسرع عملية التطوير.	تتطلب عملية التطوير معرفة عميقة بعتاد الحاسوب وآلية عمله. وتكون الأدوات المتاحة أقل تطوراً مقارنةً بلغات البرمجة عالية المستوى؛ مما يجعل عملية التطوير أكثر تحدياً.
الأداء	على الرغم من أن لغات البرمجة عالية المستوى توفر سهولة في الاستخدام والتطوير، فإن الأداء يكون أبطأ مقارنةً بلغات البرمجة منخفضة المستوى بسبب مستوى التجريد العالي.	تتميز هذه اللغات بأداء عالٍ جداً؛ لأنها تتفاعل مباشرة مع عتاد الحاسوب، وتنفذ الأوامر على نحو مباشر. وتُستخدم في التطبيقات التي تتطلب أداءاً فائقاً وسرعة عالية.
الكفاءة	لغات البرمجة عالية المستوى تكون أقل كفاءة في استخدام موارد النظام؛ لأنها تترجم الأوامر إلى لغة الآلة عبر طبقات متعددة من التجريد. وتوفر الراحة في الكتابة والتطوير على حساب الكفاءة.	توفر كفاءة عالية في استخدام موارد النظام؛ مما يجعلها مثالية للتطبيقات التي تحتاج إلى التحكم الكامل في الذاكرة والمعالج. وتُستخدم في تطوير البرمجيات التي تتطلب استغلالاً أمثل للموارد.

المهارات: أوظف مهارات التفكير الناقد، والبحث الرقمي، والتواصل للإجابة عن الأسئلة الآتية:

السؤال الأول: ما الفرق بين لغات البرمجة المبنية على الكتل ولغات البرمجة النصية من حيث كيفية تمثيل الأوامر، أعطِ مثالاً على كل منها.

■ لغات برمجية مبنية على الكتل (Block-based Programming Languages) تستخدم هذه اللغات كتلاً رسومية (Graphical Blocks) لتمثيل أجزاء البرنامج بدلاً من النصوص. ومن الأمثلة عليها: سكراتش (Scratch).

■ لغات البرمجة النصية (Text-based Programming Languages) هي التي تستخدم النصوص لتمثيل أجزاء البرنامج بدلاً من الكتل، ومن الأمثلة عليها: لغة جافا سكريبت (JavaScript).

السؤال الثاني: لماذا يعدّ المترجم أسرع في التنفيذ من المفسر؟

يكون المترجم أسرع في التنفيذ؛ لأن الشيفرة البرمجية تُترجم بالكامل مسبقاً إلى لغة الآلة؛ مما يسمح بتنفيذ الشيفرة مباشرة وبسرعة من قبل المعالج. في حين يترجم المفسر، وينفذ الشيفرة سطرًا بسطرًا في الوقت الحقيقي؛ مما يضيف وقتًا إضافيًا لكل عملية ترجمة وتنفيذ، ويجعل التنفيذ أبطأ.

السؤال الثالث: ما التحديات التي قد تواجه المبرمج عند الانتقال من استخدام لغات البرمجة المبنية على الكتل إلى لغات البرمجة النصية؟

- تعتمد لغات البرمجة المبنية على الكتل، مثل Scratch، على واجهة مرئية تتيح سحب الكتل البرمجية وإفلاتها لتكوين البرامج. في المقابل، يتطلب الانتقال إلى الكتابة النصية التعود على كتابة الشيفرة يدويًا.
- يجب على المبرمجين تعلّم القواعد النحوية الصحيحة للغة البرمجة النصية المستخدمة، والتأكد من كتابة كل حرف بدقة.

نتائج التعلم (Learning Outcomes):

- أعرّف النموذج الأولي للبرنامج.
 - أنظر في كتاب الجمل البرمجية بلغة البرمجة بايثون (Python).
 - أوضح عناصر لغة البرمجة بايثون (Python)؛ من: ثوابت، ومُتغيّرات، ورموز، وتعبيرات، وعلاقات.
 - أجري عمليات حسابية على التعبيرات الحسابية.
 - أكتب كلاً من العلاقات والعبارات الحسابية والمنطقية باستخدام لغة البرمجة بايثون (Python).
 - أعرّف الصيغة العامة لجمل الإدخال input.
- تعرّف في الدرس السابق لغات البرمجة النصية (Text-Based Programming Languages) التي تُستخدم فيها النصوص لتمثيل أجزاء البرنامج بدلاً من الكتل. وتُعدّ لغة البرمجة بايثون (Python) أحد أشهر الأمثلة على هذه اللغات؛ فما لغة البرمجة بايثون؟ وما مزاياها؟ وكيف يُمكن التعامل معها واستخدامها؟



نشاط
تمهيدي

أبحث - بالتعاون مع أفراد مجموعتي - في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن نشأة لغة البرمجة بايثون (Python)، واستخداماتها، وأهم مزاياها، ثم أناقش أفراد المجموعات الأخرى / مُعلّمي / مُعلّمتي فيما أتوصّل إليه من نتائج.

تُعرّف بايثون (Python) بأنّها لغة برمجة عالية المستوى، تُستخدم في أنظمة التشغيل المختلفة، بما في ذلك نظام التشغيل ويندوز (Windows)، ونظام التشغيل (MacOS)، ونظام التشغيل (Linux). وهي تمتاز بأنّها لغة مفتوحة المصدر؛ ما يعني إمكانية تحميل الرمز (الكود) المصدر الخاص بها، وتعديله، واستخدامه بحرية.

تثبيت لغة البرمجة بايثون (Python Setup)

تأتي معظم أنظمة التشغيل في بيئة بايثون مثبتة مسبقاً، باستثناء نظام التشغيل ويندوز (Windows)، الذي يتطلب تثبيت بايثون يدوياً. وإذا كنا نرغب في استخدام محرر نصوص، فسنحتاج إلى تثبيته أيضاً.

يُمكن تثبيت لغة البرمجة بايثون (Python) في نظام التشغيل ويندوز (Windows) باتباع الخطوات الآتية:

1. تحميل مُفسّر لغة البرمجة بايثون (Python):

أ. زيارة الموقع الإلكتروني للغة البرمجة بايثون (Python): <https://www.python.org/>

ب. الضغط على زرّ تحميل آخر إصدار متوافر للغة البرمجة بايثون (Python) كما في الشكل (1-2).

■ تتطلب لغات البرمجة النصية، مثل JavaScript, Python، فهم بناء جملة أكثر تعقيداً مقارنة باللغات المبنية على الكتل. وأي خطأ بسيط، يمكن أن يؤدي إلى فشل البرنامج.

■ يحتاج المبرمجون إلى التعمّد على استخدام أدوات البرمجة مثل نظم التحكم في الإصدارات (Git)، وأدوات تصحيح الأخطاء، وأدوات بناء المشروعات.

■ في البرمجة النصية، يجب على المبرمجين تنظيم الشيفرة على نحو منطقي وهرمي؛ مما قد يتطلب تعلّم كيفية تقسيم البرنامج إلى وحدات ودوال.

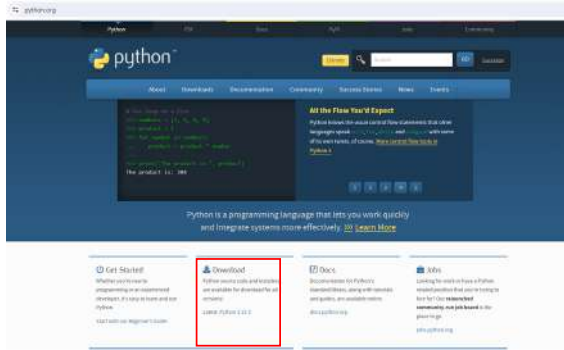
■ تتطلب البرمجة النصية تذكّر العديد من الأوامر والقواعد، على عكس لغات البرمجة المبنية على الكتل التي تقدم تلميحات بصرية، ومساعدة مستمرة.

السؤال الرابع: بعد تعلّم حول المفسر والمترجم، أيهما أفضل استخدامًا لتطوير برامج كبيرة ومعقدة، ولماذا؟

استخدام المترجم هو الأفضل لتطوير البرامج الكبيرة والمعقدة؛ لأنه يوفر أداءً عاليًا، وكفاءة في استخدام الموارد، واستقرارًا أكبر عن طريق اكتشاف الأخطاء وإصلاحها قبل التشغيل. بالإضافة إلى ذلك، يوفر المترجم ملفات تنفيذية مستقلة، يمكن توزيعها بسهولة على المستخدمين النهائيين. هذه الميزات تجعل المترجم الخيار الأمثل عند تطوير برمجيات تتطلب أداءً عاليًا، واستقرارًا ومرونة في التوزيع.

أساسيات لغة البرمجة بايثون

Basics of Python Programming



الشكل (2-1): تحميل مُفسّر لغة البرمجة بايثون (Python).

ملحوظة: يجب التأكد أن رقم الإصدار الذي يراد تحميله مُوائم لنظام التشغيل المُستخدم.

ج. الضغط على زرّ حفظ الملف (Save File) لكي تبدأ عملية التحميل.

2. تثبيت مُفسّر لغة البرمجة بايثون (Python):

أ. الضغط بزرّ الفأرة الأيمن على الملف بعد اكتمال عملية التحميل، ثمّ الضغط على خيار

(Run as Administrator) كما في الشكل (2-2).



الشكل (2-2): تثبيت مُفسّر لغة البرمجة بايثون (Python).

ب. تفعيل خيار (Use admin privileges when installing py.exe) وخيار (Add path

to python.exe) كما في الشكل (2-3).

عدد الحصص المقترح

ثلاث حصص.

المصادر: كتاب الطالب، لغة البرمجة بايثون.

الأدوات والتجهيزات:

أجهزة حاسوب، جهاز للعرض.

أولاً: التهيئة

سياق التعلّم:

أوضح للطلبة الفكرة الرئيسة من الدرس، وأضعه في قالب سياق التعلّم، وأخبرهم أنهم سيكملون رحلة التعلّم بوصفهم مبرمجين ناشئين كالتي بدؤوا بها في الصفوف السابقة؛ إذ ستسنى لهم هذه المرة فرصة تصميم ألعاب، وكتابة برامج متنوعة.

أوضح لهم منتج التعلّم المتوقع إنجازه خلال هذا الدرس، وهو فتح برنامج خاص بلعبة تخمين الأرقام في برمجة بايثون (Python)، والبَدْء بطباعة رسالة الترحيب وخيارات القائمة الرئيسة، ضمن سياق تصميم لعبة تخمين الأرقام باستخدام برمجة بايثون (Python).

نشاط تمهيدي:

لغة البرمجة بايثون: هي لغة برمجة عالية المستوى، وسهلة التعلم والاستخدام، ابتكرت على يد المبرمج الهولندي جايدو فان روسم "Guido van Rossum" في مركز العلوم والحاسب الآلي (CWI) في أمستردام، وتعدّ من أكثر اللغات انتشارًا؛ نظرًا لطبيعة الاستخدامات التي توفرها.

أهم مجالات استخدام لغة بايثون

Python:

■ تطوير مواقع الويب: إذ تمتاز لغة بايثون بوجود العديد من المكتبات، وأطر العمل الجاهزة التي يمكن استخدامها لتطوير مواقع الويب.

■ برمجة الألعاب تُستخدم لغة بايثون في تطوير الألعاب التفاعلية؛ إذ يمكن استخدام مكتبات بايثون.

■ التعلم الآلي والذكاء الاصطناعي تُعد لغة بايثون اللغة الأولى على قائمة لغات الذكاء الاصطناعي والتعلم الآلي؛ لسهولة تعلمها، وقدرتها على تنفيذ العديد من خوارزميات الذكاء الاصطناعي. ومن أشهر مكتبات بايثون المُستخدمة في هذا المجال مكتبة (Keras) و (PyTorch).

■ تحليل البيانات وعرضها على نحو مرئي: إذ يتم عن طريق مكتبات لغة بايثون تجهيز البيانات وتحويلها للشكل المطلوب، ومن ثم استخراج المعلومات المطلوبة، وإظهار الرسومات البيانية، ومن أهم المكتبات المُستخدمة في مجال تحليل البيانات مكتبة (NumPy) ومكتبة (Pandas).

■ برمجة تطبيقات سطح المكتب: يمكن استخدام لغة بايثون في برمجة تطبيقات سطح المكتب؛ إذ تتوفر مكتبة يمكن استخدامها في تطوير واجهة المستخدم.

أهم مميزات لغة البرمجة بايثون:

■ سهولة التعلم: تُعرف بايثون ببساطة قواعدها وسهولة قراءتها؛ مما يجعلها مناسبة للمبتدئين.

إضاءة

إذا لم يظهر رقم إصدار لغة البرمجة بايثون (Python) عند التحقق من تثبيت مُفسّر لغة البرمجة بايثون (Python)، فإنّ أحد أكثر الأسباب شيوعًا لذلك هو عدم تفعيل خيار (Add python.exe to PATH) أثناء عملية التثبيت كما في الشكل (2-3).



الشكل (2-3): تفعيل الخيارات، وبذلك تثبيت مُفسّر لغة البرمجة بايثون (Python).

ج. الضغط على زرّ التثبيت الآن (Install Now) لكي تبدأ عملية التثبيت.

د. الضغط على زرّ الإغلاق (Close) بعد اكتمال عملية التثبيت بنجاح (Setup Was Successful).

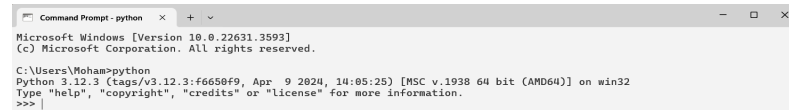
3. التحقق من تثبيت مُفسّر لغة البرمجة بايثون (Python):

أ. فتح مُوجّه الأوامر (Command Prompt)، بالذهاب إلى القائمة الرئيسية، وكتابة كلمتي مُوجّه الأوامر (Command Prompt) في مربع البحث، ثم الضغط على Command Prompt. عند اتباع الخطوات السابقة ستظهر الشاشة الموضحة في الشكل (2-4).



الشكل (2-4): شاشة تشغيل مُوجّه الأوامر (Command Prompt).

ب. كتابة كلمة بايثون (python)، ثمّ الضغط على زرّ الإدخال (Enter)؛ للتحقق من تثبيت مُفسّر لغة البرمجة بايثون (Python)، وتعرّف رقم الإصدار الخاص به كما في الشكل (2-5).



الشكل (2-5): التحقق من تثبيت مُفسّر لغة البرمجة بايثون (Python)، وتعرّف رقم إصداره.

■ المرونة: تدعم العديد من البرمجة الهيكلية والبرمجة الشيئية.

■ المجتمع والدعم: تحتوي بايثون على مجتمع كبير من المطورين والمساهمين الذين يقدمون دعمًا واسعًا ومكتبات متعددة.

■ التكامل: تتكامل بايثون بسهولة مع لغات أخرى مثل C/C++ وجافا؛ مما يعزز قدرتها على التعامل مع مختلف التطبيقات.

ثانيًا: عملية التعليم والتعلم:

إجابة محتملة

```
print('Noor')
```

ثم نحفظ الملف بالضغط على زر

Save as ثم Run Module.

ملاحظات

ب. وجوب أن يبدأ المُعرّف بأحد الحروف الكبيرة (A - Z)، أو أحد الحروف الصغيرة (a - z)، أو الشرطة السفلية (underscore) (-).

ج. عدم بدء المُعرّف بعدد.

د. منع استخدام أي كلمة من الكلمات المحجوزة.

مثال:

من أسماء المُعرّفات المقبولة في لغة البرمجة بايثون: name، number، Grade_ . ومن الأسماء غير المقبولة في لغة البرمجة بايثون: @user، -name، 2nd .

3. الكلمات المحجوزة (Reserved words): توجد كلمات محجوزة للغة البرمجة بايثون (Python)، لا يُمكن استخدامها مُعرّفات. وهذه الكلمات هي:

and	elif	from	None	return
assert	else	global	nonlocal	True
break	except	if	not	Try
class	exec	import	or	while
continue	False	in	pass	with
def	finally	is	print	yield
del	for	lambda	raise	

4. الثوابت (Constants): قيم تظل ثابتة، ولا تتعرض للتغيير أثناء تنفيذ البرنامج. وهي تُصنّف إلى نوعين اثنين، هما:

أ. الثوابت الرمزية النصية (Character Constants): سلسلة من الحروف التي تُستخدم في لغة البرمجة، وتُكتب بين علامتي اقتباس، مثل: "Hello"، و "Jordan".

ب. الثوابت العددية (Numerical Constants): سلسلة من الأعداد، تبدأ بالعدد (0)، وتنتهي بالعدد (9)، ويُمكن أن تحتوي على إشارة (+)

في مُقدّمها للدلالة على أن العدد موجب، وقد تحتوي على إشارة (-) للدلالة على أن العدد سالب. وسيقصر الحديث في هذه الوحدة على الثوابت العددية الحقيقية (real numbers)، مثل: الأعداد الصحيحة، والأعداد العشرية.

5. المتغيّرات (Variables): رموز تدلّ على القيم المُستخدمة في البرنامج؛ إذ يُخصّص للمتغيّر مساحة تخزينية في ذاكرة البرنامج، وتوضع القيمة المُرتبطة بالمتغيّر في هذه المساحة التخزينية، ويُستخدم اسم المتغيّر في الإشارة إلى تلك القيمة.

إضافة

يختلف الثابت الرمزي عن الثابت العددي في أنّه لا يُستخدم في العمليات الحسابية، وإنّما يُستخدم في تمثيل المعطيات غير العددية.

نشاط:

ستتنوع الإجابات

فعلى سبيل المثال:

إدخال قيمة نصية.

```
text_input = input("Enter some text: ")
print("You entered:", text_input)
```

input() لإدخال قيمة عددية.

```
numeric_input = int(input("Enter a number: "))
print("You entered the number:", numeric_input)
```

نشاط عملي:

إجابة محتملة.

```
city = input('What is your favorite city?')
print(city)
```

عند تشغيل البرنامج، سنحصل على النتيجة الآتية:

What is your favorite city? Mecca

Mecca

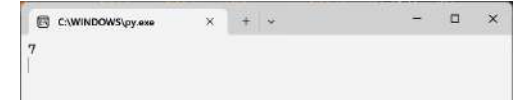
مثال:

يراد تعريف المُتغيّر المُسمّى (days) في لغة البرمجة بايثون (Python)، وإسناد القيمة (7) إليه، وطباعة قيمته.

يجب كتابة الأوامر البرمجية الآتية:

```
days = 7
print(days)
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية على شاشة جهاز الحاسوب:



مثال:

يُعرّف البرنامج الآتي مُتغيّرين، قيمة كلّ منهما (99)، ثمّ يطبع هذه القيمة لكليهما:

```
x = y = 99
print('x =', x)
print('y =', y)
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية على شاشة جهاز الحاسوب:



إضاءة:

في لغة البرمجة بايثون (Python)، يُمكن تعريف عدد من المُتغيّرات ذات القيم المتساوية في وقت واحد.

نشاط عملي

أكتب برنامجاً بلغة البرمجة بايثون (Python)؛ لتعريف مُتغيّرات يُماثل عددها عدد أيام الأسبوع، واسناد اسم يوم من أيام الأسبوع إلى كلّ منها، ثمّ طباعتها.

ملاحظات

.....

.....

.....

.....

.....

.....

إضاءة



المتغير في لغة البرمجة بايثون (Python) هو من النوع غير الثابت؛ لأنه يتغير تلقائياً بحسب نوع القيمة التي تُخزن فيه.

توجد أنواع كثيرة من المتغيرات في لغة البرمجة بايثون (Python)، ويمكن إجمال الأنواع الأساسية لهذه المتغيرات في ما يأتي:

المتغيرات العددية (Numbers)، والمتغيرات النصية (Strings)، والمتغيرات المنطقية (Booleans)، والمصفوفات ذات الحجم غير الثابت التي تُسمى القوائم (Lists)، والمصفوفات ذات الحجم الثابت والقيم الثابتة التي لا تقبل التغيير، والتي تُسمى الصفوف (Tuples)، والمصفوفات ذات الحجم غير الثابت التي لا تحتوي على قيم مكررة (Sets)، والجداول التي تُخزن فيها البيانات بصورة مفاتيح (Keys) وقيم (Values)، وتُسمى القواميس (Dictionaries).

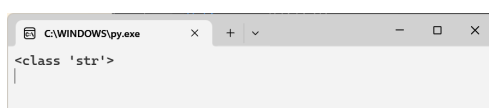
يمكنني معرفة نوع أي متغير عرّفته، وأسندت إليه قيمة ما، باستخدام الدالة type():

type()

مثال:

```
var = 'Jordan'
print(type(var))
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية على شاشة جهاز الحاسوب:



أستخدم برمجة بايثون (Python) في تحديد نوع كل متغير مما يأتي:

```
Age = 16
name = "Basem"
is_student = True
height = 1.75
```



نشاط عملي

28

نشاط عملي

إجابة محتملة.

```
name = input('What is your name?')
place_of_birth = input('What is your place of birth?')
age = int(input('What is your age?'))
print(name, 'was born in', place_of_birth, "in", 2024 - age)
```

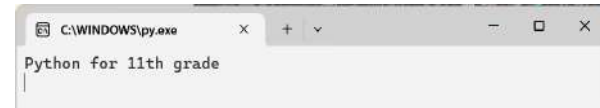
ثم نحفظ الملف بالضغط على زر Save as ثم Run Module، وإدخال البيانات التي يطلبها.

6. الرموز (Literals): يُستخدم في لغة البرمجة بايثون (Python) مجموعة من الرموز، أبرزها:
أ. النصوص (String Literals):

مثال:

```
a = '''Python for 11th grade'''
print(a)
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية على شاشة جهاز الحاسوب:

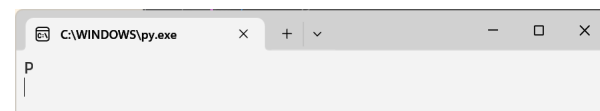


ب. الحروف (Character Literals):

مثال:

```
a = 'P'
print(a)
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية على شاشة جهاز الحاسوب:



ج. الأعداد (Numeric Literals):

مثال:

```
y = 30
print(y)
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية على شاشة جهاز الحاسوب:

30

د. الرموز المنطقية (Boolean Literals):

مثال:

```
a = (1 == True)
b = (1 == False)
c = True + 1
d = False + 4
print(a, b, c, d)
```

نشاط عملي:

إجابة محتملة.

```
day1= ' Sunday '
day2= ' Monday '
day3= ' Tuesday '
day4= ' Wednesday '
day5= ' Thursday '
day6= ' Friday '
day7= ' Saturday '
print(day1, ' , ' , day2, ' , ' , day3, ' , ' ,
      day4, ' , ' , day5, ' , ' , day6, ' , ' , day7)
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية على شاشة جهاز الحاسوب:

```
C:\WINDOWS\py.exe
True False 2 4
```

7. التعبيرات والجمل (Expression and Statements): يُعرّف التعبير بأنه سلسلة تتألف من واحد أو أكثر من القيم والمتغيرات والعوامل واستدعاءات الدوال، وينتج منها قيمة معينة. أما الجملة فهي أصغر جزء من البرنامج يقبل التنفيذ، ويؤدي إلى حدوث تأثيرات عديدة، ولا يُفْضِي إلى نتيجة مُحدَّدة أو قيمة مُعَيَّنة بعد الانتهاء من تنفيذه. ومن ثمَّ، فلا يُمكن - مثلاً - إسناد جملة إلى مُتغيِّر، أو وضعها مباشرة في دالة الطباعة (print)؛ لعدم وجود قيمة عائدة منها بعد تنفيذها.

مثال:

في الأمر البرمجي: $z = 2 + 3$ ؛
(3+2) هو تعبير، أما $(z = 2 + 3)$ فهو جملة.

8. الكتل البرمجية والمسافات الفارغة (Blocks and Indentations):

الكتل البرمجية هي مجموعة من الجمل ذات الصلة. وقد تحتوي الكتلة البرمجية على جملة واحدة فقط. أما المسافات الفارغة (Indentations) فتضاف إلى البرنامج في لغة البرمجة بايثون (Python)؛ لتحديد الكتل البرمجية وتوضيحها. وستعرف عليها بشيء من التفصيل في الدروس القادمة

```
C:\WINDOWS\py.exe
one is greater than zero.
```

قواعد إضافية لكتابة الجملة البرمجية:

1. تمييز حالة الأحرف (Case Sensitivity): تُميِّز لغة البرمجة بايثون (Python) بين الأحرف الكبيرة والأحرف الصغيرة. فمثلاً، كلمة (days) وكلمة (Days) مختلفتان، وهما لا تعنيان شيئاً واحداً؛ سواء كان استخدامهما للمتغيرات، أو الدوال، أو غير ذلك.

2. عند اختيار الأسماء (Names)، يُنصَح باتباع القواعد الآتية:

أ. اسم المُتغيِّر (Variable Name): تُستخدَم الأحرف الصغيرة أو الكبيرة عند وضع أسماء للمتغيرات. وفي حال اشتمل اسم المُتغيِّر على أكثر من كلمة، فإنَّ الشرطة السفلية (underscore) توضع بين كل كلمتين كما في المثال الآتي:

average_score = 25

نشاط عملي:

إجابة محتملة.

استخدم بايثون في تحديدي نوع كل متغير مما يأتي:

```
Age = 16
print(type(Age))
Age = '16'
print(type(Age))
name = "Basem"
print(type(name))
is_student = True
print(type(is_student))
height = 1.75
print(type(height))
```


9. العوامل والتعبير (Operators and Expressions):

تُصنّف العوامل بحسب استخداماتها إلى سبع مجموعات. وفي ما يأتي بيان لأربع منها:
أ. العوامل المُستخدمة في العمليات الحسابية (Arithmetic Operators)، وهي مُمثّلة في الجدول (2-1).

الجدول (2-1): العوامل المُستخدمة في العمليات الحسابية.

اسم العامل	الرمز	مثال توضيحي	الشرح
إضافة (Addition)	+	$x+y$	إضافة قيمة y إلى قيمة x .
الطرح (Subtraction)	-	$x-y$	طرح قيمة y من قيمة x .
الضرب (Multiplication)	*	$x*y$	ضرب قيمة x في قيمة y .
القسمة (Division)	/	x/y	قسمة قيمة x على قيمة y .
باقي القسمة (Modulus)	%	$x\%y$	إرجاع باقي قسمة قيمة x على قيمة y .
القوة (Exponentiation)	**	$x**y$	رفع قيمة x إلى أس بقيمة y .
القسمة التحتية (Floor Division)	//	$x//y$	قسمة قيمة x على قيمة y ، وإرجاع أقرب عدد صحيح إلى الناتج (أقل من الناتج، أو يساوي الناتج).

ب. العوامل المُستخدمة في المقارنات (Comparison Operators)، وهي مُمثّلة في الجدول (2-2).

الجدول (2-2): العوامل المُستخدمة في المقارنات.

اسم العامل	الرمز	مثال توضيحي	الشرح
يساوي (Equal to)	==	$x==y$	هل قيمة x تساوي قيمة y ؟
لا يساوي (Not equal to)	!=	$x!=y$	هل قيمة x لا تساوي قيمة y ؟
أكبر من (Greater than)	>	$x>y$	هل قيمة x أكبر من قيمة y ؟
أصغر من (Less than)	<	$x<y$	هل قيمة x أصغر من قيمة y ؟
أكبر من أو تساوي (Greater than or equal to)	>=	$x>=y$	هل قيمة x أكبر من قيمة y أو تساويها؟

نشاط عملي:

إجابة محتملة.

Message: This lesson is about variables. Variables can be 'numbers', 'strings', 'Booleans', lists, tuples, sets, or dictionaries.

هل قيمة x أصغر من قيمة y أو تساويها؟	$x \leq y$	$\leq$	أصغر من أو تساوي (Less than or equal to)
--------------------------------------	------------	--------	---

ج. العوامل المُستخدمة في كتابة الشروط المنطقية (Logical Operators)، وهي مُمثَّلة في الجدول (3-2).

الجدول (3-2): العوامل المُستخدمة في كتابة الشروط المنطقية.

اسم العامل	الرمز	مثال توضيحي	الشرح
(Logical AND)	and	x and y	- إرجاع (True) فقط إذا كانت قيمتي x و y هي True - إرجاع (False) إذا كانت قيمة x أو قيمة y أو قيمة كل منهما (False).
(Logical OR)	or	x or y	- إرجاع (False) فقط إذا كانت قيمة x و قيمة y هي False - إرجاع (True) إذا كانت قيمة x أو قيمة y أو قيمة كل منهما (True).
(Logical NOT)	not	not x	- إرجاع (True) إذا كانت قيمة x هي (False) وإرجاع (False) إذا كانت قيمة x هي (True).

أجد ناتج التعبيرات الحسابية والمنطقية الآتية ثم أشارك الناتج مع زملاء / الزميلات:

$$8 // 3 + 2 * 4$$

$$2 ** 3 + 5 * 2$$

$$18 \% 4 + 7 // 2$$

$$\text{not } (4 * 2 > 10)$$

$$(3 + 5 == 8) \text{ and } (4 ** 2 != 16)$$

$$(5 == 2 + 2) \text{ or } (5 > 01)$$



نشاط
فردى

نشاط:

إجابة محتملة.

passed = True

والنتيجة على الحالتين هي نفسها؛ لأن True تقابل الرقم (1) في نظام الحاسوب.

إضاءة

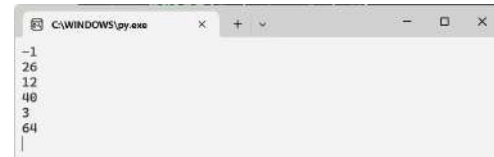
توجد في لغة البرمجة بايثون (Python) عوامل أخرى لم نتطرق إليها في هذا الدرس، مثل العوامل المُستخدمة في البحث في المصفوفات، وسيتمُّ الحديث عنها لاحقاً في هذه الوحدة.

في حال وجود عدد من العوامل لها الأولوية نفسها، فإنَّ ترتيب تنفيذ هذه العوامل يعتمد على قواعد ترابطها (Associativity Rules)، بحيث يكون التنفيذ من اليسار إلى اليمين أو العكس، أمَّا العوامل الموجودة بين الأقواس فلها الأولوية العليا بغض النظر عن نوعها.

مثال:

```
print((1 + 3) - (2 + 3))
print(10 + 2 * 8)
print(5 + 2 - 4 + 9)
print((5 + 5) * 4)
print(5 * 2 // 3)
print((2 ** 3) ** 2)
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية على شاشة جهاز الحاسوب:



أجرب بنفسي: استكشف الأوامر البرمجية الآتية، ثمَّ أتوقع ناتج تنفيذ كلٍّ منها:

```
print(4 + 3 - (2 * 5 / 10))
print(4 * 5 + 5)
print(5 * (2 // 3))
print(2 ** 3 ** 2)
```

أستخدم لغة البرمجة بايثون (Python) في إدخال الأوامر البرمجية وتنفيذها. ما النتيجة التي توصَّلتُ إليها عند تشغيل البرنامج؟ هل اختلف الناتج عن توقُّعاتي؟ أبرر سبب الاختلاف (إن وُجد).

نشاط
عملي

38

نشاط

إجابة محتملة.

الطريقة الثانية:

```
total_grades = grade_1 + \
grade_2 + \
grade_3
print(total_grades)
```

نتيجة تنفيذ البرنامج.

253

```
grade_1 = 85
grade_2 = 90
grade_3 = 78
```

الطريقة الأولى:

```
total_grades = grade_1 + grade_2 + grade_3
print(total_grades)
```

إجابة محتملة.

النتائج النهائية المتوقعة عند تنفيذ

الأوامر:

6,0

25

0

512

ملاحظات:

المواطنة الرقمية:

- استخدام المصادر الرسمية: أحمل برنامج بايثون (Python) من الموقع الإلكتروني الرسمي: python.org، وتعلم من الموارد الموثوقة، والتزم بشروط الترخيص عند استخدام المكتبات أو الرموز (الأكواد) مفتوحة المصدر.
- الأمن السيبراني: استخدم برامج مكافحة الفيروسات، وتجنب تحميل البرامج من مصادر غير موثوقة.
- التعاون المسؤول: أشارك معرفتي بمسؤولية واحترام، وأساعِد الآخرين على نحوٍ فاعل.

المشروع: تصميم برنامج وإعداده لإنشاء لعبة تخمين الأرقام باستخدام لغة البرمجة بايثون (Python) / المهمة (2).

أُكمل - بالتعاون مع أفراد مجموعتي - تنفيذ مشروع التصميم والتطوير للعبة تخمين الأرقام، وأعمل على استكمال التحضيرات اللازمة للعبة؛ بأن أفتح برنامجًا خاصًا بها في برمجة بايثون (Python)، ثم أبدأ بطباعة رسالة الترحيب وخيارات القائمة الرئيسة (تعليمات اللعبة، ابدأ اللعبة، الخروج من اللعبة)، علمًا بأن المقطع البرمجي المطلوب هو:

```
print("Welcome to Stars and Moons")
print("1. Instructions")
print("2. Start Game")
print("3. Exit")
```

أجرب كتابة برنامج يتولى تنفيذ ذلك، ثم أضيف جملة لقراءة الخيار الذي سيختاره اللاعب: (1)، أو (2)، أو (3). غير أنه يتعين الانتظار قليلًا إلى حين إضافة (قراءة مدخلات المُستخدم) إلى البرنامج:

```
option = input("Please select an option (1, 2, 3): ")
```



في هذه المرحلة، لا يُمكنني فعل أي شيء بالخيار الذي اختاره اللاعب، وأراعي الإدخال الصحيح للأوامر البرمجية أثناء كتابة المقطع البرمجي، ثم أحتفظ بما كُتب - ضمن المجموعة - في ملف حتى يُمكن لأفراد المجموعة التعديل عليه في الخطوات القادمة.



مشروع

أقيم تعلمي

المعرفة: أوظف في هذا الدرس ما تعلمته من معارف في الإجابة عن الأسئلة الآتية:
السؤال الأول:

1- ما الدالة في لغة البرمجة بايثون (Python) التي تجعل البرنامج تفاعلياً، وتُمكن المُستخدم من إدخال بيانات في البرنامج أثناء عمله؟

2- فيم تختلف كتابة البرنامج في الحالتين الآتيتين:
أ- إدخال المُستخدم نصاً في البرنامج.

ب- إدخال المُستخدم عدداً في البرنامج.

السؤال الثاني: أشرح قواعد كتابة الأسماء في لغة البرمجة بايثون (Python).

السؤال الثالث: ما الفرق بين العامل = والعامل == في لغة البرمجة بايثون (Python)؟ أَدعِم إجابتي بأمثلة.

ثالثاً: الإثراء والتأمل والتقويم.

نشاط عملي:

```
meal= "dates"
money = 0

if(meal=="dates" or meal=="sandwich")
and money >= 5:
    print("Lunch being delivered.")
else:
    print("Not able to deliver lunch.")
```

وناتجه هو:

Not able to deliver lunch.

تعليمات ذات صلة بمنتج التعلم

- أذكر الطلبة بسياق التعلم ودورهم بوصفهم ناشئين في مجال البرمجة، والمطلوب إليهم تطوير لعبة تعليمية تفاعلية باستخدام برنامج بايثون..
- أطلب إلى الطلبة في مجموعاتهم قراءة تعليمات المهمة الواردة، وأقدم لهم التوضيحات اللازمة إن لزم الأمر.
- أوجه الطلبة للتأكد من أن البرنامج يعرض رسالة الترحيب والقائمة الرئيسية، ثم يطلب إلى المستخدم اختيار خيار من بين الخيارات الثلاثة.
- أطلب إلى الطلبة الاحتفاظ بالشفيرة البرمجية المكتوبة في ملف بايثون (.py) ضمن مجموعة المشروع؛ حتى يمكن لأفراد المجموعة التعديل عليه في الخطوات القادمة.
- أوجه الطلبة للتأكد من كتابة البرنامج على نحو صحيح ومنظم، واختبار البرنامج للتأكد من أنه يعمل من دون أخطاء.

المهارات: أوظف مهارات التفكير الناقد والتواصل الرقمي والبحث الرقمي في الإجابة عن الأسئلة الآتية:

السؤال الأول: اتبع البرنامج الآتي من دون تشغيله، ثم أعدد النتيجة المترتبة على عملية التشغيل إذا أدخل المستخدم العدد (1)، ثم العدد (2) ثم العدد (3)، ثم العدد (4).

```
x = int(input("Enter x="))
print(5 * (x // 3))

A= 30 - (2*3)*x +10
B= (11*3) //7
C= x +A-B
print A,B,C
```

السؤال الثاني: اتبع البرنامج الآتي من دون تشغيله، ثم أعدد النتيجة المترتبة على عملية التشغيل.

```
print (6 + 7 < 2 -1 and 3 ==4)
print ( 2 % 4 == 0 or 7 != 5)
print (2<8 and 6>0 or 10 != 10)
```

السؤال الثالث: أكتشف الأخطاء الواردة في البرنامج الآتي من دون تنفيذه.

```
1st_funding = int(input("Enter 1st funding "))
2nd_funding = int(input("Enter 2nd funding "))
raise = 1st_funding + 2nd_funding
print("Raise =", raise)
```

القيم والاتجاهات:

أستخدم إحدى الأدوات التقنية في إعداد قاموس ناطق لمصطلحات لغة البرمجة بايثون (Python)، لمساعدة الطلبة ذوي الإعاقة ثم أعرضه على زملائي / زميلاتي في الصف.

- أوجه الطلبة لمناقشة أفراد المجموعة حول كيفية تطوير البرنامج لإضافة المزيد من الوظائف، مثل عرض التعليمات عند اختيار (1)، أو بدء اللعبة عند اختيار (2)، أو الخروج من اللعبة عند اختيار (3).
- أشارك مع الطلبة معايير تقييم المهمة:
- جودة الشيفرة البرمجية: يجب أن تكون الشيفرة خالية من الأخطاء النحوية والبرمجية، ويجب أن تكون الشيفرة البرمجية مكتوبة على نحو يتبع معايير البرمجة الجيدة.
- التعليقات: يجب أن تحتوي الشيفرة البرمجية على تعليقات توضيحية للأجزاء المهمة، مثل ما تقوم به كل دالة.
- التنظيم: يجب أن تكون الشيفرة البرمجية منظمة بطريقة تسهل قراءتها وفهمها، مع تقسيمها إلى أجزاء منطقية.
- التعاون والعمل الجماعي: يجب أن يكون هناك توزيع عادل ومنطقي للمهام بين أعضاء المجموعة بناءً على مهاراتهم وقدراتهم.

إجابات نموذجية لأسئلة أقيّم تعلمي

المعرفة: أوظف في هذا الدرس ما تعلّمته من معرفة للإجابة عن الأسئلة الآتية:
السؤال الأول: ما الدالة التي تُستخدم في لغة بايثون لجعل البرنامج تفاعلي بحيث يمكن للمستخدم إدخال بيانات إلى البرنامج في أثناء عمله؟

السؤال الثاني: اشرح قواعد كتابة الأسماء في لغة بايثون.

احتواء المُعرّف فقط على أحد الحروف (a-z)، أو أحد الحروف

(A-Z)، أو الأعداد (0-9)، أو الشرطة السفلية (_).

أن يبدأ المُعرّف بأحد الحروف الكبيرة (A-Z)، أو أحد الحروف الصغيرة

(a-z)، أو الشرطة السفلية (_).

عدم بدء المُعرّف بعدد، ومنع استخدام أيّ كلمة من الكلمات المحجوزة.



الدرس الثالث

الجمل الشرطية (Conditional Statements)

input()

الفكرة الرئيسية:

تعرّف الجمل الشرطية في لغة البرمجة بايثون (Python)، وبيان كيف يُمكن كتابتها واستخدامها في تعليق تنفيذ أوامر معينة بناءً على شروط يُحددها المُبرمج، وإجراء تطبيقات عملية لتعزيز الفهم.

المفاهيم والمصطلحات:

الجمل الشرطية (Conditional Statements).

نتائج التعلم (Learning Outcomes):

■ أكتب جملاً شرطية مركبة ومترابطة باستخدام المُعاملات المنطقية (مثل: and، or، not) في لغة البرمجة بايثون (Python).

■ أستخدم لغة البرمجة بايثون (Python) في إنشاء برامج تتضمن جملاً شرطية، وأتبع نتائجها وأنفذها.

لا بد أنك تعرفت الجمل الشرطية وطرق كتابتها في اللغة العربية واللغة الإنجليزية، هل يوجد ارتباط بين مكونات جملة الشرط في هذه اللغات ومكوناتها في لغات البرمجة؟

مُنتجات التعلم (Learning Products)

تعديل البرنامج غير إضافة الجمل الشرطية لاتخاذ القرارات بناءً على مدخلات المستخدم، ضمن سياق تصميم لعبة تضمين الأرقام باستخدام دالة int مع دالة input()
int(input())

انظر الهامش

انظر الهامش

السؤال الثالث: ما الفرق بين العامل = والعامل == في لغة بايثون؟ أوضح بأمثلة.

= : عامل الإسناد الأساسي.

مثال:

x = 5

لإعطاء المتغير x قيمة 5.

== : عامل من عوامل المقارنة.

مثال: x == y معناه هل قيمة x تساوي قيمة y.

المهارات: أوظف المهارات التي تعلّمتها للإجابة عن الأسئلة الآتية

السؤال الأول: ما نتيجة تشغيل البرنامج الآتي؛ إذا قام المستخدم بإدخال الأعداد 1 ثم 2 ثم 3 ثم 4؟ أجب عن السؤال عن طريق تتبع البرنامج من دون تشغيله.

```
x = int(input("Enter x="))
print(5 * (x // 3))
```

الإجابة:

عند إدخال 1: النتيجة 0.

عند إدخال 2: النتيجة 0.

عند إدخال 3: النتيجة 5.

عند إدخال 4: النتيجة 5.

السؤال الثاني: ما نتيجة تشغيل البرنامج الآتي؟ أجب عن السؤال عن طريق تتبع البرنامج من دون تشغيله.

السؤال الثاني: ما نتيجة تشغيل البرنامج الآتي؟ أجب عن السؤال عن طريق تتبع البرنامج من دون تشغيله.

```
meal = "dates"
money = 10

if meal == "dates" or meal == "sandwich" and money >= 5:
    print("Lunch being delivered.")
else:
    print("Not able to deliver lunch.")
```

الإجابة:

Lunch being delivered.

لعبة البطاقات الشرطية

1. أحضر - بالتعاون مع أفراد مجموعتي - عددًا من البطاقات المرقّمة من (0) إلى (01)، ثمّ أتنافس مع أفراد المجموعات الأخرى للفوز في إحدى الألعاب.

2. أبدأ اللعبة بالطلب إلى أحد زملائي / إحدى زميلاتي في المجموعة سحب بطاقة، وأتباع القواعد الآتية بناءً على العدد المُدوّن في البطاقة:

أ. إذا كان العدد المُدوّن في البطاقة أقل من (5)، فإنني أسحب بطاقة أخرى.

ب. إذا كان العدد المُدوّن في البطاقة أكبر من (5)، وكان من الأعداد الزوجية، فإن الدور ينتقل إلى زميل آخر / زميلة أخرى في مجموعتي.

ج. إذا كان العدد المُدوّن في البطاقة أكبر من (5)، وكان من الأعداد الفردية، فإن الدور ينتقل إلى مجموعة أخرى.

د. إذا كان العدد المُدوّن في البطاقة (0)، فإن المجموعة تخرج من اللعبة.

3. ألقيت انتباه الجميع إلى وجوب تكرار سحب البطاقات بالتناوب بين المجموعات، وتنفيذ القواعد السابقة.

4. أجتب مع أفراد مجموعتي - بعد انتهاء اللعبة، أو انتهاء الوقت المُحدّد لها-، ثمّ أناقشهم في ما تعلّمناه.

كيف تُشبه هذه اللعبة الجمل الشرطية في البرمجة؟ أدوّن توقّعاتي.

أنواع الجمل الشرطية في بايثون (Python)

تُستخدم الجمل الشرطية في تنفيذ مجموعة من الأوامر البرمجية في البرنامج بناءً على شروط تحددها المبرمج، وهي تمتاز بتعدد أشكالها، وتفرّد كل جملة منها بصيغة عامة تُحدّد طريقة تنفيذها في البرنامج.

أولاً: الجملة الشرطية (if statement)

تُكتب الصيغة العامة للجملة الشرطية (if) على النحو الآتي:

```
if condition:
    statements
```

السؤال الثالث: اكتشف الأخطاء الموجودة في البرنامج الآتي من دون تنفيذه.

```
1st_funding = int(input("Enter 1st funding "))
2nd_funding = int(input("Enter 2nd funding "))
raise = 1st_funding + 2nd_funding
print("Raise =", raise)
```

الإجابة

1. لا يجوز أن يبدأ اسم المتغير برقم ومن ثمَّ `1st_funding` و `2nd_funding` أسماء غير مقبولة
2. `raise` هي كلمة محجوزة.

البرنامج المصحح:

```
first_funding = int(input("Enter 1st funding="))
second_funding = int(input("Enter 2nd funding="))
total_funding = first_funding + second_funding
print("Total funding =", total_funding)
```

ثالثاً: مرحلة الإثراء والتأمل والتقويم

البحث والإثراء:

- أكلف الطلبة نشاطاً بيتياً فردياً؛ بحيث يبحثون في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن أمثلة أخرى على منصات التعلم الإلكترونية مفتوحة المصدر (MOOCs).
- أنشئ رابطاً للوصول إلى اللوح الرقمي التفاعلي مثل: (FigJam، Padlet، ..).
- أشارك الرابط مع الطلبة، وأطلب إليهم مشاركة ما يتوصلون له من نتائج عبر اللوح الرقمي التفاعلي.
- أكلف الطلبة الاطلاع على منصتين من المنصات التي يشاركنها الزملاء، وإعطاء رأيهم فيها.

الجملة الشرطية Conditional Statements

عدد الحصص المقترح:
أربع حصص.

المصادر:

كتاب الطالب، لغة البرمجة بايثون.

الأدوات والتجهيزات:

أجهزة حاسوب، جهاز للعرض.

حيث:

if: كلمة محجوزة في لغة البرمجة بايثون (Python).

condition: الشرط (تعبير منطقي).

(:): علامة يجب أن توضع بعد الشرط (condition) ونسيانها يُفضي إلى خطأ في البرنامج من نوع

.syntax error

statements: أوامر برمجية تُنفَّذ إذا كان الشرط صحيحًا (تحقق الشرط).

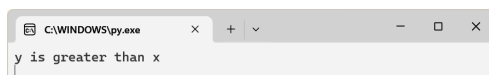
مثال:

يطبع البرنامج الآتي عبارة "y is greater than x" إذا كانت قيمة المُتغيِّر (y) أكبر من قيمة المُتغيِّر (x):

```
x = 3
y = 20

if y > x:
    print("y is greater than x")
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية على شاشة جهاز الحاسوب:



إضافة

■ يُمكن كتابة الجملة الشرطية السابقة بصورة مختصرة كما يأتي:

```
if y > x: print("y is greater than x")
```

■ يجب أن تكون المسافات البادئة صحيحة؛ لكي يعمل المقطع البرمجي (الكود) بصورة صحيحة في لغة البرمجة بايثون (Python).

■ أتأكد دائمًا أن الأسطر التي تتبع الشرط (مثل السطر الذي يحتوي على كلمة (print) تحوي مسافة بادئة (تبدأ بمسافة واحدة أو Tab واحدة).

أولاً: التهيئة

سياق التعلّم:

أوضح للطلبة الفكرة الرئيسة من الدرس، وأضعه في قالب سياق التعلّم، وأخبرهم أنهم سيكملون رحلة التعلّم التي بدؤوا بها في عالم البرمجة، وتصميم الألعاب؛ إذ سيتمكنون في هذا الدرس من تطوير لعبتهم أكثر عبر جعل اللعبة أكثر تفاعلية عن طريق إضافة الجملة الشرطية.

أوضح لهم منتج التعلّم المتوقع إنجازه خلال هذا الدرس؛ إذ يُتوقع تعديل البرنامج بإضافة الجملة الشرطية اللازمة لاتخاذ القرارات بناءً على مدخلات اللاعب، ضمن سياق تصميم لعبة تخمين الأرقام باستخدام برمجة بايثون (Python).

نشاط تمهيدي:

إجابة محتملة: ستتغير الإجابات حول وجه الشبة بين اللعبة والجمل الشرطية في البرمجة. ومن بين الإجابات الممكنة:

هذه اللعبة تشبه الجمل الشرطية في البرمجة على نحو كبير؛ إذ تقوم اللعبة على مجموعة من القواعد التي يجب اتباعها بناءً على حالة معينة، تمامًا مثل الجمل الشرطية التي تُنفَّذ عندما تتحقق شروط معينة في البرمجة. إليك التشابهات:

■ البطاقات والقواعد: في اللعبة، يعتمد ما يحدث بعد سحب البطاقة على الرقم المكتوب عليها (الشرط). في البرمجة، تُنفَّذ التعليمات بناءً على شروط محددة تتحقق أو لا تتحقق.



نشاط
عملي

أُجَرَّب بنفسني: أعدّل المقطع البرمجي (الكود) في المثال السابق لتعيين قيمة (03) للمتغير (x)، وأتبع النتيجة المُتوقَّعة من دون تشغيل البرنامج، ثمّ أتحدّق من الناتج عن طريق تنفيذ البرنامج في بيئة بايثون (Python).

ما الذي يجب تعديله في البرنامج لطباعة جملة "x is greater than y" بدلاً من جملة "y is greater than x"؟

ثانيًا: الجملة الشرطية (if else)

تُكتب الصيغة العامة للجملة الشرطية (if else) على النحو الآتي:

```
if condition:
    statements1
else:
    statements2
```

حيث:

if, else: كلمتان محجوزتان في لغة البرمجة بايثون (Python).

statements1: أوامر برمجية تُنفَّذ إذا كان الشرط صحيحًا (تحقق الشرط).

statements2: أوامر برمجية تُنفَّذ إذا لم يكن الشرط صحيحًا (عدم تحقق الشرط).

(:): علامة يجب أن توضع بعد الشرط (condition) و بعد جملة else.

مثال:

يطبع البرنامج الآتي عبارة "y is less than x" إذا كانت قيمة المتغير (y) أقل من قيمة المتغير (x)، ويطبع البرنامج عبارة "x is less than y" إذا كانت قيمة المتغير (x) أقل من قيمة المتغير (y):

```
x = 80
y = 100
if y < x:
    print("y is less than x")
else:
    print("x is less than y")
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية:



إضاءة

يُمكن كتابة الجملة الشرطية السابقة بصورة مختصرة كما يأتي:

```
x = 80
y = 100
print("y is less than x") if y < x else print("x is less than y")
```

49

- شروط مختلفة: إذا كان الرقم أقل من 5، يُنفَّذ إجراء معين (سحب بطاقة أخرى). في البرمجة، يكون لدينا شروط مختلفة، تؤدي إلى نتائج مختلفة بناءً على تحققها، مثل if-else.
- تغيير الأدوار أو الانتقال إلى مجموعة أخرى: عندما يتحقق شرط معين (مثل رقم أكبر من 5 ومن الأعداد الفردية)، يُتغيّر الدور أو يُنتقل إلى مجموعة أخرى. في البرمجة، يمكن أن تؤدي الجملة الشرطية إلى تغيير في تدفق البرنامج بناءً على الشروط.
- الخروج من اللعبة عند سحب الرقم 0: يشبه حالة إنهاء البرنامج، أو إنهاء عملية معينة عندما يتحقق شرط محدد، مثل استخدام break أو return في البرمجة.

ثانيًا: عملية التعليم والتعلم:



نشاط
فردى

ملاحظاتى

أكتب مقطعًا برمجياً بلغة البرمجة بايثون (Python) لحساب قيمة المتغير (z) وفقاً لكل معادلة مما يأتي وطباعته باستخدام جملة if:

$$z = x^2 * y, x > 5$$

$$z = x * (x + y), x == 5$$

رابعًا: الجملة الشرطية (if elif else)

تُكتب الصيغة العامة للجملة الشرطية (if elif else) على النحو الآتي:

```
if condition1:
    statements1
elif condition2:
    statements2
else:
    statements3
```

حيث:

if, else, elif: كلمات محجوزة في لغة البرمجة بايثون (Python).

statements1: أوامر برمجية تُنفَّذ إذا تحقَّق الشرط (condition1).

statements2: أوامر برمجية تُنفَّذ إذا تحقَّق الشرط (condition2).

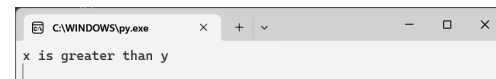
statements3: أوامر برمجية تُنفَّذ إذا لم يتحقَّق أيُّ من الشرطين (condition2, condition1).

مثال:

يطبع البرنامج الآتي عبارة "y is greater than x" إذا كانت قيمة المتغير (y) أكبر من قيمة المتغير (x)، ويطبع البرنامج عبارة "x equals y" إذا كانت قيمة المتغير (x) تساوي قيمة المتغير (y)، ويطبع البرنامج عبارة "x is greater than y" إذا كانت قيمة المتغير (y) أصغر من قيمة المتغير (x):

```
x = 50
y = 25
if y > x:
    print("y is greater than x")
elif y == x:
    print("x equals y")
else:
    print("x is greater than y")
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية:



51

البرنامج المعدل:

```
x = 30
y = 20
if y > x:
    print("y is greater than x")
```

ناتج التنفيذ هو شاشة فارغة.

التعديل الذي يجب إجراؤه على البرنامج هو تعديل الشرط ليصبح (y < x)، أو إضافة شرط آخر للشرط السابق كما يأتي:

```
x = 30
y = 20
if y > x:
    print("y is greater than x")
if y < x:
    print("x is greater than y")
```


أُجَرَّب بنفسني:

استنادًا إلى المثال السابق، أُجيب عن الأسئلة الآتية:

1. ما النتيجة المترتبة على تشغيل البرنامج إذا كانت قيمة (x) تساوي (40)؟ أعدّل المقطع البرمجي (الكود) في المثال لتعيين قيمة (40) للمتغير (x)، وأتبع النتيجة المتوقعة من دون تشغيل البرنامج، ثمّ أتحقق من الناتج عن طريق تنفيذ البرنامج في بيئة بايثون (Python).
2. ما النتيجة المترتبة على تشغيل البرنامج إذا كانت قيمة (x) تساوي (10)؟ أعدّل المقطع البرمجي (الكود) في المثال لتعيين قيمة (10) للمتغير (x)، وأتبع النتيجة المتوقعة من دون تشغيل البرنامج، ثمّ أتحقق من الناتج عن طريق تنفيذ البرنامج في بيئة بايثون (Python).
3. ما النتيجة المترتبة على تشغيل البرنامج إذا كانت قيمة (x) تساوي (25)؟ أعدّل المقطع البرمجي (الكود) في المثال لتعيين قيمة (25) للمتغير (x)، وأتبع النتيجة المتوقعة من دون تشغيل البرنامج، ثمّ أتحقق من الناتج عن طريق تنفيذ البرنامج في بيئة بايثون (Python).

نشاط
عملي

أبحث: أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن لغات برمجة أخرى، ثمّ أفرّن طريقة كتابة الجمل الشرطية فيها بطريقة كتابة الجمل الشرطية في لغة البرمجة بايثون (Python).

أبحث

المعاملات المنطقية (Logical Operators)

يُستعمل المُعامل المنطقي (and) والمُعامل المنطقي (or) لربط التعبيرات المنطقية البسيطة، وتكوين جمل منطقية مركبة، في حين يُستعمل المُعامل المنطقي (not) لنفي التعبيرات المنطقية.

1- المُعامل المنطقي (and):

قد يتوقّف تنفيذ أمر برمجي مُعيّن في البرنامج على تحقّق مجموعة من الشروط مُجمّعة.

مثال:

يراد طباعة كلمة "Excellent" إذا كان مُعدّل الطالب / الطالبة أكبر من أو يساوي (90) وأقل من أو يساوي (100).

الحل:

يجب استعمال المُعامل المنطقي (and) للدلالة على تحقّق الشرطين معًا، وتُكتب الأوامر البرمجية على النحو الآتي:

52

نشاط عملي:

الناتج:

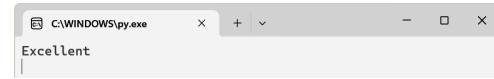
x is less than y

نشاط عملي:

الناتج: لن يطبع البرنامج أيّ شيء (ستظهر شاشة مخرجات فارغة).

```
Avg = 95
if Avg >= 90 and Avg <=100:
    print("Excellent")
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية:



أُجِرْ بنفسِي: أكتب المقطع البرمجي (الكود) السابق، ثمَّ أتحقّق من النتيجة عن طريق تنفيذ البرنامج. أعدل قيمة المُتغيّر (Avg) لتصبح (08)، ثمَّ أنفِذ البرنامج. ما الناتج الظاهر على شاشة جهاز الحاسوب؟ أحذف المسافة البادئة قبل جملة الطباعة، ثمَّ أنفِذ البرنامج. ما ناتج تنفيذ البرنامج؟

2- المُعامل المنطقي (or):

قد يتوقّف تنفيذ أمر مُعيّن في البرنامج على تحقّق شرط من مجموعة شروط.

مثال:

يراد طباعة قيمة المُتغيّر (x) إذا كانت قيمة هذا المُتغيّر تساوي (1) أو (2).

الحل:

يجب استعمال المُعامل المنطقي (or) للدلالة على تنفيذ جملة الطباعة في حال تحقّق أحد الشرطين، وتُكتب الأوامر البرمجية على النحو الآتي:

```
x = 1
if x == 1 or x == 2:
    print('you selected a valid number')
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية:



3- المُعامل المنطقي (not):

قد يتوقّف تنفيذ أمر في البرنامج على عدم تحقّق شرط مُعيّن.

نشاط فردي:

الناتج:

```
x = int(input('Enter the value of x:'))
y = int(input('Enter the value of y:'))
if x > 5:
    z = x**2 * y
elif x == 5:
    z = x * (x + y)
print('z = ', z)
```

ملحوظة: ينوه المعلم للطلبة بوجود خطأ في السؤال (تكرار إشارة المساواة في الشرطين ويصحح السؤال بإزالة المساواة من الشرط الاول ليصبح $x > 5$)

مثال:

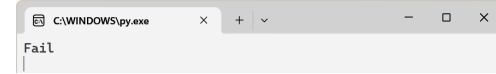
يراد طباعة كلمة "Fail" إذا لم يُحقّق الطالب / الطالبة شرط النجاح ($Avg \geq 50$).

الحل:

يجب استعمال المُعامل المنطقي (not) للدلالة على عدم تحقّق الشرط، وتُكتب الأوامر البرمجية على النحو الآتي:

```
Avg = 49
if not Avg >= 50:
    print("Fail")
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية:



- أكتب المقطع البرمجي (الكود) السابق بطريقة أخرى، ثمّ أتحدّق من النتيجة عن طريق تنفيذ البرنامج.
- أعدّل قيمة المُتغيّر (Avg) لتصبح (07)، ثمّ أنفّذ البرنامج. ما الناتج الظاهر على شاشة جهاز الحاسوب؟



نشاط
فردى

المُعاملات المنطقية والجمل الشرطية

أكتب - بالتعاون مع أفراد مجموعتي - مقطعاً برمجياً بلغة البرمجة بايثون (Python) لإدخال قيمة إحدى فواتير الشراء وعدد المشتريات، وطباعة عبارة "No discount" إذا كانت قيمة الفاتورة أقل من (001)، أو كان عدد المشتريات أقل من (3)، وطباعة قيمة الخصم الذي يساوي 5% من قيمة الفاتورة إذا كانت قيمتها أكبر من (001)، وكان عدد المشتريات أكبر من أو يساوي (3).



نشاط
جماعي

الجملة الشرطية المُركّبة (Nested Conditional Statements)

توفّر لغة البرمجة بايثون (Python) إمكانية كتابة جمل شرطية مُركّبة (مُتداخلة)؛ أيّ وضع جملة شرطية (if) داخل جملة شرطية (if) أخرى.

مثال:

يطبع البرنامج الآتي عبارة "Above ten" إذا كانت قيمة (x) أكبر من (01)، ثمّ يتحدّق إذا كانت قيمة (x) أكبر من (06). فيطبع البرنامج، إضافة إلى العبارة السابقة، عبارة "and also above 06" إذا كانت قيمة (x) أكبر من (06)، ويطبع عبارة "but not above 60" إذا كانت قيمة (x) أقل من (06).

54

نشاط عملي:

1. الناتج:

x is greater than y

2. الناتج:

y is greater than x

3. الناتج:

x equals y

جملة الشرط في لغة بايثون :

الصيغة العامة:

if condition:

الكود الذي يُنفذ إذا كانت الشرط صحيحًا.

الشرط (condition) تعبير قيمته منطقية True أو False.

الترقيم والمسافات الفارغة (Indentation) تستخدم Python لتحديد كتل البرمجة. أي كود بمسافة بادئة تحت جملة if يعدّ جزءًا من تلك الكتلة.

جملة الشرط في لغة C++ :

الصيغة العامة:

{ if (condition)

// الكود الذي يُنفذ إذا كانت الشرط صحيحًا

}

الشرط (condition) تعبير قيمته منطقية True أو False.

الأقواس المعقوفة {}: تُستخدم لتحديد بداية الكتلة البرمجية ونهايتها التابعة لجملة if.

الفروقات الرئيسة بين Python و C++ في استخدام جملة الشرط:

■ التحديد الهيكلي: في Python، تُحدّد كتل البرمجة باستخدام المسافات الفارغة (Indentation)، وهذا يعني أن المسافات الفراغية أو التبويب مهم جدًا.

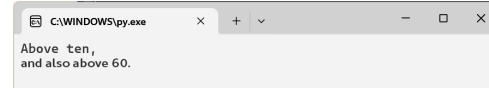
■ في C++، تُستخدم الأقواس المعقوفة {} لتحديد بداية كتل البرمجة ونهايتها.

■ إنهاء الأسطر: في Python، لا حاجة لاستخدام فاصلة منقوطة ";" لإنهاء الأسطر. في C++، يجب إنهاء معظم الأسطر بفاصلة منقوطة ";".

■ طبيعة الشروط: في Python، تُقيّم الشروط إلى قيم منطقية True أو False. في C++، أي قيمة غير صفرية تعدّ true، ويعدّ الصفر false.

```
x = 68
if x > 10:
    print("Above ten,")
    if x > 60:
        print("and also above 60.")
    else:
        print("but not above 60.")
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية:



نشاط فردي

- أُعِدَّ قيمة المُتغيِّر (x) لتصبح (52)، ثمَّ أنفِذ البرنامج. ما الناتج الظاهر على شاشة جهاز الحاسوب؟
- أُعِدَّ قيمة المُتغيِّر (x) لتصبح (5)، ثمَّ أنفِذ البرنامج. ما الناتج الظاهر على شاشة جهاز الحاسوب؟
- أهدف المسافة البادئة قبل جملة الطباعة الأخيرة. ما ناتج تنفيذ البرنامج؟ أناقش إجاباتي مع زملائي/ زميلاتي في الصف.

مثال:

يُستخدَم البرنامج الآتي في التحقق إذا كانت قيمة المُتغيِّر (y) تقبل القسمة على (3)، وتقبل القسمة على (2)، أو تقبل القسمة على أحدهما، أو لا تقبل القسمة على أي منهما. ثمَّ يطبع البرنامج العبارة الدالة على ذلك.

```
y = 9
if y%2==0:
    if y%3==0:
        print("divisible by 3 and 2")
    else:
        print("divisible by 2, but not divisible by 3")
else:
    if y%3==0:
        print("divisible by 3, but not divisible by 2")
    else:
        print("not divisible by 2 and not divisible by 3")
```

نشاط عملي:

في حال كانت قيمة Avg تساوي 80 فالناتج هو: شاشة مخرجات فارغة.
في حال حذف المسافة البادئة: لن يُنفِذ البرنامج، وستحصل على رسالة تفيد بوجود خطأ.

ملاحظات

.....

.....

.....

.....

.....

.....

.....

نشاط فردي

1. الطريقة الأخرى لكتابة البرنامج.

```
Avg=49
if Avg<50:
    print("Fail")
```

2. شاشة مخرجات فارغة.

عند تشغيل البرنامج، ستظهر النتيجة الآتية:



أُجَرَّب بنفسي:

استنادًا إلى المثال السابق، أُجيب عن الأسئلة الآتية:

1. ما نتيجة البرنامج إذا كانت قيمة (y) تساوي (8)؟

2. أكتب المقطع البرمجي (الكود) الوارد في المثال السابق في لغة البرمجة بايثون (Python)، وأتبع النتيجة المُتوقَّعة من دون تشغيل البرنامج، ثمَّ أتحقق من الناتج عن طريق تنفيذ البرنامج.

3. ما نتيجة البرنامج إذا كانت قيمة (y) تساوي (5)؟



إضاءة



إذا أردت كتابة جملة شرطية (if) فارغة لا تحتوي على أي من الجمل، فإنني أضع الجملة (pass) داخل هذه الجملة الشرطية (if)؛ لتجنُّب ظهور رسالة تفيد بوجود خطأ في البرنامج.

أُجَرَّب بنفسي:

أتبع تنفيذ البرنامج الآتي في بيئة بايثون (Python)، بوجود جملة (pass) تارة، وعدم وجودها تارة أخرى.

```
x = 4
y = 2
if x > y:
    pass
```



56

نشاط جماعي

البرنامج:

```
value = int(input('Enter the bill value:'))
items = int(input('Enter the number of items in the bill:'))
if value < 100 or items < 3:
    print('No discount')
else:
    print('The discount is', 0.05 * value)
```

ملحوظة: ألقت نظر الطلبة إلى ضرورة وجود الشرط "أكبر من أو يساوي 100 ضمنيًا في الحل، وإن لم يُذكر في نص السؤال؛ ليكون حل السؤال منطقيًا.

نشاط
جماعي

اكتشاف الأخطاء في المقطع البرمجي بلغة البرمجة بايثون (Python)
أحلل - بالتعاون مع أفراد مجموعتي - المقطع البرمجي الآتي، وأكتشف الأخطاء الواردة فيه
من دون تنفيذه، وأعمل على تصحيح هذه الأخطاء، ثم أنفذ المقطع البرمجي بعد تصحيحه.

