



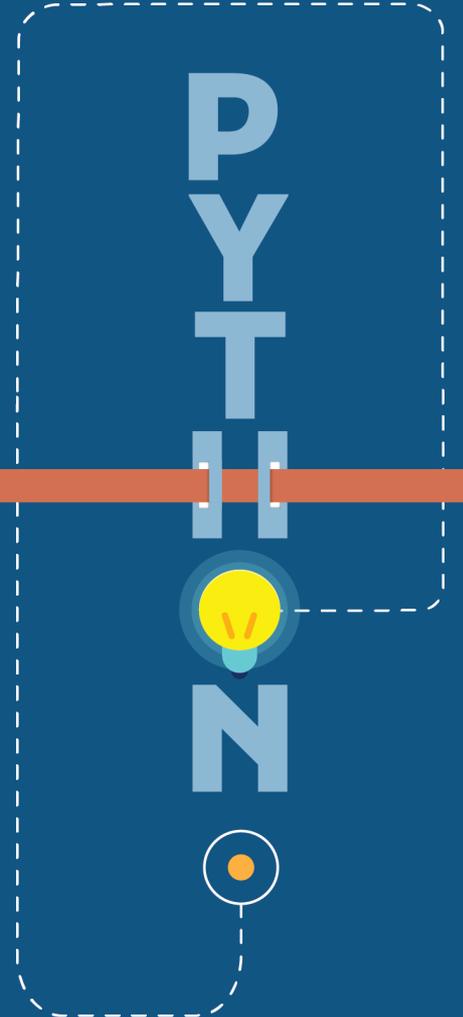
وزارة التربية

دليل تقنية

المعلم المعلومات

الصف العاشر

Cyber
Security



دليل المعلم تقنية المعلومات

الصف العاشر

إشراف

أ/ منى سالم عوض سالم

أ/ فاطمه خليف غنيم العنزي

أ/ ماجدة إبراهيم مزعل الدوسري

أ/ ساره ياسين عبدالله الامير

أ/ إيمان عبد العزيز أحمد الفارسي

إخراج

أ/ سنيه محمد علي المؤمن

المحتوى

6	مقدمة
8	1 الاستراتيجية التعليمية
19	2 الوحدة الأولى: الأمن السيبراني
20	- خطة الدروس
21	- نتائج التعلم
23	- مصطلحات حاسوبية
25	- أوراق العمل
27	3 الوحدة الثانية: برمجة بايثون
28	مدخل إلى البرمجة Introduction to programming
28	- خطة الدروس: مدخل إلى البرمجة
29	- نتائج التعلم : مدخل إلى البرمجة
30	- مصطلحات حاسوبية: مدخل إلى البرمجة
32	- الفرق بين المترجم (Compiler) والمفسر (Interpreter)
32	- ملاحظات للمعلم
34	- أوراق العمل: مدخل إلى البرمجة
39	المتغيرات Variables
39	- خطة الدروس: المتغيرات Variables
40	- نتائج التعلم: المتغيرات Variables
40	- مصطلحات حاسوبية: لمتغيرات Variables
41	- أوراق العمل Variable
44	الشروط Condition
44	- خطة الدروس: الشروط Condition
45	- نتائج التعلم: الشروط Condition
45	- مصطلحات حاسوبية: الشروط Condition
46	- أوراق العمل: الشروط Condition



57	التكرار loop	
57	- خطة الدروس: التكرار loop	
58	- نتائج التعلم: التكرار loop	
58	- مصطلحات حاسوبية: التكرار loop	
59	- حل أمثلة: التكرار loop	
61	- أوراق العمل: التكرار loop	
67	السلاسل النصية Strings	
68	- خطة الدروس: السلاسل النصية Strings	
69	- نتائج التعلم: السلاسل النصية Strings	
66	- مصطلحات حاسوبية: السلاسل النصية Strings	
71	- أوراق العمل: السلاسل النصية Strings	
76	القوائم List	
76	- خطة الدروس: القوائم List	
77	- نتائج التعلم: القوائم List	
78	- مصطلحات حاسوبية: القوائم List	
79	- ملاحظات للمعلم	
80	- أوراق العمل: القوائم List	
88	برمجة Python : المنتجات الرقمية	4
88	- نتائج التعلم: المنتجات الرقمية	
89	- مراحل إعداد المشروع	
90	- المهارات المقترحة للمشروع	
93	- أمور يجب مراعاتها في وحدة المنتجات الرقمية	
95	تحضير الدروس	5
98	خطوات إعداد الدرس	
99	خطة درس مادة الحاسوب للصف العاشر	
104	التقويم التربوي	6
105	استراتيجيات التقويم	
108	توزيع درجات مادة الحاسوب	
110	الاختبارات	7
113	الميزان النسبي/جدول مواصفات الاختبار	