```
grade = input("Enter your grade: ")

if grade >= 85:
    print("A")
elif grade >= 75:
    print("B")
elif grade >= 65:
    print("C")
else:
    print("F")
```

المواطنة الرقمية:

- التعاون والمشاركة: أتعاون مع الزملاء / الزميلات، وأشاركهم في تنفيذ الأنشطة واكتشاف الأخطاء البرمجية وتحليلها؛ ما يُعزِّز لديَّ مهارة التفكير الناقد والعمل الجماعي، ويسهم في بناء مجتمع تعليمي داعم ومُشارك.
- الأخلاقيات الرقمية: أحترم حقوق المُلكية الفكرية عند استخدام المقاطع البرمجية الموجودة في شبكة الإنترنت، أو عند التعديل عليها، وأحرص على أخذ الموافقة المُسبقة على ذلك.



مشروع

المشروع:

أكمل - بالتعاون مع أفراد مجموعتي - تنفيذ مشروع التصميم والتطوير للعبة تخمين الأرقام؛ بأن أطلع تعليمات اللعبة عند اختيار اللاعب الرقم (1) من القائمة. وبناءً على ما تعلّمناه في هذا الدرس، سأعمل - ضمن المجموعة - على تعديل ما كتبناه سابقاً (باستخدام الجمل الشرطية) كما يأتي:

- 1- طباعة شرح عن تعليمات اللعبة إن أدخل اللاعب الرقم (1).
 - 2- طباعة جملة: (لم يتم تنفيذ هذه الخصيصة بعد) إذا اختار اللاعب الرقم (2) أو الرقم (3).
 - 3- طباعة جملة: (الخيار المُدخل غير معروف، أدخل 1 أو 2 أو 3) إذا أدخل اللاعب رقماً غير الأرقام المُشار إليها.
- بعد ذلك أتأكد - مع أفراد مجموعتي - أن الشروط قد كُتبت بصورة صحيحة، وأنحقق من استجابة البرنامج للخيارات المُدخلة بشكل مناسب، ثم أحتفظ بما كُتب - ضمن المجموعة - في ملف حتى يُمكن لأفراد المجموعة التعديل عليه في الخطوات القادمة.

نشاط فردي:

1. الناتج:

2. الناتج: شاشة فارغة.

Above ten,
but not above 60.

3. الناتج: رسالة خطأ.

نشاط عملي:

4. الناتج:

divisible by 2, but not divisible
by 3

5. النتائج

divisible by 2, but not divisible
by 3

6. النتائج

not divisible by 2 and not
divisible by 3

ملاحظات

.....

.....

.....

.....

.....

.....

.....

.....

.....

أَقِيْمُ تَعَلُّمِي

المعرفة: أُوْظِفُ في هذا الدرسِ ما تعلَّمْتُهُ مِنْ معارفٍ في الإجابةِ عنِ الأسئلةِ الآتيةِ:

السؤال الأول: ما الفرق بين الجملة الشرطية (if else) والجملة الشرطية (if elif) في لغة البرمجة بايثون (Python)؟

السؤال الثاني: أذكر أمثلة على استخدام جملة (pass) داخل الجملة الشرطية (if).

المهارات: أُوظِّفُ مهارات التفكير الناقد والتواصل الرقمي والبحث الرقمي في الإجابة عن الأسئلة الآتية:

السؤال الأول: أتبّع البرنامج الآتي من دون تشغيله، ثمّ أذكر النتيجة المترتبة على تشغيله.

```
x = 20
y = 5
z = 30
if not x <= y and x < z:
    print("y < x < z")
```

السؤال الثاني: اقرأ البرنامج التالي المكتوب بلغة البرمجة بايثون (Python)، ثم أجب عن السؤالين الآتيين:

1. ما الهدف الرئيس من البرنامج؟ أصف ما يقوم به البرنامج عامةً من دون وصف وظيفة كل أمر برمجي فيه.

```
x = int(input("Enter your grade: "))
if x > 84:
    print("Excellent grade.")
```

نشاط عملي:

سينفذ البرنامج، ولن يطبع شيئاً في حال وجود جملة pass.

في حال عدم وجود جملة pass، ستظهر رسالة تفيد بوجود خطأ.

```
elif x > 76:
    print("Very good grade.")
elif x > 68:
    print("Good grade.")
elif x > 50:
    print("You passed the course.")
else:
    print("You failed the course.")
```

2. ما النتيجة المُتَربِّتة على تشغيل البرنامج إذا أدخل المُستخدِم العدد (94)، ثمَّ العدد (55)، ثمَّ العدد (86)، ثمَّ العدد (07)، ثمَّ العدد (09)، ثمَّ العدد (87)، ثمَّ العدد (67)؟ السؤال الثالث: أكتشف الأخطاء الواردة في البرنامج الآتي من دون تنفيذه.

```
n = int(input("Enter any natural number: "))

if n <= 0:
    print("Wrong input. Please enter a positive number.")
else: sum = 0
IF    n > 0:
Print N
print("The sum of the natural numbers is: ", sum)
```

القيَم والاتجاهات:

أبحث في تطبيقات الجمل الشرطية الحياتية، ثمَّ أجمعها وأوثقها في ملف حفظاً لحقوق الملكية الفكرية ثمَّ أنشرها عبر المواقع الإلكترونية للمدرسة بهدف الإسهام في نشر المعرفة.

```
grade = int(input("Enter your grade: "))

if grade >= 85:
    print("A")
elif grade >= 75:
    print("B")
elif grade >= 65:
    print("C")
else:
    print("F")
```

إثراء للمعلم والمعلمة.

لمزيد من المعلومات والأمثلة حول الجمل الشرطية، يمكنك زيارة المواقع الإلكترونية الآتية:

<https://www.geeksforgeeks.org/conditional-statements-in-python/>

<https://www.programiz.com/python-programming/if-elif-else>

تعليمات ذات صلة بمنتج التعلم

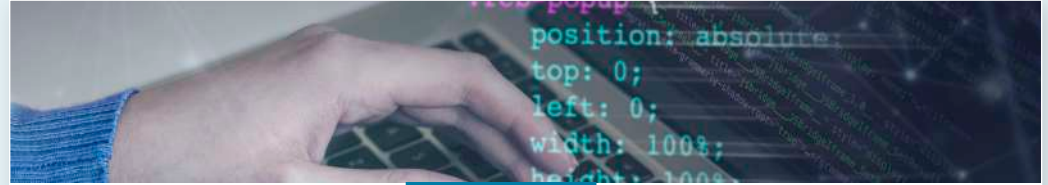
- أذكرُ الطلبة بسياق التعلم ودورهم بوصفهم ناشئين في مجال البرمجة، وأوضح لهم أن الهدف في هذه المرحلة، ضمن سياق التطوير المتكرر للبرمجيات تعديل برنامج لعبة التخمين التفاعلية باستخدام Python بإضافة الجمل الشرطية، لاتخاذ القرارات بناءً على مدخلات اللاعب.
- أطلب إلى الطلبة في مجموعاتهم قراءة تعليمات المهمة الواردة، وأقدم لهم التوضيحات اللازمة إن لزم الأمر.
- التطوير للعبة تخمين الأرقام؛ بأن أطبع تعليمات اللعبة عند اختيار اللاعب الرقم (1) من القائمة. وبناءً على ما تم تعلّمه في هذا الدرس، سيقوم بالآتي:
 - طباعة شرح عن تعليمات اللعبة إن أدخل اللاعب الرقم (1).
 - طباعة جملة: "لم تُنفذ هذه الخصيصة بعد" إذا اختار اللاعب الرقم (2) أو الرقم (3).
 - طباعة جملة: (الخيار المُدخل غير معروف، أدخل 1 أو 2 أو 3) إذا أدخل اللاعب رقمًا غير الأرقام المُشار إليها.
- أطلب إلى الطلبة الاحتفاظ بالمقطع البرمجي المكتوب في ملف بايثون (.py)، ضمن مجموعة المشروع؛ حتى يمكن لأفراد المجموعة التعديل عليه في الخطوات القادمة.
- أوجه الطلبة للتأكد من كتابة البرنامج على نحو صحيح ومنظم، واختبار البرنامج للتأكد من أنه يعمل من دون أخطاء، وأتحقق من استجابة البرنامج.
- أشارك مع الطلبة معايير تقييم المهمة، وهي:
 - الدقة في كتابة البرنامج: استخدام الجمل الشرطية على نحو صحيح. المقطع البرمجي منظم ويحتوي على تعليقات توضيحية.
 - الوظائف المطلوبة: يستجيب المقطع البرمجي للخيارات المُدخلة على نحو مناسب وصحيح.
 - تطبيق إجراءات الاختبار، والتأكد من عدم وجود أخطاء.
 - التعاون والعمل الجماعي.
 - حفظ الملف على نحو صحيح، ومشاركة أفراد المجموعة في العملية.

إجابات نموذجية لأسئلة أقيم تعلّمي

المعرفة: أوظّف في هذا الدرس ما تعلّمتُه
من معارف في الإجابة عن السؤالين
الآتين:
السؤال الأول

في جملة if else تنفذ الجمل statements1
في حال تحقق الشرط داخل جملة
if، وتنفّذ الجمل statements2 في كل
الحالات التي لا يتحقق فيها الشرط
داخل جملة if.

تشارك جملة if elif مع جملة if else
في الجزئية الأولى وهي تنفيذ الجمل
statements1 في حال تحقق الشرط
داخل جملة if، أما بالنسبة للجمل
statements2 فلا تُنفّذ في جميع الحالات
التي لا يتحقق فيها الشرط داخل جملة
if، إنما يتم تنفيذها فقط في حال تحقق
الشرط داخل جملة elif.



الدرس الرابع

الحلقات (Loops)

أنظر الهامش

الفكرة الرئيسية:

تعرف كيف تُكتب جمل التحكم في برمجية بايثون (Python) باستخدام المُعاملات المنطقية، واختيار أكثر الهياكل البرمجية مناسبة لحلّ مشكلات مُعيّنة بكفاءة وتفسير سبب استخدامها، وكتابة مقاطع برمجية بسيطة وواضحة بحيث يُسهّل على الآخرين فهمها وصيانتها.

المفاهيم والمصطلحات:

الحلقات (Loops)، جمل التحكم (Control Statements)، التهيئة (Initialization)، جمل (Statements)، الزيادة (Increment)، النقصان (Decrement)، العنصر (Element)، المصفوفة (Sequence)، النطاق (Range).

نتائج التعلّم (Learning Outcomes):

- أكتب جمل التحكم باستخدام الحلقات (مثل: For، و While) في برمجية بايثون (Python).
- أستخدم أكثر الهياكل البرمجية (مثل: الحلقات، والجمل الشرطية) مناسبة لحلّ مشكلات مُعيّنة بكفاءة.
- أكتب برامج مُتكاملة على نحو يُسهّل على الآخرين قراءتها وفهمها.
- أتعرف جمل التكرار وأذكر أنواعها.
- أقارن ناتج تنفيذ جمل التحكم (break، continue).
- تعرّفُ سابقاً الحلقات في برمجية سكراتش (Scratch)، واستخدمتها في كتابة البرامج التي تتطلّب تكرار تنفيذ

مُنتجات التعلّم أنظر الهامش (Learning Products)

تعديل البرنامج بإضافة حلقات تكرارية تُمكن من عرض القائمة بصورة مُتكررة حتّى يتم اختيار الخيار الصحيح، ضمن سياق تصميم لعبة تضمين الأرقام باستخدام برمجية بايثون (Python).

أنظر الهامش

السؤال الثاني: أذكر أمثلة على استخدام جملة (pass) داخل الجملة الشرطية (if).

إذا أردت كتابة جملة شرطية (if) فارغة لا تحتوي على أيّ من الجمل، فإنّني أضع الجملة (pass) داخل هذه الجملة الشرطية (if)؛ لتجنّب ظهور رسالة تفيد بوجود خطأ في البرنامج.

المهارات: أوظف مهارة التفكير الناقد والمهارات البرمجية والتحليل في الإجابة عن الأسئلة الآتية:

السؤال الأول: أتبّع البرنامج الآتي من دون تشغيله، ثم أذكر النتيجة المترتبة على تشغيله.

```
x = int(input("Enter your grade: "))
if x > 84:
    print("Excellent grade.")
elif x > 76:
    print("Very good grade.")
elif x > 68:
    print("Good grade.")
elif x > 50:
    print("You passed the course.")
else:
    print("You failed the course.")
```

النتيجة هو:

$y < x < z$

السؤال الثاني: اقرأ البرنامج الآتي المكتوب بلغة البرمجة بايثون (Python)، ثم أجب عن السؤالين الآتين:

1. ما الهدف الرئيس من البرنامج؟ أصف ما يقوم به البرنامج عامةً من دون وصف وظيفة كل أمر برمجي فيه.

```
x = int(input("Enter your grade: "))
if x > 84:
    print("Excellent grade.")
elif x > 76:
    print("Very good grade.")
elif x > 68:
    print("Good grade.")
elif x > 50:
    print("You passed the course.")
else:
    print("You failed the course.")
```


الإجابة: برنامج لطباعة الوصف المناسب لعلامة الطلبة، بحيث تُصنف علامة الطالب الناجح إلى واحدة من أربعة تصنيفات، وقد يكون الطالب راسبًا.

2. ما النتيجة المُرتَّبة على تشغيل البرنامج إذا أدخل المُستخدم العدد (49)، ثمَّ العدد (55)، ثمَّ العدد (68)، ثمَّ العدد (70)، ثمَّ العدد (90)، ثمَّ العدد (78)، ثمَّ العدد (76)؟

الناتج:

You failed the course.

You passed the course.

You passed the course.

Good grade.

Excellent grade.

Very good grade.

Good grade.

السؤال الثالث: أكتشف الأخطاء الواردة في البرنامج الآتي من دون تنفيذه.

البرنامج المصحح

```
n = int(input("Enter any natural number: "))

if n <= 0:
    print("Wrong input. Please enter a positive number.")
else:
    total = 0
    while n > 0:
        total += n
        n -= 1
    print("The sum of the natural numbers is:", total)
```

الحلقات Loops

عدد الحصص المقترح:
أربع حصص.

المصادر:

كتاب الطالب، لغة البرمجة بايثون.

الأدوات والتجهيزات:

أجهزة حاسوب، جهاز للعرض.

تُصنّف الحلقات في برمجة بايثون (Python) إلى نوعين، هما:

1. حلقات **while** (while loops): تُستعمل حلقة (while) لتكرار تنفيذ جملة واحدة أو أكثر طالما تحقق شرط مُعَيَّن. وفي حال لم يعد هذا الشرط مُتحققًا، فإن البرنامج يتوقّف عن تنفيذ هذه الجملة أو الجمل.

2. حلقات **for** (for loops): تُستعمل حلقة (for) لتكرار مجموعة من الجمل البرمجية عددًا مُحدّدًا من المرات.

تحتوي برمجة بايثون (Python) على جمل تحكم (Control Statements) تعمل على ضبط الآلية التي تُنفَّذ بها الحلقات. وتشمل هذه الجمل كلاً من جملة التحكم (break)، وجملة التحكم (continue):

1. جملة التحكم (break): تُستعمل هذه الجملة لإيقاف الحلقة إذا تحقق شرط مُعَيَّن، ثمّ تنفيذ الجمل التي تلي الحلقة في البرنامج.

2. جملة التحكم (continue): تُستعمل هذه الجملة لإيقاف الدورة الحالية في الحلقة، والانتقال إلى الدورة التالية فيها إذا تحقق شرط مُعَيَّن.

في ما يأتي بيان للحلقات وجمل التحكم الموجودة في برمجة بايثون (Python) وطرائق استخدامها في البرامج:

حلقات while (while loops)

تُعرّف حلقة (while) باستخدام الكلمة المحجوزة (while)، وتُكتب صيغتها العامة على النحو الآتي:

```
while condition:
    statements
    increment or decrement
```

في ما يأتي بيان مُفصّل لكل عنصر يُمثّل جزءاً من عملية التعريف بحلقة (while):

- **condition**: شرط يُحدّد استمرار تنفيذ الجمل الموجودة في حلقة (while)، ويتمّ التحقق منه في كل دورة؛ إذ يتوقّف تنفيذ الجمل حين يصبح هذا الشرط غير مُتحقق.
- **statements**: جمل توجد في حلقة (while)، ويُكرّر البرنامج تنفيذها ما دام الشرط مُتحققًا.
- **increment**، أو **decrement**: إجراء يُحدّد كيف تزداد قيمة العداد أو تنقص، وهو يُنفَّذ في كل دورة.

أولاً: التهيئة

سياق التعلّم:

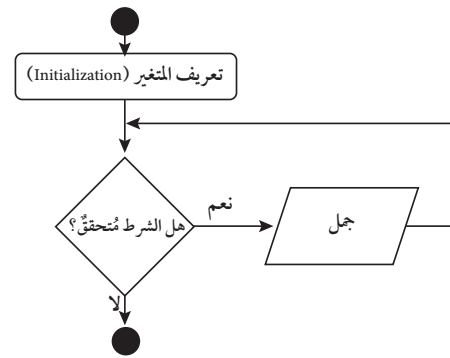
أوضح للطلبة الفكرة الرئيسة من الدرس، وأضعه في قالب سياق التعلّم، وأخبرهم أنهم سيكملون رحلة التعلّم بوصفهم مبرمجين ناشئين كالتالي بدؤوا بها، وسيكملون العمل على تطوير برامج لألعاب بسيطة، تعتمد على واجهات نصية باستخدام برمجة بايثون.

أوضح لهم منتج التعلّم المتوقع إنجازه خلال هذا الدرس، ضمن سياق التطوير المتكرر للبرمجيات، وأنهم سيطورون لعبة التخمين التي بدؤوا بها؛ بإضافة حلقات تكرارية، تُمكن من عرض القائمة بصورة مُتكرّرة ليتمّ اختيار الخيار الصحيح، ضمن سياق تصميم لعبة تخمين الأرقام باستخدام برمجة بايثون (Python).

نشاط تمهيدي:

- أطلب إلى الطلبة التفكير في الأنشطة التي يؤديونها يوميًا بانتظام مثل الاستيقاظ، وتناول الطعام، والذهاب إلى المدرسة، إلخ.
- أتيح لهم المجال للتعبير عن روتينهم اليومي كما يشاؤون؛ سواء عن طريق الكتابة، أو التعبير الشفهي.
- بعد أن يشارك الطلبة أفكارهم، أبدأ بمناقشة مفهوم الروتين، وشرح لهم أن الروتين هو نمط ثابت متكرر من الأنشطة أو الأعمال التي يؤديها الشخص يوميًا.
- أوضح للطلبة أن الروتين اليومي يشبه الحلقات في البرمجة، تمامًا كما تؤدي الأنشطة نفسها كل يوم على نحو متكرر. وتستخدم الحلقات في بايثون لتنفيذ مجموعة من التعليمات على نحو متكرر بناءً على شرط معين. وهذا ما ستتعرف إليه تفصيليًا في هذا الدرس.

يُبين الشكل (2-4) تمثيلًا لطريقة عمل حلقات (while) باستخدام مُخطّط سير العمليات.



الشكل (2-4): مُخطّط سير العمليات لحلقات (while).

مثال

يطبع البرنامج الآتي قيمة العدّاد (count) من (1) إلى (5):

```

count = 1
while count < 6:
    print("Count is", count)
    count += 1
else:
    print("Loop has ended")
  
```

إذا تبنّعت المقطع البرمجي في المثال السابق، أجد أنّ البرنامج يُنفّذه كما يأتي:

1. Initialization: بدء البرنامج بتعريف المتغير count، وإعطائه القيمة الأولية (1).
 2. Condition: تحقق البرنامج من أنّ قيمة المتغير count أقل من (6). وفي حال كان الشرط صحيحًا، فإنّ البرنامج يستمر في تنفيذ الجملة داخل الحلقة.
 3. Statement: جملة الطباعة لقيمة المتغير count.
 4. Increment، أو Decrement: عمل البرنامج على زيادة قيمة (count) بمقدار (1) في كل دورة؛ ما يؤثّر في تحقق الشرط في الدورة التالية.
- بعد ذلك سيعمل البرنامج على التحقق من الشرط مرّة أخرى؛ فإذا تبين أنّ الشرط لا يزال صحيحًا، فإنّ البرنامج سيكرّر الخطوة الثالثة والخطوة الرابعة. أمّا إذا لم يعد الشرط صحيحًا، فإنّ البرنامج يخرج من الحلقة، ويطبع عبارة "Loop has ended".

ثانيًا: عملية التعليم والتعلّم:

مناقشة:

- أشرح للطلبة مفهوم الحلقات، وأناقش معهم مبدأ عملها وأهميتها في بايثون.
- أوضح لهم أنواع الحلقات (حلقة for أو while) التي تكرر تنفيذ الكود مرات عدة؛ حتى يُستوفى شرط معين، مثلما تتكرر الأنشطة نفسها في الروتين اليومي حتى ينتهي اليوم.
- أشرح للطلبة جمل التحكم (break، continue) التي تعمل على ضبط الآلية التي تُنفذ بها الحلقات في بايثون.
- أبدأ بالشرح، مع التوضيح بأمثلة عملية لكيفية استخدام حلقات for و while في بايثون.

في المثال السابق، إذا حُذفت جملة (else)، فهل سيختلف الناتج؟ ماذا أتوقع أن يكون الناتج؟
أعدّل المقطع البرمجي بحذف جملة (else)، ثمّ أنفذ البرنامج المُعدّل باستخدام برمجية بايثون (Python)، ثمّ أفاَرِن الناتج الحالي بالناتج السابق، وألاحظ الفرق بينهما إن وجد.
أناقش زملائي/ زميلاتي في السؤالين الآتيين:
- هل أثّرت جملة (else) في الناتج؟

مثال:

يعمل البرنامج الآتي على حساب مجموع الأعداد التي أدخلها المُستخدم حتى يصل المجموع إلى (03) فأكثر، أو حتى يتسم إدخال القيمة (0). فإذا وصل المجموع إلى (03) فأكثر، طُبِعَ المجموع الحالي، وخرج البرنامج من الحلقة. أمّا إذا أُدخلت القيمة (0) قبل الوصول إلى المجموع (03)، فإنّ البرنامج يطبع رسالة مفادها أنّ المجموع أقل من (03)، ثمّ يعرض قيمة المجموع النهائية.

```
s = 0
a = int(input("Enter a number (0 to stop): "))
while a != 0:
    s += a
    if s >= 30:
        print("Sum is equal to", s)
        break
    a = int(input("Enter a number (0 to stop): "))
else:
    print("Sum is equal to", s, ".")
```

عند تشغيل البرنامج، وإدخال العدد (01)، ثمّ العدد (51)، ثمّ العدد (9)، ثمّ العدد (0)، ستظهر النتيجة الآتية على شاشة جهاز الحاسوب:

```
C:\WINDOWS\py.exe
Enter a number (0 to stop): 10
Enter a number (0 to stop): 15
Enter a number (0 to stop): 9
Sum is equal to 34
```

إذا حُذفت جملة (else) الواردة في المثال السابق، فإنّ النتيجة الآتية ستظهر على شاشة جهاز الحاسوب عند تشغيل البرنامج:

```
C:\WINDOWS\py.exe
Enter a number (0 to stop): 10
Enter a number (0 to stop): 15
Enter a number (0 to stop): 9
Sum is equal to 34
Sum is equal to 34
```

1. الناتج:

Count is 1

Count is 3

Count is 5

Loop has ended

2. إذا عُدِّلَ الشرط إلى count =

6، ستظهر رسالة تفيد بوجود

خطأ في البرنامج، إما إذا

عُدِّلَ الشرط إلى count ==

6، فسيكون الناتج:

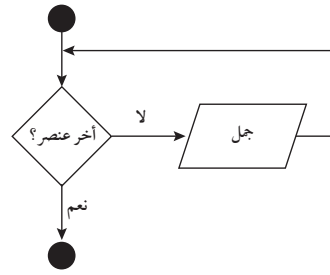
Loop has ended

3. إذا عُدِّلَ الشرط count < 6 إلى

الشرط count > 6، فسيكون

Loop has ended

يُبين الشكل (3-4) تمثيلاً لطريقة عمل حلقات (for) باستخدام مُخطّط سير العمليات.



الشكل (3-4): مُخطّط سير العمليات لحلقة (for).

الدالة range() مع حلقات (for)

تُستعمل الدالة range() لإرجاع سلسلة من الأرقام، تبدأ بالرقم (0) (ما لم يُحدّد رقم آخر)، وتنتهي برقم مُحدّد، تزداد range() بمقدار (1) (ما لم يُحدّد مقدار آخر للزيادة). والصيغة العامة لاستخدامها:

```
for element in range(a):
    statements
```

تُستخدم الدالة range() بثلاث طرائق مختلفة، هي:

■ **range(a)**: تُرجع الدالة بهذه الطريقة سلسلة من الأرقام، بدءاً بالرقم (0)، وانتهاءً بالرقم (a-1).

مثال:

إذا كانت قيمة (a) هي (5)، فإن الدالة ستُرجع سلسلة الأرقام الآتية: (0)، (1)، (2)، (3)، (4).

■ **range(a, b)**: تُرجع الدالة بهذه الطريقة سلسلة من الأرقام، بدءاً بالرقم (a)، وانتهاءً بالرقم (b-1).

مثال:

إذا كانت قيمة (a) هي (1)، وقيمة (b) هي (5)، فإن الدالة ستُرجع سلسلة الأرقام الآتية: (1)، (2)، (3)، (4).

■ **range(a, b, c)**: تُرجع الدالة بهذه الطريقة سلسلة من الأرقام، بدءاً بالرقم (a)، وانتهاءً بالرقم (b-1)، مُتزايدةً بقفزة مقدارها (c).

مثال:

نشاط عملي

لا يختلف الناتج.

نشاط جماعي :

■ لماذا يؤدي هذان المقطعان البرمجان إلى حلقة لانهاية؟

في البرنامج الأول، ستستمر الحلقة في الطباعة من دون توقف؛ لأن `i` دائماً يساوي 1، ثم سيكون شرط `while` دائماً صحيحاً.

أما في البرنامج الثاني، فستستمر الحلقة إلى ما لا نهاية؛ لأنها تعتمد على شرط `True` الذي يكون دائماً صحيحاً.

في كلا المقطعين، سيظل الشرط المرفق بالحلقات دائماً صحيحاً؛ مما يجعل الحلقات لا تتوقف.

■ ما الطرائق التي يمكن استعمالها لتجنب الحلقات اللانهائية؟

- تحديث الشرط داخل الحلقة: تغيير قيمة المتغير الذي يتحكم به داخل الحلقة.

- إضافة شرط `(break)`: استخدام `break` للخروج من الحلقة عند استيفاء شرط معين.

■ كيف يمكن تعديل المقطع البرمجي على نحو يجعل التنفيذ ينتهي في نقطة محددة؟ تعديل المقطع الأول:

```
i = 1
while i == 1:
    print("I am stuck")
    i = 0 # لإنهاء الحلقة
```

تعديل المقطع الثاني

```
count = 0
while True:
    print("I am stuck")
    count += 1
    if count == 5:
        break # إنهاء الحلقة بعد 5 مرات من الطباعة
```

ألاحظ أن وجود جملة `(else)` يتيح للمستخدم التحكم في طباعة الجملة الثانية، بحيث لا تُطبع إلا بعد خروج البرنامج من الحلقة دون أن يتحقق الشرط في جملة `(if)`.

أقرأ - بالتعاون مع أفراد مجموعتي - المقطعين البرمجين الآتين، وأحاول توقع ناتج التنفيذ، ثم أنفذ هذين المقطعين باستخدام برمجة بايثون (Python)، ثم أقارن الناتج بما توقعته:

```
i = 1
while i == 1:
    print("I am stuck")

while True:
    print("I am stuck")
```

أناقش زملائي / زميلاتي في الأسئلة الآتية:

- لماذا يؤدي هذان المقطعان البرمجان إلى حلقة لانهاية؟
- ما الطرائق التي يمكن استعمالها لتجنب الحلقات اللانهائية؟
- كيف يمكن تعديل المقطع البرمجي على نحو يجعل التنفيذ ينتهي في نقطة محددة؟

حلقات `for` (for loops)

تُعرف حلقة `(for)` باستخدام الكلمتين المحجوزتين `(for)` و `(in)` على النحو الآتي:

```
for element in sequence:
    statements
```

في ما يأتي بيان مُفصّل لكل عنصر يُمثّل جزءاً من عملية التعريف بحلقة `(for)`:

- `element`: مُتغيّر يُعرف داخل الحلقة، وتوضع فيه إحدى قيم المتابعة (sequence) التي تُجلب في كل دورة، وتكون موضوعة بعد هذا المُتغيّر.
- `sequence`: سلسلة يريد المُستخدم الوصول إلى جميع عناصرها.
- `statements`: جمل موجودة في حلقة `(for)` التي سيكرّر البرنامج تنفيذها في كل دورة.

نشاط عملي:

لا يختلف الناتج.

ملاحظات



نشاط
جماعي

عند تشغيل البرنامج، ستظهر النتيجة الآتية على شاشة جهاز الحاسوب:

```
C:\WINDOWS\py.exe
1
2
```

أُجَرِّب وأُستنتج:

في المثال السابق، إذا وُضعت جملة print() بعد جملة (if)، فماذا سيحدث؟
أَتَوَقَّع ناتج البرنامج، ثُمَّ أَنْفِذُهُ باستخدام برمجة بايثون (Python)، وَأُلَاحِظ الناتج.

جملة التحكم (continue) مع حلقات (for)

تُستخدَم جملة التحكم (continue) مع حلقات (for) بالطريقة نفسها التي استُخدِمت فيها مع حلقات (while).

مثال:

يطبع البرنامج الآتي الأعداد (1) و(3) و(4) باستخدام جملة التحكم (continue):

```
for x in range(1, 5, 1):
    if x == 2:
        continue
    print(x)
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية على شاشة جهاز الحاسوب:

```
C:\WINDOWS\py.exe
1
3
4
```

جملة (else) مع حلقات (for)

يُمكن استعمال جملة (else) مع حلقات (for) لتنفيذ مجموعة من الأوامر عند الخروج من الحلقة.

مثال:

```
for x in range(1, 5, 1):
    print(x)
else:
    print("counting is completed.")
```

نشاط جماعي:

```
for x in range(1, 5, 1):
    if x == 2:
        break
```

النتيجة:

2

المشروع: تصميم برنامج وإعداده لإنشاء لعبة تخمين الأرقام باستخدام لغة بايثون (Python)/ المهمة (4).

أكمل - بالتعاون مع أفراد مجموعتي - تنفيذ مشروع التصميم والتطوير للعبة تخمين الأرقام. وكُنّا قد انتهينا في الخطوة الأولى للمشروع من عرض القائمة الرئيسة في اللعبة، ثمّ استجبنا في الخطوة الثانية لما يُدخّله اللاعب عن طريق عرض تعليمات اللعبة أو كتابة رسالة بحسب الخيار المُدخّل. ولكنّ، ما الخطوة التي تلي عرض التعليمات وكتابة الرسالة؟ والآن سأعمل - ضمن المجموعة - على تعديل ما كُتِب سابقاً باستخدام الحلقات التكرارية؛ بأنّ أكرّر عرض القائمة الرئيسة والخيارات على النحو الآتي:

1- العودة إلى القائمة الرئيسة: سيعود اللاعب إلى القائمة الرئيسة مرّة أخرى بعد قراءة تعليمات اللعبة بدلاً من انتهاء البرنامج.

2- طلب إدخال خيار صحيح: سيعمل البرنامج على تكرار طلب إدخال خيار صحيح طالما كان الرقم المُدخّل أقل من (1) وأكثر من (3).

أستخدم حلقة تكرار مُركّبة لتنفيذ ذلك كما يأتي:

1- حلقة خارجية: حلقة لانتهائية تعمل على تكرار طباعة القائمة الرئيسة، والطلب إلى اللاعب إدخال أحد الخيارات.

2- حلقة داخلية: حلقة تعمل على تكرار طلب الإدخال إذا كان الرقم المُدخّل غير صحيح؛ أي أقل من (1) وأكثر من (3).

أتحقّق - مع أفراد مجموعتي - من كتابة المقطع البرمجي بصورة صحيحة من دون وجود أخطاء إملائية أو أخطاء منطقية. كذلك أتحقّق من أن الحلقات تعمل بصورة صحيحة، ومن تكرار القائمة الرئيسة عند الحاجة، ثمّ أحتفظ بما كُتِب - ضمن المجموعة - في ملف حتّى يُمكن لأفراد المجموعة التعديل عليه في الخطوات القادمة.



مشروع

نشاط جماعي:

لن ننفذ البرنامج جملة `else`.

```
for x in range(1, 5, 1):
    if x == 2:
        break
    print(x)
else:
    print("counting is completed.")
```

النتائج:

1

أَقِمْ تَعْلَمِي

المعرفة: أوظّف في هذا الدرس ما تعلّمته من معارف في الإجابة عن السؤالين الآتيين:
السؤال الأول: أفرّن بين حلقات (for) وحلقات (while) باستخدام مُخطّط سِير العمليات لكلّ منهما.

السؤال الثاني: ما الطرائق الثلاث التي يُمكن بها استخدام الدالة `range()` ؟

المهارات: أوظّف مهارة التفكير الناقد والمهارات البرمجية والتحليل في الإجابة عن السؤالين الآتيين:

السؤال الأول: أكتب برنامجًا لإيجاد مضروب عدد ما باستخدام حلقات (for).

السؤال الثاني: اقرأ البرنامج التالي المكتوب بلغة البرمجة بايثون (Python)، ثمّ أجب عمّا يأتي:

1. أصف ما يقوم به البرنامج عامةً من دون وصف وظيفة أيّ أمر برمجي فيه.

```
num = int(input("enter a number"))
if num % 2 != 0:
    while num > 0:
        print(num)
        num -= 2
else:
    while num > 0:
        print(num)
        num -= 1
```

القيّم والاتجاهات

أبحث في المواقع الإلكترونية الموثوقة عن تقنيات البرمجة الآمنة وطرق حماية البيانات عند مشاركة البرامج عبر مجتمعات البرمجة وألخص ما أتوصل إليه في Google Docs وأنشره عبر الموقع الإلكتروني للمدرسة.

ملاحظات

.....

.....

.....

.....

.....



75

نشاط عملي: ⚙️

لن يطبع البرنامج شيئاً في حال وجود جملة `pass`.
وفي حال عدم وجود جملة `pass`، ستظهر رسالة تفيد بوجود خطأ.

نشاط فردي: 🧑

```
for x in range(7, 11):
    print("Multiplication Table", x)
    for y in range(1, 11):
        print(x, "*", y, "=", x * y)
```

البرنامج معدل:

```
count = 0
for i in range(10):
    if i % 2 == 0:
        while count < 5:
            print("Count is:", count)
            count += 1
        else:
            print("Done with inner loop")
    if count == 5:
        break
print("Loop ended.")
```

الناتج بعد تعديل البرنامج هو:

Count is: 0

Count is: 1

Count is: 2

Count is: 3

Count is: 4

Loop ended.

ثالثاً: الإثراء والتأمل والتقويم.

أبحث:



- تسمح الحلقات لك بإعادة استخدام الجزء نفسه من الكود مرات عدة من دون الحاجة إلى كتابته مراراً وتكراراً. يقلل هذا من احتمال حدوث أخطاء، ويجعل الكود أكثر سهولة في القراءة والصيانة.
- عندما تحتاج إلى تنفيذ عملية معينة مرات عدة (مثل معالجة قائمة من العناصر)، فإن الحلقات تتيح القيام بذلك بسرعة، ومن دون تكرار الكود لكل عنصر.
- تتيح الحلقات للكود أن يكون أكثر مرونة وقابلية للتكيف مع الظروف المتغيرة، مثل عدد العناصر في قائمة، أو استمرارية عملية معينة؛ حتى يُستوفى شرط معين.

75

- يسهم استخدام الحلقات في تقليل حجم الذاكرة المستخدمة؛ إذ إننا نحتاج إلى تعريف المتغيرات مرة واحدة فقط، بدلاً من تعريف متغيرات جديدة لكل عملية متكررة.
- الحلقات ضرورية في عمليات المعالجة الآلية للبيانات الكبيرة، مثل تحليل مجموعات البيانات، أو تنفيذ عمليات حسابية معقدة على نحو متكرر.

إثراء للمعلم / المعلمة:

لمزيد من الأمثلة والمعلومات حول الحلقات في برنامج بايثون، يمكن زيارة المواقع الآتية:

<https://www.datacamp.com/tutorial/loops-python-tutorial>

<https://www.geeksforgeeks.org/loops-in-python>

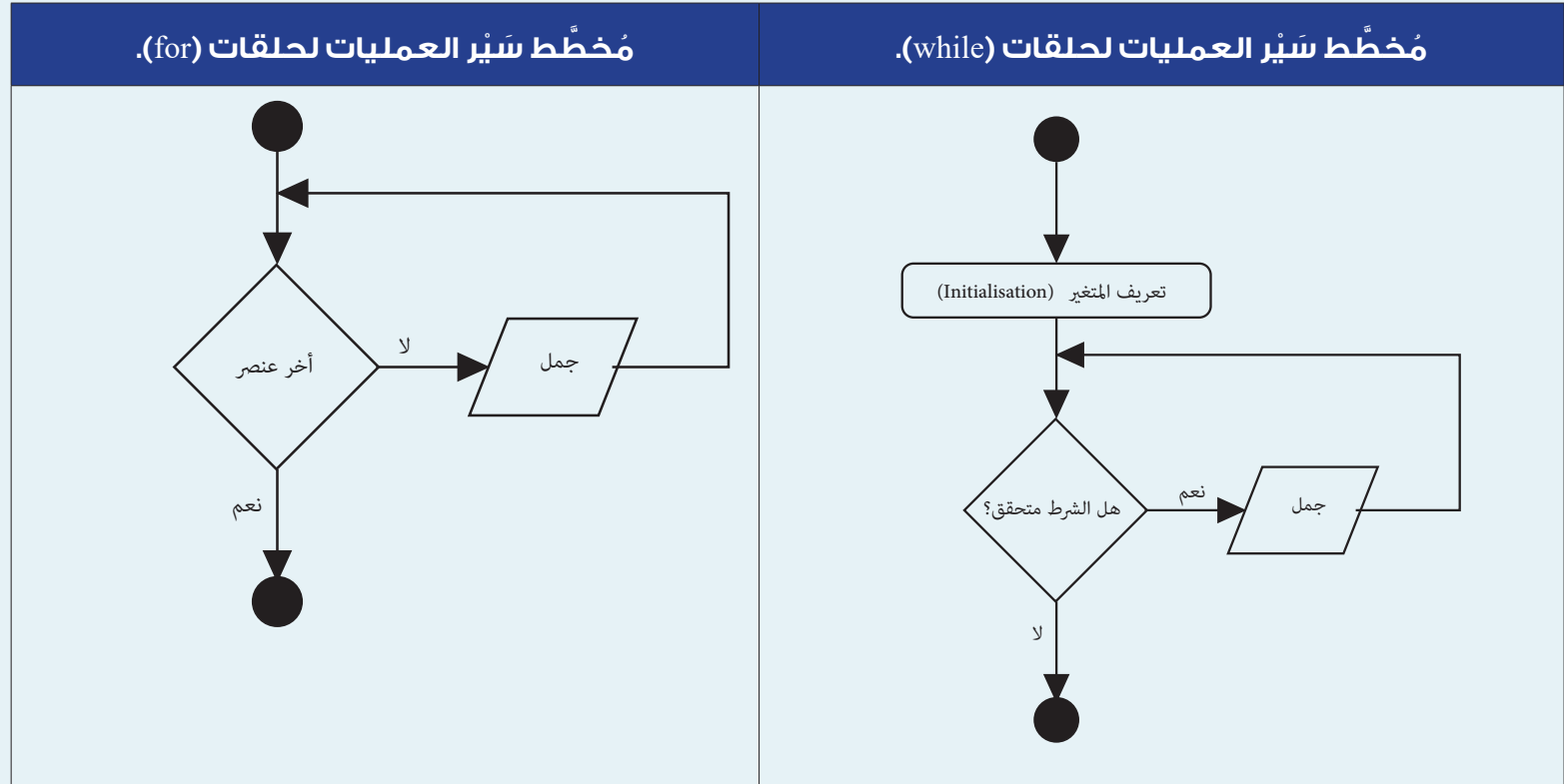


- أذكر الطلبة بسياق التعلم، وأهمية دورهم بوصفهم ناشئين في مجال البرمجة، وأنهم سيعملون برنامج لعبة التخمين التفاعلية باستخدام Python.
- أوضح أن هذا التعديل يأتي في سياق التطوير المستمر للبرمجيات.
- أذكر الطلبة بما عملوا عليه حتى الآن في اللعبة، مع التركيز على دور القائمة الرئيسة في اللعبة، وكيفية الاستجابة لمدخلات اللاعب عن طريق عرض تعليمات اللعبة، أو كتابة رسالة معينة بناءً على الخيار المُدخل.
- أوجه الطلبة لاستخدام الحلقات التكرارية لتحسين البرنامج، إذ يتم:
 - العودة إلى القائمة الرئيسة: سيعود اللاعب إلى القائمة الرئيسة مرة أخرى بعد قراءة تعليمات اللعبة بدلاً من انتهاء البرنامج.
 - طلب إدخال خيار صحيح: سيكرر البرنامج طلب إدخال خيار صحيح، مادام أن الرقم المُدخل أقل من (1) وأكثر من (3).
 - استخدام حلقة تكرار مُركبة لتنفيذ ذلك كما يأتي:
 - حلقة خارجية: حلقة لانهاية تُكرر طباعة القائمة الرئيسة، وتطلب إلى اللاعب إدخال أحد الخيارات.
 - حلقة داخلية: حلقة تُكرر طلب الإدخال، إذ كان الرقم المُدخل غير صحيح؛ أي أقل من (1) وأكثر من (3).
- أوجه الطلبة للتأكد من أن البرنامج مكتوب على نحو صحيح ومنظم، واختبار البرنامج؛ لضمان أنه يعمل من دون أخطاء، والتأكد من أن الحلقات تعمل بصورة صحيحة، والتأكد من تكرار القائمة الرئيسة عند الحاجة.
- أوجه الطلبة للاحتفاظ بالمقطع البرمجي المكتوب في ملف بايثون (.py)، ضمن مجموعة المشروع؛ حتى يمكن لأفراد المجموعة التعديل عليه في الخطوات القادمة.
- أشارك مع الطلبة معايير تقييم المهمة:
 - الدقة في كتابة البرنامج: استخدام الحلقات وجمل التحكم على نحو صحيح، وتنظيم المقطع البرمجي وإضافة تعليقات توضيحية.
 - الوظائف المطلوبة: التأكد من أن الحلقات تعمل بصورة صحيحة، وتكرار القائمة الرئيسة عند الحاجة.

- الاختبار: البرنامج يعمل من دون أخطاء.
- التعاون الجماعي: المشاركة الفعالة بين أعضاء الفريق.
- حفظ الكود: التأكد من حفظ الكود على نحو صحيح؛ ليتمكن الفريق من التعديل عليه لاحقاً.

إجابات نموذجية لأسئلة أقيم تعلّمي.

المعرفة: أُوظّف في هذا الدرس ما تعلّمته من معارف في الإجابة عن السؤالين الآتيين:
السؤال الأول: أفرّن بين حلقات (for) وحلقات (while) باستخدام مخطط سَيْر العمليات لكلّ منهما.



السؤال الثاني: ما الطرائق الثلاث التي يُمكن بها استخدام الدالة (range) ؟

- $\text{range}(a)$: ترجع الدالة سلسلة من الأرقام، تبدأ من الصفر (0) وتنتهي عند $(a-1)$.
- $\text{range}(a,b)$: ترجع الدالة سلسلة من الأرقام، تبدأ بالرقم a وتنتهي بالرقم $(b-1)$.
- $\text{range}(a, b, c)$: ترجع الدالة سلسلة من الأرقام، تبدأ بالرقم a وتنتهي بالرقم $(b-1)$ ، مع زيادة بمقدار c في كل خطوة.

المهارات: أوظف مهارة التفكير الناقد والمهارات البرمجية والتحليل في الإجابة عن السؤالين الآتيين:

السؤال الأول: أكتب برنامجًا لإيجاد مضروب الأعداد من (1) إلى (10) باستخدام حلقات (for).

```
for x in range(1, 11):
    factorial = 1
    for y in range(1, x + 1):
        factorial *= y
    print(x, ' ! = ', factorial)
```

ناتج البرنامج:

1 ! = 1

2 ! = 2

3 ! = 6

4 ! = 24

5 ! = 120

6 ! = 720

7 ! = 5040

8 ! = 40320

9 ! = 362880

10 ! = 3628800

السؤال الثاني: اقرأ البرنامج الآتي المكتوب بلغة البرمجة بايثون (Python)، ثم أجب عما يأتي:

1. ما الهدف الرئيس من البرنامج؟

```
x = 1
while x <= 6:
    y = 1
    while y <= 6:
        print(x, " * ", y, " = ", x * y)
        y += 1
    x += 1
```

يطبع البرنامج حاصل ضرب الأرقام من 1 إلى 6 بعضها ببعض.

2. أصف ما يقوم به البرنامج عامةً من دون وصف وظيفة أي أمر برمجي فيه .

ناتج البرنامج هو:

$$1 * 1 = 1$$

$$1 * 2 = 2$$

$$1 * 3 = 3$$

$$1 * 4 = 4$$

$$1 * 5 = 5$$

$$1 * 6 = 6$$

$$2 * 1 = 2$$

$$2 * 2 = 4$$

$$2 * 3 = 6$$

$$2 * 4 = 8$$

$$2 * 5 = 10$$

$$2 * 6 = 12$$

$$3 * 1 = 3$$

$$3 * 2 = 6$$

$$3 * 3 = 9$$

$$3 * 4 = 12$$

$$3 * 5 = 15$$

$$3 * 6 = 18$$

$$4 * 1 = 4$$

$$4 * 2 = 8$$

$$4 * 3 = 12$$

$$4 * 4 = 16$$

$$4 * 5 = 20$$

$$4 * 6 = 24$$

$$5 * 1 = 5$$

$$5 * 2 = 10$$

$$5 * 3 = 15$$

$$5 * 4 = 20$$

$$5 * 5 = 25$$

$$5 * 6 = 30$$

$$6 * 1 = 6$$

ملاحظات

$$6 * 2 = 12$$

$$6 * 3 = 18$$

$$6 * 4 = 24$$

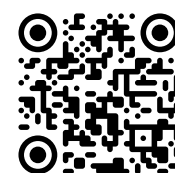
$$6 * 5 = 30$$

$$6 * 6 = 36$$

القيم والاتجاهات.

- البرمجة الآمنة هي عملية كتابة الكود بطريقة تحمي التطبيقات من التهديدات الأمنية المحتملة. عند مشاركة البرامج عبر مجتمعات البرمجة، يجب أن تكون البيانات والبرامج محمية من أي اختراقات أو استغلال غير مصرح به. في ما يأتي بعض التّقنيات الأساسية لحماية البيانات عند مشاركة البرامج في بايثون:
 - إدارة المدخلات والتحقق منها: أؤكد دائماً من التحقق من المدخلات التي تُدخل في البرنامج. يمكن أن تحتوي المدخلات غير الموثوقة على تهديدات مثل هجمات SQL injection أو هجمات XSS. واستخدام المكتبات مثل input() بحذر والتحقق من صحة البيانات المدخلة قبل معالجتها.
 - استخدام المكتبات الآمنة: تحتوي Python على مكتبات مثل hashlib و cryptography التي يمكن استخدامها لتشفير البيانات وحمايتها. أستخدمُ التشفير لحماية البيانات الحساسة مثل كلمات المرور والمعلومات الشخصية.
 - التعامل الآمن مع الأخطاء والاستثناءات: أتجنبُ كشف التفاصيل الدقيقة عن النظام عند حدوث خطأ؛ إذ يمكن أن يستخدمها المهاجمون لاكتشاف ثغرات أمنية. أستخدمُ الاستثناءات (try-except) بطريقة تجعل النظام يعالج الأخطاء على نحو آمن من دون كشف تفاصيل حساسة.
 - التحكم في الوصول: أؤكدُ من أن الأشخاص المصرح لهم فقط، يمكنهم الوصول إلى البيانات، أو استخدام الوظائف الحساسة. وأستخدمُ نظام إدارة المستخدمين والصلاحيات لتقييد الوصول إلى الأجزاء الحساسة من التطبيق.
 - تحديثات الأمان: أؤكد دائماً من تحديث جميع المكتبات والبرامج التي تستخدمها في Python؛ لضمان استخدام الإصدارات الأكثر أماناً. وأتحقق من تحديثات الأمان بانتظام؛ للحفاظ على حماية بياناتي.
- أهمية البرمجة الآمنة:
- حماية البيانات الشخصية: يمكن عن طريق البرمجة الآمنة حماية بيانات المستخدمين من الوصول غير المصرح به.
 - تجنب الهجمات: تقللُ تقنيات البرمجة الآمنة من احتمالية تعرض البرامج للهجمات الأمنية مثل اختراق البيانات.
 - الحفاظ على الثقة: تضمن البرمجة الآمنة أن المستخدمين يمكنهم الثقة في البرامج التي يستخدمونها؛ مما يساهم في نشر المطورين والمشروعات.
 - ويمكن الرجوع لقائمة التحقق من الممارسات البرمجة الآمنة عن طريق زيارة الموقع الاتي:

OWASP Foundation. (n.d.). Secure coding practices quick reference guide. OWASP. Retrieved August 10, 2024, from <https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/stable-en/02-checklist/05-checklist.html>



القوائم (Lists)

الدرس الخامس

القوائم (Lists)

الفكرة الرئيسية:

تعرف كيف يمكن التعامل مع القوائم وسلاسل الحروف في برمجة بايثون (Python)، واستخدامها في تخزين مجموعات البيانات المترابطة وإدارتها وتنفيذ عمليات فيها.

المفاهيم والمصطلحات:

القائمة (List)، سلسلة الحروف (String)، موقع العنصر في القائمة (أو الحرف في السلسلة) (Index)، إلصاق القوائم أو السلاسل (Concatenation)، القابلية للتغيير (Mutability)، القائمة المركبة (Nested List)، المصفوفة ثنائية الأبعاد (2D Matrix).

نتائج التعلم (Learning Outcomes):

- أعرف مفهوم المتغير (قائمة)، وأبين استخداماته في البرمجة.
- أنشئ قوائم على اختلاف أنواعها (متسلسلة، ومتغيرة، ومركبة)، وأستخدمها في تخزين مجموعة متنوعة من القيم.
- أحدد كيف يمكن تجميع مجموعة القيم في قائمة واحدة.
- أوضح أنواع القوائم (المتسلسلة، والمتغيرة، والمركبة)، وأنفذ عمليات مختلفة فيها، مثل: الإضافة، والحذف.
- أستخدم الدوال البرمجية الجاهزة في لغة البرمجة بايثون (Python) لمعالجة القوائم وإجراء بعض العمليات الأساسية فيها.

مُنتجات التعلم (Learning Products)

تعديل البرنامج باستخدام قوائم اللعبة في توليد الأرقام العشوائية، ضمن سياق تصميم لعبة تخمين الأرقام باستخدام برمجة بايثون (Python).

عدد الحصص المقترح :
أربع حصص.

المصادر:

كتاب الطالب، لغة البرمجة بايثون.

الأدوات والتجهيزات:

أجهزة حاسوب، جهاز للعرض

أولاً: التهيئة

سياق التعلم:

أوضح للطلبة الفكرة الرئيسية من الدرس، وهي التعرف إلى كيفية التعامل مع القوائم وسلاسل الحروف في برمجة بايثون (Python)، واستخدامها في تخزين مجموعات البيانات المترابطة، وإدارتها وتنفيذ عمليات فيها. وأخبرهم أنهم سيكملون رحلة التعلم بوصفهم مبرمجين ناشئين كالتالي بدؤوا بها، وسيكملون العمل على تطوير برامج لألعاب بسيطة، تعتمد على واجهات نصية باستخدام برمجة بايثون.

أوضح لهم منتج التعلم المتوقع إنجازه خلال هذا الدرس، ضمن سياق التطوير المتكرر للبرمجيات، وسيطورون لعبة تخمين الأرقام التي بدؤوا بها عبر تعديل البرنامج باستخدام قوائم اللعبة في توليد الأرقام العشوائية، باستخدام برمجة بايثون (Python).

تُعَدُّ القوائم إحدى المزايا المُهمَّة للغات البرمجة؛ إذ تُسهِّم في إدارة تنظيم البيانات، وتتيح للمستخدم التعامل معها واسترجاعها بسهولة. ولكن، كيف يُمكن توظيف القوائم في لغة البرمجة بايثون (Python)؟ وهل يختلف ذلك عن توظيفها في لغات البرمجة الأخرى؟



نشاط تمهيدي

أفترض أنني أريد إنشاء برنامج يتولَّى قراءة علامات (100) طالب في أحد الامتحانات، ثمَّ يعرض هذه العلامات مُرتَّبة بصورة تصاعديَّة. أناقش أفراد مجموعتي في الأسئلة الآتية بناءً على ما تعلَّمته عن لغة البرمجة بايثون (Python):

1. هل يجب عليَّ تخزين علامات هؤلاء الطلبة أم يمكنني كتابة هذا البرنامج دون الحاجة لتخزين العلامات؟
2. إذا تعيَّن عليَّ تخزين علامات هؤلاء الطلبة، فإلى كم متغيِّر أحتاج؟ وما التحدِّيات التي سأواجهها عند محاولتي تخزين العلامات؟
3. كيف يُمكنني طباعة العلامات وترتيبها تصاعديًّا؟ وهل ما تعلَّمته عن لغة البرمجة بايثون (Python) سيساعدني على ذلك؟

القوائم (Lists)

تُمثِّل القائمة في لغة البرمجة بايثون (Python) عددًا من القيم التي يُخزَّن بعضها مع بعض، وترتبط معًا بمعنى وظيفي مشترك. فمثلاً، يُمكن للمستخدم تخزين أسماء الشوارع داخل إحدى المدن في قائمة، وكذلك تخزين أسعار البضائع التي اشتراها أحد العملاء، أو تخزين علامات الطلبة في الصف الحادي عشر.

يتطلَّب استخدام القوائم في لغة البرمجة بايثون (Python) تعريف القائمة أولاً، ثمَّ تخزين العناصر داخلها.

يُمكن تعريف إحدى القوائم باستخدام الأقواس المُربَّعة []، والفصل بين عناصر القائمة بفواصل على النحو الآتي:

المقطع البرمجي التالي يعرِّف قائمة باسم List name تحتوي على أربعة عناصر، وكل عنصر في القائمة يحتوي على قيمة ثابتة واحدة تتكرر، وهي 'value 1'

```
mylist = ['value1', 'value1', 'value1', 'value1']
```

نشاط تمهيدي:

1. هل يجب عليَّ تخزين علامات هؤلاء الطلبة، أم يمكنني كتابة هذا البرنامج من دون الحاجة لتخزين العلامات؟

الإجابة: لا يمكن كتابة البرنامج من دون تخزين العلامات؛ لأننا نحتاج لقراءة العلامات كاملة قبل البدء بمعالجتها؛ لمعرفة ترتيبها (مثلاً لن أعرف ما هو أكبر رقم، قبل أن أقرأ جميع الأرقام).

2. إذا تعيَّن عليَّ تخزين علامات هؤلاء الطلبة، فإلى كم متغيِّر أحتاج؟ وما التحدِّيات التي سأواجهها عند محاولتي تخزين العلامات؟

الإجابة: سنحتاج لـ 100 متغيِّر على الأقل، وهذا سيكون صعباً للغاية.

3. كيف يُمكنني طباعة العلامات وترتيبها تصاعديًّا؟ وهل ما تعلَّمته عن لغة البرمجة بايثون، سيساعدني على ذلك؟

الإجابة المتوقعة: لم يُناقش ذلك من قبل؛ لذلك يفترض ألا يعرف الطالب كيفية القيام بذلك.

ثانيًا: عملية التعليم والتعلّم:

🔧 نشاط عملي / إثرائي:

إنشاء قائمة بأسماء الطلبة: أنشئ قائمة تحتوي على أسماء أفراد المجموعة.

إنشاء قائمة بهويات الطلبة: أنشئ قائمة ثانية تحتوي على الهويات المفضلة لكل طالب.

التحقّق من صحة الأوامر البرمجية: يمكن كتابة بعض الأوامر؛ لاختبار القوائم، والتأكد من صحتها، مثل عرض أسماء الطلبة وهوياتهم باستخدام حلقة (for).

يُمكن للمستخدم معرفة طول أيّ قائمة باستخدام الدالة (len):

```
names = ['Ali', 'Salam', 'Joud', 'Omar', 'Jana']
ages = [10, 48, 33, 17]
mix = [1, 1.5, 'Yes', 24, 5, [1, 2, 3]]

print("Length of names list:", length_of_names)
print("Length of ages list:", length_of_ages)
print("Length of mix list:", length_of_mix)
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية على شاشة الحاسوب:

```
C:\WINDOWS\py.exe
Length of names list: 5
Length of ages list: 4
Length of mix list: 6
```

نشاط عملي

أُجرب وأستكشف:

أعمل - بالتعاون مع أفراد مجموعتي - على استخدام لغة البرمجة بايثون (Python)، لإنشاء قائمة تضمّ أسماء طلبة المجموعة، وقائمة أخرى تُبيّن هوياتهم المفضّلة. بعد ذلك أكتب الأوامر البرمجية، وأتحقّق من صحتها عبر طباعة طول كل من القوائم وطباعة أسماء الطلاب مع هوياتهم المفضلة.

```
students = ["خالد", "ليلي", "محمد", "سارة", "أحمد"]
hobbies = ["البرمجة", "الرسم", "السباحة", "القراءة", "كرة القدم"]

for i in range(len(students)):
    print(f"الطالب {students[i]}: هوايته المفضلة {hobbies[i]}")
```

النتيجة: عند تنفيذ الأوامر البرمجية السابقة، ستظهر قائمة بأسماء الطلبة وهوياتهم المفضلة.

```
>>> الطالب: أحمد, هوايته المفضلة: كرة القدم
      الطالب: سارة, هوايته المفضلة: القراءة
      الطالب: محمد, هوايته المفضلة: السباحة
      الطالب: ليلي, هوايته المفضلة: الرسم
      الطالب: خالد, هوايته المفضلة: البرمجة
```

إضاءة للمعلم / المعلمة:

الجملة البرمجية في المثال السابق:

f"الطالب {students[i]}: هوايته

المفضلة {hobbies[i]} :

تستخدم ميزة f-strings في Python

لتنسيق النصوص وإدراج قيم

المتغيرات مباشرة داخل النص.

شرح الجملة:

f"...": هذا يشير إلى أن النص

داخل علامات الاقتباس هو

f-string؛ مما يسمح بتضمين

المتغيرات أو التعبيرات داخل النص.

{students[i]} هوايته المفضلة

"{hobbies[i]}":

تُستخدم الأقواس {} لإدراج

قيم المتغيرات students[i] و

hobbies[i] داخل النص.

تُستبدل {students[i]} باسم الطالب

الحالي من القائمة students،

وتُستبدل {hobbies[i]} بهوايته

المفضلة من القائمة hobbies.

```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> days = ['Su', 'Mo', 'Tu', 'We', 'Th', 'Fr', 'Sa']
>>> days[-1]
'Sa'
>>> days[-2]
'Fr'
>>> days[-7]
'Su'
>>> days[-8]
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    days[-8]
IndexError: list index out of range
>>>
```

الشكل (3-5): استخدام الأرقام السالبة في الوصول إلى العناصر.



نشاط
عملي

- أكتب المقطع البرمجي اللازم لتعريف قائمة (months)، ثم أضف العناصر إليها.
- استخدم موقع العنصر للوصول إلى العنصر ('May').
- استخدم الأعداد السالبة للوصول إلى العنصر ('Jun').
- أنفذ المقطع البرمجي للتحقق من صحته، وأتبع الأخطاء، وأعمل على تصحيحها.

إضاءة

استخدام الرقم (-1) يُقلِّل من نسبة حدوث الخطأ مقارنةً باستخدام موقع العنصر الأخير، مثل الرقم (6) في القائمة (days)؛ فالرقم (-1) يشير دائماً إلى العنصر الأخير بغض النظر عن طول القائمة، ويُسهِّل على الجميع فهم العنصر المقصود خلافاً للرقم (6) مثلاً؛ فهذا الرقم يتطلب من قارئ البرنامج معرفة أن طول القائمة هو (7)، واستنتاج أن الرقم (6) يُمثِّل موقع العنصر الأخير.

83

نشاط عملي

الإجابة:

```
months = 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep',
'Oct', 'Nov', 'Dec'
print(months[4])
print(months[-7])
```

إضاءة للمعلم / المعلمة:

في النشاط السابق قد يقع الطلبة ببعض من الأخطاء:

خطأ شائع: بدء العد من 1 بدلاً من صفر عند استخدام الأعداد الموجبة لموقع العنصر (شهر May هو الشهر الخامس؛ ولكنه موجود في المكان 4 وليس 5).

خطأ يجب الانتباه له: قد يكتب الطالب البرنامج نفسه من دون الأقواس المربعة، ويحصل على النتيجة نفسها من دون مشكلة، ولكن هذا خطأ:

```
months= 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'
print(months[4])
print(months[-7])
```

المشكلة في هذا البرنامج، أنه سيتم تعريف Tuple وليس List، وهو نوع آخر من البيانات، لا يُدرّس في الصف 11، ويمكن التحقق من ذلك:

```
>>> type(months)
```

```
<class 'tuple'>
```

أحد الفروقات بين النوعين هو أن الـ Tuple غير قابلة للتغيير، فمثلاً:

```
>>> months[0] = 'JAN'
```

Traceback (most recent call last):

File "<pyshell#145>", line 1, in <module>

```
months[0] = 'JAN'
```

TypeError: 'tuple' object does not support item assignment

ولأن البرنامج من دون الأقواس المربعة، سيعمل من دون مشكلات (لأنه لا يقوم بتعديل أي عنصر)، فقد يظن الطالب بأن الأقواس المربعة اختيارية في القوائم، وليس من الضروري إضافتها، وهذا استنتاج خطأ.

خطأ يجب الانتباه له: قد يكتب الطالب البرنامج نفسه من دون الفواصل بين العناصر:

```
months = ['Jan' 'Feb' 'Mar' 'Apr' 'May' 'Jun' 'Jul' 'Aug' 'Sep' 'Oct'
'Nov' 'Dec']
```

ومع أن هذه الجملة ليست خطأ من ناحية قواعد اللغة، إلا أنها تؤدي لنتيجة مختلفة عن المطلوب؛ إذ إن هذه الجملة ستؤدي لإنشاء قائمة مكونة من عنصر واحد، وهو نتيجة إلصاق جميع السلاسل المعطاة، أي أن البرنامج سيتحطم عند محاولة الوصول للعناصر الموجودة في المواقع 4 و 7- إذ لا يوجد إلا عنصر واحد في القائمة.

نشاط:

ما سبب استخدام جملة التحكم (break) في السطر (5)؟ وما تأثير حذفها في ناتج تنفيذ البرنامج؟

الإجابة: لن تختلف نتيجة البرنامج إن قمنا بحذف جملة break، ولكن ذلك سيؤدي لاستمرار البرنامج في العمل من دون داع؛ (لأن النتيجة وجدت) إلى أن يصل إلى نهاية القائمة، أي أن وجود جملة break، تمنع القيام بعمليات لا فائدة منها بعد إيجاد النتيجة؛ ولكنها ليست ضرورية لصحة البرنامج.

ما دلالة استخدام الجملة `range(len(readings)-1)`؟ وما تأثير تعديلها إلى `range(len(readings))` في ناتج تنفيذ البرنامج؟

```
1 found = False
2 for i in range(len(readings) - 1):
3     if readings[i] == readings[i+1]:
4         found = True
5         break
6
7 print(found)
```

الشكل (5-6): مثال على قوائم العناصر المتكررة.

في هذا المثال، عُقدت مقارنة بين كل عنصر والعنصر الذي يليه؛ فلكل قيمة من قيم (i)، تمت مقارنة العنصر الموجود في الموقع (i) بالعنصر الموجود في الموقع الذي يليه (i+1).

أحلل وأستنتج: استنادًا إلى المثال الوارد في الشكل (5-6):

■ ما سبب استخدام جملة التحكم (break) في السطر (5)؟ وما تأثير حذفها في ناتج تنفيذ البرنامج؟

■ ما دلالة استخدام الجملة `range(len(readings)-1)`؟ وما تأثير تعديلها إلى `range(len(readings))` في ناتج تنفيذ البرنامج؟

أناقش زملائي / زميلاتي ومعلمي / معلّمتي في إجاباتي.

إضاءة

الخطأ المعروف باسم "off-by-one error"، أو باسم الخطأ بخطوة واحدة، هو من أكثر الأخطاء البرمجية شيوعًا؛ قد يُخطئ المبرمج عند تحديد شرط نهاية الدوران في حلقة التكرار (while)، أو عند تحديد المدى في حلقة التكرار (for)؛ ما يدفع البرنامج إلى تنفيذ دورة واحدة أقل من المطلوب أو أكثر منه.

الإجابة:

■ إذا كانت هناك قائمة بطول 6 (مثلاً)، فإننا نستخدم `range(len(readings)-1)` لتعيين القيم 0, 1, 2, 3, 4 للمتغير i في حلقة التكرار.

■ يقارن البرنامج بين كل عنصر والعنصر الذي يليه؛ لذلك يجب أن تكون آخر مقارنة بين العنصر قبل الأخير (الموقع 4 في المثال السابق) والعنصر الأخير (الموقع 5 في المثال السابق)، أما إذا استخدمنا `range(len(readings))`، فإن آخر قيمة للمتغير i ستكون موقع العنصر الأخير (5 في المثال السابق)؛ ولذلك سيظهر خطأ في البرنامج عند محاولة تنفيذ المقارنة بين i و i+1 (لأنها بين العنصر الأخير وعنصر غير موجود).

ملحوظة: يبين المعلم للطلبة ان الإضاءة الموجودة في نهاية الصفحة يجب نقلها إلى درس الحلقات.

نشاط:

الحل الأول:

```
a = [0, 1, 2, 3, 4, 5]
for i in range(len(a)-1, -1, -1):
    print(a[i])
```

أخطاء ممكنة في هذا

الحل:

```
for i in range(len(a)-1, -1)
```

عدم تحديد حجم الخطوة (-1)، سيؤدي لعدم قيام حلقة التكرار بأي خطوة (لأن الخطوة الافتراضية موجبة).

```
for i in range(len(a), -1, -1)
```

البداية من len(a)، سيؤدي لتخطي البرنامج؛ لأنه سيحاول الوصول للعنصر الموجود في الموقع len(a)، وهو موقع خارج حدود القائمة.

```
for i in range(len(a), 0, -1)
```

الانتهاء عند صفر وليس (-1)، سيؤدي لعدم طباعة آخر عنصر؛ لأن دالة range لا تدخل الحد الأخير من المدى المعطى.

الحل الثاني:

```
a = [0, 1, 2, 3, 4, 5]
i = len(a) - 1
while i >= 0:
    print(a[i])
    i -= 1
```

ملحوظة: الحل الدارج في لغة البايثون لمثل هذا المطلوب هو بإحدى الطريقتين الآتيتين:

إثراء



يوجد نمط برمجي يستخدم متغير يلعب دور الراية (flag)، ويعتمد في كتابة البرامج على ما يأتي:

- 1- البدء بافتراض نتيجة بحث.
- 2- المرور على القائمة للتحقق من صحة الفرضية.
- 3- طباعة النتيجة بعد الانتهاء من المرور على القائمة.

في المثال السابق، أطلق على المتغير (found) اسم الراية (flag) التي تُرفع أو تُنزل عند اكتشاف وجود صفة ما أثناء عملية المرور. وهذا النمط في كتابة البرامج مفيد في عملية البحث، وله استخدامات كثيرة.

أناقش وأجرب:

كيف يمكن المرور على عناصر قائمة بالعكس؟ أناقش زملائي / زميلاتي ومعلمي / معلّمتي في هذا السؤال، ثم أجرب تنفيذ ذلك عملياً في بيئة بايثون (Python).



نشاط

العمليات في القوائم

توجد عمليات عدة يمكن تنفيذها في القوائم، مثل: الوصول إلى العناصر في قائمة ما، وإضافة عناصر جديدة إليها، وحذف عناصر منها، وترتيب العناصر فيها. وقد نُفذت بعض هذه العمليات بصورة فعلية في الأمثلة السابقة لهذا الدرس.

أجرب وأناقش: أجرب تنفيذ كل جملة من الجمل الآتية:

```
[1, 2, 3] + [98, 99, 100]
[1, 2, 3] + 5
[1, 2, 3] + [5]
[1, 2, 3] * 4
[1, 2, 3] * [1, 2, 3]
```



نشاط

```
a = [0, 1, 2, 3, 4, 5]
for e in reversed(a):
    print(e)
```

قد يتعرف الطلبة إلى دالة reversed لاحقاً في الدرس نفسه، إذا بحثوا في الإنترنت عن دوال أخرى غير المذكورة في جدول الدّوالّ.

```
a = [0, 1, 2, 3, 4, 5]
for e in a[::-1]:
    print(e)
```

تذكير: تُستخدم هذه الطريقة الاقتطاع (slicing)، وهذا ليس من منهاج الصف 110.

⚙️ نشاط:

■ أيّ الجمل يُمكن تنفيذها؟

الإجابة:

$[100, 99, 98] + [3, 2, 1]$

$[5] + [3, 2, 1]$

$4 * [3, 2, 1]$

■ أيّ الجمل لا يُمكن تنفيذها؟

الإجابة:

$5 + [3, 2, 1]$

$[3, 2, 1] * [3, 2, 1]$

ملحوظة: يبين المعلم أن اكتشاف الجمل التي يمكن تنفيذها في مفسر بايثون تكون بإضتفة دالة print قبل كل جملة.

مقارنة كل عنصر في القائمة الأولى بالعنصر الذي يُقَابِلُه في القائمة الثانية.

مثال:

القائمة [1, 2, 3] > من القائمة [3, 1]؛ لأنَّ العنصر الأوَّل (1) في القائمة الأولى أقل من العنصر الأوَّل في القائمة الثانية (3).
في حين أنَّ القائمة [1, 2, 3] < من القائمة [1, 0, 0, 0]؛ لأنَّ العنصر الثاني (2) في القائمة الأولى أكبر من العنصر الثاني (0) في القائمة الثانية.
كذلك يُمثَّلُ التحقق من وجود عنصر ما في القائمة واحدةً من العمليات المُهمَّة في القوائم. وأسهل طريقة لعمل ذلك هي استخدام العامل (in) على النحو الآتي:

```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> south = ['Aqaba', 'Maan', 'Tafeeleh', 'Karak']
>>> 'Amman' in south
False
>>> 'Karak' in south
True
>>>
```

يُمكن أيضًا استخدام العامل (not in) في التحقق من عدم وجود عنصر ما في القائمة كما يأتي:

```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> south = ['Aqaba', 'Maan', 'Tafeeleh', 'Karak']
>>> 'Amman' not in south
True
>>> 'Karak' not in south
False
>>>
```

أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن استخدامات العوامل الحسابية والعوامل المنطقية في التطبيقات العملية للحياة للغات البرمجة بايثون (Python)، ثمَّ أدوّن ما أتوصّل إليه من نتائج، وأشاركها مع زملائي/ زميلات في الصف.



أبحث

أُجَرِّب وأستكشف:

■ أعرّف قائمتين فارغتين ثم أضيف خمسة أعداد إلى كل منهما.

تذكير: خطأ شائع: عدم تحويل المدخل إلى رقم كما يأتي:

```
for i in range(5):
    num = input()
    list2 += [num]
```

هذا خطأ يصعب الانتباه إليه؛ لأن البرنامج في أغلب الأحوال سيعمل على نحو صحيح؛ ولكن المشكلة هي أن القائمة ستكون من سلاسل نصية بدلًا من أعداد، وهذا قد يؤدي لمقارنات غير متوقعة، فمثلاً 11 أكبر من 2 إذا كنا نقارن أرقامًا؛ ولكن 11 أقل من 2 إذا كنا نقارن سلاسل (لأن العنصر الأول في السلسلة الأولى '1' أقل من العنصر الأول في السلسلة الأخرى '2').

نشاط عملي: ⚙️

أكلف الطلبة العمل على النشاط كمهمة
بيئية فردية.

حل هذا التمرين موجود مباشرة في
الجدول (يمكن النسخ والاصق وتعديل
اسم القائمة فقط)؛ لذلك يفترض ألا
تكون هناك أي تحديات تواجه الطلبة عند
التطبيق؛ وألا يكون هناك شيء لمشاركته
مع الزملاء في الصف.

إضاءة 💡

من المفيد جدًا تعرّف أهم الدوال الجاهزة التي تُوفّرها اللغة، لكنّ معظم المُبرمجين المحترفين
لا يحفظون جميع هذه الدوال، وإنّما يعودون مرارًا إلى الموقع الإلكتروني للغة البرمجة
بايثون (Python)؛ للبحث عن الدوال المناسبة لبرامجهم، أو للتحقق من كيفية استخدام بعض
هذه الدوال.

أُعرّف الوظائف الجاهزة التي لها تعلق بالقوائم عن طريق الرابط الإلكتروني الآتي:

<https://docs.python.org/3/tutorial/datastructures.html>



أو عن طريق مسح الرمز سريع الاستجابة (QR Code) المجاور.



نشاط
عملي

أُجَرِّب وأستكشف:

- أستخدم الدوال الجاهزة في طباعة أكبر عنصر وأصغر عنصر في القائمتين اللتين أنشأتهما
في النشاط السابق.
- أستخدم الدوال الجاهزة في طباعة عناصر القائمة الأولى بشكل عكسي.

نشاط فردي:

■ أيُّ هذه الجمل تمكَّنت من تنفيذها؟
الإجابة:

name[0]
sorted(name)
max(name)
name += 'L'
name * 2
name < 'Zaid'
'Abd' in name

■ أيُّ هذه الجمل لم أتمكَّن من تنفيذها؟
الإجابة:

name[5] = 'h'
name.sort()

سلاسل الحروف (Strings)

يوجد تشابه بين سلاسل الحروف والقوائم، يتمثل في أنَّ كليهما تتألف من عناصر متتابعة. غير أنَّ جميع العناصر في السلسلة هي حروف (أي رموز تمثِّل حروفًا أبجدية، أو أرقامًا، أو علامات ترقيم، أو غير ذلك)؛ لذا يُمكنني تطبيق معظم ما تعلَّمته عن القوائم على سلاسل الحروف.

أُجرب وأستكشف:

أعرِّف سلسلة الحروف name = 'Amina-Abdu'، ثمَّ أُجرب تنفيذ الجمل الآتية:

```
name[0]
sorted(name)
max(name)
name += 'L'
name * 2
name < 'Zaid'
'Abd' in name
name[5] = 'h'
name.sort()
```



■ أيُّ هذه الجمل تمكَّنت من تنفيذها؟

■ أيُّ هذه الجمل لم أتمكَّن من تنفيذها؟

■ ما العامل المشترك بين الجمل التي لم أتمكَّن من تنفيذها؟

يُمكن الوصول إلى أيِّ حرف في السلسلة باستخدام الموقع (index)، ويُمكن أيضًا استخدام الدوال، مثل: sorted()، max()، len()، وcount(). وكذلك استخدام العوامل، مثل: + و* كما تعلَّمْتُ سابقًا، فضلًا عن إمكانية اختبار وجود عنصر في السلسلة باستخدام العامل (in) والعامل (not in)، بالرغم من وجود فروق بسيطة بينها؛ إذ تتحقَّق العوامل في القوائم من وجود عنصر ما في إحدى القوائم، في حين تتحقَّق العوامل في السلاسل من وجود سلسلة أُخرى داخل السلسلة نفسها.

■ ما العامل المشترك بين الجمل التي لم أتمكَّن من تنفيذها؟

الإجابة: جميعها تحاول التعديل على السلسلة.

تذكير:

الجمل الآتية لا تُعدِّل على السلسلة، وإنما تُنشئ سلاسل جديدة معدلة:

sorted(name)
max(name)
name += 'L'
name * 2

وهذا قد لا يكون واضحًا في جملة مثل a += 'L'، name، ولكن يمكن تفصيل هذه الجملة كما يأتي:

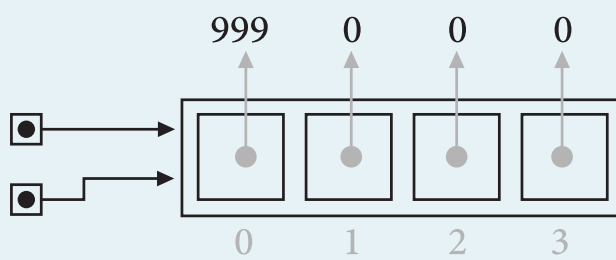
name = name + 'L'

إذ من الواضح هنا إنشاء سلسلة جديدة باستخدام خاصية الضم ثم بعد ذلك يُعَيَّن الاسم name لهذه السلسلة الجديدة بدلاً من السلسلة القديمة (أي أن السلسلة القديمة بقيت كما هي، ولكن لا يوجد اسم يشير لها الآن؛ لأن الاسم name أصبح يشير إلى السلسلة المعدلة).

```
nums1 = [0, 0, 0, 0]
nums2 = nums1.copy()
nums1[0] = 999
print(nums1)
[999, 0, 0, 0]
print(nums2)
[999, 0, 0, 0]
```

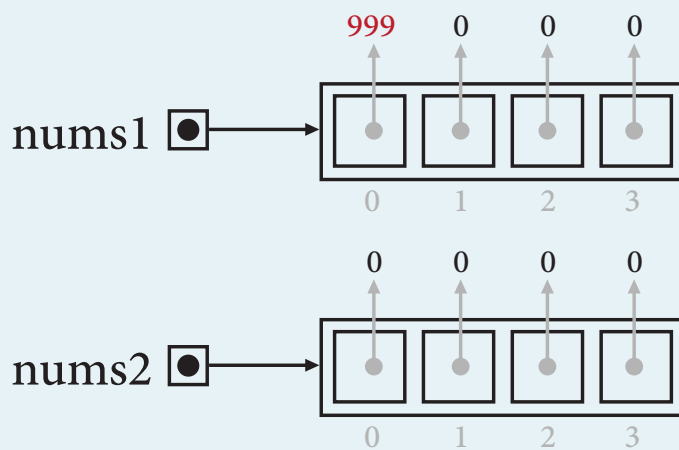
إضاءة للمعلم / المعلمة:

هناك تبعات؛ لأن القوائم قابلة للتغيير، ويجب أن نتنبه لها؛ إذ يوضح البرنامج الآتي خطأ يقع فيه كثير من المبتدئين عند



نلاحظ أيضًا أنه قد تغيرت القائمة nums2 مع أننا لم نغيرها، وإنما غيّرنا القائمة nums1، وسبب هذا هو أن nums1 و nums2 مؤشرات للقائمة نفسها، كما يوضح الرسم:

نفهم من ذلك بأن جملة `nums2 = nums1` لا تنسخ مكونات القائمة nums1 إلى القائمة num2، وإنما تعيّن المؤشر nums2 ليؤشر على القائمة نفسها التي يؤشر عليها المؤشر nums1. إذا أردنا نسخ مكونات القائمة التي يؤشر عليها المؤشر nums1 إلى قائمة جديدة، يؤشر عليها المؤشر nums2 (نقول تجاوزًا نسخ nums1 إلى nums2)، ونحتاج لاستخدام `copy()` كما يأتي:



كما هو واضح من المثال، لم تتغير قيمة العنصر الأول في nums2؛ لأن (copy) أنشأت نسخة جديدة من القائمة، وبذلك أصبح لدينا قائمتان منفصلتان لهما قيم العناصر نفسها (ما عدا العنصر الذي غيّر في nums1)، كما هو موضح في الصورة:

ويمكننا الحصول على النتيجة نفسها، إذا قمنا بعملية اقتطاع؛ لأنه عند اقتطاع جزء من القائمة (أو كامل القائمة كما في المثال الآتي)، تُنشأ قائمة جديدة تحوي العناصر المقطوعة.

لها نفس أثر `nums2 = nums1[:]` # `nums1.copy()`

لاحظ أن هذه ليست مشكلة في السلاسل؛ لأن السلاسل غير قابلة للتغيير (immutable)، أي أن وجود مؤشرين على السلسلة نفسها (غير القابلة للتغيير)، لا يمكن أن يؤدي إلى مشكلات غير متوقعة، كما حصل معنا في القوائم (القابلة للتغيير).

⚙️ نشاط عملي:

شرح الكود:

يبدأ الكود بجمع كلمة المرور من

المستخدم عبر input().

يقوم بعد ذلك بتعداد الأحرف الصغيرة

(small)، الأحرف الكبيرة (capital)،

والأرقام (number)، والرموز الخاصة

(special) داخل كلمة المرور باستخدام

حلقة for .

التحقق من الشروط:

بعد انتهاء الحلقة، يُتَحَقَّقُ من كل شرط

على حدة:

■ إذا كانت كلمة المرور أقل من 10

أحرف، تُطَبَّعُ رسالة تنبيه.

■ تُطَبَّعُ رسائل تنبيه إضافية إذا كانت

كلمة المرور تفتقر إلى الأحرف

الصغيرة، الكبيرة، الأرقام، أو الرموز

الخاصة.

التحقق النهائي:

إذا كانت جميع الشروط مستوفاة (أي كلمة المرور قوية)، تُطَبَّعُ رسالة تأكيد بأن كلمة المرور قوية.

يبدأ البرنامج المرور على أحرف كلمة السرّ، وعدّ مرّات تكرار كلّ من الأحرف الأبجدية (الصغيرة والكبيرة) والأرقام والرموز، ثمّ يتأكّد أنّ عدد مرّات التكرار لكل ما سبق لا يساوي صفراً، أنظر الشكل (5-9).

```
psw = input("Enter Password: ")
small = 0
capital = 0
number = 0
special = 0

for c in psw:
    if c.islower():
        small += 1
    elif c.isupper():
        capital += 1
    elif c.isdigit():
        number += 1
    else:
        special += 1

if len(psw) < 10:
    print("Password must be >= 10 characters long.")

if small == 0:
    print("Password must contain small letters.")

if capital == 0:
    print("Password must contain capital letters.")

if number == 0:
    print("Password must contain numbers.")

if special == 0:
    print("Password must contain special characters.")

if small != 0 and capital != 0 and number != 0 and special != 0 and
len(psw) >= 10:
    print("Strong password!")
```

الشكل (5-9): مثال على سلسلة كلمة المرور.

أنفُذ البرنامج في المثال السابق في بيئة بايثون (Python)، وألاحظ الناتج، ثمّ استكشف الأخطاء التي قد تحدث أثناء تنفيذ البرنامج، وأعمل على تصحيحها.



استكشف وأناقش:

أتبع الأوامر البرمجية في المثال السابق، واستكشف إمكانية كتابتها بطريقة أخرى، ثمّ أناقش زملائي/ زميلاتي في أهمية كل أمر، وتأثير حذفه في تنفيذ البرنامج بصورة صحيحة.



ثالثاً: الإثراء والتأمل والتقويم.

نشاط جماعي: 

أحدى الطرائق الأخرى لكتابة البرنامج:

```
psw = input('Enter password: ')
if len(psw) < 10:
    print('Password must be >= 10 characters long.')

small = 0
for c in psw:
    if c.islower():
        small += 1
if small == 0:
    print('Password must contain small letters.')

capital = 0
for c in psw:
    if c.isupper():
        capital += 1
if capital == 0:
    print('Password must contain capital letters.')

number = 0
for c in psw:
    if c.isdigit():
        number += 1
if number == 0:
    print('Password must contain numbers.')

special = 0
for c in psw:
    if not c.isalpha() and not c.isdigit():
        special += 1
if special == 0:
    print('Password must contain special characters.')

if small != 0 and capital != 0 and number != 0 and \
    special != 0 and len(psw) >= 10:
    print('Strong password!')
```

إضاءة للمعلم/ المعلمة

تعدّ شيفرة قيصر من أبسط الطرائق للتشفير، وأقلها أماناً خاصة إن كانت الرسالة المشفرة طويلة بما فيه الكفاية للقيام بتحليل إحصائي، لعدد مرات تكرار كل حرف في الرسالة؛ إذ إنه من المعروف بأن بعض الأحرف الإنجليزية، تظهر في الكتابة أكثر من غيرها بكثير؛ لذلك يمكن لشخص أن يحزر بعض الأحرف عن طريق تحديد الأحرف الأكثر تكراراً في الرسالة المشفرة، ومحاولة استبدال الأحرف بها، والتي نعلم أنها أكثر تكراراً في اللغة الإنجليزية.

ويعدّ البرنامج الذي كتبناه نوعاً خاصاً من شيفرة قيصر، ويسمى بـROT13، وهو من أقل الأنواع أماناً؛ لذلك لم تعد هذه الطريقة مستخدمة حالياً في تشفير الرسائل المهمة، ولكنها ما زالت تُستخدم لأغراض بسيطة مثل إخفاء حلول الأحاجي في المواقع الترفيهية، وما إلى ذلك.

إضاءة للمعلم/ المعلمة

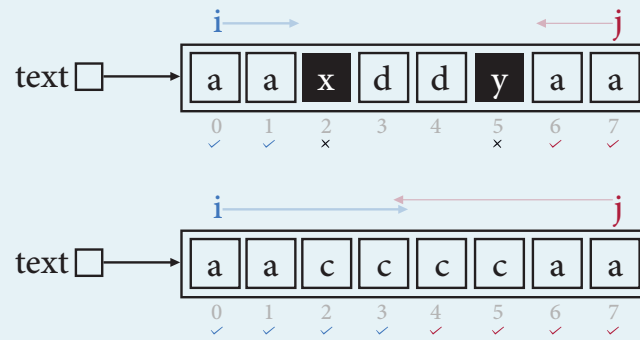
الكود البايثوني (Pythonic Code): يُطلق مبرمجو لغة البايثون وصف "بايثوني" على الكود الذي يستخدم ميزات لغة البايثون على نحو فعال؛ بحيث يكون متسقاً مع فلسفة اللغة، والطريقة التي تعارف على استخدامها أهل اللغة، فمثلاً مع أن البرنامج الذي كتبناه لشيفرة قيصر صحيح، إلا أن أي مبرمج محترف بلغة البايثون، قد لا يكون سعيداً برؤيته، ويقترح طرائق أكثر "بايثونية" لكتابته. ولتوضيح الصورة سنناقش المثال الآتي:

مثال

سنقرأ نصّاً من المستخدم، ثم نتأكد من إمكانية قراءة هذا النص من اليمين إلى اليسار، ومن اليسار إلى اليمين بالطريقة نفسها. فمثلاً كلمة "كرك" يمكن قراءتها من اليمين أو من اليسار بالطريقة نفسها، وعبارة "حصان ناصح" ينطبق عليها الشيء نفسه، ثم إن كلمات "reviver" و "rotator" و "kayak" بالإنجليزية، ينطبق ذلك عليها جميعاً.

```
1 text = input("Enter a text: ")
2
3 i = 0
4 j = len(text) - 1
5 result = True
6
7 while i < j:
8     if text[i] != text[j]:
9         result = False
10        break
11    i = i+1
12    j = j-1
13
14 print(result)
```

يستخدم هذا البرنامج متغيرين اثنين؛ أحدهما للمرور على الأحرف من اليسار إلى اليمين (يزيد في كل دورة) وآخر للمرور على الأحرف من اليمين إلى اليسار (يتناقص في كل دورة)، ويتوقف الدوران عند التقاء أو



أو تعاكس هذين المتغيرين. ويوضح الرسم الآتي كيفية حركة هذه المتغيرات في مثالين مختلفين:

يتوقف الدوران في المثال الأول عندما يكون الحرف من جهة اليمين غير مساوٍ للحرف من جهة اليسار، أما في المثال الثاني، فيتوقف الدوران عندما يصل كلٌّ من المتغيرين i و j للآخر ويتعداه؛ لأن الأحرف من اليمين واليسار جميعها متساوية.

هل هذه الطريقة الوحيدة لكتابة هذا البرنامج؟

قد يقترح عليك أحدهم الطريقة الآتية، ويصفها بأنها أكثر بايثونية:

```
text = input("Enter a text: ")
```

```
print(text == text[::-1])
```

يحقق هذان السطران (بأناقة) ما يحققه البرنامج السابق نفسه؛ إذ إن `text[::-1]` تُنشئ نسخة معكوسة من السلسلة، وعامل `==` يقارن بين النسخة الأصلية والنسخة المعكوسة.

تعليمات ذات صلة بمنتج التعلم:

- أذكر الطلبة بسياق التعلم، وأهمية دورهم بوصفهم ناشئين في مجال البرمجة، وأنهم سيعدّلون برنامج لعبة التخمين التفاعلية باستخدام Python.
- أوضح أن هذا التعديل يأتي في سياق التطوير المستمر للبرمجيات.
- أذكر الطلبة بما عملوا عليه حتى الآن في اللعبة، مع التركيز على دور القائمة الرئيسية في اللعبة، وكيفية الاستجابة لمدخلات اللاعب عن طريق عرض تعليمات اللعبة، أو كتابة رسالة معينة بناءً على الخيار المُدخل، وتعديل البرنامج مُمثلاً بتكرار عرض القائمة الرئيسية والخيارات.
- أوّجه الطلبة لاستخدام القوائم لتحسين البرنامج؛ إذ يُضافُ العنصر الرئيس في اللعبة، وهو إدخال البرنامج للرقم العشوائي.

المواطنة الرقمية:

- الأخلاقيات الرقمية: أحترم آراء الآخرين وأفكارهم عند مناقشة التعليمات البرمجية أو مناقشة المشروعات، وأقدّم النقد البناء والمساعدة للآخرين عند مراجعة تعليمات البرمجية.
- الوعي بالأمن السيبراني: أدرك أهمية استخدام برامج مكافحة الفيروسات وتحديث أنظمة التشغيل بانتظام أثناء العمل في بيئة بايثون (Python).

تصميم برنامج وإعداده لإنشاء لعبة تخمين الأرقام باستخدام لغة بايثون (Python) / المهمة (5).

أُكول - بالتعاون مع أفراد مجموعتي - تنفيذ مشروع التصميم والتطوير للعبة تخمين الأرقام. وكُنّا قد انتهينا في الخطوات السابقة من عرض القائمة الرئيسية في اللعبة، بالاستجابة لما يُدخله اللاعب عن طريق عرض تعليمات اللعبة، أو كتابة رسالة بحسب الخيار المُدخل، وتعديل البرنامج مُمثلاً بتكرار عرض القائمة الرئيسية والخيارات. والآن سأعمل - ضمن المجموعة - على إضافة العنصر الرئيس في اللعبة، وهو إدخال البرنامج للرقم العشوائي. سأستعمل - مع أفراد مجموعتي - دالة (random) لتوليد رقم مخفي في بداية اللعبة، يتألف من سلسلة تحوي (4) أعداد مختلفة، ثم أُخزّن الأرقام المُمكنة (المُحمّلة) في قائمة، وأعمل على تغيير ترتيبها باستخدام الدالة random.shuffle. أتُحقّق - مع أفراد مجموعتي - من استخدام القوائم بصورة صحيحة، ومن تفعيل عملية توليد الأرقام، ثم أحتفظ بما كُتِب - ضمن المجموعة - في ملف حتى يُمكن لأفراد المجموعة التعديل عليه في الخطوات القادمة.



مشروع

106

- أوّجه الطلبة لاستعمال وحدة (random) لتوليد رقم مخفي في بداية اللعبة؛ إذ يتألف من سلسلة تحوي (4) أعداد مختلفة، ثم تُخزّن الأرقام المُمكنة (المُحمّلة) في قائمة، وتغيير ترتيبها باستخدام الدالة random.shuffle.
- أوّجه الطلبة للتأكد من أن البرنامج مكتوب على نحو صحيح ومنظم، وإلى اختبار البرنامج؛ لضمان أنه يعمل من دون أخطاء، والتأكد من أن استخدام القوائم بصورة صحيحة، ومن تفعيل عملية توليد الأرقام.
- أوّجه الطلبة للاحتفاظ بالمقطع البرمجي المكتوب في ملف بايثون (.py)، ضمن مجموعة المشروع؛ حتى يمكن لأفراد المجموعة التعديل عليه في الخطوات القادمة.
- أشارك مع الطلبة معايير تقييم المهمة:
- الدقة في كتابة البرنامج: استخدام القوائم بصورة صحيحة، وتنظيم المقطع البرمجي وإضافة تعليقات توضيحية.
- الوظائف المطلوبة: التأكد من أن القوائم تعمل بصورة صحيحة، ومن تفعيل عملية توليد الأرقام.
- الاختبار: البرنامج يعمل من دون أخطاء.
- التعاون الجماعي: المشاركة الفعالة بين أعضاء الفريق.
- حفظ الكود: التأكد من حفظ الكود على نحو صحيح؛ ليتمكن الفريق من التعديل عليه لاحقاً.

إجابات نموذجية لأسئلة أقيم

تعلمي:

المعرفة: أستخدم ما تعلمته من معرفة في

هذا الدرس للإجابة عن الأسئلة الآتية:

السؤال الأول:

الإجابة

السلسلة هي قائمة؛ ولكنها غير قابلة للتغيير، ومكونة من رموز؛ أي أن الاختلاف بين الرئيسين في كون القائمة قابلة للتغيير، ويمكن أن تحتوي على عناصر من أي نوع (سلاسل، أرقام، قوائم أخرى، إلخ).

السؤال الثاني:

الإجابة:

```
str1 = str1[:-1]
```

أقيم تعلمي

المعرفة: أوظف في هذا الدرس ما تعلمته من معارف في الإجابة عن الأسئلة الآتية: السؤال الأول: ما أبرز أوجه التشابه وأوجه الاختلاف بين القوائم والسلاسل؟

أنظر الهامش

السؤال الثاني: أكتب جملة يمكن استخدامها في حذف الحرف الأخير من سلسلة تُسمى (str1).

أنظر الهامش

السؤال الثالث: أذكر طريقتين مختلفتين (أو أكثر) لتنفيذ كل مهمة من المهمات الآتية، ثم أحدد أيُّ هذه الطرائق أسهل أو أفضل:

1. طباعة العنصر الأول في قائمة تتألف من (10) عناصر.
2. طباعة العنصر الأخير في قائمة تتألف من (10) عناصر.
3. التأكد من وجود حرف 'a' في سلسلة ما.
4. ترتيب عناصر قائمة ما (باستخدام الدوال الجاهزة).
5. إيجاد العنصر الأكبر في قائمة ما.
6. حذف جميع عناصر قائمة ما.

أنظر الهامش

السؤال الثالث:

3-1: طباعة العنصر الأول من قائمة مكونة من 10 عناصر.

الطريقة الأولى:

```
print(mylist[0])
```

الطريقة الثانية:

```
print(mylist[-10])
```

الطريقة الثالثة:

```
print(mylist[-len(mylist)])
```

الطريقة الأولى أكثر مقروئية، وأقل عرضة للخطأ.

3-2: طباعة العنصر الأخير من قائمة مكونة من 10

عناصر.

الطريقة الأولى:

```
print(mylist[9])
```

الطريقة الثانية:

```
print(mylist[-1])
```

الطريقة الثالثة:

```
print(mylist[len(mylist)-1])
```

الطريقة الثانية أكثر مقروئية، وأقل عرضة للخطأ.

3-3: التأكد من وجود الحرف 'a' في سلسلة.

الطريقة الأولى:

'a' in mystr

الطريقة الثانية:

mystr.count('a') > 0

الطريقة الأولى أكثر مقروئية.

3-4: ترتيب عناصر قائمة (باستخدام الدوال الجاهزة)

الطريقة الأولى:

mylist.sort()

الطريقة الثانية:

mylist = sorted(mylist)

الطريقتان متشابهتان إلى حد كبير؛ ولكن الأولى أفضل؛ لأنها تُعدّل على القائمة مباشرة، بدلاً من إنشاء نسخة مرتبة، ثم تعيينها للقائمة.

3-5: إيجاد العنصر الأكبر في قائمة.

الطريقة الأولى:

max(mylist)

الطريقة الثانية:

```
while len(mylist) > 0:
    mylist.remove(mylist[0])
```

الطريقة الأولى أكثر مقروئية، وأسرع في التنفيذ.

3-6: حذف جميع عناصر القائمة

الطريقة الأولى:

mylist = []

الطريقة الثانية:

while len(mylist) > 0:

mylist.remove(mylist[0])

الطريقة الثانية سيئة للغاية، فهي بطيئة جداً في التنفيذ، وأقل مقروئية من الأولى.

المهارات:

السؤال الأول

1

```
a = [1, 3, 5, 2, 0, 4]

# 1
for i in range(len(a)):
    print(max(a[i:]))
```

النتيجة: 5 5 5 4 4 4

2

```
a = [1, 3, 5, 2, 0, 4]

# 2
for i in range(1, len(a)):
    print(max(a[:i]))
```

النتيجة: 1 3 5 5 5

المهارات: أوظف مهارة التفكير الناقد ومهارة حل المشكلات والمهارات البرمجية في الإجابة عن الأسئلة الآتية:

السؤال الأول: أتبّع البرنامج الآتي من دون تشغيله (أتبّع كل جزء من البرنامج بصورة مُنفصلة)، ثم أذكر ناتج تشغيله.

```
a = [1, 3, 5, 2, 0, 4]
# 1
for i in range(len(a)):
    print(max(a[i:]))
# 2
for i in range(1, len(a)):
    print(max(a[:i]))
# 3
for e in a[::-1]:
    a += [e]
print(a)
```

السؤال الثاني: أحدد الهدف الرئيس لكل برنامج من البرامج المُبيّنة في الجدول الآتي، ولا أصِف ما يقوم به البرنامج في كل سطر بصورة مُنفصلة، وإنما أستعمل بضع كلمات لتلخيص المهام التي يؤدّيها البرنامج بوجه عام.

<pre>result = [] for e in a: if e < 0: result = [e] + result else: result += [e] print(result)</pre>	<pre>found = False for x in a: for y in a: if x + y == 0: found = True break</pre>
<pre>for row in a: temp = row[0] row[0] = row[-1] row[-1] = temp</pre>	<pre>j = len(a) - 1 i = 0 for row in a: temp = row[i] row[i] = row[j] row[j] = temp i += 1 j -= 1</pre>

108

3

```
a = [1, 3, 5, 2, 0, 4]

# 3
for e in a[::-1]:
    a += [e]
print(a)
```

النتيجة: [1, 3, 5, 2, 0, 4, 4, 0, 2, 5, 3, 1]

السؤال الثاني:

البرنامج 1:

الإجابة: تجميع الأرقام السالبة في بداية القائمة، والموجبة في آخرها.

البرنامج 2:

الإجابة: التأكد من وجود رقمين في القائمة مجموعهم صفر.

البرنامج 3:

الإجابة: تبديل العمود الأول مع العمود الأخير

البرنامج 4:

الإجابة: تبديل القطرين بعضهما ببعض.

ملحوظة: تعتمد صحة تنفيذ البرامج السابقة على تعريف القائمة a قبل البدء بهذه الجمل البرمجية وللتأكد من صحة البرنامجين الأول والثاني تعرف القائمة a المعرفة في السؤال السابق: $[1,3,5,2,0,4]=a$

- البرنامجان الثالث والرابع، هما أفكار إثرائية تُناقش مع الطلبة في الصف، ويُقترح تطبيقها على المصفوفة التي استخدمت في نشاط لعبة $x,0$.

السؤال الثالث: أكتب برنامجًا لتحقيق كل من المهام الآتية:

المهمة 1: طباعة عدد الأرقام الزوجية الموجودة في قائمة:

```
count = 0
for e in a:
    if e % 2 == 0:
        count += 1
print(count)
```

المهمة 2: التأكد من أن أحد عناصر قائمة موجود في قائمة أخرى.

```
found = False
for e in a1:
    if e in a2:
        found = True
        break
```

المهمة 3: التأكد من وجود 3 أحرف متتابة متساوية في سلسلة أحرف.

```
found = False
for i in range(0, len(str1) - 2):
    if str1[i] == str1[i+1] and str1[i] == str1[i+2]:
        found = True
        break

print(found)
```

السؤال الرابع:

افترض أن القائمة board موجودة مسبقاً.

افترض مبدئي بأن القائمة مربعة.

```
square = True
for row in board:
    if len(row) != len(board):
        square = False
        break
if square:
    print("The board is square.")
else:
    print("The board is not square.")
```

أقدم للطلبة تعليمات لأداء المهمة وأتابع عملهم، وأحدد لهم زمنًا لتنفيذها وتسليمها.

الإجراءات:

1. توثيق الأخطاء البرمجية:

- أوجه الطلبة لجمع الأخطاء البرمجية التي تعرضوا لها بوصفهم مجموعات في أثناء التطبيق العملي لبرمجية بايثون في ملف معالج نصوص.
- أوجه الطلبة لتوثيق الحلول التي طبقتها المجموعة وكانت ناجحة، وأسهمت في حل المشكلات البرمجية في أثناء التطبيق.
- أوجه الطلبة لمشاركة المعلومات التي وثّقوها مع المجموعات الأخرى، وجمع التغذية الراجعة، والتعديل بناء عليها.

2. اختيار برنامج التصميم:

- أوجه الطلبة لاختيار برنامج لتصميم الكتيب. ومن البرمجيات المطروحة:
 - برنامج العروض التقديمية PowerPoint.
 - برنامج Canva.
 - Google Docs.
 - Online Brochure Maker.
- أو أيّ برنامج يختاره الطلبة.
- أوجه الطلبة لتصميم الكتيب، والتواصل مع الزملاء عبر منصة FigJam؛ للاتفاق على تفاصيل التصميم بين زملاء المجموعة.
- أوجه الطلبة لاختيار التصميمات الجاذبة والألوان والخطوط المتسقة.
- أوجه الطلبة لمراجعة المعلومات وتدقيقها.

3. النشر:

- أوجه الطلبة لنشر الكتيب بعد مراجعته وتدقيقه عبر موقع المدرسة، وعمل حملة إعلانية بين الطلبة؛ لإعلامهم بأهمية الاطلاع عليه.

الدوال البرمجية

Functions

عدد الحصص المقترح:

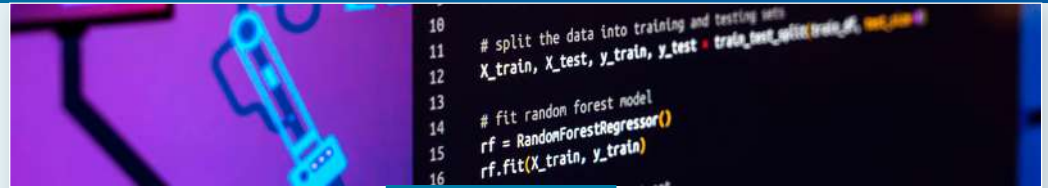
ثلاث حصص.

المصادر:

كتاب الطالب، لغة البرمجة بايثون.

الأدوات والتجهيزات:

أجهزة حاسوب، جهاز للعرض



الدرس السادس

الدوال البرمجية (Functions)

الفكرة الرئيسية:

تعرف آلية تجزئة المشكلة إلى أجزاء صغيرة، وكتابتها على أساس أنها جمل برمجية أو وحدات، وتعلم كيفية توثيق البرامج عن طريق التصميم والتطوير. كذلك تعرف الدوال البرمجية، وتعلم كيف يمكن استخدام الدوال الجاهزة واستيرادها، وكيف يمكن تعريف دوال جديدة واستخدامها في البرامج.

المفاهيم والمصطلحات:

الدالة البرمجية (Function)، الوحدة البرمجية (Module)، استيراد الوحدات (Importing Modules)، مدى المتغير (Variable's Scope)، المدخلات أو معاملات الدالة (Function Parameters)، التوثيق (Documentation)، سلاسل التوثيق (Docstrings).

نتائج التعلم (Learning Outcomes):

- أعرف المقصود بالوحدات البرمجية (Modules).
- أجزئ المشكلة إلى أجزاء صغيرة، وأعمل على تصميم كل جزء منها وبرمجته.
- أحدد الطريقة الفضلى لتمثيل أجزاء المشكلة في صورة جمل برمجية، أو روتين فرعي، أو وحدات، أو كائنات.
- أستعمل لغة البرمجة بايثون (Python) لاستدعاء روتين فرعي جاهز بناءً على وقوع حدث محدد.

مُنتجات التعلم (Learning Products)

أكتب برامج بلغة البايثون (Python) تستدعي دوال برمجية جاهزة.

أولاً: التهيئة

سياق التعلم:

أوضح للطلبة الفكرة الرئيسية من الدرس، وهي التعرف إلى كيفية استخدام الدوال الجاهزة واستيرادها، وتعريف دوال خاصة واستخدامها في برمجة بايثون (Python)، وأخبرهم أنهم سيكملون رحلة التعلم بوصفهم مبرمجين ناشئين، كالتي بدؤوا بها، وسيكملون العمل على تطوير برامج لألعاب بسيطة، تعتمد على واجهات نصية باستخدام برمجة بايثون.

أوضح لهم منتج التعلم المتوقع إنجازه خلال هذا الدرس، ضمن سياق التطوير المتكرر للبرمجيات، وأنهم سيطورون لعبة تخمين الأرقام التي بدؤوا بها عبر تنظيم المقطع البرمجي باستخدام الدوال؛ لتسهيل القراءة وإعادة الاستخدام، باستخدام برمجة بايثون (Python).

الإجابة:

دوال رأيناها مسبقًا:

print, input, range, len, max, min, sort, sorted, reverse, count, index, append, remove, upper, lower, replace, isupper, islower, isdigit, isnumeric, isalpha

تختلف هذه الدوال في أكثر من ناحية، منها:

1. هل تستقبل الدالة مدخلات؟

بعض الدوال تستقبل مدخلات

مثل:

print(...), range(...), len(...), max(...), min(...), sorted(...), count(...), index(...), append(...), remove(...), replace(...)

بعض الدوال لا تستقبل مدخلات

مثل:

sort(), reverse(), upper(), lower(), isupper(), islower(), isdigit(), isnumeric(), isalpha()

2. هل تقوم الدالة بإرجاع نتيجة؟

بعض الدوال ترجع نتيجة مثل:

input, len, max, min, sorted, count, index, replace, upper, lower, isupper, islower, isdigit, isnumeric, isalpha

بعض الدوال لا ترجع أي نتيجة مثل:

print, sort, reverse(), append, remove

3. كيف تُستدعى الدالة؟

بعض الدوال، تُستدعى بوضع اسم المتغير متبوعًا بنقطة، ثم اسم الدالة، مثل:

(على افتراض أن المتغير اسمه a).

تعرّفتُ سابقًا أنَّ الدالة البرمجية (Function) مقطع برمجي له اسم يؤدّي وظيفة ما، ويمكن استدعاؤه باستخدام اسمه؛ فكيف تُستخدم الدوال البرمجية في لغة البرمجة بايثون (Python)؟ وما أهميتها في تصميم البرامج وتسهيل قراءتها والتعامل معها؟



نشاط
تمهيدي

تعرّفتُ في الدروس السابقة الدوال الجاهزة في لغة البرمجة بايثون (Python)، التي يمكن استخدامها في معالجة القوائم والسلاسل أو أداء مهام أخرى. أكتب قائمة تحوي هذه الدوال، ثم أصنّفها بناءً على التشابه في كيفية استخدامها. بعد ذلك أبحث في أهم الجوانب التي تختلف فيها الدوال بعضها عن بعض.

الدوال البرمجية

إذا افترضتُ أنَّ الأمر print() لم يكن موجودًا في لغة البرمجة بايثون (Python)، وأنتَ يتعيّن عليّ دائمًا طباعة أيّ شيء على الشاشة بكتابة كامل الكود البرمجي الذي يقوم بالتعامل مع نظام التشغيل وأنواع البيانات المختلفة من أجل إظهارها على الشاشة بالشكل الصحيح، فإنّ ذلك سيكون مرهقًا لي بلا شك، ويجعل قراءة البرنامج عسيرة. ولهذا، فإنّ لغة البرمجة بايثون (Python) وفّرت علينا هذا الجُهد والعناء بتقديمها مقاطع برمجية جاهزة مُدقّقة وخالية من الأخطاء، بحيثُ يمكننا استدعاؤها بكل سهولة عن طريق اسمها، واستخدامها في برامجنا من دون حاجة إلى كتابة الأوامر دائمًا.

الدوال البرمجية الجاهزة

توفّر لغة البرمجة بايثون (Python) عددًا كبيرًا من الدوال البرمجية الجاهزة، وقد استخدمنا العديد منها في أمثلة سابقة ورد ذكرها في هذه الوحدة. ونظرًا إلى كثرة هذه الدوال، فإنّه يصعب على المبرمجين تذكرها جميعًا، أو حفظ كيفية استخدامها. ولهذا يُعدّ الموقع الإلكتروني للغة البرمجة بايثون (Python) الملاذ والصديق لكل مبرمج مُحترف؛ إذ يُواظِب كل منهم على زيارته باستمرار؛ للبحث عن دالة لوظيفة ما، أو تذكّر كيف تعمل إحدى الدوال.

الوحدات البرمجية (Modules)

تم تنظيم عدد الدوال الكبير في لغة البرمجة بايثون (Python) عن طريق جمعها في وحدات (modules)، تحتوي الوحدة الواحدة منها على دوال برمجية تشترك معًا في الغرض والاستخدام. فمثلًا، تحتوي وحدة (time) على دوال لها علاقة بالوقت والتاريخ، وتحتوي وحدة (math) على دوال لها علاقة بالعمليات الرياضية، وهكذا.

a.sort(), a.reverse(), a.count(...), a.index(...), a.append(...), a.remove(...), a.replace(...), a.lower(),
a.upper(), a.isalpha(), a.isnumeric()
بعض الدوال تُستدعى باسمها مباشرة مثل:
print(...), input(...), len(...), range(...), sorted(...), min(...), max(...)

ثانيًا: عملية التعليم والتعلم:

الاجابة

```
import random

def random_goodbye():
    goodbyes = ["Good Bye!!", "مع السلامة"]
    print(random.choice(goodbyes))

def random_greeting():
    greetings = ["Hello!!", "أهلاً وسهلاً"]
    print(random.choice(greetings))

name = input("What is your name? ")
random_greeting()
age = input("How old are you? ")
print("طولة العمر إن شاء الله")
random_goodbye()
```

لا يؤثر ترتيب تعريف الدوال في صحة عمل البرنامج، ما دامت تُعرّف الدالة قبل استدعائها؛ لذلك فإن نقل دالة الوداع للبداية، لا يؤثر في صحة البرنامج.

```
import random

name = input("What is your name? ")
random_greeting()
age = input("How old are you? ")
print("طولة العمر إن شاء الله")
random_goodbye()

def random_goodbye():
    goodbyes = ["Good Bye!!", "مع السلامة"]
    print(random.choice(goodbyes))

def random_greeting():
    greetings = ["Hello!!", "أهلاً وسهلاً"]
    print(random.choice(greetings))
```

عند نقل الدوال للنهاية، سيؤدي ذلك إلى إظهار رسالة خطأ، وإلى توقف البرنامج عن العمل عند وصوله لجملة الاستدعاء؛ لأن الدالة التي تُستدعى، لم تعرف بعد.

```
import random

def random_greeting():
    greetings = ["Hello!!", "أهلاً وسهلاً"]
    print(random.choice(greetings))

def random_goodbye():
    goodbyes = ["Good Bye!!", "مع السلامة"]
    print(random.choice(goodbyes))

name = input("What is your name? ")
random_greeting()
age = input("How old are you? ")
print("طولة العمر إن شاء الله")
random_goodbye()
```

الشكل (2-6): مثال على برنامج مُكامل يعمل على تعريف دالتين جديدتين واستدعائهما.

أُجرب بنفسِي: أقوم بطباعة البرنامج الموضح في الشكل 2-6 ثم أقوم بتشغيله عدة مرات وملاحظة اختلاف المخرجات في كل مرة، ثم أقوم بتجربة التالي ومناقشته مع زملائي:

- نقل كامل تعريف دالة الوداع random_goodbye لتصبح قبل دالة الترحيب random_greeting ، هل أثر ذلك على صحة عمل البرنامج ولماذا؟
- نقل كامل تعريف الدالتين ليصبحا في نهاية البرنامج بدلاً من بدايته (أي بعد الجمل التي تسأل عن الاسم والعمر وتقوم باستدعاء الدوال) ، هل أثر ذلك على صحة عمل البرنامج ولماذا؟

يتطلب استخدام بعض الدوال توافر مجموعة من البيانات، تعمل الدوال على تحليلها ومعالجتها. فمثلاً، لا يمكن استخدام دالة الجذر التربيعي قبل استقبال رقم يُمكن من حساب جذره التربيعي. ولهذا يجب تحديد عدد مدخلات (أو مُعاملات) الدالة (parameters) عند تعريفها، وإعطاء كل مدخل (مُعامل) اسماً.

عند تشغيل البرنامج، ستظهر النتيجة الآتية على شاشة جهاز الحاسوب:

78.53981633974483

78.53981633974483



نشاط
عملي

أُجَرَّب وأستنتج: أدخل البرنامج الوارد في الشكل (4-6)، ثمَّ أعدّل الأوامر البرمجية، بحيث تطلب من المُستخدم إدخال نصف القطر، ثمَّ استدعاء الدالة وطباعة النتيجة.

تعمل الدالة في المثال السابق على استقبال رقم يُمثّل نصف قطر دائرة، وإرجاع مساحة هذه الدائرة، علمًا بأنَّ عملية إرجاع النتيجة تمّت باستخدام (return)؛ إذ أنهى تنفيذ الدالة عند تلك النقطة، وتمَّ إرجاع النتيجة. ولهذا لن يُنفَّذ أيُّ أمر يأتي بعد هذه الجملة في الدالة (في حال وجود أيِّ سطر). يُمكن استخدام أكثر من جملة (return) في الدالة نفسها، بحيث تُنفَّذ كل جملة في حالة مختلفة عن الأخرى، أنظر الشكل (5-6) الذي يبيّن مثالاً على دالة تعمل على إرجاع حرف مختلف يُمثّل مُعدّل الطالب بحسب علامته:

```
def letter_grade(grade):
    if grade >= 90:
        return 'A'
    if grade >= 80:
        return 'B'
    if grade >= 70:
        return 'C'
    if grade >= 60:
        return 'D'
    return 'F'
```

لن نصل لهذا السطر إن تم تنفيذ جملة return السابقة

سنصل لهذا السطر فقط إن لم يتم تنفيذ أي من جمل return السابقة

الشكل (5-6) مثال على استخدام جملة (return) لإرجاع نتيجة ما.



نشاط
جماعي

أعمل - بالتعاون مع أفراد مجموعتي - على تعديل البرنامج السابق وتطويره، بحيث يتمكّن من جمع العلامات وإيجاد المُعدّل النهائي، ثمَّ أستخدم دالة letter_grade() في تصنيف المُعدّل.

```
import math

def area(radius):
    return math.pi * radius**2

r = float(input('Enter the circle radius: '))
print('Circle area = ', area(r))
```

الإجابة:

إضافة الكود الآتي بعد الدالة:

```
course1 = int(input('Enter grade for course 1: '))
course2 = int(input('Enter grade for course 2: '))
course3 = int(input('Enter grade for course 3: '))

avg = (course1 + course2 + course3) // 3
print('Your average is: ', avg)
print('Your grade is : ', letter_grade(avg))
```

إضاءة

يجب التحقق من وجود جملة (return) لكل حالة مُمكنة في الدالة التي تعيد نتيجة ما.

اقرأ البرنامج الآتي جيّدًا، ثمّ أحلّ - بالتعاون مع أفراد مجموعتي - الأوامر البرمجية الواردة في الشكل (6-6)؛ لتحديد الأخطاء المُحتملة، ثمّ اقترح الحلول المناسبة، وتجربتها في بيئة بايثون (Python).

```
def absolute_value(x):
    if x > 0:
        return x
    if x < 0:
        return -x
x = float(input("Enter a number to find its absolute value: "))
print("The absolute value is:", absolute_value(x))
```

الشكل (6-6): مثال على الأخطاء البرمجية.



نشاط
جماعي

مدى المتغيرات (Scope)

لكل متغير مدى (scope) يُمكن استخدامه في هذا المتغير. فمثلاً، مدى المتغير (AVOGADRO) في البرنامج الوارد في الشكل (6-7) هو جميع الأسطر التي تلي أول سطر استخدم فيه المتغير؛ لذا يُمكن استخدام هذا المتغير داخل الدوال التي تأتي بعده، وكذلك استخدامه خارج هذه الدوال. أمّا المتغير (result) فمداه هو جميع الأسطر التي تليه؛ لذا لا يُمكن استخدامه داخل الدوال (تأتي الدوال في الملف قبل أول استخدام للمتغير).

118

نشاط جماعي

الإجابة:

```
def absolute_value(x):
    if x > 0:
        return x
    if x < 0:
        return -x
x = float(input("Enter a number to find its absolute value: "))
print("The absolute value is:", absolute_value(x))
```

- الخطأ الأول: اسم الدالة يحتوي على فراغ.
- الخطأ الثاني: يجب أن تكون هناك مسافة بدء قبل محتويات الدالة.
- الخطأ الثالث: لا تعيد الدالة النتيجة المتوقعة إذا كانت قيمة x صفرًا.

نشاط جماعي:

أعدّل على البرنامج الوارد في الشكل (6-9) بتنفيذ الاستدعاء `print(atoms)` خارج الدالتين. ما تأثير ذلك في تنفيذ البرنامج؟
الإجابة: سنحصل على الخطأ.

`NameError: name 'atoms' is not defined`

لأن الاسم `atoms` معرف فقط داخل الدالتين، أي أن هناك متغيراً محلياً في الدالة الأولى اسمه `atoms`، ومتغيراً محلياً آخر في الدالة الثانية اسمه `atoms`، ولا وجود لهذه المتغيرات خارج الدالتين.

أعدّل على البرنامج نفسه بتنفيذ الاستدعاء `print(num_atoms)` داخل إحدى الدالتين. ما تأثير ذلك في تنفيذ البرنامج؟

```

AVOGADRO = 6.022 * 10**23

def atoms_to_moles(atoms):
    moles = atoms / AVOGADRO
    return moles

def moles_to_atoms(moles):
    atoms = moles * AVOGADRO
    return atoms

num_atoms = 1.204e24
result = atoms_to_moles(num_atoms)
print("Number of atoms: ", num_atoms)
print("Moles: ", result)

```

الشكل (6-7): مثال على مدى المتغيرات.

ألاحظ أن مدى كل من المتغير `(atoms)` والمتغير `(moles)` ينتهي بانتهاء الدالة؛ لذا يُطلق على هذا النوع من المتغيرات اسم المتغيرات المحلية (`local variables`)؛ فهي مُعرّفة فقط محلياً داخل الدالة. أما مدى كل من المتغير `(AVOGADRO)` والمتغير `(result)` فغير محصور داخل الدالة؛ لذا يُطلق على هذا النوع من المتغيرات اسم المتغيرات العامة (`global variables`). يُذكر أن المتغير `(atoms)` والمتغير `(moles)` المُعرّفين داخل الدالة الأولى هما متغيران مختلفان عن المتغير `(atoms)` والمتغير `(moles)` المُعرّفين داخل الدالة الثانية؛ فكل من هذه المتغيرات مُعرّفة محلياً في مدى ما، ولا علاقة له بالمتغيرات المُعرّفة في المدى الآخر.

أجرب وأستنتج:

- أعدّل على البرنامج الوارد في الشكل (6-7) بتنفيذ الاستدعاء `print(atoms)` خارج الدالتين. ما تأثير ذلك في تنفيذ البرنامج؟
- أعدّل على البرنامج نفسه بتنفيذ الاستدعاء `print(num_atoms)` داخل إحدى الدالتين. ما تأثير ذلك في تنفيذ البرنامج؟

عند استدعاء دالة ما، فإن مكاناً خاصاً لها يُحجز في الذاكرة، ويُسمّى إطار ذاكرة التكديس (`Stack Frame`)؛ إذ يستعمل هذا المكان لحفظ كل ما يتعلق بالدالة من متغيرات محلية، ثم يُنخلَص منه عند الانتهاء من تنفيذ الدالة. لتوضيح ذلك، أتتبع البرنامج الوارد في الشكل (6-8).

الاجابة: لا مشكلة في ذلك؛ إذ ستطبع قيمة `num_atoms` على نحو صحيح؛ لأن استدعاء الدالة يأتي بعد تعيين قيمة للمتغير `num_atoms` في المدى العام. في المقابل فإن البرنامج الآتي سيعطي `NameError`؛ لأن استدعاء الدالة تمّ قبل أول استخدام للمتغير `num_atoms` في المدى العام.

```

def atoms_to_moles(atoms):
    moles = atoms / AVOGADRO
    print(num_atoms)
    return moles

result = atoms_to_moles(1.204e24)
num_atoms = 1.204e24
print("Number of atoms: ", num_atoms)
print("Moles: ", result)

```


نشاط عملي:

ما نتيجة البرنامج الوارد في ما يأتي؟ كيف أفسر هذه النتيجة؟

```
x = 6
def f():
    x = 1
f()
print(x)
```

النتيجة:

6

والتفسير هو كما يأتي:

- عرّف المتغير x في المدى العام، وعُيّنَت القيمة 6 له.
- عند استدعاء الدالة، عرّف متغير جديد محلي داخل الدالة اسمه x أيضاً، وعُيّنَت القيمة 1 لهذا المتغير المحلي.
- عند الانتهاء من الدالة تم التخلص من إطار ذاكرة

```
1 def fun1(x):
2     x = 3
3
4 def fun2(x):
5     x = 5
6
7 x = 7
8 fun1(x)
9 fun2(x)
10 print('x: ', x)
```

الشكل (8-6): مثال على استدعاء الدالة وإرسال قيمة المتغير x إلى متغير محلي

- ألاحظ من البرنامج السابق ما يأتي:
7. تعريف المتغير x في المدى العام بقيمة (7).
 8. استدعاء الدالة $fun1$ ، ثم إرسال القيمة (7) إلى متغير محلي داخل الدالة، يُسمّى x .
 9. تغيير قيمة المتغير المحلي x إلى (3).
 10. انتهاء الدالة، والتخلص من المتغير المحلي x الذي يحمل القيمة (3).
 11. العودة إلى المدى العام، حيث يوجد متغير يُسمّى x ، وتبلغ قيمته (7).
 12. استدعاء الدالة $fun2$ ، ثم إرسال القيمة (7) إلى متغير محلي داخل الدالة، يُسمّى x .
 13. تغيير قيمة المتغير المحلي x إلى (5).
 14. انتهاء الدالة، والتخلص من المتغير المحلي x الذي يحمل القيمة (5).
 15. العودة إلى المدى العام، حيث يوجد متغير يُسمّى x ، وتبلغ قيمته (7).
 16. اكتمال طباعة قيمة المتغير x .
- إذن، ناتج تنفيذ البرنامج هو (7).

أجرب بنفسي:

ما نتيجة البرنامج الوارد فيما يلي؟ كيف أفسر هذه النتيجة؟

```
x = 6
def f():
    x = 1
f()
print(x)
```

نشاط
عملي

120

- التكديس الخاص بالدالة، ومن ثم التخلص من المتغير x الذي يحمل القيمة 1.
- عند طباعة قيمة x طبعت قيمة المتغير الموجود في المدى العام الذي يحمل القيمة 6.

تذكير:

- خطأ شائع: من أكثر الأخطاء شيوعاً عند استخدام المتغيرات العامة مع الدوال محاولة تغييرها على نحو خطأ داخل الدالة؛ لذلك من المهم التركيز على ذلك خلال التدريس. فمثلاً يتوقف البرنامج الآتي مع ظهور رسالة خطأ عند السطر 3:

```
x = 6
def f():
    x = 1
f()
print(x)
```

UnboundLocalError: cannot access local variable 'x' where it is not associated with a value

والسبب أنه رُبطَ الاسم `x` بالمتغير المحلي داخل الدالة، وهو ما جعل طباعته قبل تعيين قيمة له غير ممكن.
والخطأ نفسه سيحصل في البرنامج الآتي في السطر 3 وللسبب السابق نفسه؛ إذ إن الجملة `x += 1` تُترجم إلى `x = x + 1`، وهي تربط الاسم `x` بمتغير محلي، وتحاول جمع الرقم 1 معه قبل تعيين قيمة له.

```
x = 6
def f():
    print(x)
    x = 1
f()
print(x)
```

UnboundLocalError: cannot access local variable 'x' where it is not associated with a value

ثالثاً: الأثراء والتأمل والتقويم.

مثال إراثي:

سنستعرض في هذا الجزء الاختياري من الدرس مثالاً عن كيفية استخدام الدوال لتسهيل عملية تصميم الخوارزميات، وتجزئة المشكلات إلى مشكلات أصغر تمهيداً لحلها.

سنكتب في هذا المثال برنامجاً، يطلب من المستخدم إدخال رقم، ثم يطبع جميع الأعداد الأولية التي هي أصغر من ذلك الرقم. فمثلاً إن كان الرقم المدخل هو 10، فتُطبع الأرقام 1، 2، 3، 5، 7.

للتذكير: العدد الأولي هو العدد الذي لا يقبل القسمة إلا على نفسه أو على 1.

عند التفكير في هذه المسألة، يمكننا تقسيمها إلى مسألتين:

■ الأولى التأكد من كون عدد ما عدداً أولياً.

■ الثانية المرور على جميع الأعداد الأصغر من العدد المدخل؛ للتأكد من كونها أعداداً أولية.

وحتى نسهل على أنفسنا الحل، سنبدأ بكتابة دالة تستقبل رقماً، وتُرجع **True** إن كان العدد أولياً، و **False** إن كان غير ذلك؛ أي أننا سنهتم بهذا الجزء فقط من المسألة، ثم سنهتم بأمر المسألة الكاملة لاحقاً.

ستكون خطتنا للتأكد من كون العدد أولياً؛ بأن نقسمه على جميع الأعداد التي هي أصغر منه، فإن قبل القسمة على أحدها، علمنا بأن العدد غير أولي.

```
def is_prime(num):
    for x in range(2, num):
        if num % x == 0:
            return False
    return True
```

يمرُّ هذا البرنامج على جميع الأرقام من 2 إلى num، والتأكد من باقي قسمة num على تلك الأرقام، وفي حال كان باقي القسمة صفراً، نعلم بأن num ليس أولياً؛ لأنه قبلَ القسمة على أحد الأرقام.

تذكير:

لا نحتاج للمرور على جميع الأرقام التي تقل عن num، وإنما يكفي المرور على الأرقام التي تقلُّ عن أو تساوي الجذر التربيعي للرقم num؛ لأن الأرقام التي تزيد عن ذلك، لا يمكن أن تكون من عوامل num. أول من وصف هذه الطريقة للتأكد من كون العدد أولياً هو عالم الرياضيات الإيطالي فيبوناتشي (توفي 1240 ميلادي)؛ إذ ذكرها في كتابه المعروف "ليبر أباشي (Liber Abaci)" الذي اعتمد فيه على نحو كبير على أمثلة وطرائق وصفها قبله علماء الرياضيات العرب والمسلمون.

يمكننا الآن تجربة هذه الدالة على نحوٍ منفصل، والتأكد من صحتها قبل المضي قدماً للخطوة الثانية، وهي حلُّ المسألة الأصلية.

لحلُّ المسألة الأصلية، سنمرُّ على الأرقام من 1 إلى الرقم المدخل من المستخدم، ونرسل كل واحد من هذه الأرقام للدالة التي كتبناها؛ للتأكد من كون العدد أولياً.

```
n = int(input("Enter a number: "))
for i in range(1, n):
    if is_prime(i):
        print(str(i) + ' is prime')
```

من الواضح أن حل المسألة هذا قصير وسهل؛ بسبب استخدامنا لدالة is_prime؛ إذ إن البرنامج كان سيكون أطول وأكثر صعوبة في القراءة والشرح، إن كانت محتويات الدالة كاملة داخل حلقة التكرار.

إضاعة للمعلم/المعلمة:

خطأ شائع لا علاقة مباشرة له بالأنشطة: يعتقد بعض المبتدئين أن أسماء المتغيرات المستخدمة لإرسال قيمٍ لدالة، ترتبطُ بأسماء المعاملات المستخدمة في تعريف الدالة، فمثلاً:

```
def fun(a, b):
    if a > b:
        print('a is larger')
    elif b > a:
        print('b is larger')
    else:
        print('a and b are the same')

a = 5
b = 6
fun(a, b)
fun(b, a)
```

في هذا البرنامج، قد يظن بعض المبتدئين أن نتيجة الاستدعاءين هي نفسها:

b is larger

ولكن الدالة لا تهتم بأسماء المتغيرات التي تحمل القيم المرسلة لها، فالقيمة الأولى دائماً ستحمل اسم **a**، والقيمة الثانية دائماً ستحمل الاسم **b**، لذلك فإن الاستدعاء الأول نتيجته:

b is larger

والاستدعاء الثاني نتيجته:

a is larger

تعليمات ذات صلة بمنتج التعلم

- أخبر الطلبة أنه قد وصلنا الآن إلى مرحلة، نستطيع فيها تجميع الأجزاء المنفردة التي كتبناها مسبقاً لإنهاء كتابة لعبة (نجوم وأقمار).
- أوجه الطلبة لمراجعة مخطط سير العمليات الذي أنشؤوه في الدرس الأول من هذه الوحدة، ومن ثم مراجعة البرنامج للتحقق من أجزائه، وكيفية انتقاله من جزء إلى آخر.

1. بداية اللعبة:

أ- عرض رسالة الترحيب.

2. عرض القائمة الرئيسية:

أ- الخيار الأول: عرض تعليمات اللعبة.

ب- الخيار الثاني: بدء اللعبة.

ج- الخيار الثالث: الخروج من اللعبة.

3. بدء اللعبة:

أ- توليد الرقم المضمّر.

ب- بدء المحاولات (الحد الأقصى (10) محاولات):

■ إدخال التخمين.

■ التحقق من صحة التخمين.

■ حساب عدد النجوم والأقمار.

■ عرض النتائج.

4. التحقق من الفوز:

أ- في حال الفوز: عرض رسالة التهئة.

ب- في حال الخسارة: لاستمرار في المحاولات.

5. نهاية اللعبة:

أ- عرض رسالة النهاية.

■ أوجه الطلبة لمراجعة البرنامج لعبة التخمين وتعديله وتحسينه بناءً على ما تعلّمته في هذه الوحدة من تقنيات برمجية،

المشروع: تصميم برنامج وإعداده لإنشاء لعبة تخمين الأرقام باستخدام لغة بايثون (Python)/ المهمة (6).

وصلنا الآن إلى مرحلة نستطيع فيها تجميع الأجزاء المتفرقة التي كتبناها سابقاً لإنهاء كتابة لعبة (نجوم وأقمار).

أراجع مخطط سير العمليات الذي أنشأته في الدرس الأول من هذه الوحدة، ثم أنعم النظر في المخطط الآتي الذي يعرض الصورة العامة لأهم أجزاء البرنامج، وكيفية انتقاله من جزء إلى آخر.

مخطط سير العمليات (Flowchart) للعبة تخمين الأرقام:

1. بداية اللعبة:

أ. عرض رسالة الترحيب.

2. عرض القائمة الرئيسة:

أ. الخيار الأول: عرض تعليمات اللعبة.

ب. الخيار الثاني: بدء اللعبة.

ج. الخيار الثالث: الخروج من اللعبة.

3. بدء اللعبة:

أ. توليد الرقم المضمّر.

ب. بدء المحاولات (الحد الأقصى هو (10) محاولات):

■ إدخال التخمين.

■ التحقق من صحة التخمين.

■ حساب عدد النجوم والأقمار.

■ عرض النتائج.

التحقق من الفوز:

■ في حال الفوز: عرض رسالة التهئة.

■ في حال الخسارة: الاستمرار في المحاولات.

4. نهاية اللعبة:

أ. عرض رسالة النهاية.

والآن سأبدأ تعديل برنامج اللعبة وتحسينه بناءً على ما تعلّمته في هذه الوحدة من تقنيات برمجية، وأستخدم الدوال البرمجية في تسهيل قراءة البرنامج على النحو الآتي:

1. طباعة الرسالة الترحيبية للعبة، وكذلك خيارات القائمة الرئيسة (def welcome).

واستخدام الدوال البرمجية في تسهيل قراءة البرنامج على النحو الآتي:

• طباعة الرسالة الترحيبية للعبة، وكذلك خيارات القائمة الرئيسة (def welcome).

• طباعة تعليمات اللعبة (def about)؛ ذلك أن الأجزاء طويلة نسبياً، ووجودها داخل البرنامج قد يجعل قراءته مهمة عسيرة، ومن ثمّ يمكن عزلها في دوال منفصلة.

■ أوجه الطلبة لملاحظة وجود أجزاء من اللعبة، يجب إعادة استخدامها في أكثر من مكان. فمثلاً، لا يُظهر المخطط الأجزاء المتعلقة باللعب للاعبين اثنين؛ لذا يمكن افتراض أن كتابة هذا الجزء من اللعبة، سيتضمّن الأجزاء الآتية:

• التأكد أن الرقم الذي أدخله اللاعب هو من الأرقام المسموح بها (يتألف من أربعة أعداد).

• عدّ النجوم في الرقم المُدخل.

• عدّ الأقمار في الرقم المُدخل.

■ أؤكد للطلبة أهمية تجنّب كتابة هذه الأجزاء في أكثر من مكان في البرنامج، وذلك بتعريف دوال خاصة بها،



مشروع

واستدعائها عند الحاجة كما يأتي:

```
def is_valid(guess):
```

استقبال سلسلة، والتأكد أنها تتألف من (4) أرقام، وتعيد (True) أو (False).

```
def count_stars(guess,
```

```
secret):
```

استقبال الرقم المضمّر، وتوقع اللاعب، وإعادة رقم يُمثّل عدد النجوم بناءً على ذلك.

```
def count_moons(guess,
```

```
secret):
```

استقبال الرقم المضمّر، وتوقع اللاعب، وإعادة رقم يُمثّل عدد الأقمار بناءً على ذلك.

■ أوجه الطلبة للبدء بالعمل عبر كتابة هذه الدوال (يُمكن نسخ

المقطع البرمجي الذي كتبناه مسبقاً) واستخدامها، وأحاول الجمع بينها، بحيث يحاكي منطق البرنامج مخطط سير العمليات الذي اتفق عليه أفراد المجموعة.

■ أوجه الطلبة لتقسيم البرنامج إلى دوال، مثل:

```
welcome(), about(), is_valid(), count_moons(), count_stars().
```

تلميحات:

1. لا بُدّ من وجود حلقة تكرار (لانهائية) تحيط بالعبة كلها، بحيث يعود اللاعب دائماً إلى رسالة الترحيب بعد الانتهاء من أيّ لعبة.
2. لا داعي الآن لكتابة الجزء المتعلّق باللعبة للاعبين اثنين؛ إذ يُمكن تأجيل ذلك إلى وقت لاحق.
3. يُمكن فصل الجزء المتعلّق باللعبة للاعب واحد في دالة مُنفصلة بهدف تنظيم البرنامج، وكذا الحال بالنسبة إلى

2. طباعة تعليمات اللعبة (def about()؛ ذلك أنّ الأجزاء طويلة نسبياً، ووجودها داخل البرنامج قد يجعل قراءته مُهمّة عسيرة، ومن ثمّ يُمكن عزلها في دوال مُنفصلة. لاحظ وجود أجزاء من اللعبة يجب إعادة استخدامها في أكثر من مكان. فمثلاً، لا يُظهر المخطط الأجزاء المتعلّقة باللعبة للاعبين اثنين؛ لذا يُمكن افتراض أنّ كتابة هذا الجزء من اللعبة سيتضمّن الأجزاء الآتية:

1. التأكد أنّ الرقم الذي أدخله اللاعب هو من الأرقام المسموح بها (يتألف من أربعة أرقام).
2. عدّ النجوم في الرقم المُدخّل.
3. عدّ الأقمار في الرقم المُدخّل.

أُجنب كتابة هذه الأجزاء في أكثر من مكان في البرنامج، وذلك بتعريف دوال خاصة بها، واستدعائها عند الحاجة كما يأتي:

```
def is_valid(guess):
```

استقبال سلسلة، والتأكد أنها تتألف من (4) أرقام، وتعيد (True) أو (False).