الخاتمة

المقدمة

عزيزي المعلم

يسعدنا أن نقدم لك هذا الدليل الشامل الذي يهدف إلى دعمك في تقديم مادة تقنية المعلومات لمتعلمي الصف العاشر بما في ذلك مفاهيم الأمن السيبراني وبرمجة بايثون و بصورة فعالة ومبتكرة، نحن ندرك أن تقديم هذه المادة الحيوية يحتاج إلى أدوات ومعارف حديثة تتماشى مع تطورات العصر الرقمي واحتياجات المتعلمين.

يتضمن هذا الدليل استراتيجيات تعليمية متنوعة تناسب مع الفئة العمرية للمتعلمين، مثل التعلم المباشر (المحاضرة)، التعلم التعاوني، المناقشة والحوار، التعلم القائم على حل المشكلات، والتعلم القائم على المشاريع، كما يوفر الدليل أنواعًا مختلفة من التقويمات، سواء التكوينية منها أو النهائية، لضمان تقييم شامل لتقدم المتعلمين ومساعدتهم على تحسين أدائهم بشكل مستمر.

لضمان تجربة تعليمية شاملة، يحتوي هذا الدليل على خطط مفصلة لكل وحدة دراسية تشمل أهداف الوحدة، الموضوعات الرئيسية، الأنشطة التعليمية، المواد والأدوات اللازمة، وأساليب التقويم. تم تصميم نتائج التعلم بوضوح لكل وحدة لتمكين المتعلمين من معرفة ما يُتوقع منهم تحقيقه بنهاية كل درس.

فيما يخص الأمن السيبراني، ستجد في هذا الدليل توجيهات حول كيفية تقديم مفاهيم الأمن الأساسية، مثل الحفاظ على سرية البيانات وسلامتها وإمكانية الوصول إليها. سنتعلم كيفية تعريف المتعلمين على التهديدات السيبرانية المختلفة وطرق الوقاية منها.

أما بالنسبة لبرمجة بايثون، فإن الدليل يحتوي على موارد مفيدة لشرح المفاهيم البرمجية الأساسية، بما في ذلك أمثلة عملية وأنشطة تطبيقية لمساعدة المتعلمين على فهم البرمجة بطريقة سلسلة وممتعة. كما يتضمن الدليل قائمة شاملة بالمصطلحات الحاسوبية الهامة، بالإضافة إلى الحلول النموذجية لأوراق العمل لتيسير عملية التصحيح. يهدف هذا الدليل إلى تزويدك بالأدوات والمعارف اللازمة لتحسين عملية التعلم والتعليم، ودعمك في التحضير الفعال للدرس وتنفيذه بنجاح.

نحن على يقين بأن هذا الدليل سيكون مرشدك الأمثل في رحلتك التعليمية، ويساعدك على تقديم دروس ممتعة، تساهم في تطوير مهارات المتعلمين وتعزيز فهمهم للتكنولوجيا، الأمن السيبراني، وبرمجة بايثون. نأمل أن تجد في هذا الدليل كل ما تحتاجه لتحقيق أفضل النتائج في تدريس منهج الصف العاشر

مع أطيب التمنيات

الاستراتيجيات التعليمية

01



الإستراتيجيات التعليمية:

عزيزي المعلم

في ظل التطور السريع في مجال التعليم وتكنولوجيا المعلومات، أصبح من الضروري اعتماد استراتيجيات تعليمية مبتكرة ومصادر تعلم متنوعة لتلبية احتياجات المتعلمين وتحفيزهم على التعلم الفعّال ، من خلال استراتيجيات التعليم الحديثة مثل التعلم التعاوني والتعلم القائم على حل المشكلات ، بالإضافة إلى التعلم القائم على المشاريع والعصف الذهني وغيرها من الاستراتيجيات التي تتناسب مع مادة تقنية المعلومات والفئة العمرية للصف العاشر، والتي بدورها تساعد المتعلمين على تطوير مهارات التفكير النقدي والإبداعي، وتوسع من مدارك المتعلمين وتعزز فهمهم.

ومن المصادر التي تعتمد عليها هذه الاستراتيجيات الكتب الدراسية والمراجع الإلكترونية والفيديوهات التعليمية وغيرها من المصادر الحديثة، بالإضافة إلى ذلك يلعب التقويم دورًا حيويًا في العملية التعليمية، مثل التقويم التشخيصي والتكويني والختامي، واستراتيجياتها المختلفة، وأدواتها التي يجب أن يستخدمها المعلمون لتقييم تقدم المتعلمين، وتتنوع هذه الأدوات لتشمل أساليب متنوعة تلبي احتياجات المتعلمين وتوفر تغذية راجعة شاملة من اختبارات عملية ومشاريع وغيرها، حيث تلعب دورًا أساسيًا في قياس تقدم المتعلمين وتحديد نقاط القوة والضعف لديهم، وبتكامل هذه العناصر يمكننا تحقيق بيئة تعليمية شاملة تشجع على التعلم المستمر وتدعم تحقيق الأهداف التعليمية.



الإستراتيجيات التعليمية:

الاستراتيجيات التعليمية هي الخطط والأساليب والتقنيات التي يستخدمها المعلمون لتسهيل وتحسين عملية التعلم والتعليم، تهدف هذه الاستراتيجيات إلى جعل التعليم أكثر فعالية، وكفاءة، وجاذبية للمتعلم ، يمكن أن تتضمن الاستراتيجيات التعليمية مجموعة متنوعة من الأنشطة والتدريبات والتوجيهات التي تساعد المتعلمين على فهم المفاهيم واكتساب المهارات والمعرفة بطرق متنوعة تلائم احتياجاتهم التعليمية.

التعلم المباشر (المحاضرة) :



التعلم المباشر هو أسلوب تعليمي يعتمد على تقديم المعلم للمعلومات مباشرة إلى المتعلمين في بيئة صفية تقليدية، يكون المعلم هو المصدر الأساسي للمعرفة، بينما يكون دور المتعلم هو الاستماع، والتدوين، وطرح الأسئلة، والمشاركة في النقاشات، يكون أكثر فاعلية عند تدريس فكرة أو مفهوم.

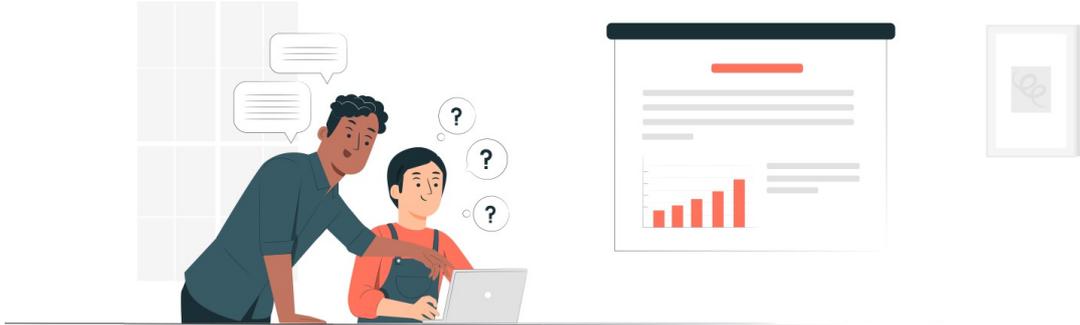
تقديم المعلم أمثلة لبعض الوظائف المستحدثة في مجال الامن السيبراني.
شرح مناطق نافذة محرر برنامج By charm.