```
def count_stars(guess, secret):
```

استقبال الرقم المضمّر، وتوقع اللاعب، وإعادة رقم يُمثّل عدد النجوم بناءً على ذلك.

```
def count_moons(guess, secret):
```

استقبال الرقم المضمّر، وتوقع اللاعب، وإعادة رقم يُمثّل عدد الأقمار بناءً على ذلك. والآن سأعمل - ضمن المجموعة - على كتابة هذه الدوال (يُمكن نسخ المقطع البرمجي الذي كتبناه سابقاً) واستخدامها، وأحاول الجمع بينها، بحيث يحاكي منطق البرنامج مخطط سير العمليات الذي اتفق عليه أفراد المجموعة.

أقسّم البرنامج إلى دوال، مثل:

```
. welcome(), about(), is_valid(), count_moons(), count_stars()
```

تلميحات:

1. لا بُدّ من وجود حلقة تكرار (لانهائية) تحيط بالعبة كلها، بحيث يعود اللاعب دائماً إلى رسالة الترحيب بعد الانتهاء من أيّ لعبة.
2. لا داعي الآن لكتابة الجزء المتعلّق باللعبة للاعبين اثنين؛ إذ يُمكن تأجيل ذلك إلى وقت لاحق.
3. يُمكن فصل الجزء المتعلّق باللعبة للاعب واحد في دالة مُنفصلة بهدف تنظيم البرنامج، وكذا الحال بالنسبة إلى الجزء الذي سيُكتب لاحقاً، ويتعلّق باللعبة للاعبين اثنين. أعمل - مع أفراد مجموعتي - على اختبار البرنامج بصورة شاملة، وأتحقّق من تصحيح الأخطاء، وأحرص على إضافة أيّ تحسينات وإبداعات شخصية تُثري المشروع (العبة).

الجزء الذي سيُكتب لاحقًا، ويتعلّق باللعب للاعبين اثنين.
العمل على اختبار البرنامج بصورة شاملة، وأتحدّق من تصحيح الأخطاء، وأحرص على إضافة أيّ تحسينات وإبداعات شخصية تُثري المشروع (اللعبة).

Project Code

```
import random

def welcome():
    print("Welcome to Stars and Moons")
    print("1. Instructions")
    print("2. Start Game (Single Player)")
    print("3. Exit")

def about():
    print("Instructions for Stars and Moons:")
    print("Guess the 4-digit number I am thinking of!")
    print("For each guess, you'll be told how many stars (correct digits in the correct position) and moons (correct digits in the wrong position) you have.")
    print("You have 10 attempts to guess the number correctly. Good luck!")
    input("Press Enter to go back")

def generate_guess():
    '''Generates a string containing 4 unique random digits'''
    digits = [x for x in range(0, 10)]
    random.shuffle(digits)
    num = "".join(str(x) for x in digits[:4])
    return num

def count_stars(guess, num):
    '''Count the number of stars (correct digits in correct positions)'''
    stars = 0
    for i in range(len(guess)):
        if guess[i] == num[i]:
            stars += 1
    return stars

def count_moons(guess, num):
    '''Count the number of moons (correct digits in incorrect positions)'''
    moons = 0
    for i in range(len(guess)):
        if guess[i] in num and guess[i] != num[i]:
            moons += 1
    return moons

def is_valid(guess):
    '''Check if the guess is valid: contains only digits and is 4 unique digits long'''
    if len(guess) != 4 or not guess.isdigit() or len(set(guess)) != 4:
        return False
    return True
```



```

def one_player():
    '''One player game'''

    print("\n\nI am hiding a number made of 4 unique digits.")
    num = generate_guess()

    # Allow 10 tries
    tries = 10
    while tries > 0:
        print("\nMake a guess (4 unique digits):")
        guess = input()

        # validate guess
        if not is_valid(guess):
            print("Invalid guess. Please enter 4 unique digits.")
            continue

        # Count stars and moons
        stars = count_stars(guess, num)
        moons = count_moons(guess, num)

        # Player won!
        if stars == 4:
            print("Congratulations! You guessed the number!")
            break

        # Print the Result
        print("Stars: " + str(stars) + ", Moons: " + str(moons))
        tries -= 1
        print("Trials left: " + str(tries))

    # Game over message
    if tries == 0:
        print("Game Over! You have guessed 10 times incorrectly")
        print("The hidden number was: " + num)
    else:
        print("Congrats! You have 4 stars (You guessed the number!)")

    input("Press Enter to go back")

#-----Program starts here-----
while True:
    welcome()

    choice = input("\n\nPick a choice (1, 2 or 3): ")
    while choice not in ['1', '2', '3']:
        print("Invalid choice!")
        choice = input("Pick a choice (1, 2 or 3): ")

    if choice == '1':
        about()
    elif choice == '2':
        one_player()
    elif choice == '3':
        print("Exiting game. Goodbye!")
        break

```

إجابات نموذجية لأسئلة أقيم تعلمي

المعرفة:

السؤال الثاني: ما الفرق بين المتغيرات المحلية ومتغيرات المدى العام؟

■ المتغير المحلي هو متغير معرف داخل دالة، ولا يمكن استخدام هذا المتغير إلا في هذه الدالة.

■ أما متغير المدى العام، فهو متغير معرف خارج جميع الدوال، ويمكن استخدامه داخل الدوال التي بعد جملة التعريف وخارجها.

السؤال الثالث: ما الفرق بين دوال مثل `sort` و `reverse` ودوال مثل `sorted` و `reversed`؟
الجواب:

129

أقيم تعلمي

المعرفة: أوظف في هذا الدرس ما تعلمته من معارف في الإجابة عن الأسئلة الآتية:
السؤال الأول: أوضّح المقصود بكل مما يأتي:

1. مُعاملات الدالة.

هي المدخلات التي تُرسل للدالة عند استدعائها.

2. استدعاء الدالة.

طلب تنفيذ الدالة داخل برنامج عن طريق ذكر اسمها، وإرسال قيم للمعاملات.

3. مدى المتغير.

هي الأماكن في البرنامج التي يمكننا فيها استخدام هذا المتغير.

4. استيراد الوحدة.

إخبار المفسر عن طريق جملة `import` بأن البرنامج سيستخدم دوال أو متغيرات في وحدة ما (وهو بمنزلة طلب سماح لاستخدام هذه الدوال والمتغيرات)
السؤال الثاني: ما الفرق بين المدى العام والمدى المحلي للمتغيرات؟

السؤال الثالث: ما الفرق بين الدالتين `sort` و `reverse` والدالتين `sorted` و `reversed`؟

المهارات: أوظف مهارة التفكير الناقد ومهارة حل المشكلات والمهارات البرمجية في الإجابة عن الأسئلة الآتية:

السؤال الأول: أكتب دالة تستقبل (3) أرقام، وترجع الرقم الوسيط، ثم أكتب برنامجاً بسيطاً يستدعي هذه الدالة.

■ الدوال `sort` و `reverse` ليس لها معاملات (مدخلات)، وتُستدعى باستخدام اسم المتغير الذي ستعمل عليه، متبوعاً بنقطة ثم اسم الدالة، ثم إن هذه الدوال لا ترجع نتيجة، وإنما تقوم تُخزن النتيجة مباشرة في المتغير الذي استخدم لاستدعاء الدالة.

■ الدوال `sorted` و `reversed` تُستدعى باسمها مباشرة مع إرسال القائمة التي نودّ معالجتها كمعامل (مدخل)، ولا تُعدّل هذه الدوال على القائمة التي تستقبلها كمدخل، وإنما ترجع نتيجة المعالجة كنسخة جديدة معدلة من القائمة.

السؤال الثاني: أكتب دالة تستقبل رقمين؛ أحدهما يمثل الطول، والآخر يمثل العرض، ثم أطلع مستطيلًا مرسومًا من علامة '#'. ثم أكتب برنامجًا بسيطًا يستدعي هذه الدالة.

السؤال الثالث: أكتب دالة تستقبل قائمتين، وتؤكد أن كل عنصر في القائمة الأولى يساوي العنصر المقابل له في القائمة الثانية (يجب أن ترجع الدالة (True) أو (False))، ثم أكتب برنامجًا بسيطًا يختبر هذه الدالة.

السؤال الرابع: أكتب برنامجًا يحتوي على دالة من تصميمي، ويطبع جدول الضرب لجميع الأرقام من (0) إلى (9)، بدءًا بطباعة جدول الصفر كاملاً، ثم طباعة جدول الرقم (1) كاملاً، وهكذا.

السؤال الخامس: أكتب برنامجًا يحتوي على دالة من تصميمي، ويطبع شكل المعين كما في الأمثلة الآتية:

```

*      *      *
***    ***    ***
***    *****
*      ***    *****
          *      *****
              *      *****
                  *

```

(تلميح: يُمكن استخدام دالتين؛ واحدة لطباعة النصف العلوي، وأخرى لطباعة النصف السفلي).

القيم والاتجاهات:

أطلق - بالتعاون مع زملائي - مبادرة (نادي البرمجة الصيفي)، وأوظف في ذلك الكُتيب الذي أنشأته سابقًا، وأسهم في تنمية مهارات طلبة مدرستي (في الصفوف 7-10) التي تتعلق بأساسيات لغة البرمجة بايثون (Python).

المهارات:

السؤال الأول:

```

# solution 1:
def median(x, y, z):
    if (x >= y and x <= z) or (x <= y and x >= z):
        return x;
    if (y >= x and y <= z) or (y <= x and y >= z):
        return y;
    return z;

# solution 2:
def median(x, y, z):
    a = [x, y, z]
    a.sort()
    return a[1]

print([1, 2, 3], "median =", median(1, 2, 3))
print([1, 2, 1], "median =", median(1, 2, 1))
print([3, 3, 1], "median =", median(3, 3, 1))

```

```
def square(height, width):  
    for i in range(height):  
        print(' #' * width)  
  
square(2, 4)  
print()  
square(5, 6)
```

```
def isEqual(a1, a2):
    if len(a1) != len(a2):
        return False

    for i in range(len(a1)):
        if a1[i] != a2[i]:
            return False

    return True

a1 = [1, 2, 3]
a2 = [1, 2, 3]

print(a1, a2, isEqual(a1, a2))

a2[1] = 3
print(a1, a2, isEqual(a1, a2))
```

```
def printTable(n):
    for i in range(10):
        print(n, ' x ', i, ' = ', n * i)

for i in range(10):
    print("Table for ", i, ":")
    print("-----")
    printTable(i)
    print("-----\n")
```

```

      *
    ***
    ***
      *

      *
    ***
  *****
  *****
    ***
      *

      *
    ***
  *****
  *****
  *****
  *****
    ***
      *

```

(تلميح: يمكنك استخدام دالتين؛ واحدة لطباعة النصف العلوي، وأخرى لطباعة النصف السفلي).

```
def upperHalf(n):
    spaces = n // 2
    for i in range(1, n+1, 2):
        print(' ' * spaces, '*' * i)
        spaces -= 1

def lowerHalf(n):
    spaces = 0
    for i in range(n, 0, -2):
        print(' ' * spaces, '*' * i)
        spaces += 1

upperHalf(5)
lowerHalf(5)
```

إجابة أسئلة الوحدة

أسئلة الوحدة:

السؤال الأول: أختار رمز الإجابة الصحيحة في كل مما يأتي:

1. إحدى الآتية تُعدُّ لغة برمجة عالية المستوى:

ج - لغة البرمجة بايثون (Python).

2. يعمل المُترجم (Compiler) على ترجمة:

أ - اللغة عالية المستوى إلى لغة الآلة دفعة واحدة.

3. ناتج `print(2**3)` في برمجة بايثون (Python) هو:

تذكير: أذكر الطلبة بوجود خلل في الطباعة في الكتاب، وتصحيح الجملة إلى `(print 3**2)`.

د - (8) الخيار الصحيح يجب استبدال الخيار به، د كتاب الطالب).

4. البيانات التي تُستعمل لتخزين النصوص في برمجة بايثون (Python)

هي من نوع:

ج - `string()`

5. إحدى الجمل الآتية تتسبب في حدوث خطأ في برمجة بايثون (Python):

د - `print("Hello" + 2)`.

6. يُمكن إنشاء قائمة في برمجة بايثون (Python) باستخدام:

أ- list = []

7. تعمل الدالة len() في برمجة بايثون (Python) على:

ج- إيجاد عدد الأحرف في سلسلة نصية، وإيجاد عدد العناصر في قائمة

ما .

8. الكلمة المفتاحية التي تُستعمل لبدء الدالة في برمجة بايثون (Python)

هي :

ب- def.

9. ناتج print(type(10)) في برمجة بايثون (Python) هو:

ب- <class 'int'>.

10. إحدى الآلية تمثل الطريقة الصحيحة لبدء حلقة (for) في برمجة بايثون

:(Python)

ب- for i in range(10)

11- الوظيفة التي تؤديها جملة التحكم (break) في حلقة (for) هي:

ب- إنهاء الحلقة بصورة كاملة.

12- ناتج print("Hello" + "World") في برمجة بايثون (Python) هو:

أ- HelloWorld.

13- ناتج 2/5 في برمجة بايثون (Python) هو:

أ- (2). (أذكر الطلبة بوجود خطأ في الكتاب).

السؤال الثاني: أُمِّز الجمل الصحيحة من الجمل غير الصحيحة في ما يأتي:

1. في لغة البرمجة بايثون (Python)، تُستعمل عبارة (if) لإنشاء حلقة. (غير

صحيحة)

2. يُمكن للقوائم في لغة البرمجة بايثون (Python) تخزين عناصر تحوي أنواعاً

مختلفة من البيانات. (صحيحة)

3. العامل = في لغة البرمجة بايثون (Python) يُستخدم في المقارنة بين قيمتين.

(صحيحة)

4. يُستخدم العامل += في إضافة قيمة إلى أحد المتغيرات، وإسناد النتيجة إلى هذا

المتغير. (صحيحة)

5. تُستخدم الكلمة المفتاحية (file) في لغة البرمجة بايثون (Python) للتعامل مع شروط مُتعدّدة. (صحيحة)

6. يجب دائماً أن تعيد الدوال قيمة في لغة البرمجة بايثون (Python). (غير صحيحة)

7. تُستعمل علامات الاقتباس الفردية والمزدوجة لتعريف سلسلة نصية في لغة البرمجة بايثون (Python). (صحيحة)

8. تتطلّب حلقة (for) في لغة البرمجة بايثون (Python) وجود مُتغيّر فهرسة صريح. (غير صحيحة)

9. في لغة البرمجة بايثون (Python)، يكون ناتج كل من $2 * 3$ و $2 ** 3$ مُتماثلاً. (غير صحيحة)

10. تبدأ التعليقات في لغة البرمجة بايثون (Python) بالرمز //. (غير صحيحة)

السؤال الثالث: أملأ الفراغ بما هو مناسب في الجمل الآتية:

1. +

2. list

3. def

4. print

5. while أو for

6. الشرط

7. int()

8. syntax error أو خطأ نحوي

9. import

10. +

السؤال الرابع: أكتب برنامجاً بلغة البرمجة بايثون (Python)، يأخذ رقماً بوصفه مدخلاً، ويطبع مُربّعه

```
# برنامج لحساب مربع عدد مدخل
number = int(input("أدخل رقماً: "))
print("مربع الرقم هو", number ** 2)
```


السؤال الخامس: أكتب برنامجًا بلغة البرمجة بايثون (Python)، يستخدم حلقة (for) في طباعة الأرقام من

```
# برنامج لطباعة الأرقام من 1 إلى 10
for i in range(1, 11):
    print(i)
```

(1) إلى (10)

السؤال السادس: أكتب برنامجًا بلغة البرمجة بايثون (Python)، يُستخدم في حساب مجموع كل الأعداد الزوجية التي تقع بين العدد (1) والعدد (50).

```
# برنامج لحساب مجموع الأعداد الزوجية من 1 إلى 50
sum_even = 0
for i in range(1, 51):
    if i % 2 == 0:
        sum_even += i
print("مجموع الأعداد الزوجية من 1 إلى 50 هو", sum_even)
```

السؤال السابع: أكتب برنامجًا بلغة البرمجة بايثون (Python)، يُدخِل قائمة من الأرقام، ويطبع أكبر رقم فيها.

```
# برنامج لطباعة أكبر رقم في قائمة
numbers = list(map(int, input("أدخل قائمة من الأرقام مفصولة بمسافات: ").split()))
print("أكبر رقم في القائمة هو", max(numbers))
```

تذكير: يتضمن السؤال فكرة إثرائية، وهي استخدام الدوال (map, split). أوضحها للطلبة، وأحلّ السؤال معهم في الغرفة الصفية.

إثراء لحل السؤال:

تُستخدم دالة map لتطبيق دالة معينة على كل عنصر في تكرار (مثل قائمة أو مجموعة)، وتقوم بإرجاع كائن من نوع map يحتوي على النتائج المحسوبة. يمكنك تحويل هذا الكائن إلى قائمة باستخدام الدالة list.

الشكل الأساسي:

map(function, iterable)

■ function: الدالة التي تريد تطبيقها على كل عنصر في التكرار.

■ iterable: التكرار الذي يحتوي على العناصر التي سيُطبق الدالة عليها.

مثال:

دالة تضاعف الرقم

def double(x):

return x * 2

numbers = [1, 2, 3, 4]

doubled_numbers = map(double, numbers)

تحويل النتيجة إلى قائمة.

print(list(doubled_numbers)) # [2, 4, 6, 8]

تُستخدم الدالة split لتقسيم سلسلة نصية إلى قائمة من العناصر بناءً على فاصل محدد. على نحو افتراضي، تُستخدم المسافة كفاصل؛ ولكن يمكن تحديد فاصل آخر إذا لزم الأمر.

الشكل الأساسي:

string.split(separator, maxsplit)

■ separator (اختياري): الفاصل الذي يستخدم لتقسيم السلسلة. إذا لم يُحدد، سيُستخدم أي عدد من المسافات كفاصل.

■ maxsplit (اختياري): الحد الأقصى لعدد التقسيمات التي يجب إجراؤها. إذا لم يُحدد، ستُقسَّم السلسلة إلى أكبر عدد ممكن من الأجزاء.

مثال:

sentence = "Hello world, this is Python"

words = sentence.split()

print(words) # ['Hello', 'world,', 'this', 'is', 'Python']

باستخدام فاصل محدد.

data = "one,two,three,four"

items = data.split(',')

print(items) # ['one', 'two', 'three', 'four']

السؤال الثامن: في ما يأتي مجموعة من المقاطع البرمجية المكتوبة بلغة البرمجة بايثون (Python). اتبّع الأوامر في هذه المقاطع، واكتشف الأخطاء الموجودة في البرنامج من دون تنفيذه، ثمّ اقترح طرائق لتصحيحها:

(a) الخطأ: `x = input("Enter a number: ")` يجب تحويل `x` إلى عدد صحيح باستخدام `int()`.
التصحيح:

```
x = int(input("Enter a number: "))
if x > 10:
    print("x is greater than 10")
else:
    print("x is less than or equal to 10")
```

(b) الخطأ: `1st_number` و `2nd_number` غير معرفة.
التصحيح:

```
first_number = int(input("Enter first number: "))
second_number = int(input("Enter second number: "))
sum = first_number + second_number
print("The sum is:", sum)
```

(c) الخطأ: استخدام حلقة `for` غير صحيحة.
التصحيح:

```
i = 1
while i <= 10:
    print(i)
    i += 1
```

تذكير: أبين للطلبة الفاصل بين المقاطع البرمجية، وأنها ليست برنامجاً واحداً

السؤال التاسع:

أعدّل المقطع البرمجي الآتي؛ لكي يتمكن البرنامج من قبول ما يدخله المُستخدم من مدخلات، ثمّ يطبع عبارة تُبيّن نوع العدد (فردى أو زوجى):

التصحيح:

```
# برنامج لقبول مدخل من المستخدم وتحديد إذا كان العدد زوجى أو فردي
number = int(input("Enter a number: "))
if number % 2 == 0:
    print("The number is even")
else:
    print("The number is odd")
```

السؤال العاشر:

اكتب برنامجاً بلغة البرمجة بايثون (Python)، تُستخدم فيه الدوال (functions) لتنفيذ مجموعة من العمليات الحسابية (الجمع، والطرح، والضرب، والقسمة) بناءً على مدخلات المُستخدم.

```
# برنامج لتنفيذ العمليات الحسابية
def add(x, y):
    return x + y
def subtract(x, y):
    return x - y
def multiply(x, y):
    return x * y
def divide(x, y):
    if y == 0:
        return "Cannot divide by zero!"
    return x / y
while True:
    # المدخلات
    num1 = float(input("Enter first number: "))
    num2 = float(input("Enter second number: "))
    # اختيار العملية
    while True:
        print("Select operation:")
        print("1.Add")
        print("2.Subtract")
        print("3.Multiply")
        print("4.Divide")
        choice = input("Enter choice (1/2/3/4): ")
        if choice == '1':
            print("Result:", add(num1, num2))
            break
        elif choice == '2':
            print("Result:", subtract(num1, num2))
            break
        elif choice == '3':
            print("Result:", multiply(num1, num2))
            break
        elif choice == '4':
            print("Result:", divide(num1, num2))
            break
        else:
            print("Invalid input, please try again.")
    # سؤال المستخدم إذا كان يرغب في إتمام عملية أخرى
    another_calculation = input("Do you want to perform another
calculation? (yes/no): ").lower()
    if another_calculation != 'yes':
        print("Goodbye!")
        break
```

السؤال الحادي عشر:

أكتب برنامجاً بلغة البرمجة بايثون (Python) لاختبار قوّة كلمات المرور بناءً على مجموعة من المعايير، مثل: الطول، ووجود الحروف الكبيرة والحروف الصغيرة، والأرقام، والرموز الخاصة، علماً بأن البرنامج سيطلب من المُستخدم إدخال كلمة المرور، ثمّ يتحقّق من درجة قوّتها وتعقيدها، ثمّ يعرض نتيجة الاختبار.

```
# برنامج لاختبار قوة كلمات المرور
import re

def check_password_strength(password):
    if len(password) < 8:
        return "Weak: Password must be at least 8 characters long."
    if not re.search("[a-z]", password):
        return "Weak: Password must contain at least one lowercase letter."
    if not re.search("[A-Z]", password):
        return "Weak: Password must contain at least one uppercase letter."
    if not re.search("[0-9-]", password):
        return "Weak: Password must contain at least one digit."
    if not re.search("[@#%$^&*!]", password):
        return "Weak: Password must contain at least one special character."
    return "Strong: Your password is strong."

password = input("Enter a password: ")
print(check_password_strength(password))
```

تذكير: أبين للطلبة ضرورة الاستكشاف لحل أسئلة الوحدة بالاعتماد على المعلومات والمهارات التي تعلّمها في الوحدة، بالإضافة إلى البحث والاستكشاف والتواصل مع الزملاء، وأحفز الطلبة لإيجاد أفكار مبتكرة للحلّ؛ بإطلاق مسابقة "مبرمج الصف" وأقدم هدايا تشجيعية للفائزين.

أدوات التقويم الخاصة بالوحدة.

تقويم منتج التعلم النهائي.

المنتج النهائي: لعبة تخمين الأرقام مصممة باستخدام لغة بايثون (Python).

سلم تقدير محكي المرجع وصفي:

المعيار	الوصف الممتاز	الوصف الجيد	الوصف المقبول	الوصف المقبول الوصف غير المقبول
وضوح فكرة اللعبة	فكرة اللعبة واضحة على نحو ممتاز، وتُعبّر بوضوح عن الهدف المطلوب من اللاعب تحقيقه.	فكرة اللعبة واضحة، لكن قد تحتاج إلى بعض التوضيحات البسيطة.	فكرة اللعبة موجودة لكنها غامضة أو تحتاج إلى تحسين كبير.	فكرة اللعبة غير واضحة أو غير موجودة.
تصميم المكونات الأساسية للعبة	صُممت جميع مكونات اللعبة الأساسية (الشاشة الرئيسية، التعليمات، بدء اللعبة، انتهاء اللعبة) على نحو جيد ومتكامل.	صُممت معظم المكونات الأساسية للعبة، لكنها قد تحتاج إلى بعض التحسينات.	صُممت بعض المكونات الأساسية، لكن هناك نقص واضح أو أخطاء.	المكونات الأساسية للعبة غير مكتملة أو غير مصممة على نحو صحيح.
الاشتمال على البنود المطلوبة	يتضمن جميع البنود المطلوبة، وتُنفذت جميع المهمات على نحو صحيح.	يتضمن معظم البنود المطلوبة، وتُنفذ معظمها على نحو صحيح.	يتضمن بعض البنود المطلوبة وتُنفذ معظمها على نحو صحيح.	يفتقر إلى العديد من البنود المطلوبة.
جودة الكود البرمجي وصحته	الكود صحيح بالكامل، وخالٍ من الأخطاء اللغوية والمنطقية.	الكود يحتوي على بعض الأخطاء البسيطة التي لا تؤثر في التشغيل الكلي.	الكود يحتوي على بعض الأخطاء اللغوية والمنطقية التي تحتاج لتصحيح.	الكود يحتوي على العديد من الأخطاء التي تعيق التشغيل.
تنفيذ منطق اللعبة	تُنفذ منطق اللعبة (توليد الرقم، حساب النجوم والأقمار، التحقق من صحة التخمين) على نحو دقيق ومن دون أخطاء.	تُنفذ منطق اللعبة على نحو جيد، مع وجود بعض الأخطاء الطفيفة التي لا تؤثر على نحو كبير في اللعب.	تُنفذ منطق اللعبة، لكنه يحتوي على أخطاء تؤثر في تجربة اللعب.	منطق اللعبة غير مكتمل أو غير صحيح؛ مما يجعل اللعبة غير قابلة للعب.
استخدام الدوال البرمجية	استُخدمت الدوال البرمجية على نحو فعال ومنظم؛ مما جعل الكود سهل الفهم والصيانة.	استُخدمت الدوال البرمجية على نحو مناسب، لكن هناك بعض الدوال التي يمكن تحسين تنظيمها أو استخدامها.	استُخدمت الدوال البرمجية، لكن الكود غير منظم أو يحتوي على دوال غير ضرورية.	لم تُستخدم الدوال البرمجية على نحو صحيح، أو لم تُستخدم على الإطلاق.

مراجعة اللعبة واختبارها	روجعت اللعبة واختبرت على نحو شامل، وصُححت جميع الأخطاء المكتشفة، وحُسّنت اللعبة بناءً على المراجعة.	روجعت اللعبة واختبرت، مع إصلاح معظم الأخطاء، وأجريت تحسينات محدودة.	روجعت اللعبة، لكن الاختبار كان محدودًا، واكتُشفت بعض الأخطاء التي لم تُصلح.	لم تُراجع اللعبة أو تُختبر على نحو كافٍ، وما تزال تحتوي على أخطاء تؤثر في اللعب.
عرض اللعبة وتقديمها	عُرِضت اللعبة وقُدِّمت على نحو احترافي، مع توضيح جميع مكوناتها وأجزائها للمستمعين.	عُرِضت اللعبة على نحو جيد، لكن هناك بعض النقاط التي كانت بحاجة إلى توضيح أفضل.	عُرِضت اللعبة، لكن العرض كان أو غير مكتمل، ولم توضح مكونات اللعبة على نحو كافٍ.	العرض كان ضعيفًا أو غير مكتمل، ولم توضح مكونات اللعبة على نحو كافٍ.
إرفاق مخطط سهمي توضيحي	المخطط السهمي شامل، ويفصل جميع جزئيات اللعبة بوضوح.	المخطط السهمي جيد، ويفصل معظم جزئيات اللعبة.	المخطط السهمي مقبول، ولكنه يفتقر لبعض التفاصيل.	المخطط السهمي غير موجود أو غير كافٍ.
العمل الجماعي وتقاسم الأدوار	وُزعت الأدوار على نحو ممتاز، وكان العمل الجماعي فعالاً للغاية.	وُزعت الأدوار على نحو جيد والعمل الجماعي كان فعالاً على نحو كبير.	وُزعت الأدوار على نحو مقبول، ولكن بعض الأعضاء لم يشاركوا بفعالية.	لم تُوزع الأدوار على نحو جيد، وكان العمل الجماعي ضعيفاً.
نشر اللعبة وتوثيق المصادر	نُشرت اللعبة بنجاح مع توثيق شامل ودقيق لجميع المصادر والمراجع المستخدمة.	نُشرت اللعبة مع توثيق جيد لمعظم المصادر والمراجع.	نُشرت اللعبة مع توثيق مقبول، ولكن ينقصه بعض المصادر المهمة.	لم تُنشر اللعبة أو لم تُوثق المصادر على نحو كافٍ؛ مما يؤثر في مصداقية المشروع.

استخدام الجدول

- وضوح المعايير: أتأكد أن الطلبة يفهمون معايير التقييم، وكيفية تحقيق أعلى الدرجات.
- التغذية الراجعة التفصيلية: أقدم تغذية راجعة مفصلة لكل معيار بناءً على الوصفات المقدمة.
- تحفيز التحسين: أستخدم التقييم لتحفيز الطلبة على تحسين أدائهم في المستقبل عن طريق تحديد نقاط القوة، ومجالات التحسين.
- تقديم التغذية الراجعة البناءة: أقدم تغذية راجعة بناءة ومشجعة للطلبة، وأركز على تقديم نصائح عملية، يمكنهم تنفيذها لتحسين عملهم وتفاعلهم.

ملاحق خاصة بالوحدة.

ملحق (1): قياس المهارات البرمجية.

مهارة التتبع.

تعريف:

هي القدرة على تتبع الجمل البرمجية وتنفيذها "ذهنيًا" جملة بجملة؛ لمعرفة أثر تشغيلها.

لتتبع جمل برمجية يحتاج المتتبع:

- معرفة معنى كل جملة من الجمل البرمجية على نحوٍ منفصل.
- القدرة على متابعة حالة المتغيرات في البرنامج من خطوة لخطوة.

لا يحتاج المتتبع أيًا من الآتي:

- تكوين فهم متكامل للهدف الذي تؤديه الجمل مع بعضها بعضًا بشكل عام.
 - اكتشاف الأخطاء في البرنامج خارج حدود مدخلات البرنامج.
 - القدرة على كتابة برامج مشابهة.
- لذلك؛ فإن المتتبع يشبه الحاسوب؛ إذ يستطيع معرفة نتيجة برنامج عن طريق تنفيذه من دون فهم الهدف منه، ومعرفة نتيجته على مدخلات أخرى، والقدرة على توليد برامج أخرى مشابهة.
- مثال:

```
def mystery1(a):  
    x = 0  
    for e in a:  
        if e > x:  
            x = e  
    print(x)  
  
a = [99, 1, 2, 0]  
mystery1(a)
```

يتطلب تتبع هذا البرنامج:

- معرفة بعض المفاهيم المتعلقة بالقوائم والدّوالّ وحلقات التكرار والجمل

الشرطية.

- القدرة على متابعة كيفية تغير قيمة المتغير x بعد كل دورة من دورات حلقة التكرار بهذا يمكن لمتتبع البرنامج معرفة أن الناتج هو الرقم (99).

ولا يتطلب تتبع هذا البرنامج بالضرورة:

- استنتاج أن الهدف من الدالة (بشكل عام) هو طباعة أكبر رقم في القائمة.
- الانتباه إلى أن الدالة لا تطبع أكبر رقم في حال كانت جميع محتويات القائمة أرقامًا سالبة.
- القدرة على كتابة دالة صحيحة، تطبع الرقم الأكبر أو الرقم الأصغر في قائمة من دون رؤية هذه الدالة ابتداءً.

قياس مهارة التتبع:

لقياس مهارة التتبع يُعطى الطالب برنامجًا أو جزءًا من برنامج (مع مدخلات محددة)، ويُطلب منه القيام بأحد (أو بعض) الآتي:

- تحديد قيمة متغير بعد انتهاء تنفيذ الجمل البرمجية.
- تحديد القيمة الراجعة من دالة عند استدعائها.
- تحديد ناتج الطباعة على الشاشة عند تشغيل البرنامج.

وتؤثر العوامل الآتية في درجة صعوبة السؤال:

- صعوبة المفاهيم المستخدمة في البرنامج (مثلًا تتبع جملة شرطية بسيطة مقارنة بتتبع دالة تستدعي نفسها).
- عدد المفاهيم المستخدمة (مثلًا تتبع برنامج يحوي جملاً شرطية مقارنة ببرنامج يحوي جملاً شرطية وحلقات تكرار وقوائم).
- درجة تعقيد التراكيب البرمجية المستخدمة (مثلًا حلقات تكرار متتابعة مقارنة بحلقات تكرار مركبة).
- عدد المتغيرات التي يحتاج المتتبع متابعتها.
- سهولة استنتاج الهدف من البرنامج (مثلًا استخدام اسم `max_in_list` في البرنامج السابق، سيجعل الهدف من الدالة واضحًا، ويسمح بمعرفة النتيجة من دون الحاجة للتتبع).
- درجة الشبه بين البرنامج الذي يراد تتبعه، وبرامج أخرى كُتبت، أو تمَّ تتبعها مسبقًا.

مثال:

عدد جميع أسباب صعوبة تتبع البرنامج الآتي:

```
# 1
x = 0
y = 0
for i in range(5):
    if i % 2 == 0:
        x += 1
    y += i

print(x, y)

# 2
s = 0
for i in range(5):
    s += i
print(s, s*2)

# 3
s = 3
for i in range(1, 10):
    s += i
print(s)
```

العلامات الجزئية:

عادة ما يُصحَّح سؤال التتبع على نحو ثنائي (علامة كاملة أو صفر)، ولكن يمكن إعطاء علامة جزئية بإحدى الطرائق الآتية (على سبيل المثال):

- إذا كان هناك أكثر من استدعاء للدالة، فيمكن تعيين علامة لمعرفة نتيجة كل استدعاء.
- مثال: يمكن استدعاء دالة `mystery1` (من الصفحة الأولى) باستخدام قائمة فيها أرقام موجبة فقط، ثم استدعاؤها باستخدام قائمة فيها أرقام سالبة فقط، وتعيين علامة جزئية لمعرفة نتيجة كل استدعاء.
- إذا كان البرنامج يحوي متغيرات متعددة، وكان المطلوب معرفة قيمة كل متغير بعد الانتهاء من تنفيذ البرنامج، فيمكن تعيين جزء من العلامة لكل واحد من المتغيرات.
- إذا كان سيُطبع على الشاشة أكثر من قيمة، فيمكن تعيين علامة جزئية لمعرفة نتيجة كل عملية طباعة.

```
# 3
s = 3
for i in range(1, 10):
    s += i
    print(s)          # 2
s = 0
for i in range(5):
    s += i
print(s, s*2)        # 1
x = 0
y = 0
for i in range(5):
    if i % 2 == 0:
        x += 1
    y += i

print(x, y)
```

- **يطبع البرنامج #1 قيمتين: 3 و 10، ولكن يمكن للطالب أن يطبع إحدى القيمتين على نحو صحيح من دون الأخرى؛ لأن كلا منهما له تعلق بمفهوم برمجي مختلف.**
- **يطبع البرنامج #2 قيمتين: 10, 20، ولكن يفترض أن يُصحح من دون علامات جزئية (إما صفر أو علامة كاملة)؛ لأن القيمة الثانية تعتمد على الأولى على نحو كامل، ولا تتطلب فهمًا مختلفًا (إلا إذا اعتبرنا أن الضرب مهارة نودّ قياسها).**
- **يطبع البرنامج #3 خمس قيم، ولكن من الصعب تخيّل أن يخطئ الطالب في بعضها من دون الآخر؛ لأن كل قيمة مطبوعة تعتمد على القيمة التي قبلها؛ لذلك إعطاء علامة جزئية متساوية لكل واحدة من القيم، لا يعبر على نحو صحيح عن تدرج في فهم الطالب، ولكن يمكن إعطاء علامة جزئية لكلّ من الحالات الآتية (على سبيل المثال):**
 - إذا طبع الطالب قيمة واحدة فقط مساوية للقيمة النهائية لـ `s` على نحو صحيح، فهذا يعني أن الطالب استطاع تتبع عملية الجمع في حلقة التكرار؛ ولكنه أخطأ بالطباعة مرة واحدة في النهاية بدلاً من الطباعة في كل دورة.
 - إذا زاد الطالب قيمة طباعة واحدة، فهذا يدل على قدرته على التتبع، والخطأ في معرفة آخر رقم في الدالة `range`.

مهارة القراءة.

تعريف:

هي القدرة على استنتاج الهدف العام لمجموعة جمل برمجية.

لقراءة مجموعة جمل برمجية يحتاج القارئ:

- معرفة معنى كل جملة من الجمل البرمجية على نحو منفصل.
- فهم الوظيفة التي تؤديها مجموعة الجمل بعضها مع بعض.

لا تتطلب مهارة القراءة بالضرورة:

- القدرة على اكتشاف جميع الأخطاء الممكنة في البرنامج.
- القدرة على كتابة برامج مشابهة.

قد تتطلب قراءة البرنامج توليد مدخلات، وتتبع نتيجة الجمل البرمجية على هذه المدخلات قبل استنتاج الوظيفة العامة للبرنامج، ولكن هذا ليس شرطاً.
مثال:

```
def mystery2(n):
    result = -1
    for i in range(n):
        if i*i == n:
            result = i
            break
    print(result)
```

تقوم هذه الدالة بطباعة الجذر التربيعي للرقم n أو الرقم -1 إن لم يكن هناك جذر تربيعي صحيح.

- قد يتمكن القارئ من استنتاج ذلك من دون الحاجة لتتبع الدالة باستخدام قيم معينة لـ n ، وقد يحتاج القارئ للتتبع باستخدام بعض القيم لـ n قبل الوصول للنتيجة.
- قد يستنتج القارئ الهدف من الدالة من دون الانتباه إلى أن نتيجة الدالة خطأ في حال كانت قيمة n هي الرقم 1.
- الاستنتاج الصحيح للهدف من الدالة، لا يعني بالضرورة أن القارئ كان سيستطيع ابتداءً كتابة دالة تحقق الهدف نفسه.

قياس مهارة القراءة:

أشهر طريقة لقياس مهارة القراءة أن يُطلب إلى الطالب التعبير عن "الهدف" من البرنامج أو الدالة على نحو عام ومجمل من دون وصف كل جملة برمجية على نحو منفصل.

توضح الأمثلة الآتية طرائق صحيحة للإجابة، وطرائق مرفوضة:

شكل صحيح للإجابة

```
def mystery3(a, b, c):
    m1 = min(a, b, c)
    m2 = max(a, b, c)
    m3 = a + b + c - m1 - m2
    print(m1, m3, m2)
```

وتطبّع الدالة المعاملات
مرتبة من الأصغر إلى
لأكبر.

شكل مرفوض للإجابة

وأصغر 1m تحفظ الدالة أكبر معامل في
ثم تُخزّن ناتج جمع ، m2 معامل في
m3 منها في m2 و m1 المعاملات، وتطرح
، m2 ثم m3 ثم m1 ثم تطبّع ،

شكل مرفوض للإجابة

تعيّن الدالة قيمة False للمتغير f ثم تقرأ n قيم، وفي حال كانت القيمة المدخلة زوجية، تُخزّن في المتغير b وتعيّن القيمة True للمتغير f، وبعد الانتهاء من حلقة التكرار، يُتأكّد من قيمة f وطباعة قيمة b إن كانت قيمة f هي True.

شكل صحيح للإجابة

وتطبّع الدالة المعاملات مرتبة من الأصغر إلى الأكبر.

```
def mystery4(n):  
    f = False  
    for i in range(n):  
        a = int(input())  
        if a % 2 == 0:  
            b = a  
            f = True  
  
    if f == True:  
        print(b)
```

شكل مرفوض للإجابة

ترجع الدالة صفرًا إذا كانت n صفرًا، عدا ذلك تستدعي الدالة ذاتها باستخدام قيمة n بعد قسمة n على 10 وتضيف 1 للنتيجة في حال كان الرقم n رقمًا زوجيًا قبل قسمته على 10،

شكل صحيح للإجابة

تقوم الدالة بإرجاع عدد المنازل في الرقم n التي تحوي أعدادًا زوجية

```
def mystery5(n):  
    if n == 0:  
        return 0  
    if n % 2 == 0:  
        return 1 + mystery(n // 10)  
    else:  
        return mystery(n // 10)
```

شكل مرفوض للإجابة

تقوم الدالة أولاً بتعيين القيمة صفر للمتغير i وطول القائمة -1 للمتغير j ثم تقوم بالدوران إلى أن تصبح قيمة i مساوية لقيمة j، وفي كل دورة تقوم بزيادة قيمة i ما دام أن العنصر i مساوٍ لصفر، ولم تتعد قيمة i قيمة j، وتقوم بتنقيص قيمة j ما دامت لم تصل لـ i، وما دام العنصر في الموقع i لا يساوي صفرًا، ثم تقوم بعد ذلك بتبديل العنصر i بالعنصر j.

شكل صحيح للإجابة

تقوم الدالة بنقل جميع الأصفار إلى بداية القائمة.

```
def mystery6(a):  
    i = 0  
    j = len(a)-1  
  
    while i < j:  
        while i < j and a[i] == 0:  
            i += 1  
        while j > i and a[j] != 0:  
            j -= 1  
  
        a[i], a[j] = a[j], a[i]
```

ألاحظ أن الإجابات المرفوضة، تدلّ على قدرة الطالب على فهم كل جملة برمجية على نحوٍ منفصل، ولكن لا تدلّ على قدرة الطالب على استنتاج فهم متكامل للهدف من كل هذه الجمل بعضها مع بعض. لذلك؛ فإن بناء سلم التقدير يجب أن يقدر مدى فهم الطالب "للهدف" من البرنامج، وليس فهم الطالب لكل جملة برمجية على نحوٍ منفصل، أو قدرة الطالب على تتبع الدالّة باستخدام قيم محددة.

مهارة اكتشاف الأخطاء وتصحيحها:

هناك مستويات عدة لمهارة اكتشاف الأخطاء وتصحيحها:

■ مخالفة قواعد اللغة:

هذا أدنى مستوى؛ إذ يتطلب اكتشاف الخطأ وتصحيحه فقط معرفة قواعد اللغة كما في المثال الآتي:

```
age = int(input())
print("You are " + age + " years old")
```

إذ يكفي لاكتشاف الخطأ معرفة أن عملية الإلصاق (concatenation) باستخدام عامل الجمع، لا تتم بين رقم وسلسلة، وإنما بين سلسلة وسلسلة أخرى.

ملاحظات

■ الخطأ البرمجي المنطقي:

هذا النوع من الأخطاء يتطلب اكتشافه تتبع البرنامج، ولكن الخطأ يظهر دائماً عند التتبع بغض النظر عن المدخلات، فمثلاً:

```
s = 0
for i in range(1, 5):
    print("\nStudent " + str(i) + ": ")
    for j in range(1, 4):
        s += int(input("Enter grade for assessment " + str(j) + ": "))
    print("Sum for student " + str(i) + ": " + str(s))
```

عند تتبع البرنامج، يتضح أن قيمة المجموع تتراكم؛ إذ يتم تصفيرها لكل طالب (الحالة الوحيدة التي لن يظهر فيها الخطأ عند التتبع هي إن كانت العلامات صفراً).

■ الخطأ البرمجي في الحالات الخاصة:

هذا النوع من الأخطاء يتطلب اكتشافه تحديد الحالات الممكنة التي يمكن للبرنامج أن يمرّ بها جميعها، وتوليد مدخلات لهذه الحالات، وتتبع البرنامج في كل حالة لمعرفة أيّ منها يؤدي إلى مشكلة. ومثال هذا دوال mystery1 و mystery2 التي رأيناها مسبقاً.

قياس مهارة اكتشاف الأخطاء وتصحيحها:

يختلف قياس هذه المهارة على ورق عن قياسها على الحاسوب، فعلى الحاسوب يمكن للطالب تشغيل البرنامج بدلاً من تتبعه، ويمكنه أحياناً استنتاج المشكلة من صيغة رسالة الخطأ التي يظهرها مفسر لغة البايثون، ثم إن الطالب يمكنه التعديل على البرنامج لاختبار بعض الفرضيات التي لديه عن سبب المشكلة؛ لمعرفة صحة الفرضية، أو إضافة جمل طباعة لتتبع قيم متغير ما، أو استخدام أدوات تكون موجودة أحياناً في بيئات البرمجة المتكاملة.

مهارة تصميم الخوارزميات لحل المسائل البرمجية:

هي القدرة على تصميم خطوات واضحة لحل مسألة برمجية على نحو صحيح؛ إذ يمكن بعد ذلك ترجمة هذه الخطوات لإحدى لغات البرمجة.

مثال:

كيف يمكننا معرفة عدد العناصر غير المكررة في قائمة ما؟

الجواب 1:

■ أنشئ عداداً بقيمة 1.

■ لكل عنصر x في القائمة: أقرن العنصر x مع جميع العناصر التي بعده في القائمة، وأضيف 1 للعداد، إن لم يكن العنصر مساوياً لأيّ من العناصر التي بعده.

الجواب 2:

- أرتب القائمة تصاعدياً (هذا سيجمع العناصر المتساوية بجانب بعضها بعضاً).
 - أنشئ عدداً بقيمة 1.
 - أقارن كل عنصر في القائمة مع العنصر الذي بعده، وأضيف 1 للعدد في حال كان العنصر لا يساوي العنصر الذي بعده مباشرة.
- مهارة تصميم الخوارزمية هي القدرة على التفكير، ابتداءً بخطوات لحل المسألة كما في المثال السابق (من دون تدريب مسبق على المسألة نفسها).

مهارة الكتابة:

تعريف:

- هي القدرة على كتابة برنامج بلغة برمجة ما لتأدية وظيفة محددة، وهي مهارة كثيراً ما تكون مركبة من مهارات أخرى:
- قد يحتاج المبرمج ابتداءً لتصميم خوارزمية لأداء الوظيفة.
 - يحتاج لكتابة كود بلغة برمجة ما لتحقيق الخوارزمية التي صُممت.
 - يحتاج للتأكد من صحة برنامجه عن طريق تخيل جميع الحالات التي يمكن أن يمرَّ بها، وتشغيل البرنامج للتأكد من صحته في جميع هذه الحالات.
 - قد يحتاج المبرمج لتتبع البرنامج الذي كتبه من أجل اكتشاف الأخطاء التي ارتكبها في أثناء الكتابة، ثم تصحيحها.

قياس مهارة الكتابة:

درجة الصعوبة:

تؤثر العوامل الآتية في صعوبة البرنامج المطلوب كتابته بناءً على:

- طولهُ.
- المفاهيم المطلوبة لكتابة الكود.
- درجة تعقيد الخوارزمية التي ستُكتب.
- حاجة المبرمج لتصميم خوارزمية جديدة مقارنة ببرمجة خوارزمية موجودة أصلاً فقط.
- درجة التشابه بينه وبين برامج أخرى رآها مسبقاً، أو كتبها مسبقاً.

فمثلاً:

سؤال 1: اكتب دالة تستقبل قائمة، وتقارن كل عنصر بالعناصر التي تأتي بعده؛ لمعرفة عدد العناصر غير المكررة في القائمة.

سؤال 2: اكتب دالة تستقبل قائمة، وتطبع عدد العناصر غير المكررة فيها.

سؤال 3: اكتب دالة تستقبل قائمة، وتستخدم الاستدعاء الذاتي من أجل معرفة عدد العناصر غير المكررة في القائمة. السؤال الأول هو الأسهل؛ لأنه لا يتطلب تصميمًا لخوارزمية، فالخوارزمية مضمنة في السؤال، أما في السؤال الثاني، فيترك الباب مفتوحًا للطالب للتفكير بالخوارزمية المناسبة وتصميمها، ثم كتابة الدالة، أما السؤال الثالث فهو الأصعب؛ لأنه أجبر الطالب على استخدام مفهوم صعب نسبيًا (الاستدعاء الذاتي)، ولم يعط أي تلميحات عن شكل الخوارزمية التي يجب أن يصممها.

لاحظ بأن جميع هذه الأسئلة ستكون سهلة إن شاهد الطالب مثيلاً لها مسبقاً.

استخدام الحاسوب:

قياسُ مهارة الكتابة على ورق، قد يختلف عن قياسها على الحاسوب من بعض النواحي، فمثلاً:

- كم الوقت المتوافر للامتحان؟
- عند وجود الحاسوب، سيشغل الطالب البرنامج، ويحاول إصلاح الأخطاء، وهذا قد يتطلب وقتاً وجهداً. أما على الورق، فقد يكتفي الطالب بكتابة البرنامج، وإن كان يحتوي على بعض الأخطاء البسيطة.
- هل نريد قياس المهارة مع مساعدات بيئة التطوير المتكاملة؟
- تعطي كثيرٌ من بيئات التطوير المتكاملة أدوات تساعد المبرمج عن طريق إعطاء اقتراحات أو تنبيهات حول الأخطاء، وهذا غير متوافر عند الكتابة على ورق. فهل المقصود قياس مهارة الطالب على نحو يتوافق مع الطريقة التي سيرمج بها في الواقع (لن يرمج على ورق)، أم المقصود قياس مدى إلمام الطالب باللغة من دون مساعدة أيٍّ من الأدوات؟
- كيف ستُقدَّر العلامة:
- عندما يكتب الطالب على ورق، لا يستطيع تشغيل البرنامج لمعرفة جميع الأخطاء؛ لذلك يمكن التجاوز عن بعض الأخطاء على نحوٍ مختلف عما إن كان الطالب قد حصل على الفرصة لتجربة البرنامج على الحاسوب، فمثلاً:
- كتابة اسم دالة على نحو خطأ على الورق مشكلة أصغر من الخطأ نفسه عند الكتابة على الحاسوب؛ لأن الطالب على الحاسوب، يستطيع التأكد من صحة اسم الدالة.
- لذلك، هل ستُقدَّر علامة الطالب عن طريق تشغيل البرنامج لمعرفة أي الحالات يكون فيها الناتج صحيحاً؟ أم سيُقرأ البرنامج لتقدير إلى أيِّ حدٍّ يعبر البرنامج الذي كتبه عن فهم (بغض النظر عن كون البرنامج يعمل أو لا يعمل)؟
- في الحالة الأولى: الاختبار على الحاسوب، يجعل التصحيح أسهل؛ لأن المدرس يمكنه تشغيل البرنامج بدلاً من تتبعه بشكل يدوي.
- في الحالة الثانية: لا يقدم الحاسوب فائدة عند التصحيح، إلا إذا كانت قراءة الحل أسهل من قراءة خط الطالب الذي قد لا يكون واضحاً.

مهارة تطوير البرمجيات:

تجمع هذه المهارة جميع المهارات السابقة، بالإضافة لمهارات أخرى مثل التخطيط والتصميم والتجريد والتوثيق والتواصل وغير ذلك، ونحتاج لقياسها مشروعات طويلة نسبياً، وسلا لم تقدير تأخذ بعين الحسبان جوانب متعددة حول جودة البرمجية المطورة كمنتج (الفائدة، وسهولة الاستخدام، واستيفاء المطلوب ، والإبداع ، إلخ) وجودة الكود المكتوب (السرعة، والمقروئية، والقابلية للتعديل، والتنظيم، والتوثيق، إلخ).

ملاحظات

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

ملحق (2): أدوات برمجية مفيدة:

بيئة التطوير المتكاملة:

في حال تعذر تثبيت بيئة التطوير المتكاملة المذكورة في المنهاج لأي سبب كان، يمكن للمعلم استخدام بيئة تطوير تعمل على المتصفح مباشرة؛ إذ إن بيئات التطوير كهذه، لا تحتاج لأكثر من الاتصال بالإنترنت، ووجود متصفح ويب.

أمثلة على بيئات التطوير التي تعمل في المتصفح (جميع هذه البيئات مجانية، ولا تحتاج لتسجيل الدخول بحساب):

■ <https://pythonhow.com/python-shell>

■ https://www.onlinegdb.com/online_python_compiler

■ <https://www.python.org/shell> بيئة تفاعلية فقط. (interactive shell)

توجد أيضًا بيئات تطوير مجانية، ولكنها تُظهر كثيرًا من الإعلانات التجارية، مثل:

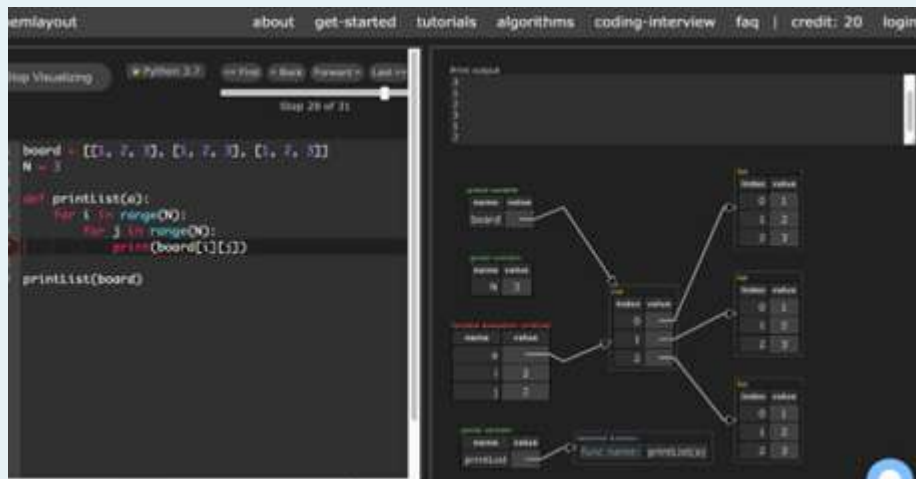
■ <https://www.online-python.com>

■ <https://www.pythonide.online>

■ https://www.online-ide.com/online_python_ide

توجد أيضًا كثير من البيئات الأخرى التي تحتاج لعمل حساب، أو للدفع، أو ليست سهلة الاستخدام.

ملحوظة مهمة: في حال وجود بعض الطلبة الذين لا يمتلكون أجهزة حاسوب في المنزل، يمكن توجيههم لاستخدام أحد هذه البيئات عن طريق متصفح الويب على أي جهاز هاتف متنقل.



تتبع أثر الكود في الذاكرة بصريًا :

تسمح الأدوات الآتية بتنفيذ برامج البايثون خطوة بخطوة من أجل تتبع أثرها في الذاكرة (بدلاً من رؤية النتيجة النهائية لتنفيذ جميع الخطوات ككل)، وتقوم أيضًا بالتمثيل البصري لحالة الذاكرة بعد كل خطوة:

■ <https://memlayout.com>

■ <https://pythontutor.com/visualize.html#mode=edit>

ملحق (3) : تعليمات ذات صلة بمشروعات التعلم (مشروعات الوجد) وأدوات التقييم الخاصة بها.

المشروع الأول: تصميم برنامج وإعداده لإنشاء لعبة الألغاز باستخدام لغة بايثون (Python).

الهدف: إنشاء برنامج بسيط للعبة الألغاز باستخدام لغة البرمجة بايثون؛ إذ تُعرض الأسئلة على المستخدم ويجب عنها، ويقارن الإجابات، ويُظهر النتائج.

تعليمات:

أتبع الخطوات والتوجيهات الآتية لفهم الأساسيات والتقنيات المطلوبة لتحقيق الهدف.

المقدمة:

أبدأ بتقديم المشروع وهدفه المتمثل في تصميم وتطوير لعبة ألغاز تعليمية باستخدام لغة بايثون. الهدف من هذا المشروع تطوير فهم الطالب لمفاهيم البرمجة الأساسية مثل الدوال، والحلقات، والشروط، ومعالجة المدخلات والمخرجات. فهم الهدف والمفهوم:

■ شرح الهدف: الهدف من المشروع بناء لعبة ألغاز تعليمية، تمكن المستخدم من الإجابة عن مجموعة من الأسئلة، مع عرض النتائج النهائية بعد انتهاء اللعبة.

■ المفهوم الأساسي: تعتمد اللعبة على عرض سلسلة من الأسئلة على المستخدم؛ إذ يقوم بإدخال الإجابة. وبعد كل إجابة، يُتحقق منها، وتُعطى ردود الفعل المناسبة، سواء أكانت الإجابة صحيحة أم خاطئة.

مرحلة التخطيط:

أوجه الطلبة للقيام بما يأتي:

■ تحديد نوعية الألغاز التي ستُضمّن في اللعبة (ثقافية، علمية، رياضية، إلخ).

• مكونات الألغاز: حتى يستطيع الطلبة تحليل برنامج الألغاز، أوجه الطلبة لمناقشة الألغاز ومكوناتها: ممّ تتكون الألغاز؟ ما المجال الذي سأضع الأسئلة فيه؟

■ جمع الأسئلة والإجابات الصحيحة وتخزينها في قائمة أو قاموس.

■ تحديد الآلية التي ستُعرض عن طريقها الأسئلة على المستخدم، وتدخل الإجابات. وهنا يجب البدء بالتخطيط لما يأتي:

• كيف ستُعرض الأسئلة على المستخدم؟

• ما الأداة التي سوف نستخدمها لإدخال الجواب لكل سؤال مطروح؟

• كيف سيقارن البرنامج الإجابات المدخلة مع الإجابة الصحيحة التي خُزنت مسبقاً في البرنامج؟

• إذا كانت الإجابة المدخلة صحيحة أو خاطئة، ماذا سوف يحدث؟

• كيف يمكن أن نزيد الدرجة الخاصة بالطالب إذا كانت الإجابة صحيحة؟

- ماذا لو كانت الإجابة خاطئة، ما الذي سيظهر للاعب؟
- كيف سينتقل اللاعب من سؤال لآخر داخل لعبة الألغاز؟
- ما معنى الفوز في هذه اللعبة؟ وكيف ستنتهي اللعبة، ومتى ستنتهي؟

مرحلة التصميم:

أوجه الطلبة للقيام بما يأتي:

- رسم مخطط انسيابي يوضح سير اللعبة بدءاً من عرض الأسئلة إلى إظهار النتيجة النهائية.
- تصميم واجهة بسيطة لعرض الأسئلة، واستقبال الإجابات من المستخدم.
- تحديد كيفية مقارنة الإجابات المدخلة بالإجابات المخزنة مسبقاً.

مرحلة التطوير:

أوجه الطلبة للقيام بما يأتي:

- كتابة الأكواد اللازمة لعرض الأسئلة، وإدخال الإجابات.
- برمجة الدوال المطلوبة لمقارنة الإجابات، وإظهار النتائج.
- إضافة ميزة حساب الدرجات، وعرض النتيجة النهائية بعد انتهاء جميع الأسئلة.

مرحلة المراجعة وتصحيح الأخطاء:

أوجه الطلبة للقيام بما يأتي:

- تجربة اللعبة: أوجه الطلبة لاختبار اللعبة بجميع المراحل؛ للتأكد من أن كل شيء يعمل على نحو صحيح، ولتحديد الأخطاء والمشكلات.
- مراجعة الأقران وجمع الملاحظات: أوجه الطلبة؛ للطلب من الزملاء تجربة اللعبة، وجمع ملحوظاتهم لتحسينها.
- التصحيح: أوجه الطلبة لتصحيح الأخطاء، وتعديل اللعبة بناءً على الملاحظات.

مرحلة التوثيق والعرض:

أوجه الطلبة للقيام بما يأتي:

- التوثيق: أجعل الطلبة يوثقون مراحل تطوير اللعبة وخطوات البرمجة على نحو منظم.
- العرض: أكلف الطلبة إنشاء عرض تقديمي يوضح فكرة اللعبة، وكيفية تطويرها، والتحديات التي واجهتها، مع إتاحة المجال لعرض اللعبة أمام الطلبة الآخرين أو خلال منافسة مدرسية.
- مشاركة اللعبة: أوجه الطلبة إلى مشاركة اللعبة مع الآخرين، سواء عبر الإنترنت أو على منصات خاصة بالمدرسة.

معايير التقييم:

- جودة الكود البرمجي وصحته: التحقق من خلو الكود من الأخطاء اللغوية والمنطقية.

- الاشتمال على البنود المطلوبة: التأكد من أن اللعبة تشتمل على جميع البنود المطلوبة.
- الاستخدام الصحيح والفعال للعناصر البرمجية: التقييم بناءً على كيفية استخدام الطلبة للعناصر البرمجية على نحو صحيح.
- توظيف سيناريوهات الفحص، وتتبع المقطع البرمجي، والتأكد من خلوه من الأخطاء: التحقق من صحة البرمجة، وتصحيح الأخطاء.
- إرفاق مخطط سهمي توضيحي: وجود مخطط شامل يوضح جميع جزئيات اللعبة.
- العمل الجماعي وتقاسم الأدوار: فعالية التعاون وتوزيع الأدوار بين أعضاء الفريق.
- عرض المشروع: وضوح العرض التقديمي للعبة وشموليته.
- نشر اللعبة وتوثيق المصادر: التأكد من نشر اللعبة مع توثيق جميع المصادر المستخدمة في المشروع.

مرحلة التقويم والتأمل:

أوجه الطلبة للقيام بما يأتي:

- تحليل الأداء: تحليل كيفية أداء اللعبة بناءً على ردود الفعل والتجربة، وتقييم اللعبة وفقاً لمعايير التقييم.
- التعلم من التجربة: مناقشة ما تعلمه الطلبة من عملية تطوير اللعبة، وكيف يمكنهم تحسين مهاراتهم في المستقبل.
- التوسع والتحسين: إعطاء توصيات لتحسينات مستقبلية يمكن تطبيقها على المشروع.
- مشاركة المشروع بشكله النهائي: نشر المشروع وعرضه على مستوى أوسع، أو مشاركته في مسابقات أو فعاليات مدرسية.

خطوات توضيحية للمساعدة في متابعة الطلبة في عملية تطوير اللعبة وإنشائها:

- خطوة 1: إنشاء ملف جديد بامتداد py في بيئة تطوير Python مثل IDLE أو Visual Studio Code.
- خطوة 2: كتابة الكود بالاستيراد اللازم: `import random`
- خطوة 3: تعريف الألغاز والإجابات المقابلة كما في المثال:

ما هو الشيء الذي يمكنه "،" لوحة المفاتيح " : " ما هو الشيء الذي لديه مفاتيح، ولكن لا يمكنه فتح الأقفال؟ " }
 puzzles = {
 وأنا صغيرة، أكون طويلة، وعندما أكبر أكون قصيرة، فمن "، "الختم" : " السفر حول العالم وفي الوقت ذاته يبقى في زاوية؟
 } "الشمعة" : " أنا؟

تذكير: هذه الأسئلة على سبيل المثال وليس الحصر.

- خطوة 4: أكمل بكتابة دالة main() التي تشمل الشرح والتفاصيل اللازمة لكيفية تنفيذ اللعبة:

```
def main():
    print("Welcome to the Puzzle Game!")
    score = 0
    num_questions = len(puzzles)
    for question, answer in puzzles.items():
        user_answer = input(question).strip().capitalize()
        if user_answer == answer:
            print("Correct!")
            score += 1
        else:
            print(f"Wrong! The correct answer is {answer}.")
            print(f"Your final score is {score} / num_questions.")

if __name__ == "__main__":
    main()
```

- تبدأ الدالة بطباعة رسالة ترحيبية باستخدام (print) تعرض نصًا يُرحَّب بالمستخدم، ويخبره أنه دخل إلى لعبة الألغاز.
- تقوم الدالة بإعداد متغير score بقيمة صفر، وهو يستخدم لتتبع عدد الإجابات الصحيحة التي يقدمها المستخدم.
- تستخدم len(puzzles) لحساب عدد الأسئلة الموجودة في قاموس الألغاز puzzles؛ إذ يمثل هذا العدد الإجمالي الألغاز في اللعبة.
- حل الألغاز:
- تبدأ دورة for بالمرور على كل عنصر في قاموس puzzles باستخدام (puzzles.items())؛ إذ يتم تعيين كل سؤال إلى question وكل إجابة إلى answer.
- طُلب من المستخدم إدخال إجابته عبر (input())، مع عرض السؤال question المناسب.
- ستخدم (.capitalize())، (.strip())، (input()) للتأكد من إزالة أي فراغات زائدة حول الإجابة، وجعل الإجابة بداية بالحرف الكبير؛ مما يساعد على المطابقة الصحيحة بغض النظر عن تنسيق الإدخال.
- التحقق من الإجابة:

- تقوم بمقارنة user_answer التي أدخلها المستخدم مع answer المحددة مسبقًا كإجابة صحيحة للسؤال الحالي.
- إذا كانت الإجابة صحيحة (user_answer == answer): تطبع "Correct!" لتوضح أن الإجابة صحيحة. تزيد قيمة score بمقدار واحد.
- إذا كانت الإجابة غير صحيحة، تطبع {Wrong! The correct answer is "answer"} لتوضح الإجابة الصحيحة للمستخدم.

■ إنهاء اللعبة:

- بعد الانتهاء من جميع الأسئلة، تطبع main() رسالة توضح النتيجة النهائية للمستخدم، مثل:
Your final score is "{score}/{num_questions}"
- حيث score هو عدد الإجابات الصحيحة التي أدلى بها المستخدم و num_questions هو عدد الأسئلة الكلي.

■ ملحوظات إضافية:

- تُعد الدالة main() مركزية في هذا البرنامج؛ إذ تنظم التفاعل بين المستخدم واللعبة.
- ينفَّذ الكود باستخدام:

```
if __name__ == "__main__":
    main()
```

للتأكد من أن الكود يعمل بوصفه برنامجًا رئيسًا عند تشغيل الملف مباشرة.

مثال لبرنامج كامل للعبة الألغاز:

```
def main():
    print("Welcome to the Puzzle Game!")
    score = 0
    num_questions = len(puzzles)
    for question, answer in puzzles.items():
        user_answer = input(question).strip().capitalize()
        if user_answer == answer:
            print("Correct!")
            score += 1
        else:
            print(f"Wrong! The correct answer is {answer}.")
    print(f"Your final score is {score}/{num_questions}.")

if __name__ == "__main__":
    main()
```

لتجربة اللعبة ألغاز: <https://www.online-python.com/o9R4nk8eBT>

المشروع الثاني: تصميم نموذج أولي وإعداده لإنشاء برنامج يساعد الطالب على إدارة المهام المنوطة به باستخدام لغة بايثون (Python).

الهدف:

إنشاء برنامج بسيط لإدارة المهام، يمكّن الطلبة من تنظيم مهامهم اليومية بفعالية. وسيتمكن الطلبة من إضافة المهام، وعرضها، وإكمالها، وحذفها من قائمة المهام.

تعليمات:

اتّبع الخطوات والتوجيهات الآتية لفهم الأساسيات والتقنيات المطلوبة لتحقيق الهدف.

المقدمة:

أبدأ بتقديم المشروع وهدفه المتمثل في تصميم برنامج لإدارة المهام وتطويره باستخدام لغة بايثون. والهدف من هذا المشروع تعزيز مهارات البرمجة لدى الطلبة عن طريق تطبيق عملي، يساعدهم في تنظيم وقتهم وإدارته.

فهم الهدف والمفهوم:

شرح الهدف: الهدف من المشروع بناءً برنامج يتيح للطلبة إضافة مهام، وعرضها، وإكمالها، وحذفها؛ مما يساعدهم على متابعة واجباتهم المدرسية والمهام اليومية.

المفهوم الأساسي: يتكون البرنامج من مجموعة من الدّوال المستقلة التي تتفاعل مع المستخدم وتنفذ العمليات المطلوبة على قائمة المهام، مثل الإضافة، العرض، والإكمال.

مرحلة التخطيط:

أوجّه الطلبة للقيام بما يأتي:

- تحديد العمليات الأساسية التي يحتاجها البرنامج (إضافة مهام، عرض مهام، إكمال مهام، حذف مهام).
يتكون البرنامج من العناصر الآتية:

- إضافة مهام: لإدخال مهام جديدة إلى القائمة.
- عرض مهام: لعرض جميع المهام الموجودة في القائمة.
- إكمال المهام: لحذف المهمة من القائمة بعد إكمالها.
- ترتيب المهام عند عرضها: لترتيب المهام بطريقة محددة عند عرضها على المستخدم.

- وضع تصوّر مبدئي لكيفية تفاعل المستخدم مع البرنامج.

طريقة تشغيل البرنامج:

- يعرض البرنامج قائمة خيارات للمستخدم على شكل قائمة منسقة؛ إذ يمكنه اختيار الإجراء المطلوب بالضغط على رقم الخيار المناسب.

طريقة إدخال الخيارات:

- لإضافة مهمة جديدة أو حذف مهمة موجودة، ستستخدم دوالّ الإضافة والحذف المعدة مسبقاً داخل البرنامج.
- لعرض المهام الموجودة، ستستخدم دالة خاصة بعرض القائمة المهام المدخلة.

مقارنة الخيارات المدخلة:

- عندما يُدخل المستخدم خياراً، يتحقق البرنامج مما إذا كان الخيار صحيحاً أم لا باستخدام جملة if-else. ، ستعرض رسالة تنبيهية للمستخدم بأن الخيار غير صحيح، مع إعادة عرض القائمة للسماح له بإدخال خيار صحيح. إعادة تشغيل البرنامج: بعد كل اختيار يقوم به المستخدم، سيُعاد عرض قائمة الخيارات له مرة أخرى؛ مما يتيح له استخدام البرنامج على مدار الوقت بسهولة.

مرحلة التصميم:

أوجه الطلبة للقيام بما يأتي:

- رسم مخطط انسيابي، يوضح كيفية تدفق البيانات بين الدوالّ المختلفة في البرنامج.
- تصميم قائمة الخيارات التي ستعرض للمستخدم، وكيفية تنفيذ كل خيار.
- تحديد طريقة إدخال البيانات والتحقق من صحة المدخلات.

مرحلة التطوير:

أوجه الطلبة للقيام بما يأتي:

- كتابة الكود الأساسي للدوال المستقلة مثل add_tasks()، view_tasks()، complete_task()، menu() .
- تنفيذ عملية إضافة المهام، والوقت لإنجاز المهام وعرضها، وإكمالها بناءً على اختيار المستخدم.
- اختبار البرنامج للتأكد من أن جميع الوظائف تعمل على نحو صحيح.

مرحلة المراجعة وتصحيح الأخطاء:

أوجه الطلبة للقيام بما يأتي:

- تجربة البرنامج: أوجه الطلبة لاختبار البرنامج بجميع المراحل؛ للتأكد من أن كل شيء يعمل على نحو صحيح، ولتحديد الأخطاء والمشكلات.
- مراجعة الأقران وجمع الملاحظات: أوجه الطلبة للطلب من زملاء تجربة البرنامج، وجمع ملحوظاتهم لتحسينه.
- التصحيح: أوجه الطلبة لتصحيح الأخطاء، وتعديل البرنامج بناءً على الملاحظات.

مرحلة التوثيق والعرض:

أوجه الطلبة للقيام بما يأتي:

- التوثيق: أجعل الطلبة يوثقون مراحل تطوير البرنامج وخطوات البرمجة على نحو منظم.
- العرض: أكلف الطلبة إنشاء عرض تقديمي، يوضح فكرة البرنامج، وكيفية تطويره، والتحديات التي واجهوها، مع إتاحة المجال لعرض البرنامج أمام الطلبة الآخرين، أو خلال منافسة مدرسية.
- مشاركة البرنامج: أوجه الطلبة إلى مشاركة البرنامج مع الآخرين، سواء عبر الإنترنت أو على منصات خاصة بالمدرسة.

معايير التقييم:

- جودة الكود البرمجي وصحته: التحقق من خلو الكود من الأخطاء اللغوية والمنطقية.
- الاشتمال على البنود المطلوبة: التأكد من أن البرنامج يشتمل على جميع البنود المطلوبة.
- الاستخدام الصحيح والفعال للعناصر البرمجية: التقييم بناءً على كيفية استخدام الطلبة للعناصر البرمجية على نحو صحيح.
- توظيف سيناريوهات الفحص، وتتبع المقطع البرمجي، والتأكد من خلوه من الأخطاء: التحقق من صحة البرمجة وتصحيح الأخطاء.
- إرفاق مخطط سهمي توضيحي: وجود مخطط شامل، يوضح جميع جزئيات البرنامج.
- العمل الجماعي وتقاسم الأدوار: فعالية التعاون وتوزيع الأدوار بين أعضاء الفريق.
- عرض المشروع: وضوح العرض التقديمي للبرنامج وشموليته.
- نشر البرنامج وتوثيق المصادر: التأكد من نشر البرنامج مع توثيق جميع المصادر المستخدمة في المشروع.

مرحلة التقويم والتأمل:

أوجه الطلبة للقيام بما يأتي:

- تحليل الأداء: تحليل كيفية أداء البرنامج بناءً على ردود الفعل والتجربة، وتقييم البرنامج وفقاً لمعايير التقييم.
- التعلم من التجربة: مناقشة ما تعلمه الطلبة من عملية تطوير البرنامج، وكيف يمكنهم تحسين مهاراتهم في المستقبل.
- التوسع والتحسين: إعطاء توصيات لتحسينات مستقبلية يمكن تطبيقها على المشروع.
- مشاركة المشروع بشكله النهائي: نشر المشروع وعرضه على مستوى أوسع، أو مشاركته في مسابقات أو فعاليات مدرسية.

خطوات توضيحية للمساعدة في متابعة الطلبة في عملية إنشاء اللعبة وتطويرها:

خطوة 1: أنشئ ملفاً جديداً بامتداد py. في بيئة تطوير Python مثل IDLE أو Visual Studio Code.

خطوة 2: فهم الدوال المستقلة:

- يبدأ الكود بتعريف مجموعة من الدوال المستقلة مثل menu()، complete_task()، view_tasks()، add_task()؛ مما يسمح لنا بإدارة قائمة المهام، والتفاعل مع المستخدم بطريقة مبسطة.
- الدالة add_task()، ويضاف لها خيار تحدي الوقت لإنجاز المهمة.
- تضيف مهمة جديدة إلى قائمة المهام التي تمثلها قائمة tasks الموجودة في النطاق العالمي، وتطبع رسالة تأكيد بنجاح إضافة المهمة.

```
def add_task(task, deadline=None):  
    task_info = {"task": task, "deadline": deadline}  
    tasks.append(task_info)  
    print("أضيفت المهمة بنجاح")
```

• الدالة view_tasks()

◦ تُعرض جميع المهام الموجودة في قائمة tasks، مع رقم كل مهمة لتمكين المستخدم من تحديدها

```
def view_tasks():  
    if not tasks:  
        print("لا يوجد مهام مدرجة")  
    else:  
        print("المهام:")  
        for index, task_info in enumerate(tasks, start=1):  
            task = task_info["task"]  
            deadline = task_info["deadline"]  
            if deadline:  
                deadline_str = deadline.strftime("%Y-%m-%d %H:%M")  
                print(f"{index}. {task} (يجب انجازها قبل: {deadline_str})")  
            else:  
                print(f"{index}. {task}")
```

◦ الدالة complete_task()

◦ تحذف المهمة المحددة استناداً إلى الفهرس المحدد من المستخدم، وتطبع رسالة تأكيد بنجاح إكمال المهمة وحذفها.

```
def complete_task(task_index):  
    if 1 <= task_index <= len(tasks):  
        del tasks[task_index - 1]  
        print("أكملت المهمة وحذفت من قائمة المهام")  
    else:  
        print("رقم المهام غير صالح")
```

• الدالة menu() - قائمة الخيارات:

- تعرض قائمة الخيارات المتاحة للمستخدم (أضافة مهمة جديدة، وعرض المهام، وإكمال مهمة، وخروج).
- وتطلب من المستخدم اختيار الخيار الذي يرغب في تنفيذه.
- تُنفَّذ الدالَّة المناسبة بناءً على الاختيار المدخل من المستخدم.

```
def menu():  
    while True:  
        print("\n( القائمة  
        print("1. إضافة مهمة جديدة")  
        print("2. عرض المهام")  
        print("3. أكمل المهمة")  
        print("4. خروج")  
        choice = input(" أدخل اختيارك ")
```

ملاحظات

.....

.....

.....

.....

.....

.....

.....

.....

- التفاعل مع المستخدم:

- عند تشغيل البرنامج، تُعرض قائمة الخيارات للمستخدم. يختار المستخدم الخيار المناسب، سواء لإضافة مهمة، أو عرض المهام الموجودة، أو إكمال مهمة محددة، أو الخروج من البرنامج.
- يستجيب البرنامج على نحو مناسب بناءً على اختيارات المستخدم، ويقدم رسائل تأكيد لكل عملية تنفيذية.

```
if choice == "1":
    task = input("أدخل نوع المهمة: ")
    deadline_input = input("هل ترغب في تحديد وقت لإنجاز المهمة؟ (نعم/لا): ")
    deadline_input = deadline_input.strip().lower()
    if deadline_input == "نعم":
        deadline_str = input("أدخل الوقت بصيغة YYYY-MM-DD HH:MM: ")
        try:
            deadline = datetime.strptime(deadline_str, "%Y-%m-%d %H:%M")
        except ValueError:
            print("التنسيق غير صحيح. سنُضاف المهمة من دون وقت محدد")
            deadline = None
    else:
        deadline = None
    add_task(task, deadline)

elif choice == "2":
    view_tasks()

elif choice == "3":
    if tasks:
        task_index = int(input("أدخل رقم المهمة المراد إكمالها: ")) - 1
        complete_task(task_index)
    else:
        print("No tasks to complete.")

elif choice == "4":
    print(".... خروج")
    break

else:
    print("اختيار غير صحيح! الرجاء إدخال خيار صحيح")
```

```
from datetime import datetime

tasks = []

def add_task(task, deadline=None):
    """إضافة مهمة جديدة إلى قائمة المهام."""
    task_info = {"task": task, "deadline": deadline}
    tasks.append(task_info)
    print("تم إضافة المهمة بنجاح")

def view_tasks():
    """عرض جميع المهام الموجودة في القائمة."""
    if not tasks:
        print("لا يوجد مهام مدرجة")
    else:
        print("المهام:")
        for index, task_info in enumerate(tasks, start=1):
            task = task_info["task"]
            deadline = task_info["deadline"]
            if deadline:
                deadline_str = deadline.strftime("%Y-%m-%d %H:%M")
                print(f"{index}. {task} (يجب إنجازها قبل: {deadline_str})")
            else:
                print(f"{index}. {task}")

def complete_task(task_index):
    """إكمال وحذف مهمة من القائمة استنادًا إلى الفهرس المحدد."""
    if 1 <= task_index <= len(tasks):
        del tasks[task_index - 1]
        print("تم أكمل المهمة وحذفها من قائمة المهام")
    else:
        print("!رقم المهام غير صالح")
```

```

def menu():
    """عرض قائمة الخيارات والتفاعل مع المستخدم."""
    while True:
        print("\nالقائمة:")
        print("1. إضافة مهمة جديدة")
        print("2. عرض المهام")
        print("3. أكمل المهمة")
        print("4. خروج")
        choice = input("أدخل اختيارك: ")

        if choice == "1":
            task = input("أدخل نوع المهمة: ")
            deadline_input = input("هل ترغب في تحديد وقت لإنجاز المهمة؟ (نعم/لا): ").strip().lower()
            if deadline_input == "نعم":
                deadline = input("أدخل الوقت بصيغة YYYY-MM-DD HH:MM: ")
                try:
                    deadline = datetime.strptime(deadline, "%Y-%m-%d %H:%M")
                except ValueError:
                    print("التنسيق غير صحيح. سيتم إضافة المهمة بدون وقت محدد.")
                    deadline = None
            else:
                deadline = None
            add_task(task, deadline)
        elif choice == "2":
            view_tasks()
        elif choice == "3":
            task_index = int(input("أدخل رقم المهمة المراد إكمالها: "))
            complete_task(task_index)
        elif choice == "4":
            print(".... خروج")
            break
        else:
            print("اختيار غير صحيح! الرجاء إدخال خيار صحيح")

    menu()

```

لتجربة قائمة المهام: <https://www.online-python.com/z354DOR6iJ>

ألعاب أخرى:

لعبة الرياضيات:

<https://www.online-python.com/3pPaXrRtEH>

لعبة "ورقة حجر مقص"

<https://www.online-python.com/J5jYG1atew>

قائمة المراجع

• مراجع حول قياس المهارات البرمجية

- ListerR.SimonB.ThompsonE.WhalleyJ. L.& PrasadC. (2006). Not seeing the forest for the trees: novice programmers and the SOLO taxonomy. ACM SIGCSE Bulletin38(3)118–122.
- VenablesA.TanG.& ListerR. (2009August). A closer look at tracingexplaining and code writing skills in the novice programmer. In Proceedings of the fifth international workshop on Computing education research workshop (pp. 117–128).
- FitzgeraldS.HanksB.ListerR.McCauleyR.& MurphyL. (2013March). What are we thinking when we grade programs. In Proceeding of the 44th ACM technical symposium on Computer science education (pp. 471–476).
- CatetéV.SniderE.& BarnesT. (2016July). Developing a rubric for a creative CS principles lab. In Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (pp. 290–295).
- LancasterT.RobinsA. V.& FincherS. A. (2019). Assessment and Plagiarism. In S. A. Fincher & A. V. Robins (Eds.)The Cambridge Handbook of Computing Education Research (pp. 414–444). chapterCambridge: Cambridge University Press.
- LiC.ChanE.DennyP.Luxton–ReillyA.& TemperoE. (2019January). Towards a framework for teaching debugging. In Proceedings of the Twenty–First Australasian Computing Education Conference (pp. 79–86).
- WeedaR.IzuC.KalliaM.& BarendsenE. (2020November). Towards an assessment rubric for eipe tasks in secondary education: Identifying quality indicators and descriptors. In Proceedings of the 20th Koli Calling International Conference on Computing Education Research (pp. 1–10).
- ChenB.AzadS.HaldarR.WestM.& ZillesC. (2020February). A validated scoring rubric for explain–in–plain–english questions. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (pp. 563–569).
- CostantiniU.LonatiV.& MorpurgoA. (2020February). How plans occur in novices' programs: A method to evaluate program–writing skills. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (pp. 852–858).
- WhalleyJ.SettleA.& Luxton–ReillyA. (2023). A think–aloud study of novice debugging. ACM Transactions on Computing Education23(2)1–38.

• مراجع حول البرمجة الحية

- RajA. G. S.PatelJ. M.HalversonR.& HalversonE. R. (2018November). Role of live–coding in learning introductory programming. In Proceedings of the 18th koli calling international conference on computing education research (pp. 1–8).
- SelvarajA.ZhangE.PorterL.& Soosai RajA. G. (2021June). Live coding: A review of the literature. In Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (pp. 164–170).
- ShahA.HoganE.AgarwalV.DriscollJ.PorterL.GriswoldW. G.& Soosai RajA. G. (2023August). An Empirical Evaluation of Live Coding in CS1. In Proceedings of the 2023 ACM Conference on International Computing Education Research–Volume 1 (pp. 476–494).

• مراجع حول استخدام التوقعات

- MillerK.LasryN.ChuK.& MazurE. (2013). Role of physics lecture demonstrations in conceptual learning. Physical Review Special Topics—Physics Education Research9(2)020113.

- BrodG. (2021). Predicting as a learning strategy. Psychonomic Bulletin & Review28(6)1839–1847.

● مراجع حول البرمجة مع قرين

- DennerJ.WernerL.CampeS.& OrtizE. (2014). Pair programming: Under what conditions is it advantageous for middle school students Journal of Research on Technology in Education46(3)277–296.
- UmapathyK.& RitzhauptA. D. (2017). A meta-analysis of pair-programming in computer programming courses: Implications for educational practice. ACM Transactions on Computing Education (TOCE)17(4)1–13.
- HawlitschekA.BerndtS.& SchulzS. (2023). Empirical research on pair programming in higher education: a literature review. Computer science education33(3)400–428.

● مراجع حول توسيم الأهداف الفرعية

- MorrisonB. B.MargulieuxL. E.& GuzdialM. (2015August). Subgoalscontextand worked examples in learning computing problem solving. In Proceedings of the eleventh annual international conference on international computing education research (pp. 21–29).
- MargulieuxL. E.& CatramboneR. (2016). Improving problem solving with subgoal labels in expository text and worked examples. Learning and Instruction4258–71.
- MargulieuxL. E.MorrisonB. B.& DeckerA. (2020). Reducing withdrawal and failure rates in introductory programming with subgoal labeled worked examples. International Journal of STEM Education71–16.

الحوسبة والحياة (Computing and Life)

الثانية 2



مُقدِّمة

تتناول هذه الوحدة مجموعة من القضايا البيئية والاجتماعية المرتبطة بالحوسبة، مع التركيز على مفهوم الحوسبة الخضراء وأهميتها، وكيفية تطبيقها. وسيتعرف الطلبة إلى النُّفايات الإلكترونية، والطرائق الصحيحة للتخلص منها، ويستكشفون في السياق الأدوات الحاسوبية الصديقة للبيئة.

ويستكشف الطلبة في ثانيا هذه الوحدة أيضا أهمية تطبيقات الحاسوب في الحياة الاجتماعية والاقتصادية، ويتعرفون أهمية التعلُّم الإلكتروني والتعليم عن بعد، واستخدام الحاسوب في مجال الصحة، والتسوق، والتسويق الإلكتروني، والحكومة الإلكترونية.

يخوض الطلبة في رحلة تعلُّمهم أيضا تجربة استكشاف تطبيقات الحاسوب المتقدمة في مجال صناعة الأفلام، والتصميم ثلاثي الأبعاد، والرسوم المتحركة، والطباعة ثلاثية الأبعاد، وكيفية استعمال هذه التطبيقات بطرائق مبتكرة ومبدعة، تخدمهم في مجالات الحياة المختلفة.

مصفوفة نتائج التعلُّم

نتائج تعلُّم الصف العاشر	نتائج تعلُّم الصف الحادي عشر	نتائج تعلُّم الصف الثاني عشر
■ يصف أهمية وسائل التكنولوجيا الحديثة للفرد والمجتمع.	■ يعرف الحوسبة الخضراء ويبين أهميتها.	■ يصف تطبيقًا لشبكة لاسلكية تُستخدم في حماية البيئة.
■ يبين مخاطر وسائل التكنولوجيا الحديثة على الفرد.	■ يسهم في تطبيق الحوسبة الخضراء عمليًا.	■ يصف تطبيقًا لشبكة لاسلكية تُستخدم لتقليل استهلاك الطاقة.
■ يبين مخاطر وسائل التكنولوجيا الحديثة على المجتمع.	■ يعرف النُّفايات الإلكترونية.	■ يصف تطبيقًا لشبكة لاسلكية تُستخدم في حماية البيئة.
■ يتعرف طرائق الوقاية من مخاطر وسائل التكنولوجيا	■ يوضح طرائق التخلص الآمنة من النُّفايات الإلكترونية.	
	■ يعدد بعض الأدوات الحاسوبية الصديقة للبيئة.	
	■ يستخدم تطبيقات الحاسوب في مجال الصحة، والتعليم، والاقتصاد، والحياة.	
	■ يوضح أثر استخدام تطبيقات الحاسوب في مجال التعليم، والصحة، والاقتصاد..	
	■ يبين أهمية تطبيقات الحاسوب (صناعة الأفلام، والتصميم ثلاثي الأبعاد، والرسوم المتحركة، والطباعة ثلاثية الأبعاد، والوسائط المتعددة) في تنفيذ المشروع.	

مصفوفة نتائج التعلم ومؤشرات الأداء حسب الوحدة

المحور	المعيار	نتائج التعلم	مؤشرات الأداء
■ الحوسبة والحياة	■ تعرّف قضايا بيئية واجتماعية متعلقة بالحوسبة. ■ بيان أهمية تطبيقات الحاسوب في الحياة الاجتماعية والاقتصادية.	■ يتعرّف الحوسبة الخضراء. ■ يعرّف النفايات الإلكترونية. ■ يبين كيفية التخلص من النفايات الإلكترونية. ■ يذكر أدوات حاسوبية صديقة للبيئة. ■ يذكر تطبيقات حاسوبية في مجال التعليم الإلكتروني. ■ يستخدم تطبيقات حاسوبية في التعلم عن بعد. ■ يذكر تطبيقات حاسوبية في مجال الصحة ويبين أهميتها. ■ يذكر تطبيقات حاسوبية في مجال التسويق والتسويق الإلكتروني ويبين أهميتها. ■ يستخدم تطبيقات حاسوبية للتسويق، والتسويق الإلكتروني. ■ يبين أهمية تطبيقات الحكومة الإلكترونية في تسهيل الحياة الاجتماعية والاقتصادية. ■ يبين أهمية تطبيقات الحاسوب (صناعة الأفلام، والتصميم ثلاثي الأبعاد، الرسوم المتحركة، الطباعة ثلاثية الأبعاد، الوسائط المتعددة) لتنفيذ المشروع	■ يُعرّف مفهوم الحوسبة الخضراء. ■ يبين أهمية الحوسبة الخضراء وفوائدها ومزاياها. ■ يسهم في تطبيق الحوسبة الخضراء في حياته اليومية.
		■ يتعامل مع النفايات الإلكترونية تعاملاً آمناً. ■ يبين أثر استخدام تطبيقات الحاسوب في الصحة، والتعليم، والاقتصاد، والحياة.	

■ يستخدم احدى تطبيقات الحاسوب (صناعة الأفلام، التصميم ثلاثي الأبعاد، الرسوم المتحركة، الطباعة ثلاثية الأبعاد، الوسائط المتعددة) لتنفيذ المشروع	■ يخطط مع الزملاء لفكرة مشروع رياضي رقمي. ■ يحدد فكرة مشروع رياضي بالتعاون مع الزملاء والمعلم. ■ يكتب خطة لتنفيذ المشروع باستخدام أحد تطبيقات Office تتضمن: الأدوات اللازمة والفترة الزمنية، وتحديد الأدوار، وطريقة العمل، والنشر. ■ ينفذ المشروع وفق الخطة الزمنية باستخدام أحد تطبيقات الحاسوب (صناعة الأفلام، والتصميم ثلاثي الأبعاد، والرسوم المتحركة، والطباعة ثلاثية الأبعاد، والوسائط المتعددة).	■ تبني مشروعات رقمية رياضية.	■ الاستخدام المتوازن للحاسوب وأدواته وتطبيقاته
--	--	--	--

مصفوفة نتائج التعلم ومؤشرات الأداء بحسب دروس الوحدة

الدرس	مؤشرات الأداء لكل درس	عدد الحصص المتوقعة	المفاهيم والمصطلحات
الدرس الأول: الحوسبة الخضراء Green Computing	■ يعرف الحوسبة الخضراء. ■ يبين أهمية الحوسبة الخضراء. ■ يسهم في تطبيق الحوسبة الخضراء	حصتان	■ الحوسبة الخضراء (Green Computing). ■ نجمة الطاقة (Energy Star).
الدرس الثاني: النفايات الإلكترونية Electronic Waste	■ يعرف النفايات الإلكترونية. ■ يبين كيفية التخلص من النفايات الإلكترونية. ■ يذكر أدوات حاسوبية صديقة للبيئة.	حصتان	■ النفايات الإلكترونية (E-Waste). ■ البصمة الكربونية الرقمية (Footprint Digital Carbon). ■ الأدوات الحاسوبية الصديقة للبيئة (Eco - Friendly Computing Tools).

المفاهيم والمصطلحات	عدد الحصص المُتَوَقَّع	مُؤَشِّرَات الأداء لكل درس	الدرس
■ التعلّم الإلكتروني (E-Learning). ■ التعلّم عن بُعد (Online Learning). ■ الحكومة الإلكترونية (E-Government). ■ الطباعة ثلاثية الأبعاد (3D-Printing).	3 حصص	■ يذكر تطبيقات حاسوبية في مجال التعليم الإلكتروني. ■ يستخدم تطبيقات حاسوبية في التعلّم عن بُعد. ■ يذكر تطبيقات حاسوبية في مجال الصحة، ويبين أهميتها. ■ يذكر تطبيقات حاسوبية في مجال التسويق، والتسويق الإلكتروني، ويبين أهميته. ■ يستخدم تطبيقات حاسوبية للتسويق، والتسويق الإلكتروني. ■ يبين أهمية تطبيقات الحكومة الإلكترونية في تسهيل المعاملات. ■ يبين أهمية تطبيقات الحاسوب (صناعة الأفلام، والتصميم ثلاثي الأبعاد، والرسوم المتحركة، والطباعة ثلاثية الأبعاد، والوسائط المتعددة) في الحياة.	الدرس الثالث: تطبيقات الحاسوب في الحياة اليومية Computer Application in our Daily Life

المفاهيم والمصطلحات	عدد الحصص المُتَوَقَّع	مُؤَشِّرَات الأداء لكل درس	الدرس
	مشروع الوحدة	■ يخطط مع الزملاء لفكرة مشروع ريادي رقمي. ■ يحدد فكرة مشروع ريادي بالتعاون مع الزملاء والمعلم. ■ يكتب خُطَّة لتنفيذ المشروع باستخدام أحد تطبيقات Office تتضمن: الأدوات اللازمة، والفترة الزمنية، وتحديد الأدوار، وطريقة العمل، والنشر. ■ ينفذ المشروع وفق الخُطَّة الزمنية باستخدام أحد تطبيقات الحاسوب (صناعة الأفلام، والتصميم ثلاثي الأبعاد، والرسوم المتحركة، والطباعة ثلاثية الأبعاد، والوسائط المتعددة).	.
	(7) حصص.	المجموع	

الوحدة 2

نظرة عامة على الوحدة

- أُنقش الطلبة في الفكرة العامة للوحدة، ثمَّ أُوْضح لهم الأهداف التي يُتَوَقَّع منهم تحقيقها بعد الانتهاء من دراسة الوحدة.
- أُوْضح للطلبة أنهم سيخوضون رحلة تعلُّم تتمحور حول تطوير مشروع ريادي رقمي وتصميمه، يتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة، وأثرها في استخدام إحدى تطبيقات الحاسوب.
- أَسْتعرض مع الطلبة الأدوات الرقمية والبرامج التي يُتَوَقَّع منهم استخدامها، وتطوير مهاراتهم فيها في أثناء دراسة

سياق التعلُّم والمهام الأصيلة

سياق التعلُّم:

خلال دراسة هذه الوحدة، سيتبنى الطلبة دور المبادرين الرياديين الذين يقودون مشروعات تهدف إلى تعزيز الوعي البيئي عن طريق مناقشة القضايا المرتبطة بالحوسبة الخضراء والنُفايات الرقمية. وسيطرقون أيضًا إلى موضوع التحوُّل الرقمي، وتأثيره في الفرد والمجتمع؛ مما يساعدهم على فهم أعمق لتلك القضايا، ضمن سياق التعلُّم الشامل.

الوحدة 2

الحوسبة والحياة (Computing and Life)

نظرة عامة على الوحدة

سأُعرِّف في هذه الوحدة مجموعة من القضايا البيئية والاجتماعية المرتبطة بالحوسبة، بما في ذلك مفهوم الحوسبة الخضراء، وأهميتها، وكيف يُمكن الإسهام في تطبيقها. وكذلك مفهوم النفايات الإلكترونية، والطرائق الصحيحة للتخلص منها، إضافةً إلى الأدوات الحاسوبية الصديقة للبيئة. سأُعرِّف أيضًا أهمية تطبيقات الحاسوب في الحياة الاجتماعية والاقتصادية، مثل: التعلُّم الإلكتروني، والتعليم عن بُعد، واستخدام الحاسوب في مجال الصحة، والتسويق الإلكتروني، والحكومة الإلكترونية. كذلك سأُعرِّف تطبيقات الحاسوب المُتقدِّمة في مجال صناعة الأفلام، والتصميم ثلاثي الأبعاد، والرسوم المُتحرَّكة، والطباعة ثلاثية الأبعاد، وكيفية استعمالها لدعم الابتكار والإبداع في مختلف المجالات.

يُتَوَقَّع مني في نهاية الوحدة أن أكون قادرًا على:

- تعريف الحوسبة الخضراء، وبيان أهميتها.
- الإسهام في تطبيق الحوسبة الخضراء عمليًا.
- تعريف النفايات الإلكترونية.
- توضيح طرائق التخلص الآمنة من النفايات الإلكترونية.
- ذكر بعض الأدوات الحاسوبية الصديقة للبيئة.
- استخدام تطبيقات الحاسوب في مجال الصحة، والتعليم، والاقتصاد، والحياة.
- توضيح أثر استخدام تطبيقات الحاسوب في مجال التعليم، والصحة، والاقتصاد.
- بيان أهمية تطبيقات الحاسوب (صناعة الأفلام، التصميم ثلاثي الأبعاد، الرسوم المُتحرَّكة، الطباعة ثلاثية الأبعاد، الوسائط المُتعدِّدة) في تنفيذ المشروع.

المشروع:

تطوير مشروع ريادي رقمي وتصميمه، يتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة، وأثرها في استخدام إحدى تطبيقات الحاسوب

مُنتَج التعلُّم:

مشروع ريادي رقمي، يتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة، وأثرها في استخدام إحدى تطبيقات الحاسوب..

استراتيجية التعليم والتعلُّم

تستند هذه الوحدة إلى استراتيجية التعلُّم بالمشروع؛ إذ يُشركُ الطلبة في مهمة تخطيط مشروعات ريادية وتنفيذها، تركز على القضايا البيئية والاجتماعية المرتبطة بالحوسبة الخضراء، وأثرها في الفرد والمجتمع. يبدأ الطلبة برحلة تعليمية تزودهم بالمعلومات الأساسية حول مفهوم الحوسبة الخضراء وأهميتها، ثم يتعاونون في كتابة خطة مشروعهم

Python

Canva

Google Forms

مشروع

MS Words

Gantt Charts

143

مُنتَجَات التعلُّم: (Learning Products)

تصميم مشروع ريادي رقمي، يقوم على استخدام أحد تطبيقات الحاسوب، ويتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة وأثرها في الفرد والمجتمع.

أختار مع أفراد مجموعتي أحد المشروعات الآتية لتنفيذه في نهاية الوحدة:

المشروع الأول: تنظيم حملة تثقيفية عن الحوسبة الخضراء؛ لتعزيز وعي المجتمع المدرسي بأهمية الحوسبة الخضراء.

المشروع الثاني: تنظيم مسابقة تحمل عنوان (البرمجة الخضراء)، وتشترط كتابة تعليمات برمجية - باستخدام لغة البرمجة بايثون (Python) - تهدف إلى الحد من استهلاك الطاقة، والتركيز على تحسين كفاءة المقطع البرمجي، والتقليل من استخدام الموارد الحاسوبية.

الأدوات والبرامج: (Digital Tools and Programs)

Canva, Python, Google Forms, MS Words, Gantt Charts

التفكير الحاسوبي، التعاون الرقمي، الابتكار العالمي، التصميم الرقمي، التعليم المستمر، التواصل الرقمي.

فهرس الوحدة

- الدرس الأول: الحوسبة الخضراء (Green Computing).
- الدرس الثاني: النفايات الإلكترونية (Electronic Waste).
- الدرس الثالث: تطبيقات الحاسوب في الحياة اليومية (Computer Application in our Daily Life).

خطة تدريس موضوعات الوحدة حسب منهجية التعلُّم القائم على المشروع

البرنامج / المهارة	المنتج	الدرس
تطبيق Google Doc.	تحديد الفكرة الرئيسة للمشروع الريادي الرقمي الذي يتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة، ثم إعداد خطة مفصلة للمشروع، تشمل تحديد الأهداف، والأدوات اللازمة، وتوزيع الأدوار، ووضع جدول زمني مفصل لتنظيم جميع الخطوات اللازمة لتنفيذ المشروع بفاعلية.	الأول: الحوسبة الخضراء
برنامج (Canva).	استكمال المرحلة التخطيطية، والتحضير لتنفيذ المشروع الريادي الرقمي الذي يتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة؛ بتجميع الموارد اللازمة.	الثاني: النفايات الإلكترونية
برنامج (Canva).	إنتاج المحتوى الرئيس للمشروع الريادي الرقمي الذي يتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة وفقاً للخطة الموضوعية، باستخدام أحد تطبيقات الحاسوب، ومراجعته ونشره في العالم الرقمي.	الثالث: تطبيقات الحاسوب في الحياة

الحوسبة الخضراء (Green Computing)

عدد الحصص المقترحة:
حصتان

المصادر والمراجع:

كتاب الطالب، تطبيق Google Docs،
تطبيق Padlet

التجهيزات

قصاصات ورقية لاصقة (Sticky Notes)،
جهاز حاسوب، أوراق بيضاء، أقلام ملونة.
تطبيق Mentimeter.



الدرس الأول

الحوسبة الخضراء (Green Computing)

الفكرة الرئيسية:

تعرف مفهوم الحوسبة الخضراء، وبيان أهميتها في الحياة، والإسهام في تطبيقها عملياً.

المفاهيم والمصطلحات:

الحوسبة الخضراء (Green Computing)، نجمة الطاقة (Energy Star).

نتائج التعلم (Learning Outcomes):

- أعرف مفهوم الحوسبة الخضراء.
- أبين أهمية الحوسبة الخضراء وفوائدها ومزاياها.
- أسهم في تطبيق الحوسبة الخضراء في حياتي اليومية.

أصبحت وسائل التكنولوجيا جزءاً لا يتجزأ من حياتنا اليومية، وغلب استخدام أدواتها على معظم أنشطتنا وممارساتنا الحياتية؛ فهل يمكن للتكنولوجيا بأدواتها ووسائلها أن تكون ضارة أو تؤثر سلباً في البيئة؟

مُنتجات التعلم (Learning Products)

تحديد الفكرة الرئيسية للمشروع الريادي الرقمي الذي يتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة، ثم إعداد خطة مفضلة للمشروع، تشمل تحديد الأهداف، والأدوات اللازمة، وتوزيع الأدوار، ووضع جدول زمني مفضل لتنظيم جميع الخطوات اللازمة لتنفيذ المشروع بفاعلية.

144

أولاً: التهيئة

سياق التعلم:

- أوضح الفكرة الرئيسية للطلبة، ومن ثم أضع الدرس في سياق تعلم، يضعهم في دور الرياديين والناشطين المجتمعين. وسأكلّفهم اقتراح مشروعات ريادية وتطويرها، تهدف إلى تعزيز وعي المجتمع بالقضايا البيئية والاجتماعية المتعلقة بالحوسبة وتنميتها. وسأشرح لهم أن رحلة تطوير فكرتهم الريادية، ستشمل عدداً من محطات التعلم التي قد تفتح أمامهم الآفاق، وتلهمهم بأفكار مميزة. ستكون محطتهم الأولى التعرف إلى مفهوم الحوسبة الخضراء، وأهميتها، والعوامل التي تؤثر فيها، بالإضافة إلى استكشاف طرائق ومعيقات تطبيقها في المجتمع.
- أبين للطلبة مُنتج التعلم الذي يُتوقع أن يُنجزوه في هذا الدرس، وهو المرحلة الأولى من المشروع الريادي الذي يُفترض اكتمال مراحلها مع نهاية هذه الوحدة؛ إذ سيكون بإمكانهم بعد هذا الدرس تحديد الفكرة الرئيسية للمشروع الريادي الرقمي الذي يتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة، ومن ثم وضع خطة، وتحديد الأهداف، والأدوات اللازمة للتنفيذ.

نشاط تمهيدي:

- أهَيُّ الطلبة وأكْلَفهم ذكر الأدوات والأجهزة الإلكترونية التي يستخدمونها في حياتهم اليومية، مع ترتيبها بحسب الأهمية.
- أجري عصفًا ذهنيًا عن هذه الاستخدامات للأدوات والأجهزة الإلكترونية: "هل تعدّ مصدرًا للخطر بحيث تؤثر سلبًا في البيئة؟"

- أوزع الطلبة في مجموعات عمل غير متجانسة، وأكْلَفهم مناقشة السؤال الآتي: "هل يمكن لاستخدام أجهزة الحاسوب والإلكترونيات المختلفة في أنشطتي اليومية أن تسهم في زيادة التلوث البيئي؟"
- أكْلَف أحد أفراد المجموعات الأولى عرض ما توصلوا إليه من نتائج لنقاشهم، ومن ثم أنتقل للمجموعة التالية، وأكمل إلى حين انتهاء جميع المجموعات من عرض آرائهم.

- أناقش الطلبة في إجاباتهم، وأوضح لهم ما يترتب على استخدام الأدوات التكنولوجية والحواسيب من أضرار؛ مما يتسبب بتفاقم مشكلة التلوث البيئي.
- أعرض للطلبة القراءات والأرقام التي تعبّر عن استهلاك الطاقة، وكم يُنفق سنويًا على تشغيل هذه الأجهزة، وما يترتب من آثار سلبية في الطاقة المستهلكة، وكيف تتسبب في انبعاث غاز ثاني أكسيد الكربون.

ستتنوع الإجابات؛ إذ من المتوقع أن يركز الطلبة على جانب سوء الاستخدام، وما يترتب عليه مثل: استهلاك الكهرباء، وإضاعة الوقت والجهد من دون الإشارة إلى أن الحواسيب التي يمكن أن تسبب ضررًا للبيئة. والإجابة عن هذا السؤال بصورة أولية، نعم يمكن لاستخدام أجهزة الحاسوب والإلكترونيات المختلفة في الأنشطة اليومية أن يسهم في زيادة التلوث البيئي بطرائق عدة:

- استهلاك الطاقة: تستهلك الحواسيب والأجهزة الإلكترونية الأخرى كميات كبيرة من الطاقة والتي يتم توليدها غالبًا من مصادر غير متجددة، مثل الفحم والنفط. ويؤدي هذا إلى انبعاث الغازات الدفيئة التي



نشاط
تمهيدي

هل يُمكن لاستخدام أجهزة الحاسوب والإلكترونيات المختلفة في أنشطتي اليومية أن يسهم في زيادة التلوث البيئي؟ كيف يُمكن لأنشطتي الحياتية أن تزيد من نسب التلوث البيئي؟ سأفكر في هاتين المسألتين، ثم أشارك زملائي / زميلاتي في أفكارتي.

الحوسبة الخضراء: تعريفها، وأهميتها (Green Computing: Definition and Importance).

يسود اعتقاد بين الناس أن أجهزة الحاسوب لا تضرّ بالبيئة، وأنّها تستهلك كمّيات قليلة من الطاقة. وهذا الاعتقاد غير صحيح؛ فهي قد تلحق ضررًا كبيرًا بالبيئة، وتُضاعف من مشكلة التلوث البيئي؛ إذ أشارت بعض الدراسات إلى أنه من بين 250 مليار دولار تُنفق سنويًا على تشغيل أجهزة الحاسوب في مختلف أنحاء العالم ما نسبته 15٪ فقط من الطاقة هو الذي يُستهلك في العمليات الحاسوبية الفعلية، في حين تُهدر بقية الطاقة أثناء عدم استخدام أجهزة الحاسوب، وتركها في وضع التشغيل. ولا شك في أن هذه الطاقة المُستهلكة تُعدّ سببًا رئيسًا لانبعاثات غاز ثاني أكسيد الكربون. ومن ثمّ، فإنّ الطاقة المُدخّرة في أجهزة الحاسوب وعمليات الحوسبة تؤدّي - في حال استخدامها - إلى تلويث البيئة بأطنان من انبعاثات الكربون سنويًا.



تسهم في التغير المناخي.

■ النفايات الإلكترونية: عند التخلص من الأجهزة الإلكترونية القديمة أو التالفة، يمكن أن تصبح نفايات إلكترونية. هذه النفايات تحتوي على مواد سامة، مثل الرصاص والزئبق، ويمكن أن تلوث التربة والمياه إذا لم تُعالج على نحو صحيح.

■ استهلاك الموارد الطبيعية: يتطلب تصنيع الأجهزة الإلكترونية استخدام كميات كبيرة من الموارد الطبيعية، مثل المعادن النادرة، التي تُستخرج في عمليات يمكن أن تكون ضارة بالبيئة.

■ التلوث عن طريق الانبعاثات: يمكن أن يؤدي التصنيع، والنقل، والتخلص من الأجهزة



هل يُمكن لاستخدام أجهزة الحاسوب والإلكترونيات المختلفة في أنشطتي اليومية أن يُسهم في زيادة التلوث البيئي؟ كيف يُمكن لأنشطتي الحياتية أن تزيد من نسب التلوث البيئي؟ سأفكر في هاتين المسألتين، ثم أشارك زملائي/ زميلاتي في أفكار.

الحوسبة الخضراء: تعريفها، وأهميتها (Green Computing: Definition and Importance)

يسود اعتقاد بين الناس أن أجهزة الحاسوب لا تضر بالبيئة، وأنها تستهلك كميات قليلة من الطاقة. وهذا الاعتقاد غير صحيح؛ فهي قد تلحق ضرراً كبيراً بالبيئة، وتضاعف من مشكلة التلوث البيئي؛ إذ أشارت بعض الدراسات إلى أنه من بين 250 مليار دولار تُنفق سنوياً على تشغيل أجهزة الحاسوب في مختلف أنحاء العالم ما نسبته 15% فقط من الطاقة هو الذي يُستهلك في العمليات الحاسوبية الفعلية، في حين تُهدر بقية الطاقة أثناء عدم استخدام أجهزة الحاسوب، وتركها في وضع التشغيل. ولا شك في أن هذه الطاقة المُستهلكة تُعد سبباً رئيساً لانبعاثات غاز ثاني أكسيد الكربون. ومن ثم، فإن الطاقة المُدخّرة في أجهزة الحاسوب وعمليات الحوسبة تؤدي - في حال استخدامها - إلى تلوث البيئة بأطنان من انبعاثات الكربون سنوياً.



145

الإلكترونية إلى انبعاث مواد كيميائية ضارة في الهواء والماء.

استخدام الأشكال والصور

■ أوجّه الطلبة إلى تأمل الصورة، وتعرّف الإشارات الظاهرة التي يمكن أن تصادفهم في حياتهم اليومية، وأناقشهم في مفهومها؛ للتوصل إلى أنها تشير إلى مفهوم التكنولوجيا الخضراء. ستتوقع الإجابات، ويمكن التركيز على كون الصورة تتضمن بعضاً من الرموز والإشارات الخضراء التي تصادفهم في حياتهم اليومية. هذه الرموز تمثل جوانب مختلفة من "التكنولوجيا الخضراء"، وهي تقنيات تهدف إلى تحسين الاستدامة البيئية. ويمكن للطلبة عن طريق التعرف إلى هذه الرموز أن يدركوا الأهمية المتزايدة للتكنولوجيا الخضراء في حياتنا اليومية. على سبيل المثال، يشير رمز إعادة التدوير إلى أهمية إعادة استخدام الموارد لتقليل النفايات، ويشير رمز الطاقة النظيفة مثل طواحين الهواء إلى استخدام مصادر الطاقة المتجددة لتقليل التأثير البيئي.

ثانيًا: عملية التعليم والتعلم:

بناء المفهوم:

- أناقش الطلبة في مفهوم الحوسبة الخضراء، وكيفية الحد من تأثيرها السلبي في تكنولوجيا المعلومات.
- أتيح المجال للطلبة للتعبير، وأناقشهم في إجاباتهم، وأوجه النقاش لتعريف الحوسبة الخضراء، والتركيز على الاستخدام المسؤول لتكنولوجيا المعلومات.

تعريف الحوسبة الخضراء

تُعرف الحوسبة الخضراء بأنها الاستخدام البيئي المسؤول لأجهزة الحاسوب والموارد التكنولوجية ذات الصلة، الذي يحد من التأثير السلبي لتكنولوجيا المعلومات والاتصالات في البيئة. وتحقيقًا لهذا الهدف؛ تُستخدم أفضل الطرائق والوسائل في تصميم أجهزة الحاسوب والخوادم، وتصنيعها، وإعادة تدويرها؛ ما يُقلل من آثارها الضارة بالبيئة. يُطلق على الحوسبة الخضراء أيضًا اسم التكنولوجيا الخضراء (Green IT)، أو التكنولوجيا المستدامة (Sustainable IT).

أهمية الحوسبة الخضراء

تُسهم الحوسبة الخضراء في تقليل استهلاك الطاقة، وتحد من انتشار النفايات الإلكترونية؛ ما يُفضي إلى خفض التكاليف التشغيلية، وتعزيز مبدأ الاستدامة البيئية. ولهذا تعمل الحوسبة الخضراء على تحسين كفاءة الطاقة، واستخدام مصادر الطاقة المتجددة، وتدوير النفايات الإلكترونية. وهي تهدف إلى الحد من نسب الانبعاثات الكربونية الناجمة عن استخدام تكنولوجيا المعلومات؛ ما يسهم في حماية البيئة، وإنتاج تكنولوجيا نظيفة ومستدامة وصديقة للبيئة.

يُمكن إجمال العوامل الرئيسة التي تحكم عمل الحوسبة الخضراء في ما يأتي:



1. كفاءة الطاقة (Energy Efficiency): يتمثل ذلك في تصنيع أنظمة لتكنولوجيا المعلومات موفرة للطاقة الكهربائية، مثل: الأجهزة الحاصلة على تصنيف نجمة الطاقة (Energy Star)، ومصادر الطاقة المتجددة، والبرمجيات التي تستهلك قليلاً من الطاقة.

التأمل والتحليل والاستنتاج

- أوزع الطلبة إلى مجموعات عمل غير متجانسة، وأكلف كل مجموعة المناقشة في ما بينهم، ومحاولة تلخيص أهمية الحوسبة الخضراء، وتوقع أهدافها.
- أكلف كل مجموعة انتداب شخص لعرض نتائج نقاشهم.
- أناقش الطلبة في ما توصلوا له، وأجمل النقاش مع التركيز على أهم النقاط ذات الصلة بأهمية الحوسبة الخضراء والهدف منها.
- تسهم الحوسبة الخضراء في تقليل استهلاك الطاقة، وتحد من انتشار النفايات الإلكترونية، وخفض التكاليف التشغيلية، وتعزيز مبدأ الاستدامة البيئية.
- أهداف الحوسبة الخضراء: تحسين كفاءة الطاقة، واستخدام مصادر الطاقة المتجددة، وتدوير النفايات الإلكترونية، والحد من نسب الانبعاثات الإلكترونية؛ مما يسهم في حماية البيئة، وإنتاج تكنولوجيا نظيفة ومستدامة وصديقة للبيئة.

مناقشة:

■ أكلّف كل مجموعة دراسة عاملين من العوامل الرئيسة التي تحكم عمل الحوسبة الخضراء، ومن ثم إتاحة المجال لكل مجموعة لعرض هذه العوامل، وبيان المقصود منها.

■ ستتّوَع إجابات الطلبة. وهنا يمكن التركيز على ما ورد في صفحة 147 فقط.

أخطاء شائعة

يعتقد معظم الطلبة أن الحاسوب لا يضرّ البيئة، ولا توجد نفايات إلكترونية. تُوضّح هذه النقطة عن طريق النشاط التمهيدي والتعرف إلى أضرار الحواسيب والتكنولوجيا على البيئة من حولنا. أذكر الطلبة أن الدرس الثاني، يتناول موضوع النفايات الإلكترونية بالتفصيل.

2. ترشيد الموارد (Resource Reduction): يتمثل ذلك في تقليل استخدام المواد الخطرة والموارد غير المتجددة، وتعزيز فكرة إعادة التدوير.
3. افتراضية الخوادم (Virtualization Servers): يتمثل ذلك في استعمال خادم مادي واحد لإدارة أنظمة التشغيل المتعددة واستخدام التجزئة الافتراضية لتشغيل مجموعة متنوعة من التطبيقات على الخادم نفسه وتقليل عدد الخوادم في مراكز البيانات، ما يؤدي إلى ترشيد استهلاك الطاقة.
4. الحوسبة السحابية (Cloud Computing): يتمثل ذلك في استخدام الموارد المشتركة في مراكز البيانات المركزية على السحابة الإلكترونية؛ ما يُوفّر كثيرًا من الطاقة مقارنة باستخدام الخوادم الفردية ومراكز البيانات الفعلية.
5. تصميم مراكز البيانات (Data Center Design): يتمثل ذلك في تصنيع أنظمة تبريد موفّرة للطاقة، وإعداد ترتيبات أفضل للخوادم، وتوزيع الطاقة على نحو يحدّ من استهلاكها.
6. إدارة النفايات الإلكترونية (E-Waste Management): يتمثل ذلك في التخلص الآمن من النفايات الإلكترونية، وإعادة تدويرها؛ ما يحول دون تلويث البيئة بالمُكوّنات الخطرة، مثل المعادن الثقيلة.
7. المشتريات المستدامة لتكنولوجيا المعلومات (Sustainable IT Procurement): يتمثل ذلك في شراء الأنظمة التكنولوجية الصديقة للبيئة، مثل: الأجهزة الموفّرة الطاقة، والأجهزة التي تحوي على القليل من المُكوّنات الخطرة.
8. العمل عن بُعد، والتعاون الافتراضي (Telecommuting and Virtual Collaboration): يتمثل ذلك في تقليص عمليات السفر والتنقل؛ ما يُفضي إلى تحجيم البصمة الكربونية وخفضها.
9. كفاءة البرمجيات (Software Efficiency): يتمثل ذلك في ابتكار حلول برمجية تُستخدم فيها الموارد بأكثر الطرائق فاعلية.
10. دعم مصادر الطاقة المتجددة (Promotion of Renewable Energy Sources): يتمثل ذلك في استخدام موارد الطاقة المتجددة، والإفادة منها في تشغيل البنية التحتية لتكنولوجيا المعلومات.

طرائق تطبيق الحوسبة الخضراء

تتعدّد أوجه تطبيق الحوسبة الخضراء، ويُمكن إجمالها في ثلاثة مستويات؛ الأوّل يُمثّله الأفراد، والثاني يُمثّله المجتمع، والثالث يُمثّله المؤسسات والشركات. وفي ما يأتي بيان لذلك:

مناقشة

- أوضح للطلبة أن هناك مستويات لتطبيق الحوسبة الخضراء، وأنه يمكن إجمالها في ثلاثة مستويات: الأفراد، والمجتمع، والمؤسسات، والشركات.
- أجهز رابطاً على منصة بادلت، وأشاركه مع الطلبة.
- أطرح على الطلبة السؤال الآتي: كيف يمكن أن يسهم الفرد في تحسين بيئة التكنولوجيا المستدامة؟ وأوجّه الطلبة لكتابة أفكارهم على المنصة.
- أناقش مع الطلبة أهم البنود ذات الصلة بتطبيق الحوسبة الخضراء على مستوى الفرد التي كتبوها، وألخص معهم البنود على شكل خريطة مفاهيمية..

1. طرق تطبيق الحوسبة الخضراء على مستوى الأفراد:

يُمكن للأفراد الإسهام في تحسين بيئة التكنولوجيا المستدامة عالمياً باتّباع الخطوات الآتية:



- أ. إطفاء الأجهزة غير المستخدمة: يجب فصل أجهزة الحاسوب والأجهزة الإلكترونية عن مصدر التيار الكهربائي بعد الانتهاء من استخدامها؛ للحد من استهلاك الطاقة.
- ب. ضبط الأجهزة على وضع السكون: يجب برمجة الأجهزة على وضع السكون في حال عدم استخدامها مُدَّةً طويلة؛ ما يُسهم في ترشيد استهلاك الطاقة.
- ج. ضبط إعدادات الطاقة: يجب اختيار الوضع المُوفِّر للطاقة في الأجهزة الإلكترونية عند ضبط إعدادات الطاقة فيها؛ ما يحول دون استهلاك كثير من الطاقة.
- د. استخدام الأجهزة الموفرة للطاقة: يُفضَّل شراء الأجهزة التي تحمل ملصق نجمة الطاقة (Energy Star)، وتستهلك طاقة أقل، وتُحافظ - في الوقت نفسه - على الطاقة المُستخدمة بكفاءة عالية.
- هـ. إعادة التدوير: يجب التخلص من الأجهزة الإلكترونية التالفة بصورة آمنة وصحيحة، تتمثل في إعادة التدوير؛ ما يُقلِّل من انتشار النفايات الإلكترونية، ويحدُّ من تلوث البيئة.
- و. التقليل من عمليات الطباعة وإعادة تعبئة أحبارها: يُنصَح بالطباعة على وجهي الورقة، وتصغير حجم الخط عند الطباعة؛ ما يُرشِّد استهلاك الورق والحبر، علماً بأنَّ إعادة تعبئة حبر الطابعة أفضل من شراء القطعة الخاصة بذلك (Cartridges) في تطبيق الحوسبة الخضراء.
- ز. تخطيط عمليات شراء لأجهزة الحاسوب والأجهزة الإلكترونية: يجب التفكير ملياً قبل شراء أجهزة الحاسوب والأجهزة الإلكترونية، والتأكُّد أنَّها تُناسب طبيعة الاستخدام، وتفي بالغرض المنشود.

مناقشة

■ أوزع الطلبة في مجموعات عمل غير متجانسة، وأخصص لكل مجموعة إحدى طرائق تطبيق الحوسبة الخضراء على مستوى المجتمع.

■ أكلف كل مجموعة مناقشة الطريقة المخصصة لهم، وكتابة أفكار ومشاركتها على منصة بادلت، ومن ثم مناقشتها مع الزملاء.

■ أثنى على مشاركات المجموعات، وأركز على الطرائق المتبعة في تطبيق الحوسبة الخضراء، وألخص الأفكار الرئيسة.



أجمع بعض المعلومات عن المراكز المحلية لإعادة تدوير الإلكترونيات، ثم أنظم حملة لجمع الأجهزة الإلكترونية القديمة من الطلبة ومرافق المدرسة، وإرسالها إلى مراكز إعادة التدوير.

إضاءة



نجمة الطاقة (Energy Star): برنامج حكومي أطلقته الولايات المتحدة الأمريكية عام 1992م بوساطة وكالة حماية البيئة (EPA) ووزارة الطاقة (DOE)؛ لتعزيز الكفاءة في استخدام الطاقة، والحد من آثارها الضارة بالبيئة.

تضع الشركات الصانعة ملصق نجمة الطاقة (Energy Star) على منتجاتها بعد الوفاء بالمعايير والضوابط الصارمة بهذا الخصوص؛ للدلالة على أن هذه المنتجات تستهلك طاقة أقل، وتسهم في حماية البيئة من التلوث. تشمل المنتجات الحاصلة على هذا الاعتماد كلاً من أجهزة الحاسوب، والشاشات، والأجهزة المنزلية، وأنظمة الإضاءة، وما شابه.

2. طرائق تطبيق الحوسبة الخضراء على مستوى المجتمع: يُمكن تطبيق الحوسبة الخضراء على مستوى المجتمع بالتباعد الخطوات الآتية:

ب تشجيع السياسات البيئية

أ التوعية والتثقيف

ج التعاون مع المنظمات البيئية

نشاط جماعي :

■ أطلب إلى الطلبة العمل في مجموعات صغيرة للتعاون على تصميم ملصقات توعوية، تهدف إلى نشر الوعي عن جانب من جوانب الحوسبة الخضراء، مثل توفير الطاقة، أو إدارة النفايات الإلكترونية.

■ على كل مجموعة أن تختار موضوعاً محدداً من موضوعات الحوسبة الخضراء. ويمكن للطلبة البحث عن معلومات حول الموضوع الذي اختاروه؛ للتأكد من دقة المعلومات التي ستُضمّن في الملصق.

■ أوجّه الطلبة إلى استخدام برامج تصميم ملصقات مثل Canva، أو أي برامج تصميم أخرى مناسبة. أشجعهم على اختيار الألوان والتصاميم التي تعبر تعبيراً فعالاً عن الرسالة البيئية.

- أوجّه الطلبة للتأكد من أن يتضمن الملصق رسالة واضحة ومباشرة عن كيفية تبني ممارسات الحوسبة الخضراء، مع تقديم نصائح عملية يمكن للآخرين اتباعها في حياتهم اليومية.
- أطلب إلى الطلبة مراجعة الملصقات معاً، وتقديم ملحوظاتهم بعضهم إلى بعض؛ لتحسين التصميم والمحتوى، ثم أراجع الملصقات النهائية؛ لضمان جودتها.
- بعد الانتهاء من التصميم، أوجه الطلبة إلى تعليق الملصقات في أنحاء المدرسة، مثل الممرات، والمكتبة، والمقصف؛ بهدف زيادة الوعي الطلبة والمجتمع المدرسي عن أهمية الحوسبة الخضراء.

- أ. التوعية والتثقيف: يُقصد بذلك نشر الوعي بأهمية الحوسبة الخضراء وفوائدها؛ ما يُحفّز المجتمع على اتخاذ خطوات فاعلة للحدّ من الآثار التكنولوجية الضارّة بالبيئة.
- ب. تشجيع السياسات البيئية: يتمثل ذلك في دعم الأنظمة والتشريعات التي تُعزّز استخدام التكنولوجيا الصديقة للبيئة، مثل القوانين الضريبية المُحفّزة للشركات التي تنتهج مبادئ الحوسبة الخضراء في أعمالها وأنشطتها.
- ج. التعاون مع المنظّمات البيئية: تحرص المؤسسات والشركات والدوائر الحكومية والخاصة على العمل مع المنظّمات التي تهتمّ بالبيئة، وتُسهّل سُبل تطبيق مبادرات الحوسبة الخضراء؛ ما يُعزّز الجهود المشتركة لتحقيق أهداف التنمية البيئية المستدامة.

تصميم ملصقات إرشادية لأحد تطبيقات الحوسبة الخضراء.

أنشر - بالتعاون مع أفراد مجموعتي - الوعي بأهمية الحوسبة الخضراء في المجتمع، وأعمل على تصميم ملصقات إرشادية لأحد تطبيقات الحوسبة الخضراء (مثل: توفير الطاقة، وإدارة النفايات الإلكترونية) باستخدام برنامج (Canva)، أو غيره من برامج التصميم الخاصة بإنشاء الملصقات، ثم أعلّق الملصقات في مختلف مرافق المدرسة؛ سعياً لزيادة وعي الطلبة ومجتمع المدرسة بأهمية الحوسبة الخضراء.

3. طرائق تطبيق الحوسبة الخضراء على مستوى المؤسسات والشركات: يُمكن للمؤسسات والشركات أن تُسهّم في تطبيق الحوسبة الخضراء بالتزام جملة من الإجراءات، أبرزها:



مناقشة

■ أترح على الطلبة السؤال الآتي: كيف يمكن للمؤسسات أن تسهم في تطبيق الحوسبة الخضراء؟

■ أتيح المجال للطلبة للتعبير عن أفكارهم ومشاركتها عبر منصة بادلت.

■ أعرض مشاركات الطلبة، وأناقشها معهم، ثم أخص أهم البنود ذات الصلة بتطبيق الحوسبة الخضراء على مستوى المؤسسات، مع التركيز على تأكيد تبعات الالتزام بمثل هذه الخطوات في تحسين البيئة التكنولوجية الخضراء، وما تحقّقه من أهداف الحوسبة الخضراء.

■ أتيح المجال للطلبة؛ لإعطاء أمثلة من المجتمع، تعدّ تطبيقاً

للحوسبة الخضراء على مستوى الشركات. حملة شركة زين في تعزيز الحوسبة الخضراء، ومنصة الحوسبة السحابية للحكومة الأردنية (<https://www.oracle.com/jo-ar>).



يمكن توجيه الطلبة لزيارة موقع نجمة الطاقة الذي يقدم بعض المعلومات عن الأجهزة واستهلاكها للطاقة (<https://www.energystar.gov/products/computers>).

- تصميم الأجهزة بكفاءة عالية: يؤدي اعتماد مواصفات خاصة في تصميم أجهزة الحاسوب إلى ترشيد استهلاك الطاقة، وضمان عمل الأجهزة مددًا طويلاً؛ ما يزيد من أمد التحديث المستمر، ويُخفّض استهلاك الموارد بصورة كبيرة.
- استخدام الطاقة المُتجدّدة: يؤدي استعمال مراكز البيانات لمصادر الطاقة المُتجدّدة إلى تقليل الاعتماد على الوقود الأحفوري، وخفض نسب الانبعاثات الكربونية.
- إدارة النفايات الإلكترونية: يتمثل ذلك في سنّ تشريعات تُعزّز إعادة تدوير الأجهزة الإلكترونية، وإتلافها بصورة صحيحة تُحدّ من أثارها السلبية في البيئة، إضافة إلى استخدام المواد الخام المستدامة والقابلة لإعادة التدوير.
- افتراضية الخوادم: يُقصد بذلك تخصيص خوادم افتراضية مُتعدّدة لجهاز واحد؛ سعياً لتقليل استهلاك الطاقة والمساحة.
- تحسين كفاءة البرمجيات: يتمثل ذلك في تطوير برمجيات تستهلك طاقة أقل، وتعمل بكفاءة أكثر، فضلاً عن إطالة عمر الأجهزة؛ لتقليل الحاجة إلى استبدالها المُتكرّر.
- تصميم مبانٍ خضراء ومستدامة: يُسهم التصميم الجيد للمؤسسات والمباني في ترشيد استهلاك الطاقة، ويتمثل ذلك في اعتماد أنظمة حديثة للتدفئة والتبريد والتهوية، تتضمن استخدام ممرّات باردة أو ممرّات ساخنة بحسب الحاجة.
- شراء الأجهزة والمعدّات التي تُرشّد استهلاك الطاقة: يتمثل ذلك في شراء أجهزة حاسوب مُوفّرة للطاقة، مثل أجهزة الحاسوب المحمولة (Laptop) التي تستهلك طاقة أقل ممّا تستهلكه الأجهزة المكتبية (Desktop).



أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن أنواع أجهزة الحاسوب المختلفة، وأقارن بينها من حيث مُعدّل استهلاكها للطاقة في الساعة الواحدة، ثمّ أشارك زملائي / زميلاتي ومُعلّمي / مُعلّمتي في النتائج التي أتوصّل إليها.

مُعوّقات تطبيق الحوسبة الخضراء

تواجه الحوسبة الخضراء تحديات عديدة تُؤثّر سلباً في تطبيقها، وتتمثل أبرزها في ما يأتي:

1. التكلفة العالية: تُحجّم بعض الشركات عن انتهاز طريق الحوسبة الخضراء بسبب التكلفة المادية المرتفعة بالنسبة إليها. فقد يتطلّب استخدام الأجهزة والتكنولوجيا المُوفّرة للطاقة وجود استثمارات أولية ضخمة؛ ما يُمثّل عائقاً أمام الشركات والأفراد.

نشاط بحثي

- أناقش الطلبة في أنواع الحواسيب التي يفتنونها ونوعيتها، وأيها تعدّ صديقة للبيئة.
- أكلف الطلبة البحث -في واجب فرديّ بيتيّ- عن أنواع أجهزة الحاسوب المختلفة، ومن ثم المقارنة بينها من حيث معدل استهلاكها للطاقة في الساعة الواحدة.
- أوفق مع الطلبة على موعد لعرض نتائج البحث أمام الطلبة، أو مشاركته عبر منصة بادلت.

مناقشة

- **أناقش الطلبة في المعوقات والتحديات التي تواجه تطبيق الحوسبة الخضراء، وآثارها السلبية.**
- **أقسّم اللوح إلى عمودين، وأضع عنواناً للعمود الأول (المعوقات والتحديات)، وعنواناً (الحلول) للعمود الثاني، وأطلب إلى الطلبة كتابة أفكارهم على قصاصات ورقية، وإصاقها في المكان الصحيح على اللوح.**
- **أناقش الطلبة في آرائهم واقتراحاتهم.**
- **أناقش الطلبة في تطبيق الحوسبة الخضراء في الأردن، والإجراءات المتبعة من الدولة لتدعيم تطبيق الحوسبة الخضراء، ومواجهة تحديات التطبيق، والتغلب عليها.**

2. التدريب: يتطلّب تطبيق مبادئ الحوسبة الخضراء تدريب الموظفين على كيفية استخدام التكنولوجيا الخضراء؛ ما يُرهق كاهل بعض المؤسسات والشركات.
3. التحديث المستمر: يتعيّن على المؤسسات والشركات - في ظلّ التطوّر السريع للتكنولوجيا- متابعة آخر التحديثات والتقنيات الجديدة في مجال الأجهزة الإلكترونية وأجهزة الحاسوب؛ لضمان ديمومة ترشيد الاستهلاك في الطاقة. وهذا يُحتّم عليها الاستغناء عن الأجهزة القديمة التي لديها، وشراء أجهزة جديدة؛ ما يُمثّل تحدّيًا رئيسًا لها.
4. البنية التحتية: قد تكون البنية التحتية القائمة غير مُلائمة لتطبيق مبادئ الحوسبة الخضراء؛ ما يُحتّم على المؤسسات والشركات إدخال كثير من التعديلات والتحديثات الإضافية.
5. ثقافة الوعي البيئي: يُعدّ الجهل بأهمية الحوسبة الخضراء أحد أبرز التحديات التي تحدّ من تطبيق مبادئ الحوسبة الخضراء؛ إذ لا تحفل كثير من المؤسسات والشركات بالمشكلات البيئية (مثل التغيّر المناخي) عند تصنيع الأجهزة التكنولوجية أو شرائها.

تطبيق الحوسبة الخضراء في الأردن

- يبدّل الأردن كثيرًا من الجهود الدؤوبة لتطبيق مبادئ الحوسبة الخضراء في مختلف المؤسسات والوزارات الحكومية، مثل: وزارة الاقتصاد الرقمي والريادة، ووزارة البيئة، ووزارة الطاقة والثروة المعدنية. كذلك تبنت العديد من الشركات في القطاع الخاص فكرة الحوسبة الخضراء، وأخذت تُعَدُّ الخُطى نحوها؛ سعيًا لتحسين كفاءة مواردها، وتقليل انبعاثات غاز الكربون، وتوفير الطاقة. وقد تمثّل ذلك في اتّخاذ العديد من الإجراءات والمبادرات التي تدعم الاستدامة البيئية، مثل:
1. مبادرات التوعية: يتمثّل ذلك في تنظيم حملات توعية تُعرّف الناس بأهمية الحوسبة الخضراء وفوائدها.
 2. التشريعات والسياسات: يتمثّل ذلك في وضع القوانين والتشريعات التي تُعزّز استخدام التكنولوجيا الصديقة للبيئة، وتُحفّز المؤسسات والشركات على تطبيق مبادئ الحوسبة الخضراء، بما تُقدّمه لها من حوافز وتسهيلات.
 3. الاستثمار في الطاقة المُتجدّدة: يتمثّل ذلك في تنفيذ مشروعات الطاقة الشمسية التي تُزوّد مراكز البيانات بحاجتها من الطاقة، وتُخفّف العبء والضغط على الشبكة التقليدية للطاقة.
 4. التعاون مع الشركات: يتمثّل ذلك في تحفيز الشركات على تبني مبادئ الحوسبة الخضراء، عن طريق تقديم الحوافز المالية والتقنية لها؛ ما يُسهّم في تعزيز الاستدامة البيئية، ويحدّ من تبّعات البصمة الكربونية.

إضاءة:

- ألفت انتباه الطلبة إلى مربع الإضاءة، وأناقشهم في البرمجة الخضراء، ومفهومها، ودورها في المحافظة على البيئة مثل (استهلاكها طاقة أقل، وتحسين استخدام البيانات، وتقليل النفايات الإلكترونية).
المستند الذي أعده مع بقية المجموعات عبر البريد الإلكتروني.

بحث ومشاركة

نشاط فردي بيتي

- أطلب إلى الطلبة البحث عبر الإنترنت عن تقنيات وممارسات الترميز الأخضر التي يمكن أن يتبعها المطورون للبرمجيات.
- أطلب إلى الطلبة تقديم أمثلة محددة على ممارسات الترميز الأخضر التي يمكن تنفيذها.

إضاءة:

ظهر في الآونة الأخيرة مصطلح يُسمّى الترميز الأخضر أو البرمجة الخضراء (Green Coding)، ويُقصد به اعتماد التعليمات البرمجية (تُعرف أيضًا باسم البرامج) التي تُسهّم في المحافظة على البيئة، ولا تُلحق ضررًا كبيرًا بها. ويتمثل ذلك في اعتماد تعليمات برمجية فاعلة تستهلك طاقة أقل، وتحسين استخدام البيانات، وتقليل النفايات الإلكترونية.



أبحث وأشارك:

توجد تقنيات عديدة يُمكن للمُطوّرين استخدامها في تنفيذ الأسس التي يقوم عليها الترميز الأخضر. أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن تلك التقنيات، وعن بعض الأمثلة التي تُعزّز هذا الجانب، ويُمكن تنفيذها للحدّ من الآثار التكنولوجية الضارة بالبيئة، ثمّ أشارك زملائي / زميلاتي ومُعَلّمي / مُعَلّمتي في النتائج التي أتوصّل إليها.



نشاط



153

- أتأكد من أن الطلبة يدونون المصادر التي استخدموها في بحثهم.
- بعد جمع المعلومات، أطلب إلى كل الطلبة إعداد تقرير مختصر (1-2 صفحة)، يلخص النتائج التي توصلوا إليها، مع تضمين أمثلة وتقنيات مذكورة.
- يجب أن يتضمن التقرير أيضًا أهمية ممارسات الترميز الأخضر، وتأثيرها الإيجابي في البيئة.
- أتفق مع الطلبة على موعد لتقديم التقرير، ومشاركة النتائج مع زملائهم في الصف عبر منصة بادل.

ستتنوع الإجابات والمشاركات، ومن بينها: عادةً ما تُصمَّم مبادئ الترميز الأخضر لتكمّل معايير الاستدامة في تكنولوجيا المعلومات والممارسات المستخدمة في جميع أنحاء المؤسسة. ويتطلب الترميز الأخضر تغييرات هيكلية وثقافية على حد سواء..

■ **تحسين استخدام الطاقة في جوهرها:** يمكن برمجة التطبيقات القائمة على المعالجات متعددة النواة لزيادة كفاءة استخدام الطاقة. على سبيل المثال، يمكن للكود البرمجي أن يوجّه المعالجات إلى الإيقاف والتشغيل مرة أخرى خلال ميكروثانية بدلاً من استخدام إعدادات توفير الطاقة الافتراضية التي قد لا تكون فعالة بالقدر نفسه.

■ **الكفاءة في تكنولوجيا المعلومات:** يُشار إليها أحياناً باسم تكنولوجيا المعلومات الخضراء أو الحوسبة الخضراء، تهدف هذه المنهجية إلى تحسين الموارد، وتوحيد أعباء العمل لتقليل استخدام الطاقة. وعن طريق تحسين بنية تكنولوجيا المعلومات باستخدام أدوات حديثة مثل الأجهزة الافتراضية (VMS) والحاويات، يمكن للمنظمات تقليل عدد الخوادم الفعلية المطلوبة للعمليات؛ مما يقلل من استهلاك الطاقة وكثافة الكربون.

■ **الخدمات المصغرة:** تعدّ الخدمات المصغرة نهجاً شائعاً لبناء التطبيقات التي تقسم البرامج المعقدة إلى عناصر أصغر تُسمى خدمات، تُستدعى هذه الخدمات الصغيرة فقط عند الحاجة، بدلاً من تشغيل برنامج ضخم تشغيلاً كاملاً. النتيجة هي أن التطبيقات تعمل بكفاءة أكبر.

■ **DevOps القائم على السحابة:** تقلل التطبيقات التي تعمل على بنية تحتية سحابية موزعة من كمية البيانات المنقولة عبر الشبكة، وتقلل من استهلاك الطاقة الإجمالي للشبكة.

■ **تمكين الإدارة والموظفين:** يكون التغيير فعالاً فقط عندما يكون الموظفون والإدارة متعاونين ومتفهمين على ذلك. يساعد ذلك في دعم برنامج الاستدامة، ويجعل الناس يشعرون بأنهم جزء من الحل.

ثالثاً: الإثراء والتأمل والتقويم

معلومات إثرائية:



- أعرّض بعض البطاقات التي تلصق على الأجهزة الإلكترونية، وأعرف هل تتوافق مع التكنولوجيا الخضراء أم لا؟
- أطرح التساؤل: ما هي المواصفات التي يجب أن تتوفر في الأجهزة الإلكترونية؛ لتتوافق مع التكنولوجيا الخضراء؟

- أستمع إلى تفسير الطلبة للملصق، وأناقشهم في مواصفات الأجهزة التي يجب أن يختاروها صديقة للبيئة.
- رمز (A): يشير هذا الرمز إلى أن الجهاز من أفضل الأجهزة الموفرة في استهلاك الكهرباء؛ فإذا كان الجهاز غسالة ملابس مثلاً، فهذا يعني أنها تستهلك 14 كيلو وات / ساعة.
- رمز (B): يشير هذا الرمز إلى أن الجهاز في المرتبة الثانية من حيث استهلاكه للكهرباء؛ فإذا كان الجهاز ثلاجة مثلاً فهذا يعني أنها تستهلك 68 كيلو وات / ساعة.
- رمز (C): يشير هذا الرمز إلى أن الجهاز يستهلك كمية كبيرة؛ فإذا كان الجهاز غسالة، فإنها تستهلك 18 كيلو وات / ساعة.
- رمز (E): يشير هذا الرمز إلى أن الجهاز يستهلك كمية كبيرة من الكهرباء؛ فإذا كان الجهاز ثلاجة، فهذا يعني أنها تستهلك 128 كيلو وات / ساعة.

ملاحظات:

.....

.....

.....

.....

.....



أراعي ما يأتي بعد دراسة موضوع (الحوسبة الخضراء):

■ الاستخدام المسؤول للتكنولوجيا:

- أطفئ الأجهزة الإلكترونية في حال عدم استخدامها، وأفعل أوضاع توفير الطاقة فيها (مثل ضبطها على وضع السكون)، وأحدث أنظمة التشغيل والبرامج بانتظام واستمرار؛ لتحسين كفاءة هذه الأجهزة، وتقليل استهلاك الطاقة.
- أعيد تدوير الأجهزة الإلكترونية القديمة بصورة صحيحة.
- أختار الأجهزة الموفرة للطاقة، وأشتري منها فقط ما يفي بحاجاتي؛ لتقليل استهلاك الطاقة، والحد من الفضلات الإلكترونية.

■ التعليم والتوعية:

- أشارك الأصدقاء والعائلة في ما أعرفه من معلومات عن الحوسبة الخضراء.
- أتحقق نفسي والآخرين بخصوص التأثير السلبي للتكنولوجيا في البيئة، وكيف يمكن التقليل من هذا التأثير.

■ الاستخدام الذكي للتكنولوجيا:

- أحدد وقتاً لاستخدام الأجهزة الإلكترونية؛ سعياً لتقليل استهلاك الطاقة.
- أعتمد الحوسبة السحابية؛ لتقليل الحاجة إلى الأجهزة الفردية القوية.
- أشارك في المبادرات والحملات التي تُعزز الوعي بأهمية الحوسبة الخضراء والاستدامة البيئية.



- تنظيم جلسة حوارية لمناقشة المسؤولية الاجتماعية التي تقع على عاتق الطلبة في تطبيق الحوسبة الخضراء، ودورهم المسؤول في نشر التوعية ويتمثل ذلك في:

- مناقشة السلوكيات الفردية التي يجب على الطالب اتباعها لتحقيق الحوسبة الخضراء.

- مناقشة دور الطلبة في نشر التوعية وتصميم الملصقات في المدرسة.

- أنظم جلسة نقاش قصيرة مع الطلبة؛ للتعبير عن تجربتهم بعد تطبيق النشاط والتغذية الراجعة التي حصلوا عليها من ردود أفعال الطلبة عند تنفيذهم حملة التوعية.

التعليم والتوعية:

- إتاحة المجال للطلبة للتعبير عن تجربتهم، ومشاركة معلوماتهم عن الحوسبة الخضراء.
- إتاحة المجال للطلبة لنقاش التأثير السلبي للتكنولوجيا في البيئة، وكيف يمكن تقليل التأثير، والسماح لهم بتبادل الأفكار

الاستخدام الذكي للتكنولوجيا:

- يمكن تحقيق هذا المجال عن طريق تكليف الطلبة بتنفيذ مهمة دفتر اليوميات الوارد في الصفحة 157.

نشاط علاجي:

أوجّه الطلبة إلى إعداد قائمة تتضمن مجموعة من أجهزة الحاسوب والأجهزة التكنولوجية الحديثة في المنزل، وتحديد كيفية الإسهام في توعية أفراد الأسرة في تطبيق الحوسبة الخضراء عن طريق كتابة مجموعة من التعليمات وعرضها على زملاء في الصف.

التعليمات الخاصة بمُنْتَجِ التعلّم:

- أذكر الطلبة بسياق التعلّم، ودورهم بوصفهم ربايين يقودون مشروعات تهدف إلى تعزيز الوعي البيئي عن طريق مناقشة القضايا المرتبطة بالحوسبة الخضراء والنفايات الرقمية.
- أذكر الطلبة أنه حان وقت البدء بالمرحلة الأولى من المشروع الريادي الذي يُفترض اكتمال مراحله مع نهاية هذه الوحدة، وهو تحديد الفكرة الرئيسة للمشروع الريادي الرقمي الذي يتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة، ومن ثم وضع خطة، وتحديد الأهداف والأدوات اللازمة للتنفيذ.
- أوزع الطلبة في مجموعات عمل غير متجانسة، وأوجههم لقراءة تعليمات المهمة المنوطة بهم، وأقدم لهم النصح والإرشاد عند الحاجة.
- أوضح للطلبة معايير التقييم الآتية للمهمة المنشودة:



المشروع: تنفيذ مشروع ريادي رقمي يتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة، باستخدام إحدى تطبيقات الحاسوب / المهمة (1).

ضمن إطار التحضيرات لإطلاق مشروع ريادي رقمي يتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة، سأتعاون مع أفراد مجموعتي على تنفيذ المرحلة الأولى من المشروع. المرحلة الأولى: التصميم والتخطيط.

تحديد فكرة المشروع: أختار فكرة مشروع ريادي رقمي، تُركّز على أحد الموضوعات التي تتناولها الوحدة، مثل: الحوسبة الخضراء، والنفايات الإلكترونية، والبرمجة الخضراء، واستخدام تطبيقات الحاسوب في مجال الصحة، والتعليم، والاقتصاد، والحياة. بعد ذلك أجتمع مع أعضاء الفريق لمناقشة الفكرة، وتحديد الهدف الرئيس للمشروع.

وضع خطة المشروع: أستخدم في كتابة خطة المشروع تطبيقات (Office)، مثل: (Microsoft Word)، و (Google Docs)؛ على أن تتضمن الخطة ما يأتي:

- الأهداف؛ أي ما نريد تحقيقه من المشروع.
 - الأدوات اللازمة؛ أي الأجهزة والبرامج التي سنستخدمها في تنفيذ المشروع.
 - الجدول الزمني؛ أي تخصيص الوقت اللازم لتنفيذ كل مرحلة من مراحل المشروع.
 - توزيع الأدوار؛ أي تحديد المهام لكل عضو في الفريق.
- أتحقق من توثيق جميع مراحل المشروع، وأحتفظ بالملفات اللازمة لاستكمال المراحل اللاحقة.

- تحديد الفكرة الرئيسة للمشروع: يجب أن تكون الفكرة أصيلة ترتبط بتطبيق الحوسبة الخضراء، وتعلق بقضية بيئية واجتماعية، ترتبط بعلاقة مباشرة بالحوسبة الخضراء.
- إعداد خطة مفصلة للمشروع: يجب تحديد الأهداف المراد تحقيقها خلال مراحل تنفيذ المشروع، وتحديد الأدوات اللازمة للتنفيذ وتوزيع الأدوار.
- وضع جدول زمني مفصل لتنظيم خطوات المشروع: يجب ترتيب الأهداف بحسب الأولوية، وتحديد جدول زمني مفصل لها خلال فترة التنفيذ.
- كتابة خطة المشروع باستخدام برنامج (Google Docs): يجب كتابة خطة المشروع بلغة واضحة، ونصوص واضحة، وبتمييز العناوين.

استراتيجية التقويم: التقويم المعتمد على الأداء. أداة التقويم: سُلم التقدير.

المعيار	مؤشر الأداء	نعم	لا	نوعاً ما	ملاحظات
تحديد الفكرة الرئيسة للمشروع	■ الفكرة أصيلة، وترتبط ارتباطاً مباشراً بتطبيق الحوسبة الخضراء، وتتعلق بقضية بيئية أو اجتماعية مهمة.				
إعداد خطة مفصلة للمشروع	■ تحتوي الخطة على أهداف واضحة، وتحدد الأدوات اللازمة، وتوزع الأدوار توزيعاً منظماً.				
وضع جدول زمني مفصل	■ الجدول الزمني منظم، ويحدد الأهداف بحسب الأولوية، ويحتوي على جدول واضح لفترة التنفيذ.				
كتابة خطة المشروع باستخدام Google Docs	■ الخطة مكتوبة بوضوح، مع تنظيم جيد للنصوص، والعناوين واضحة، والأقسام مميزة.				
مقياس الأداء: ■ نعم: استيفاء الطالب / الطالبة جميع مُتطلّبات المؤشّر على نحوٍ ممتاز. ■ لا: عدم استيفاء الطالب / الطالبة مُتطلّبات المؤشّر. ■ نوعاً ما: استيفاء الطالب / الطالبة بعض مُتطلّبات المؤشّر؛ لكنّه يحتاج إلى تحسين في جوانب أخرى.					

ملاحظات

إجابات أسئلة أقيم تعلّمي

المعرفة: أوظف في هذا الدرس ما تعلّمته من معارف في الإجابة عن الأسئلة الآتية:

السؤال الأول:

الحوسبة الخضراء هي الاستخدام البيئي المسؤول لأجهزة الحاسوب والموارد التكنولوجية ذات الصلة، والذي يحد من التأثير السلبي لتكنولوجيا المعلومات والاتصالات في البيئة. وتظهر أهميتها عن طريق إسهامها في تقليل استهلاك الطاقة، والحد من انتشار النفايات الإلكترونية، وخفض التكاليف التشغيلية، وتعزيز مبدأ الاستدامة البيئية؛ مما يحسن من كفاءة الطاقة، ويوجه نحو استخدام الطاقة المتجددة، وتدوير النفايات الإلكترونية، ثم إنها تهدف إلى الحد من نسبة الانبعاثات الكربونية الناجمة عن استخدام تكنولوجيا المعلومات؛

أقيم تعلّمي

المعرفة: أوظف في هذا الدرس ما تعلّمته من معارف في الإجابة عن الأسئلة الآتية:

السؤال الأول: أوضّح المقصود بالحوسبة الخضراء، وأبين أهميتها.

انظر الهامش

السؤال الثاني: ما الإجراءات التي أستطيع تنفيذها وحدي لتطبيق مبدأ الحوسبة الخضراء؟

- إطفاء الأجهزة غير المستخدمة.
- استخدام الأجهزة الموفرة للطاقة.
- ضبط الأجهزة على وضع السكون.
- إعادة التدوير للأجهزة.
- ضبط إعدادات الطاقة.

السؤال الثالث: ما الأمور التي يجب على المجتمع مراعاتها في تبني مبادئ الحوسبة الخضراء؟

انظر الهامش

السؤال الرابع: أعلّل ما يأتي:

1. يُعدُّ استخدام جهاز الحاسوب المحمول (Laptop) أفضل من استخدام جهاز الحاسوب المحمول (Desktop) في تطبيق مبدأ الحوسبة الخضراء.

انظر الهامش

2. يساعد تصميم ممرّات باردة أو ممرّات ساخنة في المؤسسات والمباني على تطبيق مبدأ الحوسبة الخضراء.

انظر الهامش

156

السؤال الثالث:

- التوعية والتثقيف ونشر الوعي بأهمية الحوسبة الخضراء وفوائدها.
- تشجيع السياسات البيئية التي تتمثل في دعم الأنظمة والتشريعات التي تعزز استخدام التكنولوجيا الصديقة للبيئة.
- التعاون مع المنظمات البيئية التي تهتم بالبيئة، وتسهل سبل تطبيق الحوسبة الخضراء.

السؤال الرابع: أعلل ما يأتي:

1. أجهزة الحاسوب المحمولة مصممة لتكون أكثر كفاءة في استخدام الطاقة مقارنة بالحواسيب المكتبية، فهي تستهلك طاقة أقل بسبب مكوناتها المصغرة والمصممة للعمل بكفاءة عالية لفترات طويلة على طاقة البطارية. ثم إن الحواسيب المحمولة، تعتمد على تقنيات تبريد متقدمة وفعالة؛ مما يقلل من الحاجة إلى أنظمة التبريد الكبيرة التي تستهلك طاقة كبيرة كما هو الحال في الحواسيب المكتبية. وأجهزة الحاسوب المحمولة عادة ما تحتوي على عدد أقل من المكونات الكبيرة والمستهلكة للطاقة عند تصنيعها مقارنة بالحواسيب المكتبية؛ مما يقلل من الحاجة إلى موارد إضافية في التصنيع، ويقلل من الأثر البيئي أيضًا. والبطاريات المستخدمة في أجهزة الحاسوب المحمولة مُصممة لتكون فعالة وذات عمر طويل؛ مما يقلل من الحاجة إلى استبدالها على نحو متكرر، ومن ثمّ تقليل النفايات الإلكترونية.
2. للتمكن من التعليل، نوضح في البداية المقصود بكل من (الممر البارد، والممر الساخن).

ممر الهواء البارد: (Cold Aisle) تُصمَّم الممرات الباردة؛ بحيث يُوجَّه الهواء البارد من أجهزة التبريد مباشرة إلى مداخل خوادم الحاسوب.

ممر الهواء الساخن (Hot Aisle): تُصمَّم الممرات الساخنة؛ بحيث يُوجَّه الهواء الساخن الخارج من الخوادم بعيدًا عن مداخل الهواء البارد؛ مما يقلل من اختلاط الهواء الساخن والبارد، ويزيد من فعالية التبريد. وعن طريق الفصل بين الهواء البارد والساخن، يمكن تحسين كفاءة أنظمة التبريد في مراكز البيانات؛ مما يقلل من استهلاك الطاقة اللازمة لتبريد المعدات. كذلك فإنّ تنظيم تدفق الهواء تدفقًا صحيحًا، يساعد في تقليل الحاجة إلى التبريد المفرط؛ مما يقلل من استهلاك الطاقة، ومن ثمّ يقلل من الأثر البيئي. ويساعد التبريد الفعّال أيضًا في الحفاظ على درجات حرارة مناسبة للمعدات؛ مما يزيد من عمرها، ويقلل من الحاجة إلى استبدالها على نحو متكرر.

المهارات: أوظف مهارات التواصل والبحث الرقمي للإجابة عن السؤال الآتي:

السؤال الأول:

خطوات كتابة تقرير عن دور المؤسسات الحكومية والشركات في تطبيق مبدأ الحوسبة الخضراء:

■ تحديد الفكرة الرئيسة للتقرير: دور المؤسسات الحكومية والشركات في القطاعين العام والخاص في تعزيز الحوسبة الخضراء.

■ جمع المعلومات: البحث عن مصادر موثوقة على الإنترنت، توضيح كيف تسهم الحكومات والشركات في تطبيق ممارسات الحوسبة الخضراء. ويمكن الاستعانة بمواقع مثل IBM Green Coding أو OWASP للحصول على معلومات حول المبادرات والتقنيات المستخدمة في الحوسبة الخضراء.

المهارات: أوظف مهارة التواصل الرقمي ومهارة البحث الرقمي في الإجابة عن السؤال الآتي:

أكتب - بالتعاون مع أفراد مجموعتي - تقريراً عن دور المؤسسات الحكومية والشركات العامة والخاصة في تطبيق مبدأ الحوسبة الخضراء، وأستعين لذلك بمواقع ذات صلة بالموضوع، بإشراف مُعلّمي / مُعلّمتي.

انظر الهامش

القيم والاتجاهات:

أنشئ دفتر يوميات (Dairy Book) باستخدام تطبيق (CANVA)، ثم أطبق إجراءات الحوسبة الخضراء في المنزل والمدرسة، ثم أدون ما أقوم به يومياً على مدار أسبوع كامل، وأقيم الإجراءات التي نفذتها، وأقيس درجة تأثيرها في توفير الطاقة.

■ تنظيم التقرير:

- المقدمة: توضيح أهمية الحوسبة الخضراء ودورها في حماية البيئة.
- دور المؤسسات الحكومية: شرح المبادرات والسياسات الحكومية التي تدعم الحوسبة الخضراء.
- دور الشركات: عرض الأمثلة على كيفية تنفيذ الشركات لممارسات الحوسبة الخضراء.

■ مراجعة وتحرير:

- مراجعة التقرير؛ للتأكد من دقة المعلومات وصحة اللغة والتنظيم.
- التأكد من توثيق جميع المصادر التي استخدمت
- عند كتابة المسودة الأولى، يمكن الاستعانة بالمعلم؛ للحصول على ملحوظات حول كيفية تحسين التقرير.

■ التقديم والمشاركة: مناقشة التقرير مع الزملاء، وتقديمه وفق الموعد المحدد.

القيم والاتجاهات

أقدم للطلبة التعليمات اللازمة لتنفيذ النشاط، وأحدّد لهم الوقت اللازم للتنفيذ والتسليم.

خطوات التطبيق:

- تدوين الأعمال اليومية، وتحديد إجراءات تطبيق الحوسبة الخضراء.
- تقييم الأعمال ومدى تطبيق الحوسبة الخضراء فيها.
- تحديد الدور الذي يقوم به وتأثيره في توفير الطاقة.
- اقتراح تحسينات وأعمال يمكن القيام بها لتحسين تطبيق الحوسبة الخضراء، وتقليل استهلاك الطاقة.
- تطبيق الأعمال اليومية أسبوعاً آخر، والمقارنة بين الأسبوعين الأول والثاني.

الدرس الثاني:

النفايات الإلكترونية

(Electronic Waste)

عدد الحصص المُقترحة:
حصتان.

المصادر والمراجع: كتاب الطالب، تطبيق
Google Docs.

الأدوات والتجهيزات:
قصاصات ورقية لاصقة (Sticky Notes)،
جهاز حاسوب، أوراق بيضاء، أقلام مُلوّنة



الدرس الثاني

النفايات الإلكترونية (Electronic Waste)

الفكرة الرئيسية:

تعرّف مفهوم النفايات الإلكترونية (E-Waste)، وطرائق التخلص الآمنة منها، وكذلك تعرّف بعض الأدوات الحاسوبية الصديقة للبيئة.

المفاهيم والمصطلحات:

النفايات الإلكترونية (E-Waste)، البصمة الكربونية الرقمية (Carbon Digital Footprint)، الأدوات الحاسوبية الصديقة للبيئة (Echo-Friendly Computing Tools).

نتائج التعلم (Learning Outcomes):

- أعرّف مفهوم النفايات الإلكترونية.
- أوضّح طرائق التخلص الآمنة من النفايات الإلكترونية.
- أذكر أدوات حاسوبية صديقة للبيئة.

هل سبق أن سمعتُ بمفهوم النفايات الإلكترونية؟ هل تكوّنت لديّ فكرة عن هذا المفهوم؟ لماذا تُعدُّ الأجهزة الإلكترونية التالفة من النفايات؟

مُنتجات التعلم (Learning Products)

استكمال المرحلة التخطيطية، والتحضير لتنفيذ المشروع الريادي الرقمي الذي يتناول القضايا البيئية والاجتماعية المُتعلّقة بالحوسبة، وذلك بتجميع الموارد اللازمة.

أولاً: التهيئة

سياق التعلم:

- أوضّح للطلبة فكرة الدرس الرئيسة التي تركز على النفايات الإلكترونية من حيث المفهوم، وطرائق التخلص الآمنة منها، وتعرّف الأدوات الحاسوبية الصديقة للبيئة.
- أوضّح للطلبة النتائج الرئيسة التي ستُحقّق خلال رحلة تعلّمهم لهذا الدرس، والتي تركز على الأفكار الرئيسة من الدرس.
- أبين للطلبة مُنتج التعلم الذي يُتوقّع أن يُنجزوه في هذا الدرس؛ إذ يمثل المحطة الثانية من المشروع الريادي الذي اختير؛ إذ يقوم الطلبة بعد إنهاء هذا الدرس باستكمال المرحلة التخطيطية للمشروع، والتحضير لتنفيذه عن طريق تجميع المواد اللازمة.

نشاط تمهيدي:

- أوزع الطلبة في مجموعات عمل غير متجانسة.
- أكلف الطلبة نقاش الأسئلة في ما بينهم، والتحضير لعرضها ومناقشتها مع الزملاء في الصف.

- أتيح المجال للطلبة للمناقشة والعرض للإجابات، وأدير الحوار بين الطلبة؛ للتوصل الى الإجابات الصحيحة.
- إجابات محتملة:

- ماذا أفعل بالأجهزة الإلكترونية بعد تلفها أو استبدالها؟
- التخلص منها بالطريقة المعتادة مع النفايات، أو دفنها أو حرقها؛ مما يتسبب في التلوث.
- الإجابة المفترضة هي: إعادة تدويرها أو التخلص منها بطريقة آمنة.



نشاط
تمهيدي

أفكر في الأسئلة الآتية، ثم أشارك أفراد مجموعتي في تجربتي الشخصية المتعلقة بموضوع النفايات الإلكترونية:

- ماذا أفعل بالأجهزة الإلكترونية بعد تلفها أو استبدالها؟
 - في رأيي، ما الطريقة الصحيحة للتخلص من أجهزة الحاسوب والأجهزة الإلكترونية التالفة؟
 - كيف تؤثر هذه الطريقة في البيئة والصحة والمجتمع؟
- أستمع لتجارب أفراد مجموعتي بهذا الخصوص، وأبادل معهم الأفكار والمقترحات والحلول، ثم أدون النتائج التي نتوصل إليها في المجموعة. بعد ذلك أعد - بالتعاون مع أفراد مجموعتي - عرضاً تقديمياً قصيراً يتناول تلك النتائج.

تعريف النفايات الإلكترونية (E- Waste Definition)



تُعرف النفايات الإلكترونية باسم المُخلفات الإلكترونية أو النفايات الرقمية؛ وهي أجهزة إلكترونية قديمة أو شكك عمرها الافتراضي على الانتهاء، واستُبدل بها أجهزة أخرى جديدة أو حديثة. ومن الأمثلة عليها: الحواسيب، والهواتف المحمولة، وأجهزة التلفاز، والأجهزة الإلكترونية المنزلية.

تحتوي النفايات الإلكترونية على مواد خطيرة وسامة (مثل: الزئبق، والرصاص)، يُمكنها أن تلوث البيئة، وتضر بالصحة العامة إذا لم يحسن التخلص منها بصورة آمنة. وهي تُعدُّ مصدرًا رئيسًا ومعيّنًا لا ينضب للنفايات الصلبة العالمية، التي تتراكم بكميات ضخمة سنوياً، ولا يعاد تدوير معظمها بطرائق صحيحة؛ ما يلحق ضرراً كبيراً بالبيئة.

تشير كثير من الدراسات المتخصصة إلى أنَّ النفايات الإلكترونية هي أسرع نموّاً بين النفايات الصلبة على مستوى العالم؛ إذ تزداد بمعدل يفوق نموَّ السُّكَّان بنحو ثلاثة أضعاف. وبحسب بيانات مُنظمة الصحة العالمية، فقد شهد عام 2019م تدوير أقل من ربع النفايات الإلكترونية على المستوى الرسمي في مختلف دول العالم، علماً بأنَّ هذه النفايات تحوي موارد قيّمة يُمكن استعادتها والاستفادة منها إذا أُعيد تدويرها بصورة صحيحة؛ ما يجعلها مصدرًا مهمّاً للدخل. غير أنَّ البلدان ذات الدخل المنخفض أو الدخل المتوسط لا تُلقي بالآ إلى هذا الجانب، وتعاني نقصاً في القوانين وضعفاً في التدريب وخللاً في البنية التحتية؛ ما يُعرّض سُكَّانها لمخاطر جَمّة.

159

- ما الطريقة الآمنة للتخلص من أجهزة الحاسوب والأجهزة الإلكترونية الآمنة؟
- التخلص من البطاريات واستخدام إعادة التدوير، أو بيع الأجهزة لمتاجر الأجهزة المستعملة للاستفادة منها، أو استخدام برامج الاسترداد وإعادة الشراء التي تتبعها بعض الشركات.
- كيف تؤثر هذه الطريقة في الصحة والبيئة والمجتمع؟
- الطريقة الاعتيادية؛ إذ تُدفن المخلفات الإلكترونية في الأرض أو تُحرق؛ مما يؤدي الى تسرب الزئبق والرصاص إلى الأرض والهواء، ويتسبب ذلك مشكلات صحية وبيئية كبيرة.
- أدير الحوار بين الطلبة، وأستمع إلى إجابات المجموعات المختلفة، وأوجه النقاش نحو تعريف النفايات الإلكترونية.

- أذكر الطلبة بما تعلموه عن الحوسبة الخضراء.
- أذكر الطلبة بالقضية البيئية التي عُدَّ فيها استخدام الحاسوب سبباً لتلوث البيئة، وتراكم النفايات الإلكترونية.
- أسأل الطلبة عن تجارب خاصة مرّوا بها في التعامل مع الأجهزة الإلكترونية، وكيف تخلصوا من التالف منها. واستمع إلى إجابات الطلبة وأسألهم: هل يعدّون هذه الطريقة صحيحة في التعامل مع هذه الأجهزة والتخلص من التالف منها؟
- أذكر بأن الحكم على هذه السلوكيات، سيتم إرجاؤه إلى ما بعد الانتهاء من هذا الدرس، وسيُتاح المجال لهم لتأمل سلوكياتهم.

ثانيًا: عملية التعليم والتعلم

بناء المفهوم

- أناقش الطلبة في إجاباتهم، وأوضح لهم ما يترتب على النفايات التكنولوجية من أضرار، تفاقم مشكلة التلوث البيئي.
- أدير جلسة عصف ذهني، وأسأل الطلبة عن تعريف النفايات الإلكترونية، وأستمع إلى إجاباتهم، وأدير الحوار للتوصل إلى أن النفايات الإلكترونية هي أجهزة إلكترونية قديمة، أو شك عمرها الافتراضي على الانتهاء، واستبدل بها أجهزة أخرى جديدة أو حديثة.
- أطرح قضية كون الأجهزة القديمة فعالة وغير تالفة أحياناً؛ لكنها غير حديثة، وتصبح غير صالحة للاستعمال. أناقش الطلبة في الإجراء الصحيح الذي يجب اتخاذه في هذه الحالة.
- أوجه النقاش نحو تراكم النفايات الإلكترونية، والأرقام والنسب التي تشير إلى تدويرها خلال الأعوام السابقة.
- أعرض الأرقام والدراسات التي تشير إلى النمو السريع للنفايات الإلكترونية مقابل إعادة تدويرها.

استخدام الأشكال والصور:

■ أعرض الشكل (1-1) وأوجه الطلبة إلى قراءة الأرقام ومقارنة النمو السريع في النفايات الإلكترونية.

■ ألفت نظر الطلبة إلى مقارنة اللون البرتقالي مع اللون الأزرق، وتعرّف النسبة بين كمية تراكم النفايات الإلكترونية، وجمعها، وإعادة تدويرها.

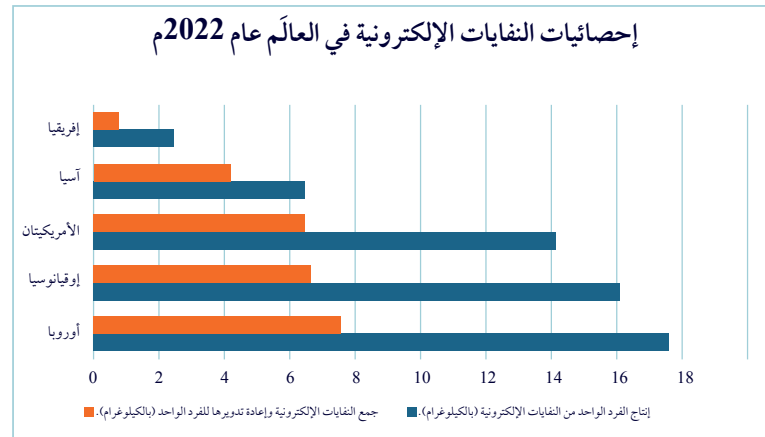
■ التركيز على التحديات التي تواجهها الدول، والتي أسهمت في اتساع الهوة بين توليد النفايات والجهود المبذولة في إعادة تدويرها، وهي (التقدم التكنولوجي، وزيادة الاستهلاك، ودورة الحياة القصيرة للمنتجات).

■ إتاحة الفرصة للطلبة لإعطاء أمثلة على هذه التحديات. عن تجربة التعلّم الإلكتروني ومدى

تأثيرها في تعليمهم. مثل التطور الهائل الذي يحدث في عالم الاتصالات والأجهزة الخلوية، وتطوّر الحواسيب وأقراص التخزين، والسرعة الهائلة في التقدم التكنولوجي.

من الممكن إتاحة المجال للطلبة للمقارنة بين التطور التكنولوجي في مراحل تعليمهم المختلفة، والحديث عن تجربة التعلّم الإلكتروني التي لم تكن متوافرة من قبل.

ثمّ جاء تقرير الأمم المتحدة الرابع عن النفايات الإلكترونية (GEM) مُبيّناً أنّ توليد النفايات الإلكترونية وتكدّسها ينمو بسرعة تفوق خمسة أضعاف مُعدّل إعادة التدوير المُوثّقة، أنظر الشكل (1-1)؛ ففي عام 2022م، أنتج العالم قرابة (62) مليون طن من النفايات الإلكترونية، بزيادة نسبتها 82٪ على عام 2010م. ومن المُتوقّع أن يصل الرقم إلى نحو (82) مليون طن بحلول عام 2030م. أمّا ما جُمع وأُعيد تدويره من هذه النفايات فكان أقل من الربع، بما نسبته 22.3٪ من المجموع الكلي للنفايات الإلكترونية؛ ما تسبّب في هدر كثير من الموارد الطبيعية التي بلغت قيمتها (62) مليار دولار، وزاد من مخاطر التلوّث بصورة كبيرة. ولا شكّ في أنّ التحديات التي تُواجهها كثير من دول العالم (مثل: التقدّم التكنولوجي، وزيادة الاستهلاك، ودورة الحياة القصيرة للمنتجات) قد أسهمت في زيادة الفجوة واتّساع الهوة بين توليد النفايات والجهود المبذولة لإعادة تدويرها.



Source: The Global E-waste Monitor 2024

الشكل (1-1): إحصائيات النفايات الإلكترونية في مختلف دول العالم عام 2022م بحسب تقرير الأمم المتحدة (مُراقب النفايات الإلكترونية العالمي لعام 2024م).

■ أوجّه الطلبة إلى قضية الإثراء والتعرّف إلى طرائق إعادة تدوير النفايات الإلكترونية، والطريقة المبتكرة في صناعة ميداليات الألعاب الأولمبية. ويمكن توجيههم لزيارة الرابط الآتي: ميداليات طوكيو 2020 الأولمبية - التصميم والتاريخ والصور (olympics.com).

■ أعرّض الصور الآتية لمبادرة الفن والنفايات الإلكترونية، وأكّلف الطلبة تأمل هذه الصور.



شهدت دورة الألعاب الأولمبية في طوكيو عام 2020م كثيرًا من التحضيرات والتجهيزات، وكان لافتًا فيها اعتماد مُقترح صنع الميداليات من مواد أُعيد تدويرها، بوصف ذلك جزءًا من مبادرة أوسع تهدف إلى تعزيز الاستدامة البيئية والمحافظة على موارد البيئة. ومن ثمّ، فقد أمكن صنع الميداليات الذهبية والفضية والبرونزية من مواد توجد في النفايات الإلكترونية التي يعاد تدويرها، مثل: الهواتف المحمولة القديمة، والأجهزة الإلكترونية الصغيرة. بدأ القائمون على هذا المُقترح حملتهم عام 2017م، وتمكّنوا من جمع (16.5) كلف من الذهب، وهو ما يُمثّل 54% من الكميّة المطلوبة، و(1800) كلف من الفضة، بما نسبته 43.9% من الكميّة اللازمة لطلاء ميداليات أصحاب المركز الثاني في البطولة الأولمبية. وتحقيقًا لهذا الهدف؛ فقد بدأ العمل على تفكيك الأجهزة والمعدّات، وتحويلها إلى معادن خام؛ ما زاد من حصيلة ما جُمع من المعادن النفيسة؛ إذ بلغ مجموع الكميّة المُستخرّجة من البرونز نحو (2700) كلف بحلول عام 2018م، في حين أسهمت التبرّعات في زيادة كمّيّات الذهب والفضة المُستخرّجة لتصل إلى (28.4) كلف من الذهب، و(3500) كلف من الفضة.

إدارة النفايات الإلكترونية (E-waste Management)



تهدف إدارة النفايات الإلكترونية إلى استعادة النفايات الإلكترونية، ومعالجتها، وإعادة تدويرها، أو تجديدها؛ للاستفادة منها، واستخدامها في مختلف مناحي الحياة مرّة أخرى. غير أنّ عملية إعادة التدوير الإلكتروني تُواجه تحدّيًا كبيرًا؛ نظرًا إلى طبيعة هذه الأجهزة؛ فهي مُعقّدة، ومصنوعة من الزجاج والمعدن والبلاستيك بنسب مُتفاوتة.

تشمل عملية إدارة النفايات الإلكترونية المراحل الآتية:

1. جمع النفايات؛ إذ يتمّ تجميع النفايات الإلكترونية من مصادر مختلفة.
2. تفكيك النفايات؛ إذ يتمّ فصل مكونات النفايات الإلكترونية بعضها عن بعض؛ لتحديد ما يُمكن أن يعاد استخدامه.
3. تنظيف البيانات؛ أي التأكد أنّ البيانات لم تُعدّ صالحة للاستخدام.
4. إعادة التدوير؛ أي فصل الأجزاء والمواد لاستخدامها في مُنتجات جديدة.
5. التجديد؛ أي إعادة استخدام الأجزاء القيّمة لإطالة أمد عمر المعدّات الأخرى.

- أناقش مع الطلبة المواد التي صنعت منها هذه الفراشات، وما هي الرسالة التي تريد الفنانة إيصالها.
- أوجّه النقاش نحو طرائق إدارة النفايات الإلكترونية، وأركز على الهدف منها وهو: استعادة النفايات الإلكترونية، ومعالجتها، وإعادة تدويرها، وتجديدها للاستفادة منها، واستخدامها في مناحي الحياة المختلفة مرّة أخرى.
- أكّلف الطلبة تأمل الرمز، وهو يشير إلى إعادة تدوير المخلفات الإلكترونية والاستفادة منها مرّة أخرى.

مناقشة

- أكتب على بطاقات مراحل عملية إدارة النفايات الإلكترونية.
- أوزع الطلبة في خمس مجموعات غير متجانسة، وأزود كل مجموعة ببطاقة من البطاقات؛ بحيث تحدد كل مجموعة ترتيب المرحلة، وتناقش ما يحدث فيها.
- أناقش مع الطلبة ترتيب هذه الخطوات بحسب أولوية المراحل.
- أتوقف عند بعض المراحل لمناقشة أمور مهمة مثل:
- جمع النفايات: ضرورة اقتراح فكرة لجمع النفايات، أهمها الفصل، وتخصيص وعاء خاص للنفايات الإلكترونية.
- تفكيك النفايات: أناقش الطلبة

تمر عملية معالجة النفايات الإلكترونية بالمرحلة الآتية:

1. التفكيك؛ أي إزالة المكونات المهمة من النفايات الإلكترونية لتجنب التلوث بالمواد السامة خلال العمليات اللاحقة.
 2. المعالجة الميكانيكية، وهي تشمل عملية السحق وعملية الفرز للنفايات الإلكترونية؛ ما يُسهل استخراج المواد القابلة لإعادة التدوير، وفصل المواد الخطرة.
 3. التكرير؛ إذ يساعد التكرير على استعادة المواد الخام من دون إلحاق ضرر كبير بالبيئة. وفي هذه المرحلة، يتم تنقية الكسور أو تعديلها؛ استعداداً للبيع بوصفها مواد خام ثانوية، أو للتخلص منها بصورة آمنة.
- يُذكر أن عملية التفكيك تُفضي إلى إزالة المكونات الأساسية، في حين تؤدي المعالجة الميكانيكية إلى فصل المواد القابلة لإعادة التدوير، ومعالجة المواد الخطرة، وتصفية الانبعاثات الغازية، ومعالجة المخلفات؛ ما يُحد من تأثيرها الضار بالبيئة.

أبحث

أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن مصطلح إعادة الاستخدام (Reuse)، ومصطلح إعادة التدوير (Recycle)، ومصطلح إعادة التصنيع (Upcycling)، ومصطلح التقليل (Reduction)، ثم أعد تقريراً عن ذلك، وأشاركه زملائي / زميلاتي في الصف.

استراتيجيات إدارة النفايات الإلكترونية

تعد إدارة النفايات الإلكترونية عملية مهمة لضمان تطبيق مبادئ الحوسبة الخضراء، والحد من التأثير البيئي الضار للأجهزة الإلكترونية المهملة.



162

حول تفكيك النفايات، ومتى يتاح ذلك "بعض النفايات يمكن إعادة تدويرها من دون تفكيك، إذا كانت جميع أجزائها صالحة مثلاً".

■ تنظيف البيانات: من الضروري تنبيه الطلبة أن التخلص من النفايات الإلكترونية يتطلب البحث عن صلاحية الجزء الذي تُخزن فيه البيانات، والتأكد من حذف البيانات، أو عدم صلاحيتها للاستخدام، أو عدم القدرة على استردادها؛ لحماية خصوصية بياناتهم.

■ إعادة التدوير: أناقش الطلبة في المجالات التي يمكن إعادة استخدام النفايات الإلكترونية فيها.

■ أطرح السؤال الآتي: متى تخضع النفايات الإلكترونية لمرحلة التجديد؟ (إذا كان هناك إمكانية لإعادة استخدامها وإطالة عمر المعدات الأخرى التي تعمل بالتزامن معها).

■ ألقت نظر الطلبة إلى أن النفايات الإلكترونية تُعالج قبل إعادة تدويرها، وأسأل الطلبة: "ما سبب اللجوء إلى معالجة النفايات الإلكترونية قبل إعادة تدويرها؟"

■ إجابة محتملة: تضمن المعالجة التخلص من أي مواد خطيرة، قد تؤدي إلى انبعاثات غازية، وتضر البيئة، بالإضافة إلى فصل المكونات غير القابلة للتدوير.

- أوضح للطلبة مراحل معالجة النفايات الإلكترونية، وما يحدث في كل مرحلة، وأهمية كل مرحلة في عملية إعادة التدوير.
- أطرح السؤال الآتي: ما الفرق بين عملية إدارة النفايات الإلكترونية وعملية معالجتها؟

بحث ومناقشة:

- أكلف الطلبة مهمةً فرديةً بيتية، تتمحور حول البحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن مصطلح إعادة الاستخدام (Reuse)، ومصطلح إعادة التدوير (Recycle)، ومصطلح إعادة التصنيع (Upcycling)، ومصطلح التقليل (Reduction).
- أطلب إلى الطلبة كتابة نتائج بحثهم ومشاركتها مع زملاء عن طريق اللوح الرقمي التفاعلي Padlet.
- أشارك مع الطلبة رابط اللوح الرقمي التفاعلي الخاص بالنشاط.
- أكلف الطلبة الاطلاع على مشاركات زملائهم، والتفاعل مع ثلاث مشاركات على الأقل.
- أشرف على مشاركات الطلبة، وأقدم التغذية الراجعة؛ بتزويدهم بتعريفات نموذجية، والطلب إليهم مقارنتها بمشاركاتهم.

إجابة محتملة:

- مصطلح إعادة الاستخدام (Reuse): إجراء أو ممارسة لاستخدام عنصر ما، سواء لغرضه الأصلي (إعادة الاستخدام التقليدية)، أو لتنفيذ وظيفة مختلفة مثل (إعادة التدوير للأفضل أو إعادة توظيفها لأغراض أخرى).
- مصطلح إعادة التدوير (Recycle): مصطلح عام، يصف عديدًا من العمليات التي تُعالج فيها النفايات، وتُفكك إلى المواد الخام المختلفة المركبة منها؛ إذ يمكن إعادة استخدام المواد الخام نفسها لإنتاج مُنتجات جديدة.
- مصطلح إعادة التصنيع (Upcycling): هي عملية تحويل النفايات أو المنتجات عديمة الفائدة إلى مواد أو منتجات جديدة، جودتها أفضل، أو لها فائدة بيئية أفضل.
- مصطلح التقليل (Reduction): تساعد تقليل كمية النفايات المنتجة في المجتمع في القضاء على توليد النفايات الضارة والمستمرة، ودعم الجهود المبذولة لتعزيز مجتمع أكثر استدامة.

استخدام الأشكال والصور:

- أوجه الطلبة لمشاهدة الصورة الواردة في كتاب الطالب التي تتمحور حول "استراتيجيات إدارة النفايات الإلكترونية".
- أناقش الطلبة بالاستراتيجيات والهدف من تطبيقها.
- أركز على أن إدارة النفايات الإلكترونية، تتضمن تحقيق أهداف الحوسبة الخضراء، والتقليل من النفايات الإلكترونية.
- أطرح سؤالاً عن إيجابيات إدارة النفايات الإلكترونية بعد مناقشة الاستراتيجيات، ثم أسأل عن الآثار الناجمة عن سوء إدارة النفايات الإلكترونية.
- أتيح المجال للطلبة لبناء التوقعات والتعبير عنها.

تشمل استراتيجيات إدارة النفايات الإلكترونية ما يأتي:

1. التقليل، وإعادة الاستخدام، وإعادة التدوير: يُقصد بذلك تحفيز الأشخاص على تقليل استهلاكهم للأجهزة الإلكترونية، وإعادة استخدام الأجهزة قدر الإمكان، وتعزيز عملية إعادة التدوير المسؤولة للأجهزة التالفة أو المستهلكة.
2. مسؤولية المُنتج طويلة المدى: يشمل ذلك التزام الشركات المُصنّعة للأجهزة الإلكترونية بتطبيق البرامج التي تُحمّلها مسؤولية دورة الحياة لمنتجاتها، بما في ذلك التخلص الآمن منها بعد انتهاء عمرها الافتراضي.
3. التشريعات والتنظيمات: يجب سَنُّ القوانين التي تدعم إدارة النفايات الإلكترونية، بما في ذلك ضوابط التخلص الآمن من النفايات الإلكترونية، وأهداف إعادة التدوير.
4. التوعية العامة والتعليم: يتمثل ذلك في تثقيف أفراد المجتمع، وتوعيتهم بأهمية إدارة النفايات الإلكترونية على نحوٍ مسؤول، وتعريفهم بمزايا إعادة التدوير.
5. البنية التحتية اللازمة لجمع النفايات الإلكترونية وإعادة تدويرها: يكون ذلك بإنشاء نقاط تجميع مُعتمدة للنفايات الإلكترونية.
6. أمان البيانات: يُقصد بذلك مسح جميع البيانات الشخصية والبيانات المُهمّة قبل إعادة تدوير الأجهزة الإلكترونية.
7. التجديد والتبرُّعات: يُقصد بذلك تجديد الأجهزة الإلكترونية، ثمّ التبرُّع بها للمدارس، أو للمنظمات غير الربحية، أو لفئات المجتمع المحرومة.
8. التصميم البيئي: يُقصد بذلك تبني ممارسات التصميم البيئية التي تُسهّل عملية إعادة التدوير.
9. التعاون والشراكات: يجب تعزيز أو أصر التعاون بين الشركات المُصنّعة، والتجّار، والمنظمات ذات العلاقة، والحكومات.
10. البحث والابتكار: يجب دعم البحوث العلمية التي تهدف إلى تطوير تقنيات إعادة التدوير.

الآثار البيئية الناجمة عن سوء إدارة النفايات الإلكترونية

إنّ التعامل الخطأ مع المُخلفات الإلكترونية، وغياب شروط السلامة العامة والوقاية الضرورية أثناء التعامل مع المواد السّامة في هذه المُخلفات؛ يُمثّل خطراً على الصحة، وتهديداً للموارد الطبيعية، وبخاصة التربة والمياه. يُبيّن الجدول (1-2) أبرز العناصر والمواد السّامة الموجودة في النفايات الإلكترونية بحسب ما أوردته وزارة البيئة الأردنية.

مناقشة

- أطرح السؤال الآتي: ماذا يحدث إذا تم التخلص من النفايات الإلكترونية على نحوٍ خطأ؟
- أستمع إلى إجابات الطلبة وأناقشهم فيها، ثم أناقش الطلبة في احتمالات التلوث الناتج عن سوء إدارة النفايات الإلكترونية، والأطراف المتضررة.

استخدام الأشكال والصور:

■ أجهز بطاقات بعدد المجموعات في الصف، مكتوب عليها المواد الواردة في الجدول (2-1)، وبطاقات أخرى مكتوب عليها أماكن وجودها.

■ أوزع البطاقات على المجموعات، وأستخدم استراتيجية الرؤوس المرقمة لاختيار إحدى المجموعات التي تحمل بطاقة باسم المواد، وأطلب إليها قراءة اسم المادة، ثم أطلب إلى المجموعات التي تحمل بطاقات بمكان وجود المواد السامة أن ترفع بطاقتها، ثم أسألهم: أيّ هذه البطاقات هو المكان الصحيح لوجود للمادة؟ ثم ألصق الأوراق الصحيحة بجانب بعضها بعضاً على اللوح أو على أوراق Flip Chart.

الجدول (2-1): أبرز العناصر والمواد السامة في النفايات الإلكترونية.

اسم المادة السامة	آثارها ومخاطرها	مكان وجودها
الزئبق	- اضطراب في النمو. - أمراض القلب. - الأمراض السرطانية. - داء السكرى.	- الميكروويف. - لوحات الدارات الإلكترونية. - عاكس التيار. - المحركات.
الكاديوم	- فقدان الكالسيوم. - هشاشة العظام. - تلف الرئتين. - الوفاة.	- بطاريات الهواتف المحمولة.
الكروم	- تهيج الجلد. - الطفح الجلدي.	- صناعة البلاستيك.
النحاس	- التهاب الحلق والرئتين. - تلف الكبد والكلى.	- الأسلاك النحاسية. - لوحات الدارات الإلكترونية.
الرصاص	- اضطراب في النشاط المعرفي واللفظي. - الشلل. - الغيبوبة. - الوفاة.	- أجهزة الحاسوب. - الشاشات. - أجهزة التلفاز. - البطاريات.
النيكل	- الأمراض السرطانية.	- البطاريات القابلة للشحن.
الفضة	- مرض Argyria (بقع زرقاء وبقع رمادية تنتشر على الجلد).	- الهواتف المحمولة.
البريليوم	- الأمراض السرطانية.	- الموصلات.
البلاستيك، والبولفينيل كلوريد	- الإضرار بجهاز المناعة. - الأمراض السرطانية.	- الشاشات. - لوحات المفاتيح. - الفأرة. - جهاز الحاسوب المحمول. - مفتاح (USB).

164

■ أعرض الجدول (2-1) الذي يلخص العناصر والمواد السامة في النفايات الإلكترونية.

■ ألفت نظر الطلبة إلى أن هذه العناصر توجد في أدوات وأجهزة متوافرة غالباً للجميع، وأناقش معهم الأضرار والآثار الخطيرة التي تسببها، وأركز على ضرورة التوعية من هذه السموم خاصة أن جميع الفئات تتعامل مع هذه المواد.

ملاحظات

نشاط فردي:

- أكلف الطلبة مهمةً فرديةً بيتية، وهي تصميم ملصق للتوعية بمخاطر المواد السامة في النفايات الإلكترونية باستخدام أحد برامج التصميم.
- أناقش الطلبة في أهم الأمور التي يجب أن تتوفر في الملصق؛ بحيث يُراعى أن يكون واضحًا وسهلاً، ولا تجد الفئة المستهدفة صعوبة في فهمه، وأن تكون الفكرة والمحتوى المقصود منه واضحين، وأن يُحقق الملصق الهدف الذي صمم لأجله.
- أساعد الطلبة في اختيار برامج لتصميم الملصق، واقترح لهم بعضها مثل CANVA.
- أوجه الطلبة إلى ضرورة تعميم الملصق على الطلبة وأولياء الأمور عبر وسائل التواصل المتوفرة لديهم.



أصمّم ملصقاً للتوعية بمخاطر المواد السامة في النفايات الإلكترونية باستخدام أحد برامج التصميم، ثمّ أشارك الطلبة وأولياء الأمور في الملصق عبر الوسائل الإلكترونية المتوفرة.

الإدارة الفردية للنفايات الإلكترونية

1. في ما يأتي بعض النصائح التي تسهم في التخلص من النفايات الإلكترونية بطرائق صحيحة وآمنة: الوعي بمفهوم النفايات الإلكترونية: يتعين على إدراك مخاطر النفايات الإلكترونية، مُمثلة في المواد السامة التي تحويها، والتي قد ينتهي المطاف بمعظمها إلى مدافن النفايات. ولهذا، فإنّ تعرّف مكوّنات النفايات الإلكترونية يُعدّ أولى خطوات التخلص منها.
2. تقليل كمّ النفايات الإلكترونية: يُمكنني الحدّ من النفايات الإلكترونية بشراء ما يلزمي فقط، واختيار المُنتجات طويلة الأجل، والمُنتجات المُوفّرة للطاقة، وإطالة أمد عمر الأجهزة بإصلاحها بدلاً من استبدال أجهزة جديدة بها.
3. التعاون مع المؤسسات والوزارات، والمشاركة في المشروعات التي تُعنى بتدوير النفايات الإلكترونية على المستوى المحلي.

إضاءة



مشروع (تفكيك): مشروع استثماري أردني، أنشئ للتخلص من النفايات الإلكترونية بصورة آمنة وصحيحة. أعرّف مزيداً من التفاصيل عن هذا المشروع، وأزور الموقع الرسمي الإلكتروني للمشروع؛ بمسح الرمز سريع الاستجابة (QR Code) المجاور:

أبحث

أبحث عن مشروعات محلية في محافظتي، تُعنى بإدارة النفايات الإلكترونية، ثمّ أشارك النتائج التي أتوصّل إليها مع زملائي/ زميلاتي في الصف.

مناقشة

عمل مجموعات

- أوزع الطلبة في مجموعات صغيرة، وأشرح لهم أن الهدف من النشاط اقتراح نصائح، تسهم في التخلص من النفايات الإلكترونية بطرائق صحيحة وآمنة على مستوى الأفراد.
- أتيح لهم المجال للنقاش، وتلخيص النصائح، والتحضير لعرضها ومناقشتها مع بقية زملاء.
- أكلف كل مجموعة اختيار مندوب للعرض، وأفسح المجال للنقاش.
- أدير الحوار؛ بحيث يُتوصّل لبعض النصائح التي تسهم في التخلص من النفايات الإلكترونية بطرائق صحيحة وآمنة على مستوى الأفراد، مركزاً على البنود الواردة في كتاب الطالب.
- يمكن إثراء النقاش عن طريق طرح أمثلة للإدارة الفردية للنفايات في المدرسة أو المنزل، وإتاحة المجال للطلبة للتعبير عن ممارساتهم التي تتمحور حول النقاط أعلاه.

- أوجه الطلبة للتعرف إلى مشروع تفكيك الاستثماري الأردني الذي أنشئ خصيصاً للتخلص من النفايات الإلكترونية عبر زيارة الموقع الإلكتروني؛ عن طريق مسح رمز الاستجابة السريع المرفق مع الإضاءة.

بحث ومناقشة:

نشاط فردي بيتي

- أكلّف الطلبة مهمة بحثية فردية، تتمثل في البحث عن المشروعات المحلية التي تُعنى بإدارة النفايات الإلكترونية.

- أزود الطلبة ببعض العناوين لمساعدتهم على التوصل إلى مراكز تدوير النفايات الإلكترونية في الأردن (مثل : <http://pops.moenv.gov.jo> / (Default /Ar

- يمكن مشاركة الريبورتاج الآتي: https://youtu.be/ya0plbfD_kElsi=BEcDwzq1CIKpt73u مع الطلبة وتوجيههم للبحث عن أماكن وجود مثل هذه اللوحات الإرشادية، وكيف تُجمع النفايات الإلكترونية في الحدائق.

- أكلّف الطلبة تلخيص نتائج البحث باستخدام أحد البرامج الحاسوبية، ومشاركتها مع زملاء عبر اللوح الرقمي التفاعلي (Padlet).

- أشارك مع الطلبة رابط اللوح الرقمي التفاعلي الخاص بالنشاط.

- أكلّف الطلبة الاطلاع على مشاركات زملائهم، والتفاعل مع ثلاث مشاركات على الأقل.

- أشرف على مشاركات الطلبة، وأفعل النقاش.

البصمة الكربونية الرقمية (Carbon Digital Footprint)



تُعرف البصمة الكربونية الرقمية بأنها التأثير السلبي في البيئة الناجم عن استخدام التكنولوجيا الرقمية وممارسة الأنشطة الرقمية عبر شبكة الإنترنت، مُمثلاً في انبعاثات الكربون، واستهلاك الطاقة، فكل عمل نقوم به في شبكة الإنترنت، أو في أجهزة الرقمية، ينتهي به الحال إلى التخزين، وهو جزء من بصمتنا الكربونية الرقمية التي تؤثر سلباً في البيئة.

مثال:

إذا اعتدت مشاهدة جهاز التلفاز مدة ساعة واحدة أو ساعتين يومياً كل عام، فهذا يعني أنني أستخدم ما يكفي من الكهرباء لتشغيل ثلاجتي مدة تصل إلى نصف عام تقريباً. وفي عام 2020م، بلغت البصمة الكربونية لإحدى القنوات ما يُعادل تشغيل مدينة تحوي (150000) منزل.

قياس بصمتي الكربونية الرقمية (Digital Carbon Footprint).

يُمكِنني قياس بصمتي الكربونية الرقمية باتباع الخطوات الآتية:

1. زيارة موقع (Digital Carbon Footprint) عن طريق الرابط الإلكتروني الآتي:

<https://www.digitalcarbonfootprint.eu>



أو مسح الرمز سريع الاستجابة (QR Code) المجاور:

2. اختيار الجهاز الذي سأستخدمه.

3. تعديل بيانات الاستخدام.

4. تأمل كمية غاز ثاني أكسيد الكربون CO_2 الذي أُسهم في إطلاقه في البيئة.



نشاط
فردى

■ أطلب إلى الطلبة تحديد الأعمال اليومية التي يقومون بها عادة (المكوث في المنزل ومشاهدة التلفاز أو التنقل بالسيارة، أو إضاءة المنزل ...)، وأوضح للطلبة أن هذه الأعمال تترك لهم بصمة كربونية على كوكب الأرض. وأسألهم في ما إذا كانوا يعدونها إيجابية أو سلبية.

■ أسأل الطلبة عن البصمة الكربونية الرقمية وماذا يعرفون عنها.

■ أعرض للطلبة الفيديو في الرابط الآتي:

<https://youtu.be/IpKgpcFwdx4>

■ أطلب إلى الطلبة تعريف البصمة الكربونية بحسب المعلومات في الفيديو.

■ أوضح لهم تعريف البصمة الكربونية الرقمية، وأنها التأثير السلبي في البيئة الناجم عن استخدام التكنولوجيا الرقمية، وممارسة الأنشطة الرقمية عبر الإنترنت، وأن البصمة الكربونية الرقمية هي إجمالي كمية انبعاثات الغازات الدفيئة (GHG) الناتجة عن نمط الحياة.

■ أوجه الطلبة لقراءة المثال المطروح في الكتاب، وأناقشهم بما يترتب على هذا المثال البسيط، وأكلفهم المقارنة مع أنشطة أخرى، تترك فيها الأجهزة في وضع التشغيل من دون استخدام، وأطلب إليهم طرح أمثلة من حياتهم، وتأمل سلوكياتهم.

■ أسأل الطلبة: "هل تعدّ البصمة الكربونية الرقمية من النفايات الإلكترونية؟ ولماذا؟"

■ أتيح المجال للطلبة للتعبير عن آرائهم وبناء توقعاتهم، ثم أكلفهم التعرف إلى قيمة بصمتهم الكربونية الرقمية عن طريق توجيههم للنشاط الفردي. (يمكن تنفيذ النشاط في غرفة الصف إن توافرت أجهزة الحاسوب لجميع الطلبة لقياس البصمة الكربونية الرقمية).

- أدير حوارًا عن ضرورة التقليل من قيمة البصمة الكربونية، وكيف يمكننا أفرادًا، تحسين الممارسات للتقليل من النفايات الإلكترونية والانبعاث الكربوني.

إثراء للمعلمة / المعلم

- يمكن تزويد الطلبة بالموقع الآتي لقياس البصمة الكربونية، ومن ثم التعرف إلى كيفية التقليل منها بحسب الأعمال التي يقوم بها الأفراد.

<https://survey.steadypace.sa/runner/evjYVXVilukXvKUXGnIr>



- يتم تعبئة البيانات، وإكمال الاختبار الذي يقدم في النهاية نصائح؛ للتقليل من أثر البصمة الكربونية الرقمية.

نشاط جماعي:

■ أوزع الطلبة في مجموعات عمل غير متجانسة .

■ أوزد الطلبة بالمهمة البحثية خارج الغرفة الصفية حول البحث عن أدوات حاسوبية صديقة للبيئة باستخدام مواقع بحث موثوقة، وأقدم لهم التوضيحات اللازمة.

■ وهنا بعض المواقع التي يمكن البحث فيها للحصول على بعض المعلومات:

<https://www.ecocertify.org/ar/eco-friendly/ecofriendly-tech-accessories-and-devices/>
<https://sustainablykindliving.com/eco-friendly-laptops/>

نشاط فردي:

167

- أبدأ بمناقشة كيفية الحد من انتشار النفايات الإلكترونية مع الطلبة، وربط ذلك بالإدارة الفردية لمعالجة هذه النفايات، وأشجع الطلبة على التفكير في كيفية إدارة النفايات الإلكترونية مسؤولية في حياتهم اليومية.
- أكلف الطلبة مهمة فردية، تتضمن التفكير والبحث في تعديل يمكن إجراؤه على أحد مكونات الحاسوب ليصبح صديقاً للبيئة، وأشجعهم على التفكير في الابتكارات الصغيرة التي يمكن أن تحدث فرقاً كبيراً في تقليل الأثر البيئي.
- أوجه الطلبة لمشاركة أفكارهم ومقترحاتهم حول التعديلات البيئية عبر اللوح الرقمي التفاعلي (Padlet).
- أشارك مع الطلبة رابط اللوح الرقمي التفاعلي الخاص بالنشاط، وأحثهم على البدء في مشاركة أفكارهم بعد انتهاء البحث.
- أكلف الطلبة الاطلاع على مشاركات زملائهم، والتفاعل مع ثلاث مشاركات على الأقل عن طريق تقديم تعليقات بناءة، أو طرح أسئلة لتوسيع النقاش.
- أشرف على المشاركات والنقاشات بين الطلبة، وأفعل الحوار عن طريق طرح أسئلة محفزة تساعد على التفكير النقدي وتوليد أفكار جديدة، وتأكد أن يبقى النقاش منضبطاً وموجهاً نحو التعلم والتطوير.

نشاط جماعي

أبحث - بالتعاون مع أفراد مجموعتي - في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن أدوات حاسوبية صديقة للبيئة، ثم أعد عرضاً تقديمياً عنها باستخدام إحدى الأدوات الرقمية (مثل تطبيق العروض التقديمية (Google Slide)، ثم أعرضه أمام زملائي / زميلاتي في الصف.

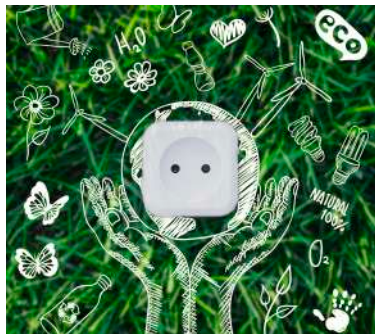
نشاط فردي

أفترض أنني أريد الإسهام في الحد من انتشار النفايات الإلكترونية؛ بأن أفكر في إجراء تعديل على أحد مكونات الحاسوب ليصبح صديقاً للبيئة. أشارك أفكارني مع زملائي / زميلاتي في الصف، ثم أناقشهم فيها.

إضاءة



التأثير السلبي في الاستهلاك والإنتاج
يُغلب على أنماط الاستهلاك والإنتاج اليوم الاعتماد على مصادر الطاقة التقليدية (غير المتجددة). ولا شك في أن الاستخدام المفرط للمياه والأراضي، وانبعاثات الغازات الدفيئة، وتوليد النفايات وإدارتها، والتخلص غير الآمن من النفايات السامة؛ يُلقي بظلاله القاتمة على البيئة. ولهذا يجب اتخاذ قرارات مُهمّة، وتطبيق إجراءات صارمة؛ لخفض الهدر في الغذاء إلى النصف، وضمان الإدارة السليمة للنفايات الكيميائية، والحد من الاستهلاك غير المسؤول، وتشجيع السياحة الخضراء.



ستتووع الإجابات والمقترحات، وفي ما يأتي بعض الأفكار والأسئلة التحفيزية:

- تحسين كفاءة الطاقة في وحدات المعالجة المركزية: البحث في كيفية تصميم وحدات المعالجة المركزية؛ بحيث تستهلك طاقة أقل في أثناء العمليات اليومية، والتفكير في إمكانية استخدام معالجات متعددة النواة، تدعم وضع السكون (Sleep Mode) لتقليل استهلاك الطاقة.
- استخدام مواد قابلة لإعادة التدوير في صناعة الهياكل والأجزاء: استكشاف إمكانية تصنيع هياكل الحاسوب من مواد معاد تدويرها، أو قابلة للتحلل الحيوي بدلاً من البلاستيك التقليدي. ومن المواد التي انتشر استخدامها لصناعة أدوات الحاسوب صديقة البيئة "خشب البامبو".
- تبريد أكثر كفاءة وصداقة للبيئة: النظر في استخدام تقنيات تبريد مبتكرة، مثل التبريد بالماء، أو باستخدام المواد الصديقة للبيئة التي تقلل من الاعتماد على المراوح الكهربائية ذات الاستهلاك العالي للطاقة.
- تصميم بطاريات صديقة للبيئة: البحث في تطوير بطاريات حواسيب محمولة، تكون أكثر كفاءة في استخدام الطاقة، وأطول عمراً، مع استخدام مواد أقل سُمية، وأكثر قابلية لإعادة التدوير.
- تقليل استهلاك الطاقة في شاشات الحاسوب: استكشاف كيفية تحسين شاشات الحاسوب لتستهلك طاقة أقل، مثل استخدام تقنيات LED منخفضة الاستهلاك، أو الشاشات القابلة للإطفاء تلقائياً عند عدم الاستخدام.

أسئلة تحفيزية:

- كيف يمكننا تقليل استهلاك الطاقة في الحاسوب من دون التأثير في أدائه؟
- هل هناك مواد جديدة يمكن استخدامها في تصنيع الحاسوب لتكون أكثر صداقة للبيئة؟
- كيف يمكن تحسين عملية التخلص من الحواسيب القديمة لتقليل الأثر البيئي؟
- ما هي الابتكارات التي يمكن إدخالها في أنظمة التبريد لجعلها أكثر كفاءة وأقل استهلاكاً للطاقة؟
- كيف يمكن تصميم المكونات الإلكترونية لتكون قابلة للتدوير بسهولة بعد انتهاء عمرها الافتراضي؟

ثالثاً: مرحلة الإثراء والتأمل والتقويم

- أطرح على الطلبة السؤال الآتي: بناءً على ما ورد في الدرس، ما هي الأمور الواجب مراعاتها عند شراء الأجهزة التكنولوجية؟
- أطلب إلى الطلبة تلخيص ما يتوصلون إليه من نتائج على اللوح التفاعلي (Padlet) حول الطرائق الصحيحة للتعامل الآمن مع النفايات الإلكترونية وإعادة تدويرها.
- أشارك الرابط مع الطلبة للعمل التشاركي، والسماح لهم بالنقاش.
- إجابة محتملة: شراء الأجهزة الضرورية فقط، واختيار الأجهزة الموفرة للطاقة التي تمتاز بعمرها الافتراضي الطويل.
- أوجه الطلبة إلى تلخيص أهم المعلومات المتعلقة بإعادة التدوير، والتبرع بالأجهزة.
- أوجه الطلبة لمناقشة ذويهم وعائلاتهم، والتركيز على إعادة التدوير الآمن.

إضاءة

الأردن لإعادة تدوير أجهزة الحاسوب والأجهزة الإلكترونية: مؤسسة حديثة تهدف إلى التعامل الصحيح مع جميع أجهزة الحاسوب والنفايات الإلكترونية التي تضر بالبيئة، واستخدام طرائق احترافية لإعادة التدوير والتجديد؛ ما يسهم في المحافظة على البيئة، وحمايتها من مخاطر التلوث.

وقد أبدت المؤسسة استعدادها لمساعدة مختلف الشركات والمؤسسات على التخلص من النفايات الإلكترونية بصورة آمنة، وزيادة الوعي بالآثار السلبية التي تُخلفها هذه النفايات على البيئة.

أتعرف مزيداً من التفاصيل عن هذه المؤسسة بمسح الرمز سريع الاستجابة (QR Code) الآتي:



المواطنة الرقمية:

أراعي ما يأتي بعد دراسة موضوع (النفايات الإلكترونية):

- الاستخدام المسؤول للتكنولوجيا: أحرص على شراء الأجهزة الإلكترونية الضرورية فقط، وأختار المنتجات التي تُوفّر قدرًا كبيرًا من الطاقة، وتمتاز بعمرها الافتراضي الطويل.
- إعادة التدوير والتبرع: أتبرع بالأجهزة الإلكترونية التي لا تزال تعمل، أو أعيد تدويرها بصورة صحيحة في مراكز إعادة التدوير المُعتمدة.
- التوعية بالمخاطر والتثقيف: أشارك أفراد العائلة والأصدقاء في المعلومات المُتعلقة بمخاطر النفايات الإلكترونية وأهمية إعادة التدوير الآمن لها.

168

■ أذكر الطلبة بدورهم بوصفهم رياديين في المشروع الذي اختير في الدرس السابق، وأن عليهم استكمال المراحل المختلفة للمشروع. يُتطلب إتمام هذا المشروع الريادي مع نهاية الوحدة الدراسية.

■ أوضح للطلبة أن المطلوب في هذه المرحلة استكمال التخطيط والتحضير لتنفيذ المشروع الرقمي الذي يتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة.

■ أوضح للطلبة أن عليهم تجميع الموارد اللازمة، مثل النصوص، والصور، ومقاطع الفيديو، والأدوات التقنية التي ستساعد في تنفيذ المشروع.

■ أوجه الطلبة إلى قراءة تعليمات المهمة المنوطة بهم بدقة، وأقدم لهم النصح والإرشاد عند الحاجة.

■ أؤكد ضرورة عقد اجتماعات دورية مع أعضاء الفريق؛ لمتابعة تقدم العمل في المشروع، وتبادل الأفكار. تساعد هذه الاجتماعات في تعزيز التعاون، وضمان سير المشروع وفق الخطة الموضوعية.

■ أحرص على التأكد من أن كل عضو في الفريق يؤدي المهام المنوطة به وفقًا للخطة الزمنية المرصودة؛ فالتعاون والتنظيم مفتاح نجاح المشروع.

■ أناقش مع الطلبة معايير التقييم الخاصة بالمهمة، وأشاركهم أداة التقييم التي ستستخدم لتقييم عملهم:

• استكمال جميع الموارد المطلوبة (النصوص، والصور، والفيديوهات، والأدوات التقنية).

• مستوى التعاون بين أعضاء الفريق.

• الالتزام بالخطة الزمنية.

• جودة الاجتماعات الدورية ومخرجاتها.

• مستوى الإبداع والابتكار في استخدام الموارد.

المعيار	مؤشر الأداء	نعم	لا	نوعاً ما	ملاحظات
وضوح مراحل المشروع وخطواته	■ وضوح الخطوات والأهداف وارتباطها بالقضية البيئية التي يتناولها المشروع.				
تجميع الموارد اللازمة	■ جمع جميع الموارد المطلوبة (النصوص، الصور، ومقاطع الفيديو، والأدوات التقنية) بشكل كامل.				
التعاون بين أعضاء الفريق	■ تعاون فعال بين أعضاء الفريق، ومراعاة توزيع الأدوار، وتكليف أفراد المجموعة بمهام واضحة.				
الالتزام بالخطّة الزمنية	■ الالتزام بالخطّة الزمنية المحددة لتنفيذ المهام.				
جودة الاجتماعات الدورية	■ الاجتماعات الدورية مفيدة، وأسهمت في تقدم المشروع.				
الإبداع في استخدام الموارد	■ استخدام الموارد بطرائق إبداعية وفعالة لتحقيق أهداف المشروع.				
مقياس الأداء:					
■ نعم: استيفاء الطالب/ الطالبة جميع مُتطلّبات المؤشّر على نحوٍ ممتاز.					
■ لا: عدم استيفاء الطالب/ الطالبة مُتطلّبات المؤشّر.					
■ نوعاً ما: استيفاء الطالب/ الطالبة بعض مُتطلّبات المؤشّر؛ لكنّه يحتاج إلى تحسين في جوانب أُخرى.					

إجابات أسئلة أقيّم تعلّمي

المعرفة: أستخدم ما تعلّمته من معارف في هذا الدرس للإجابة عن الأسئلة الآتية:

■ السؤال الثالث: ما الطرائق الصحيحة والأمنة للتخلص من النفايات الإلكترونية؟

يتم ذلك عن طريق استعادة النفايات الإلكترونية، ومعالجتها، وإعادة تدويرها، أو تجديدها؛ للاستفادة منها، واستخدامها في مناحي الحياة المختلفة مرة أخرى عن طريق إدارة هذه النفايات ومعالجتها باتباع الخطوات الآتية:

- جمع النفايات.
- تفكيك النفايات.
- تنظيف البيانات.
- إعادة التدوير.
- التجديد.
- معالجة النفايات.
- التفكيك.
- المعالجة الميكانيكية.
- التكرير.

المهارات: أوظف مهارات التفكير الناقد، والتواصل الرّقمي، والبحث الرّقمي في الإجابة عن السؤال الآتي:

- أهَيُّ الطلبة لسياق التعلّم بالمشروع، والبدء بتنفيذ الخطة؛ بحيث تتضمن ما يأتي: الأهداف، وأسماء المشاركين، والخطة الزمنية، والإجراءات، وتوزيع المهام، وخطوات التنفيذ، والمصادر والمراجع، والتقييم.
- تحديد بداية تطبيق المشروع، ومتابعة تنفيذ الطلبة له.
- تطبيق المشروع على مستوى المدرسة، وتنفيذ فعاليات تحقق الأهداف المحددة له.

أقيّم تعلّمي

المعرفة: أوظف في هذا الدرس ما تعلّمته من معارف في الإجابة عن الأسئلة الآتية:

السؤال الأول: ما المقصود بالنفايات الإلكترونية؟

النفايات الإلكترونية هي أجهزة إلكترونية قديمة أو شك عمرها الافتراضي على الانتهاء، واستبدل بها أجهزة أخرى جديدة أو حديثة.

السؤال الثاني: ما تأثير النفايات الإلكترونية في البيئة؟

تحتوي النفايات الإلكترونية على مواد خطيرة وسامة مثل: الزئبق، والرصاص (يُمكنها أن تلوث البيئة، وتضرّ بالصحة العامة إذا لم يُحسن التخلص منها بصورة آمنة).

السؤال الثالث: ما الطرائق الصحيحة والأمنة للتخلص من النفايات الإلكترونية؟

أنظر الهامش

المهارات: أوظف مهارات التفكير الناقد والتواصل الرّقمي والبحث الرّقمي في الإجابة عن السؤال الآتي:

بعد الاطلاع على مشروعات إدارة النفايات الإلكترونية المحلية، والتواصل مع أفراد مجموعتي ومُعَلِّمي/ مُعَلِّمتي، أعد خطة مشروع لإدارة النفايات الإلكترونية في المدرسة، وأضمنه ما يأتي:

1. الأهداف.

أنظر الهامش

2. أسماء المشاركين/ المشاركات.

أنظر الهامش

■ تقديم المساعدة للطلبة إن لزم الأمر، والاتفاق معهم على آلية الترويج للمشروع، وكيفية البدء بتطبيقه .

مثال توضيحي لخطة مشروع لإدارة النفايات الإلكترونية في المدرسة

1. الأهداف:

- وعية الطلبة والمعلمين بأهمية إدارة النفايات الإلكترونية.
- تقليل كمية النفايات الإلكترونية في المدرسة عن طريق إعادة التدوير والتخلص الآمن منها.
- تعزيز روح المسؤولية البيئية لدى الطلبة عن طريق المشاركة الفعالة في المشروع .

2. المشاركون:

3. الخطة الزمنية.

أنظر الهامش

4. الإجراءات.

أنظر الهامش

5. توزيع المهام.

أنظر الهامش

6. خطوات التنفيذ.

أنظر الهامش

7. المصادر والمراجع.

أنظر الهامش

8. التقييم.

أنظر الهامش

القيّم والاتجاهات

أنشئ وثيقة باستخدام برمجية (Ms-Word)، ثم أدوّن فيها الخطوات التي أتبعها يومياً للتقليل من بصمتي الكربونية الرقمية.

- الطلبة: مجموعة مختارة من الطلبة من مختلف الفصول الدراسية.
- المعلمون: مشرفو المشروع، ومعلمو المواد العلمية والتقنية.
- إدارة المدرسة: لدعم تنفيذ المشروع، وتوفير الموارد اللازمة.
- مؤسسات محلية: جهات مختصة في إعادة تدوير النفايات الإلكترونية.

3. الخطة الزمنية:

- الأسبوع 1: التحضير والتخطيط، وعقد اجتماع أولي مع المشاركين لتحديد الأهداف وتوزيع المهام.
- الأسبوع 2-3: جمع المعلومات والموارد، والتواصل مع المؤسسات المحلية المتخصصة.
- الأسبوع 4-5: تنفيذ الإجراءات، بما في ذلك جمع النفايات الإلكترونية وفرزها.
- الأسبوع 6: تقييم النتائج، وإعداد تقرير المشروع النهائي.

- حصر الأجهزة الإلكترونية التالفة في المدرسة.
- جمع الأجهزة من الطلبة والمعلمين؛ للتخلص منها أو إعادة تدويرها.
- التواصل مع مؤسسات إعادة التدوير؛ للتأكد من التخلص الآمن من النفايات الإلكترونية.
- إعداد حملات توعية داخل المدرسة عن أهمية إدارة النفايات الإلكترونية.

5. تقسيم العمل بين الزملاء / الزميلات:

- فريق التوعية: مسؤول عن إعداد حملات التوعية وتنفيذها.
- فريق جمع النفايات: مسؤول عن جمع الأجهزة الإلكترونية وفرزها.
- فريق التواصل: مسؤول عن التنسيق مع المؤسسات المحلية وإدارة المدرسة.
- فريق التوثيق: مسؤول عن توثيق جميع الأنشطة، وإعداد التقرير النهائي.

6. خطوات التنفيذ:

- الخطوة 1: عقد اجتماعات دورية مع جميع الفرق لتحديد المهام، والتأكد من سير العمل.
- الخطوة 2: جمع الأجهزة الإلكترونية من أقسام المدرسة المختلفة وفرزها.
- الخطوة 3: إعداد جدول زمني لجمع النفايات الإلكترونية بالتنسيق مع المؤسسات المتخصصة.
- الخطوة 4: تنفيذ حملة توعية شاملة تشمل محاضرات ومواد إعلامية.
- الخطوة 5: تقييم النتائج عن طريق متابعة كمية النفايات التي أُعيد تدويرها أو التخلص منها.

7. المصادر:

- مواقع إدارة النفايات الإلكترونية المحلية.
- منظمات بيئية محلية.
- مواد تعليمية حول أهمية إدارة النفايات الإلكترونية.

8. التقييم:

- مراجعة مدى تحقيق الأهداف المحددة في بداية المشروع.
- تحليل تأثير المشروع في الوعي البيئي لدى الطلبة والمعلمين.
- تقديم توصيات لمشروعات مستقبلية بناءً على نتائج هذا المشروع.
- جمع التغذية الراجعة من المشاركين لتقييم فعالية توزيع المهام وتنفيذ الخطة.

القيم والاتجاهات

أقدم للطلبة تعليمات للقيام بالمهمة، وأحدد لهم موعدًا للتنفيذ والتسليم.
تعليمات القيام بالمهمة المطلوبة:

■ أقدم للطلبة التعليمات لأداء المهمة عن طريق تطبيق ما يأتي:

- أوجه الطلبة لكتابة أهم الأعمال التي يؤديها الطلبة خلال اليوم، ولمدة شهر، وكيف تتكرر.
- أوجه الطلبة لقياس البصمة الكربونية عن طريق المواقع التي تقدّم حاسبة البصمة الكربونية، ورصد الأعمال التي تسبب زيادة البصمة الكربونية.
- أكلّف الطلبة كتابة الوثيقة، مع مراعاة الوضوح في اللغة، وكتابة أهم الأعمال التي تزيد من البصمة الكربونية، وتبين كيفية تقليلها.
- أزود الطلبة ببعض المواقع التي تقدم النصائح لتقليل البصمة الكربونية.
- أمثلة مقترحة:

<https://oceanfdn.org/ar/calculator/reduce>

<https://www.un.org/ar/actnow/ten-actions>

■ أطلب إلى الطلبة مشاركتي الوثيقة عبر البريد الإلكتروني.

الدرس الثالث:

تطبيقات الحاسوب في الحياة اليومية Computer Application in our Daily Life

عدد الحصص المُقترحة:
3 حصص.

المصادر والمراجع:

كتاب الطالب، تطبيقات حاسوبية،
Google Docs , Padlet , Jamboard.

الأدوات والتجهيزات:

قصاصات ورقية لاصقة (Sticky
Notes)، أوراق بيضاء، أقلام ملونة.



الدرس الثالث

تطبيقات الحاسوب في الحياة (Computer Applications in our Daily Life)

الفكرة الرئيسية:

تعرف تطبيقات حاسوبية في مجال التعلم الإلكتروني، والتعلم عن بُعد، والصحة، والتسويق والتسويق الإلكتروني، وغير ذلك من مختلف مجالات الحياة، وبيان أهميتها في الحياة اليومية.

المفاهيم والمصطلحات:

التعلم الإلكتروني (E-Learning)، التعلم عن بُعد (Online Learning)، الحكومة الإلكترونية (E-Government)، الطباعة ثلاثية الأبعاد (3D-Printing).

نتائج التعلم (Learning Outcomes):

- أذكر تطبيقات حاسوبية في مجال التعلم الإلكتروني.
- أستخدم تطبيقات حاسوبية في مجال التعلم عن بُعد.
- أذكر تطبيقات حاسوبية في مجال الصحة، وأبين أهميتها.
- أستخدم تطبيقات حاسوبية في مجال التسويق والتسويق الإلكتروني.
- أبين أهمية تطبيقات الحكومة الإلكترونية في تسهيل المعاملات.
- أوضح أهمية بعض تطبيقات الحاسوب في الحياة، مثل: صناعة الأفلام، والتصميم ثلاثي الأبعاد، والطباعة ثلاثية الأبعاد، والرسوم المتحركة، والوسائط المتعددة.
- أستخدم بعض تطبيقات الحاسوب في تنفيذ مشروع ريادي.

مُنتجات التعلم (Learning Products)

إنتاج المحتوى الرئيس للمشروع
الريادي الرقمي الذي يتناول
القضايا البيئية والاجتماعية
المتعلقة بالحوسبة وفقاً
للخطة الموضوعية، باستخدام
أحد تطبيقات الحاسوب،
ومراجعته، ونشره في العالم
الرقمي.

172

أولاً: التهيئة

سياق التعلم:

- أوضح للطلبة فكرة الدرس الرئيسة، وأنه سيتعرف خلال الدرس إلى التطبيقات الحاسوبية في مجال التعلم الإلكتروني، والتعلم عن بُعد، والصحة، والتسويق والتسويق الإلكتروني، وغير ذلك في مجالات الحياة المختلفة، وبيان أهميتها في الحياة اليومية.
- أبين للطلبة مُنتج التعلم، وأنهم سيعملون على استكمال مشروعهم الريادي، وتنفيذ المرحلة الثالثة فيه؛ إذ سيعملون على إنتاج محتوى المشروع الرئيس الذي يتعلق بإحدى القضايا البيئية والاجتماعية المتعلقة بالحوسبة، باستخدام أحد تطبيقات الحاسوب، بالإضافة إلى مراجعته ونشره في العالم الرقمي.

يستخدم كثير من الأشخاص تطبيقات حاسوبية مُتنوعة؛ كلٌ بحسب حاجاته واهتماماته. وفي ظلّ تطوّر العالم الرقّمي واتّساعه، ظهرت تطبيقات أخرى تتسق مع المستجدات التكنولوجية؛ فما مستقبل تطبيقات الحاسوب؟ وكيف أتخيل العالم الرقّمي في المستقبل القريب والمستقبل البعيد؟

أفكر في الأسئلة الآتية:

- ما التطبيقات الحاسوبية التي أستخدمها في حياتي اليومية؟
 - ما مجالات الحياة التي أستخدم فيها التطبيقات الحاسوبية؟
 - كيف تُسهّم هذه التطبيقات في تسهيل شؤون حياتي؟
- أشارك تجربتي مع زملائي / زميلاتي في الصف، ثمّ أناقشهم في تجاربهم.

نشاط
تمهيدي

تؤدّي التطبيقات الحاسوبية المختلفة اليوم دوراً مهمّاً في إنجاز المهام اليومية على نحوٍ أكثر سرعة وفاعلية؛ سواء كان ذلك في مجال التعلم، أو التسوّق، أو الصحة، أو غير ذلك من المجالات. وقد أسهمت هذه التطبيقات إسهاماً كبيراً في تحسين مختلف مناحي الحياة، وزيادة إنتاجية الأفراد والمؤسسات، وتوفير سبل الراحة في العديد من جوانب الحياة اليومية.

استكشاف التطبيقات الحاسوبية في مختلف مجالات الحياة.

أختار - بالتعاون مع أفراد مجموعتي - واحداً من المجالات الآتية:

التعليم، الصحة، المعاملات الحكومية، التسوّق والتسويق الإلكتروني.

ثمّ أبحث - بالتعاون معهم - عن التطبيقات الحاسوبية المُستخدمة في المجال المختار، وأجمع أمثلة على تطبيقات حاسوبية شائعة في هذا المجال، وأوضّح أهمية استخدام هذه التطبيقات في المجال المختار.

بعد ذلك ألخصّ - بالتعاون معهم - النتائج التي توصّلنا إليها، ثمّ أعدّ معهم عرضاً تقديمياً عن المجال الذي اخترناه، ثمّ أعرض نتائج البحث أمام أفراد المجموعات الأخرى، وأناقشهم فيها.

نشاط
جماعي

173

■ أهَيُّ الطلبة للحديث عن التكنولوجيا، وكيف غزت حياتنا، وعن تطوّر العالم الرّقميّ السريع واتّساعه، وظهور التطبيقات الحاسوبية التي تتسق مع المستجدات التكنولوجية.

■ أطرح على الطلبة الأسئلة الآتية: "ما هو مستقبل التطبيقات الحاسوبية؟"، "كيف تتخيلون المستقبل الرّقميّ القريب والبعيد؟"

■ أستمع إلى الإجابات، ثمّ أسأل عن تطبيقات حاسوبية في حياتهم، لا يمكنهم الاستغناء عنها، وأهَيُّ الطلبة للنشاط التمهيدي، وأوزعهم في مجموعات عمل غير متجانسة لتنفيذه.

- أتيح المجال لبعض الطلبة للتعبير بشكل سريع عن تجاربهم وأفكارهم حول الأضرار والأخطار التي يعرفونها.
- أخبر الطلبة أننا في هذا الدرس، سنتناول واحدة من هذه الأخطار، وهي الجريمة الإلكترونية.

نشاط تمهيدي:

■ أوضّح للطلبة أنّ الهدف من النشاط تعرّف تجاربهم مع التطبيقات الحاسوبية في حياتهم ومجالات استخدامها، وأتيح المجال أمامهم لمشاركة بعضهم بعضاً في التجارب والأفكار.

■ أوجه الأسئلة الآتية التي قد تحفز الطلبة للحديث عن تجاربهم:

- ما التطبيقات الحاسوبية التي أستخدمها في حياتي اليومية؟
- ما مجالات الحياة التي أستخدم فيها التطبيقات الحاسوبية؟
- كيف تسهم هذه التطبيقات في تسهيل شؤون حياتي؟

■ أتيح المجال للطلبة للتعبير عن تجاربهم واستخداماتهم للتكنولوجيا.

■ أسأل الطلبة: "كيف تتخيلون حياتكم من دون توافر هذه التطبيقات؟"

- افسح المجال للتعبير عن التوقعات ومناقشة وجهات النظر.
- أركز في الإجابة على أن التطبيقات الحاسوبية تؤدي دوراً مهماً في إنجاز المهام اليومية على نحو أكثر سرعة وفاعلية، وهي تسهم في تحسين مناحي الحياة المختلفة، وزيادة إنتاجية الأفراد، بالإضافة إلى توفير سبل الراحة في عديد من جوانب الحياة. وأؤكد ارتباط هذه الإيجابيات بالاستخدام المتوازن لهذه التطبيقات.

الربط بالمعرفة السابقة:

- أطلب إلى الطلبة تذكّر ما تعلّموه عن الحوسبة الخضراء وأهميتها وكيفية تطبيقها، وتقليل النفايات الإلكترونية. وأكلفهم تذكّر بعض الإجراءات المتبعة لإدارة النفايات الإلكترونية.
- أوجه النقاش نحو التطبيقات الحاسوبية التي تتعلق بمجالات الحياة، وتطبّق عن طريقها الحوسبة الخضراء.
- أحدد مع الطلبة بعض المجالات المهمة التي تُستخدم فيها التطبيقات الحاسوبية، وأوجههم نحو تطبيق النشاط الجماعي.

ثانياً: عملية التعليم والتعلّم

نشاط جماعي:

- أحدد مع الطلبة المجالات المختلفة للبحث عن تطبيقات حاسوبية تستخدم فيها.
- أكلّف كل مجموعة مجاًلاً معيناً للبحث عن التطبيقات المحوسبة المستخدمة فيه.
- أكلّف الطلبة إعداد عرض تقديمي في إطار المجموعة، يلخص أهم الأفكار حول التطبيقات الحاسوبية، من حيث أهمية استخدامها في هذا المحدد، وأهداف تطبيقها، وأمثلة عليها.
- أتيح المجال للطلبة لعرض ملخصاتهم ومناقشتها مع زملاء.
- أوضح للطلبة أنه سيتناول كل جانب على حدة؛ للوقوف على هذه التطبيقات، وتعرّف أهميتها.

إجابة مقترحة

المجال	أمثلة	التطبيقات المستخدمة	أهمية التطبيقات
التعليم	أنظمة إدارة التعلّم. المنصات التعليمية. التطبيقات التعليمية المحمولة في الهواتف والحواسيب الذكية.	Google Classroom Moodle Microsoft Teams Google Meet Khan Academy Kahoot Flipgrid Quizlet	استدامة التعليم. توفير الوقت والجهد. تحسين جودة التعليم. المرونة. التعلّم الذاتي.

المجال	أمثلة	التطبيقات المستخدمة	أهمية التطبيقات
الصحة	السجل الصحي الإلكتروني. تطبيقات الصحة الرقمية. الذكاء الاصطناعي والتحليل الضخم للبيانات. الروبوتات والأتمتة. الاستشارات عبر شبكة. الإنترنت والتطبيب عن بعد. الطباعة ثلاثية الأبعاد والتخصيص	نظام حكيم. تطبيق Fitbit وتطبيق Apple Health. برنامج تحليل البيانات هدى. منصة Med Jordan . الطباعة ثلاثية الأبعاد للأجهزة التعويضية. المُتخصّصة التي تُعتمد فيها قياسات دقيقة جدًا.	تحسين مستوى الرعاية الصحية، وتمكين المستشفيات والمراكز الصحية من تقديم خدمات أكثر كفاءة وفاعلية، فضلاً عن توفير الوقت والجهد، وتقليل التكاليف والنفقات، والدقة في التشخيص والعلاج، وتقديم أفضل خدمات الرعاية الصحية، لا سيّما في المناطق النائية.
المعاملات الحكومية	إصدار شهادة عدم محكومية. الاستعلام عن دفع المخالفات. الاستعلام عن ضريبة الأبنية. تجديد رخصة المهن ولوحة الإعلانات إلكترونياً. خدمة إصدار شهادة الميلاد المسجلة مسبقاً.	الموقع الرسمي للحكومة الأردنية.	تحسين مستوى تقديم الخدمات. تطوير المهارات. توفير البنية التحتية والتكنولوجية اللازمة لتقديم الخدمات إلكترونياً، وتحسين مستوى أمن المعلومات وحمايتها. رفع إنتاجية القطاع وزيادة كفاءته. تقديم خدمات أفضل للأفراد وقطاع الأعمال. توفير المعلومات المطلوبة بدقة عالية في الوقت المناسب. زيادة عائدات الاستثمار.
التسوق والتسويق الإلكتروني	عمليات البيع والشراء والمعاملات التجارية	تطبيق أمازون. إعلانات جوجل. نظام إدارة علاقات العملاء. مواقع التواصل الاجتماعي. تطبيق السوق المفتوح.	زيادة المنافسة. تحسين تجربة العملاء. زيادة الكفاءة.

مناقشة

أناقش الطلبة في التوجه نحو التحول الرقمي في مجال التعليم، والحاجة الملحة له في أثناء جائحة كورونا التي خلقت واقعاً جديداً، تطلب الانتقال السريع إلى التعلم عن بعد كحل لضمان ديمومة التعليم.

■ أشارك مع الطلبة رابطاً لمنصة بادلت، وأطلب إليهم كتابة فكرة واحدة تدل على مفهوم التحول الرقمي.

■ أناقش الطلبة في الأفكار التي طرحوها، وأربطها بتجربتهم الفعلية للتعلم عن بعد في أثناء فترة الجائحة، وأناقشهم في مزايا التحول الرقمي في التعليم عن طريق ما طبقوه فعلياً.

تطبيقات حاسوبية في مجال التعلم الإلكتروني (E-Learning) ومجال التعلم عن بُعد (Online Learning)

أصبح التحول الرقمي في مجال التعليم ضرورة لا مفر منها في ظل التطورات التقنية المستمرة. وقد بدأ هذا التحول في الظهور منذ استخدام الحاسوب في مجال التعليم خلال عقد التسعينيات من القرن الماضي، ثم تزايدت أهميته أثناء جائحة كورونا التي أفضت إلى واقع جديد تطلب إيجاد حلول تعليمية عن بُعد؛ لضمان ديمومة العملية التعليمية التعلمية.

مزايا التحول الرقمي في التعليم



يمكن إجمال مزايا التحول الرقمي في التعليم في ما يأتي:

1. تعزيز مهارات الطلبة التقنية: يساعد التحول الرقمي الطلبة على اكتساب المهارات التقنية اللازمة لمواكبة التطورات الحديثة في سوق العمل، مثل: مهارات استخدام الحواسيب، والبرمجة، والتعامل مع البرمجيات المختلفة.
2. تسهيل الوصول إلى المعلومة: يتيح استخدام التكنولوجيا الوصول إلى المعلومات بسهولة وسرعة، إذ يمكن للطلبة والمعلمين/المعلمات الاطلاع على الموارد التعليمية عبر شبكة الإنترنت في مختلف الأحوال والأماكن والأوقات.
3. المرونة في عملية التعلم والتعليم: يمتاز التعليم الرقمي بمرونة كبيرة، تتيح للطلبة والمعلمين/المعلمات تحديد أوقات الدراسة والتدريس التي تناسبهم، فضلاً عن إتاحة المجال أمام الطلبة للتعلم بالوتيرة التي تفي بحاجاتهم، وتراعي أحوالهم؛ ما يعزز جانب الفهم لديهم.

174

استخدام الأشكال والصور:

- أوجه الطلبة لمشاهدة الشكل، وأناقش معهم أهم مزايا التحول الرقمي في قطاع التعليم.
- أطرح أمثلة توضيحية حول مزايا التحول الرقمي في قطاع التعليم، مستوحياً أمثلة من مشاركات الطلبة وتجاربهم.
- أطرح السؤال الآتي على الطلبة: "هل يقتصر التحول الرقمي على إدخال التكنولوجيا في الغرف الصفية؟ ما الذي يترتب على ذلك؟"

ستنوع الإجابات، ويمكن التركيز على البنود الآتية:

لا يقتصر التحول الرقمي فقط على إدخال التكنولوجيا في الغرف الصفية. إنه عملية شاملة تؤثر في جميع جوانب التعليم، بدءاً من المناهج الدراسية والتقييم، وصولاً إلى البنية التحتية المدرسية، والتواصل مع المجتمع.

ما يترتب على التحول الرقمي:

- **تغيير أساليب التدريس:** يتطلب التحول الرقمي إعادة التفكير في طرائق التدريس التقليدية. ويجب على المعلمين تطوير مهارات جديدة لاستخدام الأدوات التكنولوجية بفعالية، وتعزيز التعلم القائم على التكنولوجيا.
- **تخصيص التعليم:** عن طريق التحول الرقمي، يمكن تخصيص التعليم وفقاً لاحتياجات كل طالب. ويمكن للمعلمين استخدام البيانات لتحديد نقاط القوة والضعف لدى الطلبة، وتوفير موارد تعليمية مخصصة.
- **تعزيز الكفاءة:** يمكن أن تجعل التكنولوجيا العمليات المدرسية أكثر كفاءة، مثل إدارة الوقت، وإعداد الدروس، وتقييم الطلبة. وتساعد الأنظمة الرقمية في أتمتة العديد من المهام الإدارية.
- **إتاحة التعلم المرن:** مع التحول الرقمي، يمكن للطلبة الوصول إلى المحتوى التعليمي في أي وقت، ومن أي مكان. يفتح هذا أبواباً جديدة للتعلم الذاتي والتعلم عن بعد.
- **تحسين الاتصال والتواصل:** تعزز التكنولوجيا التواصل بين المعلمين، والطلبة، وأولياء الأمور عن طريق منصات التعليم الإلكتروني، والبريد الإلكتروني، والتطبيقات التفاعلية.
- **تحفيز الابتكار:** يشجع التحول الرقمي على الابتكار داخل المدرسة؛ إذ يمكن للمعلمين والطلبة استكشاف أدوات وبرامج جديدة؛ لتحسين التجربة التعليمية.

العواقب المترتبة على التحول الرقمي:

- **زيادة الحاجة للتدريب:** سيحتاج المعلمون والموظفون إلى تدريب مكثف لمواكبة التغييرات التكنولوجية.
- **الفجوة الرقمية:** يمكن أن تتوسع الفجوة بين الطلبة الذين لديهم وصول إلى التكنولوجيا، والذين ليس لديهم هذا الوصول؛ مما يخلق تحديات في تحقيق المساواة في التعليم.
- **الاعتماد على التكنولوجيا:** قد يؤدي الاعتماد المفرط على التكنولوجيا إلى إهمال الجوانب التقليدية من التعليم التي ما زالت ذات قيمة.

- أوزع الطلبة في مجموعات، وأكلفهم كتابة أهمية التحول الرقمي على ورقة بيضاء (Flipchart).
- أكلف كل مجموعة عرض ما توصلت إليه، وإجمال أهمية التحول الرقمي في قطاع التعليم في ثلاث نقاط وهي: استدامة التعليم، وتوفير الوقت، وتحسين جودة التعليم.

أدوات التحول الرقمي:

■ أسأل الطلبة عن الأدوات الرقمية التي يستخدمونها، وسبب استخدامها لها.

■ أناقش الطلبة في أدوات التحول الرقمي في التعليم الواردة في الدرس، متيحاً لهم المجال للتعبير، وذكر ما يعرفونه من أدوات.

■ أخص ما يورده الطلبة، وأتوصل معهم لأبرز الأدوات التي تُوظف في قطاع التعليم، ويمكن إبرازها في ما يأتي: أنظمة إدارة التعلم، والمنصات التعليمية، والتطبيقات التعليمية المحمولة في الهواتف والحواسيب الذكية.

■ رسم خريطة ذهنية للطلبة عن أدوات التحول الرقمي في قطاع التعليم، وأطلب إليهم التعرف إلى التطبيقات التي تساعد في عملية التعليم، وأكلفهم تصنيفها بحسب الأدوات:

يوجد العديد من أدوات التحول الرقمي في التعليم، ويمكن إجمال أبرزها في ما يأتي:

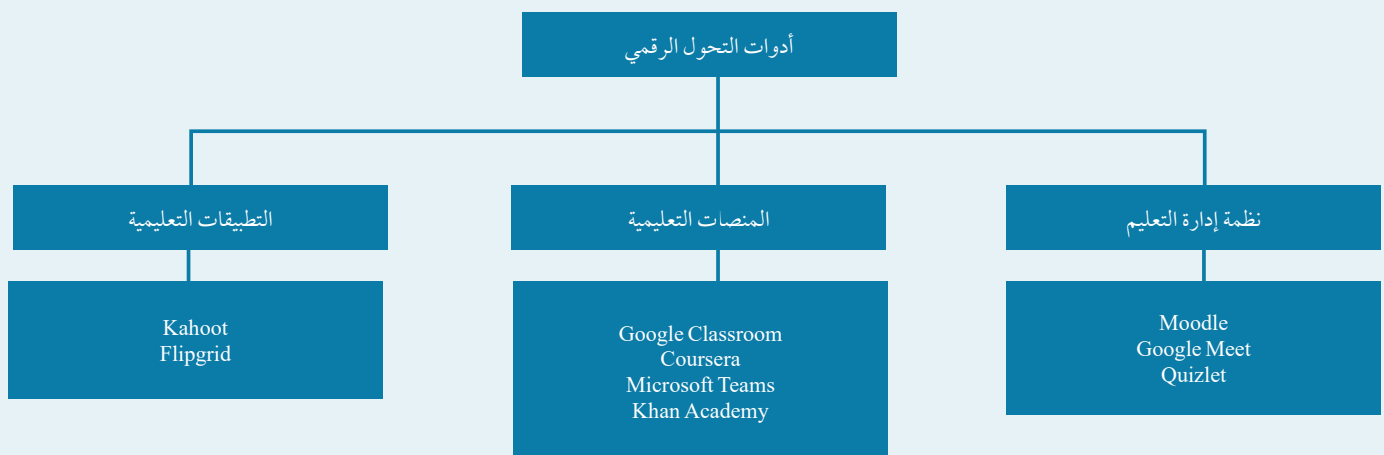
1. أنظمة إدارة التعلم (Learning Management Systems: LMS): تُعرف أنظمة إدارة التعلم بأنها برامج حاسوبية مُصمَّمة لإدارة عملية التدريب والتعليم ومتابعتها وتقييمها.
2. المنصات التعليمية: تُوفّر هذه المنصات دورات تعليمية عبر شبكة الإنترنت.
3. التطبيقات التعليمية المُحمّلة في الهواتف والحواسيب الذكية: تُسهّل هذه التطبيقات عملية الوصول إلى المواد التعليمية، وتساعد الطلبة على التعلم الذاتي.

من الأمثلة الشائعة على التطبيقات الحاسوبية في هذا مجال التعلم الإلكتروني:

1. (Google Classroom): منصة تعليمية تُعزّز سبل التواصل والتعاون بين المُعلِّمين / المُعلَّمتات والطلبة، وتتيح للمُعلِّمين / المُعلَّمتات إنشاء صفوف افتراضية، ودعوة الطلبة إلى الانضمام إليها. وكذلك مشاركة الموارد التعليمية والواجبات، وإدارة النقاشات، وإجراء التقييمات إلكترونياً، فضلاً عن متابعة الطلبة وتوجيههم وإرشادهم.
2. (Moodle): نظام لإدارة التعلم مفتوح المصدر. وفيه يُقدّم العديد من الدروس والموارد التعليمية عبر شبكة الإنترنت.
3. (Microsoft Teams): منصة للتعلم الإلكتروني والتواصل بين المجتمعات المختلفة. وفيها يُمكن للمستخدم إجراء محادثات نصية ومرئية وصوتية، وعقد اجتماعات عبر شبكة الإنترنت. كذلك تتيح المنصة للمستخدم مشاركة الموارد، وإدارة العديد من المهام، وهي تُعنى أساساً بتقديم خدمات تعليمية وتربوية.
4. (Google Meet): أداة لعقد الاجتماعات والمحاضرات الافتراضية، وهي تدعم التفاعل المباشر بين المُعلِّمين / المُعلَّمتات والطلبة.
5. (Coursera): منصة تُقدّم دورات تدريبية عبر شبكة الإنترنت بالتعاون مع جامعات عالمية؛ ما يمنح الطلبة تعليماً فريداً يَخُصّ النظر عن المكان والزمان.
6. (Khan Academy): منصة تُقدّم دورات تعليمية مجانية عبر شبكة الإنترنت في مجموعة مُتنوعة من الموضوعات.
7. (Kahoot): تطبيق يتيح للطلبة إنشاء ألعاب ومسابقات تعليمية تفاعلية، ثمّ مشاركتها عن طريق أجهزة الهواتف الذكية والأجهزة اللوحية وأجهزة الحاسوب. كذلك يعرض التطبيق النتائج والترتيب العام للمتسابقين بعد كل سؤال، ويتيح للطلبة الاندماج في العملية التعليمية التفاعلية عن طريق اللعب التفاعلي. ويُعدّ التطبيق أداة شائعة للتعلم النشط.
8. (Flipgrid): تطبيق يتيح للمُعلِّمين / المُعلَّمتات والطلبة تسجيل مقاطع فيديو قصيرة لمشاركة الأفكار والمناقشات؛ ما يُعزّز سبل التفاعل والنقاش داخل الغرف الصفية الافتراضية.

176

الإجابة المحتملة:



■ أناقش الطلبة في أهمية هذه الأدوات ودورها في عملية التعليم والتعلم.

■ أفسح المجال للطلبة للحديث عن استخداماتهم لبعض هذه التطبيقات.

البحث والمشاركة

■ أوجه الطلبة إلى تنفيذ مهمة فردية بيتية، تتمثل في البحث عن تطبيقات أخرى تستخدم في مجال التعلم عن بعد، ومجال التعلم الإلكتروني.

■ أوجه الطلبة إلى مشاركة إجاباتهم/ والنقاش بعضهم مع بعض عن طريق استخدام اللوح التفاعلي (Jamboard).

■ أشارك رابط اللوح التفاعلي الخاص بالنشاط مع الطلبة.

■ أوجه الطلبة للتفاعل مع مداخلات الزملاء، وأتابع تنفيذهم لمهمة البحث، وأعطي التغذية الراجعة لهم.

أثناء للمعلم/ المعلمة

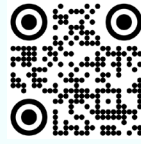
في ما يأتي رابط لموقع إلكتروني، يتضمن أمثلة على تطبيقات حاسوبية ذات صلة بقطاع التعليم.

9. (Quizlet): أداة تعليمية تتيح للمعلمين/ للمُعلّّمت والطلبة إنشاء بطاقات تعليمية، واختبارات، وألعاب تعليمية، تحسّن عمليتي الفهم والتذكر، وتدعم مجموعة مُتنوّعة من الموضوعات.

أبحث

أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن تطبيقات حاسوبية أخرى تُستخدم في مجال التعليم عن بُعد، ومجال التعلم الإلكتروني، ثم أشارك ما أتوصّل إليه من نتائج مع زملائي/ زميلاتي في الصف عن طريق اللوح التفاعلي (Jamboard).

نشاط فردي



أزور منصّة التعلم الأردنية (JoLearn) عن طريق الرابط الإلكتروني الآتي:

<https://jolearn.jo>

أو مسح الرمز سريع الاستجابة (QR Code) المجاور:

ثم أبحث في كيفية الدخول إلى حسابي في الصفحة الرئيسة للمنصة؛ لاستكشاف الموارد التعليمية الإلكترونية التي تقدّمها المنصة لي بوصفي طالباً/ طالبة، ثم أشارك ما أتوصّل إليه من نتائج مع زملائي/ زميلاتي في الصف.

نشاط فردي

التسجيل في مساق إلكتروني.

1. أختار إحدى المنصات التعليمية، مثل: منصّة إدراك (Google)، أو منصّة (Coursera)، أو منصّة (Udacity).

2. أبحث عن مساق إلكتروني أو دورة تدريبية ذات صلة بموضوعات الوحدة الدراسية، مثل: الحوسبة الخضراء، وإدارة النفايات الإلكترونية.

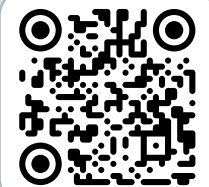
3. أسجل في المساق، ثم أبدأ رحلة التعلم.

4. أنهي جميع الدروس والمهام المطلوبة في المساق.

5. أشارك تجربتي مع زملائي/ زميلاتي في الصف.

6. أناقش زملائي/ زميلاتي في إيجابيات المساق والتحديات التي واجهتها أثناء عملية التعلم.

رابط مختصر: <https://short-link.me/L3ow>



نشاط فردي:

■ أوجه الطلبة لتنفيذ النشاط الفردي، وهو الدخول إلى منصة التعلم الإلكتروني Jolearn

■ خارج أوقات الحصة، وأقدم لهم الدعم المناسب عند الحاجة.

■ أوجه الطلبة لاستكشاف المنصة وكتابة إيجابياتها وسلبياتها، واقتراح التحسينات من وجهة نظرهم، ومشاركتها عبر اللوح التفاعلي.

■ أوجه الطلبة للتفاعل مع مداخلات الزملاء

- أوضح للطلبة المطلوب من النشاط الفردي البيتي.
- أوضح لهم كيفية التسجيل في منصات التعلم والمسابقات.
- أقدم لهم الدعم المناسب عند الحاجة.
- أناقش معهم المدة التي يحتاجها المساق، ومن ثم أئفّق معهم على موعد يناسب الجميع؛ للحدّث عن تجربتهم، وإيجابيات المساق وسلبياته.

مناقشة

- اسأل الطلبة عن زيارتهم لمركز صحي، أو وزارة الصحة من قبل، وهل استخدم أحدهم التطبيقات الحاسوبية لوزارة الصحة.
- أناقش الطلبة في الخدمات المتوقعة التي يمكن أن تقدمها التطبيقات الحاسوبية في مجال الصحة.
- أوجه النقاش نحو أهمية التحوّل الرّقميّ في مجال الصحة والهدف منه.
- الإجابة المحتملة: يسهم التحوّل الرّقميّ في مجال الصحة في تحسين مستوى الرعاية الصحية، وتمكين المستشفيات والمراكز الصحية من تقديم خدمات أكثر كفاءة وفاعلية، فضلاً عن توفير الوقت والجهد، وتقليل التكاليف والنفقات، وتوخي

تطبيقات حاسوبية في مجال الصحة

يشهد قطاع الرعاية الصحية في العالم تحوّلاً تقنيّاً مُهمّاً؛ ما جعل مفهوم التحوّل الرّقمي ركيزة أساسية لتطوير القطاع الصحي مسبقاً، وغدت تطبيقات التحوّل الرّقمي في هذا القطاع أكثر تنوعاً وشمولاً، وهو ما أسهم بفاعلية في تحسين مستوى الرعاية الصحية، ومكّن المستشفيات والمراكز الصحية من تقديم خدمات أكثر كفاءة وفاعلية، فضلاً عن توفير الوقت والجهد، وتقليل التكاليف والنفقات، وتوخي الدقّة في التشخيص والعلاج، وتقديم أفضل خدمات الرعاية الصحية، لا سيّما في المناطق النائية.

في ما يأتي بعض الأمثلة على هذه التطبيقات:

1. السجل الصحي الإلكتروني: تعمل تطبيقات السجل الصحي الإلكتروني على تخزين المعلومات الطبية ومشاركتها بين مُقدّمي الرعاية الصحية بصورة آمنة؛ ما يتيح للأطباء الوصول السريع إلى بيانات المرضى، ومُمكنهم من توخي الدقّة في التشخيص واختيار العلاج المناسب. يُعدّ نظام حكيم (Hakeem) في الأردن واحداً من الأمثلة على السجلات الصحية الإلكترونية؛ إذ يُوفّر سجلات طبية إلكترونية متكاملة، تُسهّل على الأطباء الوصول إلى معلومات المرضى على نحوٍ سريع وآمن؛ ما يزيد من دقّة التشخيص وفاعلية العلاج. يُسهم هذا النظام أيضاً في تحسين إدارة الرعاية الصحية، وتقليل الأخطاء الطبية.
2. تطبيقات الصحة الرقمية: تشمل تطبيقات الصحة الرقمية تتبّع اللياقة البدنية، والصحة الذكية، والمراقبة الذاتية للصحة؛ إذ تُمكن المرضى من متابعة حالتهم الصحية بأنفسهم، والتفاعل مع مُقدّمي الرعاية الصحية بصورة أفضل. ومن أمثلتها: تطبيق (Fitbit)، وتطبيق (Apple Health) التي يتيح تتبّع النشاط البدني والنوم، ومراقبة الصحة العامة.
3. الذكاء الاصطناعي والتحليل الضخم للبيانات: يُمكن للذكاء الاصطناعي والتحليل الضخم للبيانات مساعدة المؤسسات الصحية على تحليل كمّ البيانات الهائل، والكشف عن الأنماط الصحية المُتعدّدة، فضلاً عن الإسهام في تحري



الدقّة في التشخيص والعلاج، وتقديم أفضل خدمات الرعاية الصحية، لا سيّما في المناطق النائية.

- أسترّض مع الطلبة بعض التطبيقات التي تُستخدم في المجال الصحي، والخدمات التي توفرها هذه التطبيقات.
- أركز على استخدامات هذه التطبيقات وبعض الأمثلة عليها، ثم أوجه الطلبة إلى تنفيذ النشاط الفردي.

نشاط فردي

- أكلف الطلبة مهمةً فرديةً بيتية، تتمحور حول تصفح تطبيق حكيم، وجمع معلومات حوله.
- أتيح لهم المجال للبحث، وتلخيص أهم النقاط بحسب التعليمات الواردة.
- اتفق معهم على موعد لعرض نتائج البحث أمام زملائهم.

نشاط بحثي

- أكلف الطلبة مهمةً فرديةً بيتية، تتمحور حول البحث عن تطبيقات أخرى في مجال الصحة.
- أوجه الطلبة إلى مشاركة إجاباتهم، والنقاش بعضهم مع بعض باستخدام اللوح التفاعلي (Padlet).
- أشارك رابط اللوح التفاعلي الخاص بالنشاط مع الطلبة.
- أوجه الطلبة للتفاعل مع مداحلات الزملاء، وأتابع تنفيذهم مهمة البحث، وأعطي التغذية الراجعة لهم.
- الإجابات المحتملة:

- موقع وزارة الصحة:

<https://www.moh.gov.jo/Default/Ar>



أبحث

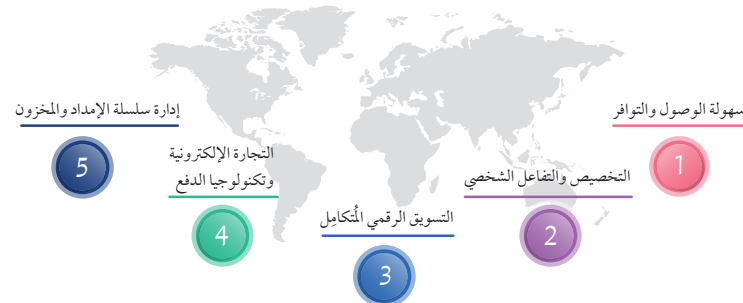
أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن أمثلة أخرى على التطبيقات الحاسوبية في مجال الصحة، ثم أشارك ما أتوصل إليه من نتائج مع زملائي / زميلاتي في الصف عن طريق اللوح التفاعلي (Jamboard).

تطبيقات حاسوبية في مجال التسوق والتسويق الإلكتروني

يشهد العالم اليوم تزايداً ملحوظاً في استخدام التكنولوجيا الرقمية في عمليات البيع والشراء والتسوق والتسويق عبر شبكة الإنترنت، والهواتف الذكية، ووسائل التواصل الاجتماعي، والتطبيقات المخصصة لذلك؛ ما يساهم في دفع عجلة الاقتصاد، والنهوض بحركة التجارة، لا سيما في ظل ارتفاع أسعار الوقود، وصعوبة الوصول إلى المتاجر التقليدية، وتعذر زيارة الأسواق المحلية البعيدة والأسواق العالمية.

مزايا التحول الرقمي في مجال التسوق والتسويق الإلكتروني

مزايا التحول الرقمي في مجال التسوق والتسويق الإلكتروني



أخذ التسوق والتسويق الإلكتروني يُنافس الطرائق التقليدية في عمليات البيع والشراء والمعاملات التجارية؛ نظراً إلى ما يتصف به من مزايا، أبرزها:

1. سهولة الوصول والتوافر:
أ. إمكانية التسوق الدائم للمستهلكين والعملاء؛ ما يوفر عليهم الوقت والجهد، ويهيئ لهم سبل الراحة والدعة.
- ب. تمكين المنصات الإلكترونية المتاجر من التعامل مع عدد كبير من المستهلكين والعملاء دون حاجة إلى إنشاء بنية تحتية مادية.

- بوابة الخدمات الإلكترونية للأفراد سند:

<https://signflow.sanad.gov.jo/>



- أوزع الطلبة في مجموعات عمل غير متجانسة.
- أناقشهم حول تجاربهم في عملية الشراء عبر الإنترنت.
- أتيح المجال لهم للتعبير عن التجربة، وكيف تمت عملية الشراء والدفع.
- أسألهم عن التطبيقات التي استخدمت للتسوق الإلكتروني، وكيف كانت طبيعة عملية التسوق.
- أناقشهم في المجالات الأخرى التي استخدمت فيها تطبيقات التسوق غير عملية الشراء.
- أستعرض معهم مزايا التحول الرقمي في مجال التسوق والتسويق الإلكتروني.

2. التخصيص والتفاعل الشخصي:
 - أ. استخدام البيانات الضخمة وتحليلات العملاء في تقديم توصيات مُتخصّصة وعروض خاصة بناءً على تفضيلات العملاء وسلوكياتهم.
 - ب. الاستفادة من الدردشة التفاعلية وخدمات العملاء عبر شبكة الإنترنت في تقديم دعم فوري للعملاء.
 3. التسويق الرقمي المُتكامل:
 - أ. استخدام استراتيجيات التسويق الرقمي (مثل: تحسين مُحركات البحث (SEO)، والتسويق عبر البريد الإلكتروني، والتسويق عبر وسائل التواصل الاجتماعي، والإعلانات المدفوعة) في الترويج وجذب مزيد من العملاء.
 - ب. إسهام التحليلات الرقمية في قياس فاعلية حملات التسويق وضبطها؛ لتحقيق أفضل النتائج.
 4. التجارة الإلكترونية وتكنولوجيا الدفع:
 - أ. تمكين المواقع الإلكترونية وتطبيقات التسوق العملاء من شراء المُنتجات بسهولة عبر شبكة الإنترنت.
 - ب. استخدام تقنيات الدفع الرقمية (مثل: المحافظ الإلكترونية، وبطاقات الائتمان، والتحويلات البنكية) في المعاملات التجارية؛ ما يجعلها أكثر سهولة وأماناً.
 5. إدارة سلسلة الإمداد والمخزون:
 - أ. استخدام الأنظمة الرقمية في إدارة المخزون وتتبع الشحنات؛ ما يزيد من الفاعلية والكفاءة، ويعمل على تخفيض الكُلف التشغيلية.
 - ب. تتبع الطلبات بصورة مباشرة؛ ما يُمكن العملاء من تعرّف سير الإجراءات التي تمرُّ بها طلباتهم، وتحديد الإجراء الذي وصلت إليه بدقة.
- تأثير التحول الرقمي في مجال التسوق والتسويق الإلكتروني:
- في ما يأتي أبرز آثار التحول الرقمي في مجال التسوق والتسويق الإلكتروني:
1. زيادة المنافسة: يُسهّل التحول الرقمي على المُنافسين الجُدد دخول السوق، ما يزيد من وتيرة التنافس، ويُحفّز الشركات على تحسين خدماتها ومُنتجاتها.
 2. تحسين تجربة العملاء: يُعزّز التفاعل الفوري والتخصيص من رضا العملاء، ويزيد من ولائهم للعلامة التجارية.
 3. زيادة الكفاءة: يؤدي استخدام الأنظمة الرقمية في إدارة العمليات إلى التقليل من الأخطاء، وزيادة كفاءة العمليات التشغيلية.

- أكلفهم الاطلاع والبحث عن التطبيقات الإلكترونية في مجال التسويق والتسوق.
- أناقشهم في تأثير التحول الرقمي في مجال التسويق والتسوق الإلكتروني، وأركز على الجوانب الآتية: **زيادة المنافسة، وتحسين تجربة العملاء، وزيادة الكفاءة.**
- أركز على كل تأثير، وكيف يحسّن عملية التسوق والتسويق ويعززها.

نشاط: ⚙️

- أكلف الطلبة نشاطاً فردياً بيتياً، يتمحور حول تصفح الإنترنت، وفتح موقع السوق المفتوح.
- أتيح لهم المجال لتصفح الموقع، والتعرف إليه، وإلى الخدمات التي يقدمها: مثل السلع المتوفرة في الموقع، ومزايا الموقع، ومزايا السلع المتوفرة، والفئة المستهدفة.
- أناقش الطلبة في الفرق بين عمليتي التسوق الإلكتروني والتسوق العادي، ولماذا صار التحول والتوجه أكثر نحو التسوق الرقمي.
- أسأل الطلبة: هل توجد تحديات أمام عملية التسوق الرقمي، وكيف يمكن مواجهتها؟

التحديات: الفترة الزمنية، وتفاوت الأسعار، واختيار مواقع التسوق المناسبة، وعدم تطابق السلع، واسترداد المشتريات وإرجاعها.

- أمثلة على التطبيقات والتحويلات الرقمية في مجال التسوق والتسويق الإلكتروني:
1. أمازون (Amazon): منصة تجارة إلكترونية وحوسبة سحابية، تُقدم حلولاً متنوعة؛ للوفاء بحاجات العملاء في مختلف دول العالم، وتعرض تجربة تسوق شاملة، تتضمن توصيات متخصصة، وتعليقات للعملاء، وخيارات شحن متنوعة.
 2. إعلانات جوجل (Google Ads): أدوات تسويق رقمي، تتيح استهداف الجمهور بدقة عن طريق الإعلانات المدفوعة التي تظهر للمستخدمين بناءً على اهتماماتهم وسلوكياتهم في شبكة الإنترنت.
 3. نظام إدارة علاقات العملاء (Salesforce: CRM): يساعد هذا النظام الشركات على تتبع تفاعلات العملاء وتحليلها، وإدارة حملات التسويق والمبيعات بفاعلية.
 4. مواقع التواصل الاجتماعي: شاع في الآونة الأخيرة إنشاء التجار - الذين يملكون متاجر حقيقية - صفحات للبيع والشراء الإلكتروني في مواقع التواصل الاجتماعي، مثل: صفحات فيسبوك (Facebook)، وإنستغرام (Instagram)؛ نظراً إلى سهولة التواصل الدائم مع المستهلكين والعملاء في هذه المواقع، علماً بأن ذلك لا يقتصر فقط على تسويق الملابس والمواد الغذائية، وإنما يمتدّاه إلى خدمات النقل، وحجوزات الرحلات، والترفيه، وغير ذلك.
 5. تطبيق السوق المفتوح (OpenSooq): يُعدّ السوق المفتوح أكبر تطبيق للإعلانات المُبَوَّبة باللغة العربية؛ إذ يتيح هذا التطبيق لملايين المستخدمين تنفيذ عمليات بيع وشراء للعديد من المُنتَجات والخدمات عبر شبكة الإنترنت من دون وسيط، ويُمكن المشتريين من مشاهدة السلع والخدمات المعروضة، مثل: السيارات، والعقارات، والإلكترونيات، والأثاث.

أزور الموقع الإلكتروني للسوق المفتوح عن طريق الرابط الإلكتروني الآتي:

<https://jo.opensooq.com/ar>



أو مسح الرمز سريع الاستجابة (QR Code) المجاور، ثمّ أستمع عرض السلع المتوفرة في الموقع، وأستكشف أهم مزايا الموقع في ما يخصّ مجال التسوق، والفئات التي يستهدفها.



نشاط

- أوضح للطلبة أن هذه التحديات تعتمد على سياسة هذه التطبيقات، وكيفية اختيار التطبيق المناسب للفئة المستهدفة.

ملاحظات

تطبيقات الحكومة الإلكترونية

أولى الأردن عملية التحول الرقمي اهتمامًا كبيرًا، وتمثل ذلك في أتمتة الخدمات الحكومية المُقدَّمة للمواطنين، بالإعلان عن برنامج الحكومة الإلكترونية عام 2001م، الذي أُطلق برعاية ملكية سامية، وكُلِّفت وزارة الاتصالات وتكنولوجيا المعلومات بتنفيذه وقتئذٍ، ثم تولّت إكماله اليوم وزارة الاقتصاد الرقمي والريادة، بتوفيرها عددًا من التطبيقات للهواتف الذكية، ومجموعة من القنوات الرقمية عبر شبكة الإنترنت؛ بُغية إنجاز المعاملات الحكومية التي تخصّ المواطنين، أنظر الشكل (1-3) الذي يُبين أهداف برنامج الحكومة الإلكترونية.



الشكل (1-3): أهداف برنامج الحكومة الإلكترونية.

في ما يأتي أبرز الخدمات التي تُقدِّمها الحكومة الإلكترونية للمواطنين:

1. إصدار شهادة عدم محكومية: تتيح هذه الخدمة للمواطنين إصدار شهادة عدم المحكومية إلكترونياً، وإمكانية تقديم الطلب والدفع بصورة إلكترونية.
2. الاستعلام عن دفع المخالفات: تُوفّر هذه الخدمة قناة إلكترونية تُمكنّ المواطنين والمقيمين من الاستعلام عن مخالفات المركبات، ودفع قيمتها إلكترونياً، والاطّلاع على تفاصيل كل مخالفة منها.
3. الاستعلام عن ضريبة الأبنية (المُسَقَّفات): تُوفّر هذه الخدمة الاستعلام عن ضريبة الأبنية، ودفع قيمتها إلكترونياً.
4. تجديد رخصة المهن ولوحة الإعلانات إلكترونياً: تُوفّر هذه الخدمة قناة إلكترونية تُمكنّ أصحاب رخص المهن والأعمال الحرة من تجديد رخصهم، ودفع رسومها إلكترونياً، إضافة إلى تسلم هذه الرخص إما عن طريق البريد الأردني، وإما شخصياً.
5. خدمة إصدار شهادة الميلاد المُسجَّلة مُسبقاً.

■ أقسم الطلبة إلى مجموعات.

■ أوجه الطلبة لقراءة النص في الكتاب وتأمل الشكل (1-3) الذي يبين أهداف برنامج الحكومة الأردنية، وكتابة الأفكار الرئيسة على قصاصات ورقية.

■ أوجه المجموعات للنقاش في ما بينهم حول أهداف برنامج الحكومة الإلكترونية وأبرز خدماتها.

■ أختار طالباً من كل مجموعة لعرض أفكارهم ومناقشتها مع بقية المجموعات، والتركيز على أهمية وجود تطبيقات تسهم في إنجاز المعاملات الحكومية للمواطنين خاصة مع التحول الرقمي، والحاجة إلى أتمتة الخدمات الحكومية للمواطنين.

■ أوجه الطلبة لتوقع خدمات أخرى تقدمها الحكومة إلكترونياً للمواطنين، وأوجههم للتأكد من توافرها عن طريق استكشاف موقع الحكومة الإلكترونية.

ملاحظات

■ أوجه الطلبة لتطبيق النشاط بوصفه مهمة بيتية فردية، تهدف لتعرّف موقع الحكومة الأردنية واستكشافه.

■ أزود الطلبة بالتعليمات، وأكلفهم إجابة الأسئلة المطروحة بالتزامن مع تصفّحهم للموقع.

الإجابات المحتملة:

1. هل يستطيع السائح أو المُستثمر استخدام هذا الموقع؟ أوضح ذلك.

يستطيع المستثمر استخدام الموقع لتعرّف جميع الأنشطة التي تتعلق ببدء نشاطه التجاري، وإدارته وإنهائه، بالإضافة إلى توضيح الاحتياجات كلها التي تلزمه في كل مرحلة، مع وجود رابط يحوّل المستثمر إلى الجهات المختصة

بكل خطوة في أثناء تنفيذه المشروع. ويتوافر لدى المستثمر الخيارات الثلاثة، وتُفصّل كل مرحلة من بداية تسجيله للمشروع التجاري إلى إنهاء المشروع.

السائح: يساعد الموقع السائح على تعرّف الأماكن التي يمكنه زيارتها، وكيفية تنظيم زيارته للأردن عن طريق تقديم معلومات حول الحصول على تأشيرة والإقامة، وكيفية التنقل، وأرقام السفارات، بالإضافة إلى الأمور المهمة من أجل مغادرة الأردن، ويُفضل أن يتصفح السائح أو المستثمر هذا الموقع؛ لمعرفة ما يترتب عليه من مسؤوليات وأمور لإنجازها في أثناء فترة السياحة. أما بقية الخدمات، فلا يمكن للسائح أو المستثمر الحصول عليها؛ لأنها تتطلب الرقم الوطني، وكلمة المرور الخاصة به

2. ما أهم الخدمات التي تقدمها صفحة المواطن في الموقع؟

يقدم الموقع للمواطن بعض الخدمات التي تتعلق بالعائلة، والتعليم، والعمل، والتقاعد، والسفر، والشيخوخة، وخدمات

أزور الموقع الإلكتروني الرسمي للحكومة الإلكترونية:

[/https://portal.jordan.gov.jo/wps/portal/Home](https://portal.jordan.gov.jo/wps/portal/Home)

ثم أجب عن الأسئلة الآتية بعد تصفّح الموقع:

1. هل يستطيع السائح أو المُستثمر استخدام هذا الموقع؟ أوضّح ذلك.
2. ما أهم الخدمات التي تُقدّمها صفحة المواطن في الموقع؟
3. استكشف الخدمات التي يُمكن للحكومة الإلكترونية أن تُقدّمها لي - بوصفي طالبًا- بعد دخولي على صفحة الخدمات، ثم الضغط على خيار مواطن، ومنه على خيار التعليم كما في الشكل الآتي:



تطبيقات حاسوبية للوسائط المتعدّدة

يوجد العديد من التطبيقات الحاسوبية الخاصة بإعداد الوسائط المتعدّدة وتطويرها وطباعتها. وهذه أبرزها:

صناعة الأفلام

تُعَدُّ برامج صناعة الأفلام وتحريرها (Movie Maker) أحد أكثر التطبيقات الحاسوبية انتشارًا في الهواتف المحمولة وأجهزة الحاسوب. أما الجديد في هذا المجال فهو تدخل الذكاء الاصطناعي

خاصة بذوي الاحتياجات الخاصة. ويتفرع منها جميع الخدمات التي تلزم المواطن في أثناء انخراطه في الحياة في الأردن.

3. أستكشفُ الخدمات التي يمكن للحكومة الإلكترونية أن تقدمها لي بوصفي طالباً بعد دخولي على صفحة الخدمات، ثمَّ الضغط على خيار مواطن، ثمَّ على خيار التعليم.

■ إصدار وثيقة معادلة أصلية بدل مفقود/ إصدار نسخة إضافية أصلية.

■ الاستفسار عن الاعتراف بمؤسسات التعليم العالي غير الأردنية.

■ الاستدعاءات والاستشارات للقضايا المستجدة، والإجابة عن الاستفسارات والأسئلة اليومية لمتلقي الخدمة.

■ الاستفسار عن البيانات اللازمة للالتحاق في الجامعات الأردنية الرسمية لمرحلة البكالوريوس والتجسير.

■ إصدار شهادات (بدل تالف - مفقود، كشف علامات، ترجمة).

■ تمويل الأقساط الجامعية.

■ المشاركة في مسابقة الإبداع الشبابي (قصة، شعر، الفن التشكيلي، النص المسرحي، الرواية، الرسم، التصوير الفوتوغرافي).

■ إنشاء الرقم الوطني.

■ إصدار وثيقة معادلة أصلية بدل مفقود/ إصدار نسخة إضافية أصلية.

<https://short-link.me/L4aM> رابط لكل الخدمات التي تُقدم للمواطن

في مجال التعليم:

نشاط عملي:

- أوزع الطلبة في مجموعات عمل غير متجانسة، وأناقشهم في تطبيقات الوسائط المتعددة التي يعرفونها، أو سبق أن استخدموها في مجال صناعة الأفلام.
- أوضح دور التطور التكنولوجي واستخدام الذكاء الاصطناعي في تطوير الأفلام من دون الحاجة إلى كاميرا أو إلى توفير مشاهد معينة.
- أوجه الطلبة نحو تنفيذ النشاط على شكل مجموعات؛ إذ تتخيل كل مجموعة سيناريو حول الحوسبة الخضراء، ويُطبّق عملياً؛ لتعرّف كيفية إنتاج الأفلام بالذكاء الاصطناعي.
- أتابع تنفيذ الطلبة النشاط، ومن ثم أتيح لهم المجال لعرض أعمالهم.
- أوجه أفراد المجموعة لمناقشة مزايا استخدام الذكاء الاصطناعي وسليباته،

في صناعة الأفلام؛ إذ توجد تطبيقات كثيرة للذكاء الاصطناعي سهّلت عملية صناعة الأفلام من دون حاجة إلى استخدام آلة تصوير (كاميرا) عالية الجودة، ومن دون حاجة إلى تصوير أي شيء أساساً، أو إرفاق صوت أو موسيقى مع الفيلم. فكل ما هو مطلوب تزويد برنامج الذكاء الاصطناعي بالموضوع المطلوب ونبذة عنه، ليقوم البرنامج بإعداد فيلم فائق الجودة، ومُكتمل العناصر؛ من: صوت، وموسيقى تصويرية، وصور مُتحرّكة.

من تطبيقات الذكاء الاصطناعي في صناعة الأفلام: تطبيق (Invideo AI) الذي يتطلّب استخدامه إنشاء حساب في الموقع، وإدخال كلمات مفتاحية عن موضوع الفيلم، فيتولّى الموقع صنع فيلم كامل.



أزور الموقع الإلكتروني لتطبيق (Invideo AI) عن طريق الرابط الإلكتروني الآتي:

<https://ai.invideo.io/login>



أو مسح الرمز سريع الاستجابة (QR Code) المجاور، ثم أنفذ الإجراءات الآتية:

1. إنشاء حساب خاص بي في الموقع.
 2. إعداد مقطع فيديو عن الحوسبة الخضراء باستخدام الذكاء الاصطناعي.
 3. عرض مقطع الفيديو أمام طلبة الصف.
- أفكر في المزايا والسليبات لاستخدام الذكاء الاصطناعي في صناعة الأفلام، ثم أناقش ذلك مع زملائي / زميلاتي في الصف.

الطباعة ثلاثية الأبعاد (3D Printing):

الطباعة ثلاثية الأبعاد هي عملية إنشاء كائن ثلاثي الأبعاد من ملف رقمي. ولكن، ما مبدأ العمل الذي تقوم عليه الطباعة ثلاثية الأبعاد؟

تتمثل هذه العملية أولاً في بناء نموذج ثلاثي الأبعاد باستخدام برامج خاصة لهذا الغرض، مثل برنامج (Tinker CAD) الذي يمتاز بأنه مجاني، ولا يلزم تحميله في جهاز الحاسوب الخاص بي، فضلاً عن إعطائه المُبتدئين دروساً في هذا المجال، واحتوائه على ميزة تصدير النموذج الذي يُنشأ بوصفه ملفاً قابلاً للطباعة بامتداد (OBJ).

185

وأتيح لهم المجال لتبادل الأفكار، والنقاش ولا سيما بعد عرض أعمال المجموعات المختلفة.

مناقشة

- أناقش الطلبة في كيفية الطباعة ثلاثية الأبعاد، ومبدأ عملها..

ثالثاً: مرحلة الإثراء والتأمل والتقييم

- أوضح للطلبة التعليمات الخاصة بالنشاط.
- أوزع الطلبة في مجموعات، وأكلّفهم البحث عن التطبيقات التي تتعلق بصناعة الأفلام والطباعة ثلاثية الأبعاد، ويمكن توزيع العمل، وتبادل الأفكار في ما بين المجموعات.
- أوكد ضرورة الدخول إلى رابط اللوح التفاعلي ومشاركة الأفكار المختلفة.
- أتابع أعمال الطلبة وكيفية تطبيقهم؛ حتى يحصلوا على النموذج ثلاثي الأبعاد.

- أوضح للطلبة أن عملية الطباعة لهذه الملفات تحتاج إلى تقطيع، وفي بعض الأحيان يُستفاد من برامج خاصة للتقطيع.

تطبيقات مقترحة للرسوم المتحركة: <https://www.guru99.com/ar/best-animation-software.html>



تطبيقات مقترحة للطباعة ثلاثية الأبعاد: <https://geeksville.com/tutorial/3d-printer-software/5/>



إثراء:

- ألقت نظر الطلبة إلى مبادرة مصنع الأفكار والهدف منها؛ إذ تشجع الشباب الأردني، وتحثهم على الإبداع والابتكار.

زود الطلبة بالموقع الإلكتروني للموقع؛ لتعرف الموقع ومجالاته وبعض الصور

<https://www.designinjordan.com/articles/04-ar/>



المواطنة الرقمية:



أشارك مع الطلبة التوصيات الآتية لتحقيق المواطنة الرقمية، وتحقيق الأمان على الإنترنت خاصة عند التعامل مع المواقع التي تتطلب معلومات خاصة بهم، وأناقشهم في البنود الواردة، وأقدم لهم التوضيحات اللازمة إن لزم الأمر.

■ أناقش الطلبة في موضوع التَّحَقُّق من المصادر الرقمية وموثوقيتها، مستعينا بالتوضيحات الآتية:

- التَّحَقُّق من نطاق الموقع (Domain): المواقع ذات النطاقات مثل edu. (للمؤسسات التعليمية) و gov. (للحكومات) و org. (للمنظمات غير الربحية) التي تميل إلى أن تكون أكثر موثوقية مقارنة بالمواقع ذات النطاقات التجارية مثل com. ولكن هذا ليس ضماناً كاملاً.



نشاط
جماعي

استكشاف تطبيقات حاسوبية في مجال الرسوم المتحركة والتصميم ثلاثي الأبعاد.

أبحث - بالتعاون مع أفراد مجموعتي - في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن تطبيقات حاسوبية تُستخدم في الرسوم المتحركة والتصميم ثلاثي الأبعاد، ثم أتعرف هذه التطبيقات، وأتعلم أساسيات استخدامها، ثم أكتب ملخصاً عنها، وأشاركه أفراد المجموعات الأخرى عن طريق اللوح الإلكتروني (Padlet).

بعد ذلك أنصفح مقترحات زملائي / زميلاتي في المجموعات الأخرى، ثم أبدي رأيي فيها.

المواطنة الرقمية:



أراعي ما يأتي بعد دراسة موضوع (تطبيقات الحاسوب في الحياة):

- الخصوصية والأمان: أتجنّب مشاركة الآخرين في معلوماتي الشخصية بمنصات التعليم الإلكتروني والتطبيقات الصحية، وأتحقق من أمانها قبل التفكير في مشاركتها.
- استخدام كلمات مرور مُعقَّدة ومُحكَّمة: أتأكد أن كلمات المرور الخاصة بي مُعقَّدة، وأحرص على تغييرها بصورة دورية.
- الموارد التعليمية: أستخدم الموارد التعليمية الإلكترونية على نحو مسؤول، وأحترم حقوق الملكية الفكرية.
- التفاعل الرقمي: أتعامل باحترام مع الآخرين في البيئات التعليمية والبيئات الصحية الرقمية.
- التَّحَقُّق من المصادر: أتحقق من موثوقية المصادر التي أستخدمها، ومن المواقع الإلكترونية التي أنصفحها قبل إدخال بياناتي الشخصية الخاصة.

- التَّحَقُّق من البروتوكول الأمني (HTTPS): التأكد من أن الموقع يستخدم بروتوكول HTTPS بدلاً من HTTP؛ مما يدل على أن الموقع أكثر أماناً لنقل المعلومات.
- البحث عن سياسة التحرير أو الإشراف: التأكد من أن الموقع يحتوي على سياسة تحريرية واضحة أو إشراف على المحتوى. والمواقع التي تفتقر إلى هذه السياسات قد تكون أقل موثوقية.
- التفكير في دوافع الموقع: تحليل الغرض من الموقع. هل هو موقع إخباري، أو أكاديمي، أو تجاري، أو ترويجي؟ ومحاولة فهم الدوافع المحتملة وراء المعلومات التي يقدمها الموقع.
- التَّحَقُّق من التحديثات الأخيرة: التأكد من أن الموقع يُحدَّث بانتظام وأنه حديث. قد تحتوي بعض المواقع على معلومات قديمة وغير محدثة.
- مراجعات المستخدمين وآراؤهم: تحقّق من مراجعات المستخدمين الآخرين وآرائهم حول الموقع أو المقالة. مراجعات إيجابية ومتسقة، يمكن أن تعزز مصداقية الموقع.

- استخدام أدوات التَّحَقُّق الرِّقْمِيّ: هناك عديد من الأدوات والخدمات المتاحة عبر الإنترنت؛ للتحقق من صحة الأخبار والمعلومات، مثل Snopes و FactCheck.org. استخدم هذه الأدوات للتحقق من صحة المعلومات.
- التَّحَقُّق من المصادر المربوطة: التَّحَقُّق من روابط المصادر التي يستشهد الموقع بها؛ للتأكد من موثوقيتها ودعمها المعلومات المقدمة.
- التعرف إلى علامات الاحتيال أو المعلومات الكاذبة: كن على دراية بالعناوين المبالغ فيها، واللهجات المشيرة للعواطف، أو المطالبات التي تبدو غير واقعية. هذه غالباً ما تكون علامات على معلومات غير موثوقة.
- استخدام محركات البحث المتخصصة: عند البحث عن معلومات معينة، قد يكون من الأفضل استخدام محركات بحث متخصصة، أو قواعد بيانات أكاديمية مثل Google Scholar أو PubMed للوصول إلى مصادر موثوقة.
- التأكد من عدم التحيز الخوارزمي: الوعي للتحيز الخوارزمي في وسائل التواصل الاجتماعي ومحركات البحث. قد تدفع الخوارزميات بمحتوى معين بناءً على التفضيلات السابقة؛ مما قد يحد من تنوع المصادر التي نصل إليها.
- كما هو الحال مع المصادر التقليدية، يجب التَّحَقُّق من المعلومات الآتية:
 - المؤلف: التَّحَقُّق من خلفية المؤلف، وخبرته، ومؤهلاته الأكاديمية أو المهنية في الموضوع المعني.
 - الجهة الناشرة: معرفة المنظمة أو الجهة التي نشرت المحتوى. هل هي جامعة، أو مؤسسة بحثية، أو دار نشر موثوقة؟
 - تاريخ النشر: التأكد من أن المعلومات حديثة وملائمة للموضوع الذي نبحث فيه، خاصة إذا كانت مرتبطة بمجالات تتغير بسرعة، مثل التكنولوجيا أو الطب.
 - المصادر المقتبسة: التَّحَقُّق مما إذا كانت المقالة أو الكتاب، يستند إلى مصادر موثوقة أخرى. فوجود مراجع وارتباطات لمصادر معروفة، يعزز من مصداقية العمل.
 - جودة اللغة والأسلوب: المصادر الموثوقة عادة ما تكون مكتوبة بلغة سليمة وواضحة، مع تجنب الأخطاء اللغوية والإملائية. يعكس الأسلوب المهني والمنظم جدية المحتوى.
 - مراجعة النظراء (Peer Review): في حال المصادر الأكاديمية، يجب التَّحَقُّق مما إذا كانت المقالة قد خضعت لمراجعة من قبل خبيرين في المجال؛ فالمقالات المراجعة عادة ما تكون أكثر مصداقية.
 - مقارنة المعلومات: مقارنة المعلومات الواردة في المصدر مع ما هو موجود في مصادر أخرى؛ فإذا كانت المعلومات متسقة عبر مصادر موثوقة عدة، فهذا يعزز من مصداقيتها.
 - الوصول إلى المصدر الأصلي: الرجوع إلى المصدر الأصلي للمعلومة بدلاً من الاعتماد على مصدر ثانوي، فقد يكون المصدر الأصلي أكثر تفصيلاً ودقة.
 - التَّحَقُّق من الروابط والمراجع: إذا كان المصدر يزود بروابط أو مراجع لمصادر أخرى، يجب التَّحَقُّق من صحتها، والتأكد من أنها تؤدي إلى معلومات صحيحة وموثوقة.

نشاط علاجي:

- أوزع الطلبة في مجموعات عمل غير متجانسة.
- أحضر بطاقات من الكرتون من اللون نفسه، وأكتب عليها المجالات التي تُستخدم فيها التطبيقات الحاسوبية.
- أحضر بطاقات من لون آخر، وأكتب عليها أمثلة على التطبيقات في مجالات مختلفة.
- أختار لوناً آخر، وأكتب عليه أهمية استخدام هذه التطبيقات في كل مجال من المجالات (التعليم، الصحة، الحكومة).
- أكلّف الطلبة ترتيب البطاقات؛ إذ يتكون لديهم خريطة متفرعة، تُكتب فيها المجالات التي تُستخدم فيها التطبيقات الحاسوبية وأهميتها، ومثال عليها، إذ تشكل ملخصاً للدرس.
- يُراعى في توزيع المجموعات توزيع مستويات الطلبة؛ حتى يقدم الأقران المساعدة بعضهم لبعض.

نشاط إثرائي (1):

تكليف الطلبة تصميم موقع إلكتروني للصف، وتحميل ملفات العمل والبحث التي أنشأها الطلبة في الوحدة، ومشاركتها مع طلبة المدرسة.

نشاط إثرائي (2):

تكليف الطلبة تصميم استبانة إلكترونية؛ لمعرفة مدى معرفة طلبة المدرسة بالحواسبة الخضراء، ونشرها عبر موقع المدرسة الإلكتروني، ودعوة الطلبة لتعبئتها، ثم تحليل نتائجها، والتخطيط لحملة توعية مناسبة وفقاً لهذه النتائج.

التعليمات الخاصة بمنتج التعلم

- أذكر الطلبة بسياق التعلم، وبأن عليهم تنفيذ المهمة الثالثة لإنجاز المشروع الخاص بهم، وهي مهمة التنفيذ والإنتاج.
- أوضح المطلوب من الطلبة في المرحلة الثالثة من تطوير المشروع، وهو إنتاج المحتوى الرئيس للمشروع الريادي الرقمي الذي يتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة وفقاً للخطة المحددة، باستخدام إحدى تطبيقات الحاسوب، ومراجعته ونشره في العالم الرقمي.
- أوجه الطلبة لقراءة التعليمات الخاصة بالمهمة، وأقدم التوضيحات اللازمة.
- أوضح للطلبة أن عليهم التركيز على التنفيذ والإنتاج، ومن ثم الاختبار والتحسين، وكيفية النشر والتقييم، ويمكنهم أيضاً مشاركة زملائهم في هذه المرحلة بعد اطلاعهم على



مشروع

- المشروع: تنفيذ مشروع ريادي رقمي يتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة، باستخدام إحدى تطبيقات الحاسوب / المهمة (3).
- أعمل - بالتعاون مع أفراد مجموعتي - على تنفيذ المرحلة الثانية من المشروع، وهي مرحلة التنفيذ والإنتاج:
1. أبدأ بإنتاج المحتوى الرئيس للمشروع وفقاً للخطة الموضوعية، مثل كتابة السيناريوهات وتحريرها؛ سواء كان المشروع مقطع فيديو، أو ملصقاً، أو رسوماً متحركة، أو تصميمًا ثلاثي الأبعاد.
 2. أتحقق من تجهيز البرنامج المناسب، ثم أبدأ العمل بإنتاج المشروع في نسخته الأولى.
 3. أتعاون مع أفراد مجموعتي، وأعد معهم اجتماعات دورية؛ لمتابعة سير العمل في المشروع، وتبادل الأفكار.
- الاختبار والتحسين:
1. أعرض المشروع - في نسخته الأولى - على مجموعة محدودة من الزملاء / الزميلات، أو المعلمين / المعلمات؛ للحصول على التغذية الراجعة اللازمة.
 2. أستخدم ملاحظات المجموعة في إجراء التحسينات اللازمة.
- النشر:
1. أجهز المشروع للنشر في المنصات المناسبة، مثل يوتيوب (YouTube)، أو مواقع التواصل الاجتماعي، أو منصة المدرسة.
 2. أكتب وصفاً موجزاً للمشروع وهدفه؛ لجذب الانتباه.
 3. أنشر المشروع في المنصات المحددة، وأستعمل الرسوم المناسبة لجذب مزيد من الزوار.
 4. أحتفّ زملائي وأصدقائي على مشاهدة المشروع ومشاركته.
- التقييم والتحسين:
1. أستخدم نماذج جوجل (Google Forms) في إنشاء استبانة لجمع التغذية الراجعة من الزوار.
 2. أحلّل الردود والملاحظات، ثم أحدد مواطن القوة ومواطن الضعف في المشروع.
 3. أجري - بناءً على التغذية الراجعة - التعديلات النهائية لتحسين المشروع.
 4. أعرض النسخة المعدلة مرة أخرى (عند الحاجة)، وأشاركها من جديد.

المشروع على نحو كامل؛ للحصول منهم على التغذية الراجعة.

- أزود الطلبة بمعايير التقييم للمهمة المنشودة في هذه المرحلة.
- ما يأتي معايير تقييم المهمة التي تتعلق بمراحل التنفيذ، والإنتاج، والنشر، والتقييم في مشروع ريادي رقمي:

1. إنتاج المحتوى:

- جودة السيناريوهات والمحتوى: وضوح النصوص وارتباطها الوثيق بأهداف المشروع، إضافة إلى دقة المعلومات وصحة المحتوى البيئي والاجتماعي.
- الإبداع والابتكار: مدى استخدام الأفكار المبتكرة والتقنيات الحديثة في إنتاج الفيديو، والملصقات، والرسوم المتحركة، أو التصميم ثلاثي الأبعاد.

- الالتزام بالخطة الزمنية: تنفيذ المهام ضمن الجدول الزمني المحدد، والالتزام بالمواعيد النهائية.

2. التعاون والمشاركة:

- فعالية الاجتماعات: مدى تكرار الاجتماعات الدورية وفعاليتها مع الفريق؛ لتبادل الأفكار ومراجعة التقدم.
- إسهام الفريق: مستوى التعاون والمشاركة الفعالة لكل عضو في الفريق خلال مرحلة الإنتاج.

3. الاختبار والتحسين:

- جمع التغذية الراجعة: شمولية عملية عرض المشروع وفعاليتها على مجموعة صغيرة؛ للحصول على ملحوظات بناءة.
- تطبيق التحسينات: مدى الاستجابة للتغذية الراجعة، وإدخال التعديلات المناسبة لتحسين جودة المشروع.

4. النشر والتوزيع:

- الجاهزية للنشر: جودة إعداد المشروع للنشر، بما في ذلك اختيار المنصات المناسبة والوسوم التي تزيد من الوصول.
- الترويج والمشاركة: مدى فعالية الترويج للمشروع، وتشجيع الآخرين على مشاهدته ومشاركته.

5. التقييم والتحسين النهائي:

- استبانة التغذية الراجعة: دقة تصميم الاستبانة باستخدام (Google Forms)، وسهولة استخدامها لجمع آراء المشاهدين.
- تحليل الردود: تحليل التغذية الراجعة، وتحديد نقاط القوة والضعف في المشروع.
- لتحسين النهائي: إجراء التحسينات بناءً على التغذية الراجعة، وإعادة نشر المشروع بفعالية بعد التعديلات.

6. التوثيق والتقرير النهائي:

- توثيق المراحل: جودة توثيق جميع مراحل العمل من الإنتاج إلى التقييم والتحسين.
- التقرير النهائي: تقديم تقرير نهائي شامل، يعرض الخطوات المتبعة، والتحديات التي تم التغلب عليها، والنتائج النهائية.

إجابات أسئلة أقيم تعلّمي

المعرفة: أوظّف في هذا الدرس ما تعلّمته من معارف في الإجابة عن الأسئلة الآتية:
السؤال الأول:

1. التعلّم الإلكتروني (E-Learning)

: التعلّم الإلكتروني هو نظام تفاعلي للتعليم، يقدم للمتعلّم باستخدام تكنولوجيا الاتصال والمعلومات، ويعتمد على بيئة إلكترونية رقمية متكاملة، تعرض المقررات الدراسية عبر الشبكات الإلكترونية، وتوفر سبل الإرشاد والتوجيه، وتنظيم الاختبارات، وكذلك إدارة المصادر والعمليات وتقويمها.

2. الحكومة الإلكترونية

(E-Government): أتمتة الخدمات الحكومية المقدمة للمواطنين عن طريق توفير عدد من التطبيقات والقنوات الرقمية عبر شبكة الإنترنت؛ لإنجاز المعاملات الحكومية التي تخص المواطنين.

3. الطباعة ثلاثية الأبعاد (3D-Printint): هي عملية إنشاء كائن ثلاثي الأبعاد من ملف رقمي.

السؤال الثاني:

- إصدار شهادة عدم محكومة: تتيح هذه الخدمة للمواطنين إصدار شهادة عدم المحكومة إلكترونياً، وإمكانية تقديم الطلب والدفع بصورة إلكترونية
- الاستعلام عن دفع المخالفات: تُوفّر هذه الخدمة قناة إلكترونية، تُمكن المواطنين والمقيمين من الاستعلام عن مخالفات المركبات، ودفع قيمتها إلكترونياً، والاطّلاع على تفاصيل كل مخالفة منها.
- الاستعلام عن ضريبة الأبنية (المُسَقَّفات): تُوفّر هذه الخدمة الاستعلام عن ضريبة الأبنية، ودفع قيمتها إلكترونياً.
- تجديد رخصة المهن ولوحة الإعلانات إلكترونياً: تُوفّر هذه الخدمة قناة إلكترونية، تُمكن أصحاب رخص المهن والأعمال الحرة من تجديد رخصهم، ودفع رسومها إلكترونياً، إضافةً إلى تسلم هذه الرخص؛ إمّا عن طريق البريد الأردني، وإمّا شخصياً.

أقيم تعلّمي

المعرفة: أوظّف في هذا الدرس ما تعلّمته من معارف في الإجابة عن الأسئلة الآتية:
السؤال الأول: أوضّح المقصود بكُلّ ممّا يأتي:
1. التعلّم الإلكتروني (E-Learning).

أنظر الهامش

2. الحكومة الإلكترونية (E-Government).

أنظر الهامش

3. الطباعة ثلاثية الأبعاد (3D-Printint).

أنظر الهامش

السؤال الثاني: ما الخدمات التي تُقدّمها الحكومة الإلكترونية للمواطنين؟

أنظر الهامش

السؤال الثالث: كيف تُسهم التكنولوجيا في تحسين جودة التعليم؟

أنظر الهامش

المهارات: أوظّف مهارات التفكير الناقد والتواصل الرقمي والبحث الرقمي في الإجابة عن الأسئلة الآتية:

السؤال الأول: أصف كيف يعمل تطبيق السجل الصحي الإلكتروني على تحسين خدمات الرعاية الصحية.

أنظر الهامش

السؤال الثالث:

1. استدامة التعليم: جعلُ التحوُّل الرِّقْمِيّ التعليم متاحًا ومتوافرًا للطلبة كافةً في مختلف الأوقات والأحوال، لا سيَّما الطارئة منها، مثل جائحة كورونا.
2. توفير الوقت: أتاح التحوُّل الرِّقْمِيّ في التعليم للطلبة والمُعَلِّمين/ المُعلِّمات توفير الوقت الذي كان يُقضى في الانتقال إلى المدارس والمؤسسات التعليمية.
3. تحسين جودة التعليم: أسهم استخدام الأدوات التكنولوجية المُتقدِّمة في تحسين جودة التعليم، بما وفَّرتَه من تجارب وخبرات ومهارات تعليمية مُتنوِّعة، تُناسب حاجات الطلبة المختلفة.

المهارات: أُوظِّف مهارات التفكير الناقد والتواصل الرِّقْمِيّ والبحث الرِّقْمِيّ في الإجابة عن الأسئلة الآتية:

السؤال الأوَّل:

تعمل تطبيقات السجل الصحي الإلكتروني على تخزين المعلومات الطبية، ومشاركتها بين مُقدِّمي الرعاية الصحية بصورة آمنة؛ مما يتيح للأطباء الوصول السريع إلى بيانات المرضى، ويُمكنهم من توجُّهِ الدِّقَّة في التشخيص، واختيار العلاج المناسب

السؤال الثاني:

1. زيادة المنافسة: يُسهِّل التحوُّل الرِّقْمِيّ على المُنافسين الجُدُد دخول السوق؛ مما يزيد من وتيرة التنافس، ويُحفِّز الشركات على تحسين خدماتها ومنتجاتها.
2. تحسين تجربة العملاء: يُعزِّز التفاعل الفوري والتخصيص من رضا العملاء، ويزيد من
3. ولائهم للعلامة التجارية.
4. زيادة الكفاءة: يؤدِّي استخدام الأنظمة الرِّقْمِيَّة في إدارة العمليات إلى تقليل من الأخطاء، وزيادة كفاءة العمليات التشغيلية.

إجابة أسئلة الوحدة

السؤال الأول: اكتب المصطلح المناسب أمام كل عبارة من العبارات الآتية:

مؤشر الأداء	العبارة
الحوسبة الخضراء.	تصميم أو تصنيع أو استخدام أجهزة الحاسوب والأجهزة التكنولوجية الأخرى بطريقة تحد من الآثار الضارة على البيئة كانبعاثات الكربون واستهلاك الطاقة.
النفايات الإلكترونية	أي شيء يحتوي على مقابس وأسلاك ومكونات إلكترونية، كأجهزة التلفاز والحاسوب والهواتف المحمولة ومكيفات الهواء، وألعاب الأطفال الإلكترونية، وغيرها من الأجهزة.
الطباعة ثلاثية الأبعاد.	عملية إنشاء كائن ثلاثي الأبعاد من ملف رقمي.
تطبيق حكيم (Hakeem).	أحد البرامج الوطنية المهمة لحوسبة قطاع الصحة في الأردن الذي أطلق في عام 2009.
جهاز الفروت.	جهاز حاسوب بلا شاشة ولا لوحة مفاتيح، والاعتماد على مبدأ جهاز العرض (Projector).

السؤال الثاني: ما تحديات تطبيق الحوسبة الخضراء؟

- قلة الوعي بأهمية الحوسبة الخضراء من أبرز التحديات والمعوقات التي تؤثر في تطبيق مبدأ الحوسبة الخضراء، فقليل من الجهات تفكر في المشكلات البيئية، كالتغير المناخي عند تصنيع أو أجهزة تكنولوجية شرائها.
- تطور التكنولوجيا وتغيرها بسرعة كبيرة، وهذا يمثل تحدياً في تمديد عمر حياة المنتجات الإلكترونية.
- الكلفة المادية للتحويل إلى الحوسبة الخضراء، ربما تكون كبيرة لبعض الشركات.
- تشكل الاحتياجات المتنوعة للمستخدمين النهائيين للتكنولوجيا صعوبة في اتخاذ القرار، فبعض مراكز البيانات، تأخذ السرعة والأداء، ضمن أولوياتها على الأمور البيئية.
- قد يكون امتلاك جهاز أصغر حجماً أكثر أهمية من امتلاك جهاز قابل لإعادة التدوير لدى طلبة المدارس والجامعات.

السؤال الثالث: علل ما يأتي:

1. استبدال الخيزران (Bamboo)، بمادة البلاستيك التي تدخل في صناعة ملحقات الحاسوب. حلها لمشكلة البلاستيك، وتقليل النفايات الإلكترونية.
 2. أجهزة الحاسوب من نوع (Notebooks) أقل تأثيراً في البيئة من أجهزة الحاسوب المحمولة (Laptops). وأجهزة الحاسوب المحمولة أقل تأثيراً في البيئة من أجهزة الحاسوب المكتبية (Desktop).
- تصميم المعالج في أجهزة Notebook عادةً ما تستخدم معالجات ذات كفاءة طاقة أعلى والتي تهدف إلى تقليل استهلاك الطاقة مقارنةً بالمعالجات المستخدمة في أجهزة اللاب توب والأجهزة المكتبية.
- حجم الشاشة: أجهزة Notebook غالباً ما تأتي بشاشات أصغر مقارنةً باللاب توب والأجهزة المكتبية؛ مما يعني استهلاك طاقة أقل من الشاشات الأكبر.

المكونات الداخلية: أجهزة Notebook قد تستخدم مكونات ذات استهلاك طاقة أقل، مثل وحدات التخزين الصلبة SSD، بدلاً من الأقراص الصلبة التقليدية HDD التي تستهلك طاقة أقل.

تقنيات توفير الطاقة: تتضمن أجهزة Notebook تقنيات متقدمة لإدارة الطاقة، مثل وضعيات توفير الطاقة التلقائية، أو أنظمة تبريد أكثر كفاءة؛ مما يساعد في تقليل استهلاك الطاقة.

التخلص من النفايات الإلكترونية باتباع إجراءات، كالحرق الذي يُعدّ من الإجراءات الرديئة التي تؤثر سلباً في البيئة. لأنه ينتج موادّ سامة كالرصاص في الجو، وهي تعدّ من أحد الأسباب الملوثة للبيئة.

السؤال الرابع: ما هي الخطوات التي يمكن أن تتبعها؛ للتخلص من النفايات الإلكترونية على نحو سليم؟

- الوعي بمفهوم النفايات الإلكترونية: فبعضهم لا يدرك ما تتضمنه النفايات الإلكترونية من مواد سامة، قد ينتهي معظمها في مدافن النفايات؛ لذا فإن معرفة مكونات النفايات الإلكترونية، يعد الخطوة الأولى نحو التخلص منها.
- تقليل النفايات الإلكترونية: إن تقليل النفايات الإلكترونية بشراء ما تحتاجه فقط، واختيار المنتجات طويلة الأجل والموفرة للطاقة، وإطالة عمر أجهزتك بإصلاحها بدلاً من استبدالها بأجهزة جديدة بها، يُعدّ من النصائح المهمة التي تساهم في تقليل النفايات الإلكترونية.
- التعاون مع المؤسسات والوزارات والمشاروعات التي تُعنى بتدوير النفايات الإلكترونية على المستوى المحلي، ويجدر الذكر أن هناك عديداً من المشاروعات الريادية والشركات والجهات في الأردن، تُعنى بجمع النفايات الإلكترونية وإعادة تدويرها على نحو سليم، ويتطلب منك ذلك التواصل مع هذه الجهات.

السؤال الخامس: وضح أثر التطبيقات الآتية في حياتنا:

- تطبيق حكيم (Hakeem) في القطاع الصحي. يهدف إلى تحسين الإدارة الطبية وتقديم الخدمات للمواطنين بأفضل المعايير الدولية عن طريق إنشاء ملف طبي إلكتروني للمريض على التطبيق، وتيسير وصول مستخدم النظام إليه من أي منشأة طبية باستخدام رقمه الوطني، ويحتوي ملف المريض الإلكتروني على تقارير المريض، والأدوية التي يتناولها، والتاريخ الطبي، وغيرها من البيانات الطبية.
- منصة التعلم الأردنية في قطاع التعليم. توفير المواد والأدوات التعليمية الإلكترونية اللازمة للمعلم والطالب، بما يساعد على تعزيز العملية التعليمية بمكوناتها الأساسية.

■ تطبيق السوق المفتوح (Opensooq) في مجال التسويق.

أكبر تطبيق للإعلانات الممولة باللغة العربية؛ إذ يتمكن المستخدم من بيع عديد من المنتجات والخدمات لملايين المستخدمين عبر الإنترنت أو شرائها من دون وسيط، ويمكن للمشتريين استعراض السلع والخدمات، كالسيارات والعقارات والإلكترونيات والأثاث... إلخ.

السؤال السادس: لماذا يعدّ التخلص السليم من النفايات الإلكترونية ضروريًا؟

إن التعامل الخطأ مع المخلفات الإلكترونية، وغياب شروط السلامة والوقاية الضرورية، ووجود مواد سامة في هذه المخلفات، وغياب شروط السلامة والوقاية الضرورية في أثناء التعامل معها، يشكل خطرًا على الصحة والموارد الطبيعية، خاصة التربة والمياه.

المهارات:

السؤال الأول: كيف يمكن تطبيق مبادئ الحوسبة الخضراء في المنزل؟

- إطفاء الأجهزة غير المستخدمة: إطفاء الحواسيب والأجهزة الإلكترونية عند عدم استخدامها يمكن أن يسهم إسهامًا كبيرًا في تقليل استهلاك الطاقة.
- وضع الأجهزة في وضع السكون: ضبط الأجهزة على وضع السكون أو الإسبات عند عدم الاستخدام فترات طويلة، يقلل من استهلاك الطاقة.
- ضبط إعدادات الطاقة: ضبط إعدادات الطاقة على الوضع الموفر للطاقة في الأجهزة الإلكترونية، يسهم في تقليل استهلاك الطاقة.
- استخدام الأجهزة الموفرة للطاقة: شراء الأجهزة التي تحمل ملصق Energy Star، تستهلك طاقة أقل، وتتميز بكفاءة طاقة عالية.
- إعادة التدوير: التخلص من الأجهزة الإلكترونية بطريقة صحيحة عن طريق إعادة التدوير، يقلل من النفايات الإلكترونية ويحمي البيئة.
- تقليل الطباعة وإعادة تعبئة حبر الطابعة: استخدام الطابعة على وجهي الورقة، وتقليل حجم الخط عند الطباعة لتقليل استهلاك الورق والحبر، وإعادة تعبئة حبر الطابعة، يعدّ إجراءً أفضل من شراء الجزء الخاص بتعبئة حبر الطابعة (Cartridges) في تطبيق الحوسبة الخضراء.
- دراسة الاحتياجات قبل شراء أجهزة الحاسوب أو الأجهزة الإلكترونية: التأكد من شراء الأجهزة التي تلبي الاحتياجات وتفي بالغرض.

السؤال الثاني: أناقش الفوائد والتحديات المرتبطة بتطبيق مبادئ الحوسبة الخضراء في الشركات.

- التكلفة العالية: إن التكلفة المادية للتحويل إلى الحوسبة الخضراء، ربما تكون كبيرة لبعض الشركات، فقد تتطلب الأجهزة والتكنولوجيا الموفرة للطاقة استثمارات أولية عالية؛ مما قد يكون عائقاً أمام الشركات والأفراد.
- الحاجة إلى التدريب: يتطلب تطبيق الحوسبة الخضراء تدريباً إضافياً للموظفين على استخدام التكنولوجيا الخضراء وفهم فوائدها.
- التحديث المستمر: الحاجة إلى متابعة التحديثات والتقنيات الجديدة لضمان كفاءة الطاقة المستمرة، يمكن أن يكون تحدياً، خاصة مع التطور السريع للتكنولوجيا، وتغيرها بسرعة كبيرة، وهذا يمثل تحدياً في تمديد عمر حياة المنتجات الإلكترونية.
- البنية التحتية: قد تكون البنية التحتية القائمة غير ملائمة لتبني ممارسات الحوسبة الخضراء؛ مما يتطلب تعديلات وتحديثات إضافية.
- قلة الوعي: قلة الوعي بأهمية الحوسبة الخضراء من أبرز التحديات والعوائق التي تؤثر في تطبيق مبدأ الحوسبة الخضراء، فقليل من الجهات تفكر في المشكلات البيئية، كالتغير المناخي عند تصنيع أو شراء أجهزة تكنولوجيا.

السؤال الثالث: أحلّ تأثير التطبيقات الحاسوبية في الاقتصاد المحلي، ممثلاً على ذلك بتطبيق السوق المفتوح.

يعدّ أكبر تطبيق للإعلانات المبنية باللغة العربية؛ إذ يتمكن المستخدم من بيع عديد من المنتجات والخدمات لملايين المستخدمين عبر الإنترنت من دون وسيط وشرائها، ويمكن للمشتريين استعراض السلع والخدمات، كالسيارات والعقارات والإلكترونيات والأثاث... إلخ. ويسهم هذا التحوّل الرّقميّ في:

- زيادة المنافسة: يسهل التحوّل الرّقميّ دخول العملاء الجدد إلى السوق؛ مما يزيد من التنافس ويحفز الشركات على تحسين خدماتها ومنتجاتها.
- تحسين تجربة العملاء: التفاعل الفوري والتخصيص يعزز من رضا العملاء ويزيد من ولائهم للعلامة التجارية.
- زيادة الكفاءة: استخدام الأنظمة الرّقمية في إدارة العمليات، يقلل الأخطاء، ويزيد كفاءة العمليات التشغيلية.

السؤال الرابع: ما التحديات التي تواجه تطبيق الحوسبة الخضراء على مستوى المجتمع؟

- كثير من الأفراد في المجتمع، قد لا يكونون على دراية بمفهوم الحوسبة الخضراء وأهميتها في تقليل التأثير البيئي ويؤدي. نقص الوعي إلى قلة الدعم والمشاركة في المبادرات الصديقة للبيئة.
- الاستثمار في تكنولوجيا أكثر كفاءة في استخدام الطاقة، مثل الأجهزة الصديقة للبيئة أو مراكز البيانات ذات الكفاءة العالية، ويمكن أن يكون مكلفاً في البداية؛ مما يشكل عائقاً، خاصة للأفراد والشركات الصغيرة.
- بعض التقنيات الخضراء قد لا توفر عائداً مالياً ملموساً على المدى القصير؛ مما يجعل بعضهم متردداً في تبنيها.
- يتطلب تطبيق الحوسبة الخضراء التطوير المستمر للابتكارات التكنولوجية التي تدعم كفاءة الطاقة، وإدارة النفايات الإلكترونية.

- الحاجة إلى توافق الأنظمة القديمة مع التَقْنِيَّات الجديدة الأكثر كفاءة في استهلاك الطاقة، قد يسبب مشكلات في التكامل والتحديث.
 - تُمثل إدارة النُفَايَات الإلكترونية تحديًا كبيرًا؛ إذ إن عديدًا من الأجهزة الإلكترونية القديمة غير قابلة لإعادة التدوير بسهولة، وقد تحتوي على مواد ضارة بالبيئة.
 - البنية التحتية الداعمة: ضعف البنية التحتية لإعادة التدوير وإدارة النُفَايَات الإلكترونية، يجعل من الصعب تطبيق الحوسبة الخضراء تطبيقًا فعالًا.
 - عدم وجود قوانين وتشريعات قوية تدعم الحوسبة الخضراء، يمكن أن يعرقل الجهود الرامية إلى تعزيزها.
 - غياب الحوافز الاقتصادية مثل الإعفاءات الضريبية أو الدعم المالي للانتقال إلى حلول صديقة للبيئة، يمكن أن يقلل من الدافع لتبني الحوسبة الخضراء.
 - قد يواجه تطبيق الحوسبة الخضراء مقاومة من المجتمع بسبب الاعتياد على الممارسات التقليدية، وعدم الرغبة في تغييرها.
 - قد تؤدي التكاليف الإضافية لتبني تَقْنِيَّات الحوسبة الخضراء إلى زيادة الفجوة بين الفئات الاجتماعية المختلفة؛ إذ قد يصعب على الفئات الأقل دخلًا تبني هذه التَقْنِيَّات.
 - على الرغم من أن الحوسبة الخضراء تهدف إلى تقليل الأثر البيئي، فإن إنتاج الأجهزة الصديقة للبيئة، قد يتطلب استهلاك موارد طبيعية كبيرة، ويتسبب في تلوث خلال عملية التصنيع.
 - يتطلب تطبيق مبادرات الحوسبة الخضراء التزامًا طويل الأمد من المجتمع والمؤسسات، وضمان استمرارية الدعم المالي واللوجستي لهذه المبادرات.
- السؤال الخامس: استخدم أمثلة من النص لشرح كيف يمكن أن تسهم التطبيقات الحاسوبية في تحسين عملية التعليم عن بعد.

يتيح التحوُّل الرِّقْمِيّ في التعليم الوصول للجميع، خاصة في الأوقات الطارئة مثل جائحة كورونا، ويوفر الوقت الذي يُستهلك في الانتقال. ويعزز استخدام التكنولوجيا من جودة التعليم؛ بتقديم تجارب متنوعة تناسب مع احتياجات الطلبة، ويساعد أيضًا في اكتساب المهارات التقنية الضرورية لسوق العمل. ويمكن الوصول إلى الموارد التعليمية بسهولة وفي أي وقت؛ مما يوفر مرونة في أوقات الدراسة والتدريس.

ويقلل التعليم الرِّقْمِيّ من التكاليف؛ بتقليل الحاجة إلى الكتب التقليدية، ويشجع على التفاعل بطرائق مبتكرة مثل الوسائط المتعددة. ويناسب التحوُّل الرِّقْمِيّ الأساليب المفضلة للجيل الجديد، ويزيد من إنتاجية العملية التعليمية.

القيم والاتجاهات:

مثال توضيحي:

تعهد التخلص من النفايات الإلكترونية:

أنا الموقع أدناه، أتعهد بتبني ممارسات مسؤولة في التعامل مع الأجهزة الإلكترونية والنفايات الناتجة عنها، وذلك عن طريق:

- التقليل من النفايات الإلكترونية: استخدام الأجهزة الإلكترونية بأكثر قدر ممكن من الفعالية والكفاءة قبل استبدالها، والتبرع بها عند عدم الحاجة إليها.
 - إعادة التدوير: التخلص من الأجهزة الإلكترونية التالفة أو غير المستخدمة عن طريق القنوات المناسبة لإعادة التدوير، بما يتوافق مع المعايير البيئية.
 - الوعي والمشاركة: المشاركة في نشر الوعي حول أهمية التخلص من النفايات الإلكترونية بطرائق صديقة للبيئة، وتشجيع زملائي وأفراد المجتمع على القيام بذلك.
 - الابتكار والاستدامة: البحث عن حلول تكنولوجية، واستخدام طرائق مبتكرة تدعم الاستدامة، وتقلل الأثر البيئي.
- بتوقيعي على هذا التعهد، ألتزم بالعمل على تحقيق هذه الأهداف والإسهام في حماية بيئتنا والمحافظة على كوكبنا للأجيال القادمة.

ملاحظاتي

الاسم: _____
 التوقيع: _____
 التاريخ: _____

أدوات التقويم الخاصة بالوحدة

تقويم منتج التعلم النهائي

المنتج النهائي: مشروع ريادي رقمي يتناول القضايا البيئية والاجتماعية المتعلقة بالحوسبة وأثرها باستخدام إحدى تطبيقات الحاسوب.

سلم تقدير محكي المرجع وصفي

المعيار	ممتاز (4)	جيد جدًا (3)	جيد (2)	يحتاج إلى تحسين (1)
جودة السيناريوهات والمحتوى	السيناريوهات واضحة ومتراصة، والمحتوى دقيق ومناسب لأهداف المشروع بشكل كامل.	السيناريوهات واضحة إلى حد كبير، والمحتوى دقيق إلى حد ما ومناسب للأهداف.	السيناريوهات بحاجة لبعض التحسين في الوضوح، والمحتوى به بعض الأخطاء البسيطة.	السيناريوهات غير واضحة، والمحتوى غير دقيق أو غير مناسب للأهداف.
الإبداع والابتكار	يتضمن المشروع أفكارًا يتضمن وتقنيات مبتكرة جدًا، مع استخدام فني بارز.	يتضمن المشروع أفكارًا وتقنيات مبتكرة، مع استخدام فني جيد.	يتضمن المشروع بعض الأفكار المبتكرة، لكن الاستخدام الفني عادي.	يفتقر المشروع إلى الابتكار والتجديد، والاستخدام الفني ضعيف.
الالتزام بالخطة الزمنية	نفذت المهام ضمن الجدول الزمني المحدد بشكل دقيق.	نفذت المهام ضمن الجدول الزمني مع بعض التأخير البسيط.	نفذت المهام مع تأخير ملحوظ عن الجدول الزمني.	نفذت المهام مع تأخير كبير، أو تجاوز للجدول الزمني بشكل ملحوظ.
فعالية الاجتماعات	كانت الاجتماعات دورية وفعالة جدًا، وتم تبادل الأفكار بنجاح.	كانت الاجتماعات فعالة إلى حد كبير، وتم تبادل الأفكار بشكل جيد.	كانت الاجتماعات قليلة الفعالية، وتم تبادل الأفكار بشكل محدود.	كانت الاجتماعات غير فعالة، وتبادل الأفكار كان ضعيفًا جدًا.

إسهام الفريق	كل أعضاء الفريق أسهموا بشكل فعال في الإنتاج.	معظم أعضاء الفريق أسهموا بشكل فعال في الإنتاج.	بعض أعضاء الفريق أسهموا بشكل محدود في الإنتاج.	إسهامات الفريق كانت ضعيفة جداً أو غير متساوية بين الأعضاء.
تصميم الاستبانة	الاستبانة مصممة بشكل ممتاز وسهلة الاستخدام، وجمعت معلومات مهمة.	الاستبانة مصممة بشكل جيد وسهلة الاستخدام؛ لكنها جمعت معلومات محدودة.	الاستبانة بحاجة إلى تحسينات في التصميم، وواجهت صعوبة في جمع المعلومات.	الاستبانة غير فعالة وصعبة الاستخدام، ولم تجمع معلومات ذات قيمة.
جمع التغذية الراجعة وتحليلها	جمعت التغذية الراجعة بشكل شامل من مجموعة متنوعة من الأفراد.	جمعت التغذية الراجعة من مجموعة جيدة من الأفراد.	جمعت التغذية الراجعة من عدد محدود من الأفراد.	لم تُجمع التغذية الراجعة بشكل كافٍ، أو تم تجاهلها.
التحسين النهائي	أُجريت التحسينات النهائية بشكل ممتاز والمشروع أصبح جاهزاً للنشر النهائي.	أُجريت التحسينات النهائية بشكل جيد، والمشروع بحاجة لبعض اللمسات النهائية.	أُجريت بعض التحسينات، لكن المشروع ما زال بحاجة إلى تعديلات إضافية.	لم تُجرَ التحسينات النهائية بشكل كافٍ، والمشروع غير جاهز للنشر.
الجاهزية للنشر	المشروع جاهز للنشر بجودة عالية، مع استخدام منصات ووسوم مناسبة.	المشروع جاهز للنشر بجودة جيدة، مع استخدام منصات ووسوم مناسبة.	المشروع جاهز للنشر، لكن يمكن تحسين جودة العرض أو الوسوم.	المشروع غير جاهز للنشر أو يحتاج إلى تعديلات كبيرة.
الترويج والمشاركة	رُوجَّ للمشروع بشكل فعال جداً، وحظي بمشاركة واسعة.	رُوجَّ للمشروع بشكل جيد، وحظي بمشاركة معقولة.	رُوجَّ للمشروع بشكل محدود، وكانت المشاركة ضعيفة.	الترويج كان غير فعال، والمشروع لم يحقق أي مشاركة تذكر.
توثيق المراحل	جميع مراحل العمل موثقة بشكل ممتاز ومنظم.	معظم مراحل العمل موثقة بشكل جيد.	بعض مراحل العمل موثقة، لكن التوثيق بحاجة إلى تحسين.	التوثيق ضعيف أو غير موجود.
التقرير النهائي	التقرير شامل ومفصل ويعكس جميع جوانب المشروع بشكل ممتاز.	التقرير جيد ويغطي معظم جوانب المشروع بشكل واضح.	التقرير محدود ويغطي بعض جوانب المشروع فقط.	التقرير ضعيف، ولا يعكس جوانب المشروع بشكل كافٍ.

ملحق (1) : تعليمات ذات صلة بمشروعات التعلّم (مشروعات الوحدة) وأدوات التقييم الخاصة

بها.

المشروع الأول: تنظيم حملة تثقيفية عن الحوسبة الخضراء؛ لتعزيز وعي المجتمع المدرسي بأهمية الحوسبة الخضراء.

الهدف: تنظيم حملة تثقيفية لتعزيز وعي المجتمع المدرسي بأهمية الحوسبة الخضراء، واستخدام التّقنيّات المستدامة. **تعليمات:**

أوجه الطلبة لاتباع الخطوات والتوجيهات الآتية لفهم الأساسيات والتّقنيّات المطلوبة لتحقيق الهدف.

المقدمة: أبدأ بتقديم المشروع وهدفه وهو توعية المجتمع المدرسي بأهمية الحوسبة الخضراء، وكيفية تطبيق ممارسات صديقة للبيئة في استخدام التكنولوجيا.

فهم الهدف والمفهوم:

- شرح الهدف: الهدف من المشروع تنظيم حملة تثقيفية تفاعلية، تهدف إلى تعزيز فهم المجتمع المدرسي لمفهوم الحوسبة الخضراء وتطبيقاتها.
- المفهوم الأساسي: تعتمد الحملة على نشر الوعي عن طريق مواد تعليمية، وورش عمل، وأنشطة تفاعلية توضح تأثير الحوسبة الخضراء في البيئة، وأهمية تبنيها في الحياة اليومية.

مرحلة التخطيط:

أوجه الطلبة للقيام بما يأتي:

- تحديد الأهداف والمحتوى: تحديد الأهداف التعليمية للحملة والفئة المستهدفة.
- البحث والتحليل: جمع المعلومات المتعلقة بالحوسبة الخضراء وتحليلها، مثل فوائدها البيئية والاقتصادية. والمقارنة بين تقنيّات الحوسبة التقليدية والحوسبة الخضراء من حيث الكفاءة، والاستدامة، والتأثير البيئي.
- إنشاء خطة الحملة: إعداد جدول زمني، يتضمن الأنشطة المختلفة والمراحل التي ستم فيها الحملة.

مرحلة التصميم: أوجه الطلبة للقيام بما يأتي:

- تطوير محتوى الحملة: إنشاء مواد تعليمية مثل إعداد كتيبات، وعروض تقديمية، وملصقات توعوية تُبرز أهمية الحوسبة الخضراء، وكيفية تطبيقها في الحياة اليومية.
- إعداد ورش العمل: تصميم محتوى ورش العمل التفاعلية التي ستُعقد خلال الحملة، مع التركيز على أمثلة عملية وتطبيقات حقيقية.
- تصميم الشعار والهوية البصرية: ابتكار شعار للحملة، وتصميم المواد البصرية لجذب الانتباه، وتحقيق التأثير المطلوب.

مرحلة التنفيذ: أوجه الطلبة للقيام بما يأتي:

- تنفيذ الحملة في المدرسة: تنظيم الفعاليات المجدولة مثل المحاضرات، وورش العمل، والأنشطة التفاعلية لتعزيز الفهم، ونشر الوعي.
- توزيع المواد: توزيع الكتيبات والملصقات على الطلبة والمعلمين، وضمان وصول الرسالة إلى جميع أفراد المجتمع المدرسي.

مرحلة المراجعة وتصحيح الأخطاء: أوجه الطلبة للقيام بما يأتي:

- جمع الملحوظات: دعوة المشاركين لتقديم ملحوظاتهم حول الحملة وفعاليتها، وتقييم فعالية الحملة عن طريق استبانات ومقابلات مع المشاركين؛ لتحليل مدى تحقيق الأهداف المرجوة.
- مراجعة الحملة: مراجعة الأنشطة والمواد المستخدمة، وتحديد النقاط التي تحتاج إلى تحسين أو تعديل.

مرحلة التوثيق والعرض:

- أوجه الطلبة للقيام بما يأتي:
 - توثيق مراحل الحملة: إعداد تقرير يوثق جميع خطوات العمل، بما في ذلك الأنشطة التي نُفذت والمواد التي أُعدت.
 - إعداد عرض تقديمي: تحضير عرض، يلخص نتائج الحملة، ويشرح أثرها في المجتمع المدرسي.
 - تقديم الحملة: عرض نتائج الحملة وتوصياتها المستقبلية أمام الإدارة المدرسية، أو في الفعاليات المدرسية.
- معايير التقييم:

- الإبداع والابتكار في تصميم الحملة: مدى تميز المواد التعليمية والأنشطة المستخدمة في توصيل الرسالة.
- التفاعل والمشاركة: مدى تفاعل الطلبة والمعلمين مع الحملة ومدى مشاركتهم في الأنشطة.
- التأثير التعليمي: قدرة الحملة على تحقيق الهدف، وزيادة الوعي بالحوسبة الخضراء بين المشاركين.

مرحلة التقويم والتأمل:
أوجه الطلبة للقيام بما يأتي:

- التأمل في تجربة الحملة: التفكير في ما تم تعلّمه خلال عملية تنظيم الحملة، وما يمكن تحسينه في المشروعات المستقبلية.
- جمع التغذية الراجعة: جمع آراء المشاركين بعد انتهاء الحملة، وتحديد النقاط القوية، والمجالات التي تحتاج إلى تحسين.
- تحديد الخطوات الآتية: وضع خطة لتحسين الأنشطة التثقيفية المستقبلية بناءً على الملحوظات، والتفكير في كيفية تطوير حملات جديدة، تعزيز الوعي البيئي باستخدام التكنولوجيا، ووضع خطة لتحسين الأنشطة التثقيفية المستقبلية بناءً على الملحوظات، والتفكير في كيفية تطوير حملات جديدة، تعزيز الوعي البيئي باستخدام التكنولوجيا.

ملاحظات

.....

.....

.....

.....

.....

.....

.....

.....

.....

المشروع الثاني: تنظيم مسابقة تحمل عنوان (البرمجة الخضراء)، واشتراط كتابة تعليمات برمجية باستخدام لغة البرمجة بايثون (Python) – تهدف إلى الحد من استهلاك الطاقة، والتركيز على تحسين كفاءة المقطع البرمجي، والتقليل من استخدام الموارد الحاسوبية.

الهدف: تشجيع الطلبة على استخدام مهارات البرمجة لتطوير حلول مبتكرة، تقلل من استهلاك الطاقة، وتحسّن كفاءة الكود باستخدام لغة بايثون.
تعليمات:

توجيه الطلبة لاتباع الخطوات والتوجيهات الآتية لفهم الأساسيات والتقنيات المطلوبة لتحقيق الهدف.
المقدمة: أبدأ بتقديم المشروع وهدفه وهو توعية الطلبة بأهمية البرمجة الخضراء، وكيفية تطبيقها لتقليل استهلاك الطاقة، وتحسين كفاءة الكود.

فهم الهدف والمفهوم:
شرح الهدف:

- إنشاء تطبيق بلغة بايثون، يقيس استهلاك الطاقة في الكود البرمجي.
- تحليل البيانات المستخرجة؛ لتحديد النقاط التي يمكن تحسينها.
- اقتراح حلول مبتكرة؛ لتحسين الكفاءة، وتقليل استهلاك الطاقة في البرمجة.

المفهوم الأساسي: يعتمد المشروع على تصميم كود برمجي فعال وتطويره باستخدام لغة بايثون، بحيث يحقق الأهداف المحددة بأقل استهلاك ممكن للطاقة والموارد.

مرحلة التخطيط:

أوجه الطلبة للقيام بما يأتي:

- جمع المعلومات: دراسة تأثير الكود البرمجي في استهلاك الطاقة، وكيفية تحسين كفاءة البرمجة.
- العصف الذهني: توليد أفكار حول كيفية تصميم تطبيق، يمكنه قياس استهلاك الطاقة، وتحليل أداء الكود.
- استكشاف المكتبات المتاحة: البحث عن مكتبات بايثون التي يمكن استخدامها، مثل psutil لقياس استهلاك الموارد.
- إنشاء خطة عمل: وضع خطة زمنية للمشروع، تتضمن جميع المراحل من البحث حتى التقديم.

مرحلة التصميم:

أوجه الطلبة للقيام بما يأتي:

- تصميم مخطط سير العمل: رسم مخططات توضح عن كيفية عمل الكود البرمجي، والميزات الأساسية التي سيحتوي عليها.
- تحديد الميزات: تحديد الميزات الرئيسة للبرنامج، مثل قياس استهلاك الطاقة، وعرض ملخص الأداء، وتقديم توصيات لتحسين الكود.
- إعداد بيئة العمل: تجهيز بيئة العمل البرمجية التي سيستخدمها الطلبة لتطوير التطبيق.

مرحلة التطوير:

أوجه الطلبة للقيام بما يأتي:

- كتابة البرنامج: البدء في كتابة المقطع البرمجي باستخدام لغة بايثون، مع التركيز على تحسين كفاءة الكود، واستخدام الموارد بأفضل طريقة ممكنة.
- التحقق من المقطع البرمجي: التأكد من أن البرنامج يعمل بشكل صحيح، ويتماشى مع الأهداف المحددة.
- إدماج الميزات: تنفيذ الميزات المتفق عليها داخل البرنامج، مثل قياس استهلاك الطاقة، وعرض ملخصات حول الأداء.

مرحلة المراجعة وتصحيح الأخطاء:

أوجه الطلبة للقيام بما يأتي:

- اختبار البرنامج: إجراء اختبارات شاملة للتأكد من فعالية البرنامج في قياس استهلاك الطاقة، وتحسين المقطع البرمجي.
- جمع الملحوظات: دعوة زملاء أو معلمين آخرين لتجربة التطبيق، وتقديم ملحوظاتهم.
- تصحيح الأخطاء: مراجعة الكود وإجراء التعديلات اللازمة بناءً على الملحوظات واختبارات الأداء.

تحليل البيانات وتقديم حلول:

أوجه الطلبة للقيام بما يأتي:

- جمع البيانات: جمع البيانات من جميع القياسات التي يقوم بها البرنامج.
- استخدام أدوات التحليل: استخدام أدوات مثل pandas لتقديم تقارير بصرية حول استهلاك الطاقة.
- تقديم الحلول: بناءً على التحليل، تطوير استراتيجيات لتحسين الكفاءة مثل تحسين الكود لتقليل استهلاك الذاكرة، واستخدام خوارزميات أكثر كفاءة.

مرحلة التوثيق والعرض: أوجه الطلبة للقيام بما يأتي:

- توثيق المشروع: إعداد وثيقة توضح جميع مراحل العمل والخطوات المتبعة في تطوير التطبيق، مع إرفاق البرنامج والتصميمات.
- إعداد عرض تقديمي: تحضير عرض تقديمي يشرح فكرة المشروع، وكيفية تصميم البرنامج وتطويره، ونتائج الاختبارات والتحليل، والمقترحات لتحسين الكفاءة.
- تقديم المشروع: عرض المشروع والتطبيق النهائي أمام اللجنة أو الزملاء، مع توضيح أهمية البرمجة الخضراء وتأثيرها.

معايير التقييم:

- الإبداع: الابتكار في الحلول المقدمة لتحقيق أهداف البرمجة الخضراء.
- جودة الكود البرمجي وصحته: التحقق من خلو الكود من الأخطاء اللغوية والمنطقية.
- الاستخدام الصحيح والفعال للعناصر البرمجية: التقييم بناءً على كيفية استخدام الطلبة للعناصر البرمجية استخدامًا صحيحًا.
- الاشتمال على البنود المطلوبة: التأكد من أن البرنامج يشتمل على جميع البنود المطلوبة.
- إرفاق مخطط سهمي توضيحي: وجود مخطط شامل يوضح جميع جزئيات البرنامج.
- توظيف سيناريوهات الفحص: تتبع المقطع البرمجي والتأكد من خلوه من الأخطاء.
- الكفاءة: مدى كفاءة التطبيق في تحقيق الأهداف بأقل استهلاك للطاقة.
- العمل الجماعي وتقاسم الأدوار: فعالية التعاون وتوزيع الأدوار بين أعضاء الفريق.
- عرض المشروع: وضوح العرض التقديمي للبرنامج وشموليته.
- نشر البرنامج وتوثيق المصادر: التأكد من نشر البرنامج، وتوثيق جميع المصادر المستخدمة في المشروع.

النتائج النهائية المتوقعة:

- تطبيق يعمل على قياس استهلاك الطاقة.
- تحليل شامل لنتائج استهلاك الطاقة.
- مجموعة من الحلول لتحسين الكفاءة، وتقليل الاستهلاك.

للوصول إلى psutil واستخدامها في المشروعات البرمجية، يجب أولاً تثبيتها في بيئة بايثون الخاصة بالمستخدم عن طريق اتباع الخطوات الآتية:

تثبيت مكتبة psutil

يمكنك تثبيت المكتبة باستخدام أداة pip، التي تأتي مدمجة مع بايثون. افتح سطر الأوامر (Command Prompt) أو الطرفية (Terminal) واستخدم الأمر التالي لتثبيت psutil:

```
pip install psutil
```

بعد الانتهاء من التثبيت، يمكن التحقق من أن المكتبة قد تُثبِتَ بنجاح عن طريق تشغيل بايثون، ومحاولة استيراد المكتبة:

```
import psutil
```

```
__print(psutil.version) # يجب أن يظهر رقم الإصدار الحالي من المكتبة
```

إذا لم يظهر أي خطأ، فهذا يعني أن المكتبة قد تُثبِتَ بنجاح، ويمكن استخدامها في البرامج.

ملاحظات

```

1 import psutil
2 import time
3 # دالة للعثور على عملية معينة حسب اسمها
4 def find_process_by_name(name):
5     for proc in psutil.process_iter(['pid', 'name']):
6         if proc.info['name'] == name:
7             return proc
8     return None
9 # لعملية معينة (والذاكرة CPU) دالة لقياس استهلاك الموارد
10 def measure_process_performance(proc):
11     try:
12         cpu_usage = proc.cpu_percent(interval=1) / psutil.cpu_count()
13         memory_info = proc.memory_info()
14         memory_usage = memory_info.rss / (1024 ** 2) # تحويل الذاكرة إلى ميجابايت
15         return cpu_usage, memory_usage
16     except psutil.NoSuchProcess:
17         return None, None
18 # دالة لعرض التقرير النهائي
19 def display_report(cpu_usage, memory_usage):
20     print("\n--- Performance Report ---")
21     print(f"CPU Usage: {cpu_usage:.2f}%")
22     print(f"Memory Usage: {memory_usage:.2f} MB")
23     print("-----")
24 # الدالة الرئيسية
25 def main():
26     process_name = input("Enter the name of the process to monitor (e.g., chrome.exe): ")
27     # العثور على العملية
28     proc = find_process_by_name(process_name)
29     if proc is None:
30         print(f"Process {process_name} not found.")
31         return
32     print(f"Monitoring process: {process_name} (PID: {proc.pid})")
33     # قياس أداء العملية
34     cpu_usage, memory_usage = measure_process_performance(proc)
35
36     if cpu_usage is None or memory_usage is None:
37         print("The process has ended.")
38     else:
39         # عرض التقرير
40         display_report(cpu_usage, memory_usage)
41 if __name__ == "__main__":
42     main()

```

مثال آخر بسيط للبرنامج:

في إطار مشروع "البرمجة الخضراء"، نقدم برنامجًا بسيطًا مصممًا لتوعية الطلبة بأهمية قياس أداء العمليات البرمجية وتحليلها باستخدام لغة بايثون. يركز البرنامج على تنفيذ عمليات بسيطة مثل الحلقات والجمل الشرطية، وقياس الوقت المستغرق في تنفيذ هذه العمليات. عن طريق هذا البرنامج، يُشجّع الطلبة على التفكير في كيفية تحسين كفاءة الكود، وتقليل استهلاك الموارد الحاسوبية عن طريق تجربة الحلقات، والجمل الشرطية، والدوال البرمجية وتعديلها.

مميزات البرنامج:

- بسيط وسهل الفهم: يعتمد على مبادئ البرمجة الأساسية، مثل الجمل الشرطية والحلقات؛ مما يجعله مناسبًا للمبتدئين.
- قياس الأداء: يقدم طريقة لقياس الوقت المستغرق في تنفيذ عمليات معينة؛ مما يسمح للطلبة بتقييم كفاءة الكود.
- قابل للتطوير: يمكن للطلبة تعديل البرنامج لتجربة عمليات مختلفة، وتحسين الأداء بناءً على النتائج.

ملاحظات

```

1 import time
2
3 # دالة لحساب الوقت المستغرق في تنفيذ عملية معينة
4 def measure_time(func):
5     start_time = time.time() # تسجيل وقت البدء
6     func() # تنفيذ الدالة المطلوبة
7     end_time = time.time() # تسجيل وقت النهاية
8     return end_time - start_time # حساب الوقت المستغرق
9
10 # for عملية بسيطة تتضمن حلقة
11 def simple_operation():
12     total = 0
13     for i in range(1000000): # for حلقة
14         total += i
15     return total
16
17 # دالة لعرض التقرير
18 def display_report(time_taken):
19     print("\n--- Performance Report ---")
20     print(f"Time Taken: {time_taken:.6f} seconds")
21     print("-----")
22
23 # الدالة الرئيسية
24 def main():
25     print("Measuring performance...")
26
27     # قياس الوقت المستغرق في تنفيذ العملية البسيطة
28     time_taken = measure_time(simple_operation)
29
30     # عرض التقرير
31     display_report(time_taken)
32
33 if __name__ == "__main__":
34     main()
35

```


قائمة المراجع

- Dayal, B. (2032). Going green: The future of software development. Retrieved May 28, 2032, from <https://www.linkedin.com/pulse/going-green-future-software-development-bipin-dayal/>
- Electronic Health Solutions (EHS). (n.d.). ميكدج مانربلا - ميكدج مانربلا. Retrieved May 28, 2024, from <https://ehs.com.jo/ar/البرنامج-الحكيم-البرنامج-الوطني-لحوسبة-القطاع-الصحي>
- Geneva Environment Network. (n.d.). The growing environmental risks of e-waste. Retrieved May 28, 2024, from <https://www.genevaenvironmentnetwork.org/resources/updates/the-growing-environmental-risks-of-e-waste/>
- Greenly. (2023). What is green computing and what does it imply? Retrieved May 28, 2024, from <https://greenly.earth/en-us/blog/company-guide/what-is-green-computing-and-what-does-it-imply#anchor-3>
- International Telecommunication Union, & United Nations Institute for Training and Research. (2024). The Global E-waste Monitor 2024. Retrieved May 28, 2024, from https://ewastemonitor.info/wp-content/uploads/202403//GEM_2024_1803-_web_page_per_page_web.pdf
- Jordan Government. (n.d.). Home. Retrieved May 28, 2024, from <https://portal.jordan.gov.jo/wps/portal/Home#/>
- Lancaster University. (n.d.). Green computing: A contribution to save the environment. Retrieved May 28, 2024, from <https://www.lancaster.ac.uk/data-science-of-the-natural-environment/blogs/green-computing-a-contribution-to-save-the-environment>
- Ministry of Digital Economy and Entrepreneurship. (n.d.). تمينورتكلالا تموكحلا جمانرب قراڊا. Retrieved May 28, 2024, from https://www.modee.gov.jo/Ar/Pages/الإلكترونية_الحكومة_إدارة_برنامج
- Ministry of Environment, Jordan. (2024). ببساولحلا ريودت قداعا عورشم: قراولما تاعورشم. Retrieved May 28, 2024, from https://moenv.gov.jo/AR/ListDetails/5/35/الوزارة_مشروعات
- ObjectBox Limited. (2023, April 19). Green coding: Developing sustainable software for a greener future. Retrieved May 28, 2024, from <https://objectbox.io/green-coding/>
- Paul, S. G., et al. (2023). A comprehensive review of green computing: Past, present, and future research. IEEE Access, 11, 87445-87494. <https://doi.org/10.1109/ACCESS.2023.3304332>
- Shalini, K., & Prasanthi, N. (2013). Green computing. IOSR Journal of Computer Engineering (IOSR-JCE), 2(17), 25-29. ISSN: 2278-0661, ISBN: 2278-8727. Retrieved May 28, 2024, from <https://www.iosrjournals.org/iosr-jce/papers/sicete-volume217/.pdf>
- Shalini, K., & Prasanthi, Naga. (2013). Green Computing. Journal of Telematics and Informatics, 1, 1-13. <https://doi.org/10.12928/jti.v1i1.113->
- Tang, M., Wu, X., Xu, J., Liu, J., Li, Z., Gao, J., & Tian, Z. (2023). Indoor temperature control of radiant ceiling cooling system based on deep reinforcement learning method. Buildings, 13(9), 2281. <https://doi.org/10.3390/buildings13092281>
- Vikram, S. (2015). Green computing. 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), 767-772. <https://doi.org/10.1109/ICGCIoT.2015.7380566>
- World Health Organization. (2023, October 18). Electronic waste (e-waste). Retrieved May 28, 2024, from [https://www.who.int/ar/news-room/fact-sheets/detail/electronic-waste-\(e-waste\)](https://www.who.int/ar/news-room/fact-sheets/detail/electronic-waste-(e-waste))

تم بحمد الله