مثال:



الإستراتيجيات التعليمية:

استراتيجية المناقشة والحوار



هي أسلوب تعليمي يهدف إلى تعزيز التعلم من خلال تبادل الأفكار والمعلومات بين المعلم والمتعلمين أو بين المتعلمين أنفسهم. تعتبر هذه الاستراتيجية من الطرق الفاعلة التي تشجع على التفكير النقدي، وتطوير مهارات التواصل، وتعميق الفهم من خلال النقاشات المفتوحة والموجهة.

طرح سؤال على المتعلمين

ما هي مميزات برنامج Python؟ (لماذا ندرس برنامج Python؟)

- سهولة التعلم والاستخدام.
- تطوير برمجيات معقدة من خلالها.
- تستخدم في مجالات متعددة مثل تطبيقات الويب، الذكاء الاصطناعي.
- يحتوي على العديد من المكتبات الجاهزة.
- تتوافق مع العديد من أنظمة التشغيل.
- التحديث المستمر.

مثال :



الإستراتيجيات التعليمية:

التعلم التعاوني



التعلم التعاوني هو أسلوب تعليمي يعتمد على تقسيم المتعلمين إلى مجموعات صغيرة للعمل معًا لتحقيق أهداف تعليمية مشتركة. بقصد التعاون فيما بينهم والمشاركة في حل المشكلات، من خلال المشاركة في المعلومات والموارد، مما يعزز التعلم النشط والتفاعل.

تقسيم المتعلمين إلى مجموعات كل مجموعة لا تزيد عن ثلاثة متعلمين ويتم توزيع أسئلة مطبوعة على أوراق يطلب من كل مجموعة حلها مثل كتابة تعليمات:

- ايجاد باقي القسمة.
- ادخال رقمين ثم طباعة ناتج الجمع.
- حساب مساحة الدائرة.

مثال:



الإستراتيجيات التعليمية:

التعلم القائم على حل المشكلات

هو أسلوب تعليمي يركز على تقديم مشكلات معقدة وواقعية للمتعلمين للعمل على حلها، يهدف إلى تطوير مهارات التفكير النقدي، وحل المشكلات، والتعلم الذاتي، والعمل الجماعي، يتم تشجيع المتعلمين على البحث، والتحليل، والتفكير بشكل مستقل وجماعي للتوصل إلى حلول مناسبة.

مثال:

برنامج تحويل وحدة القياس الزمن من الدقيقة الى الثانية والعكس.

مشكلة البرنامج:

تحويل وحدة القياس الزمن من الدقيقة الى الثانية والعكس

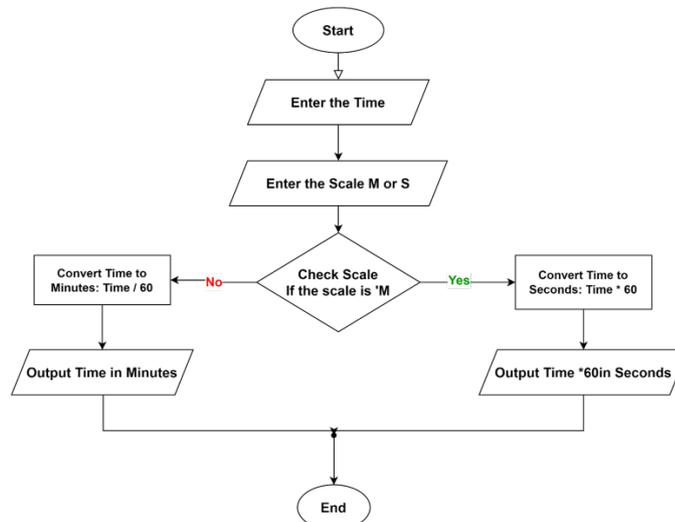
الخوارزمية:

- استقبال من المستخدم قيمة الزمن.
- استقبال من المستخدم وحدة قياس الزمن المطلوب التحويل اليها سواء (دقيقة أو ثانية)
- استخدم المعادلة التالية لتحويل وحدة قياس الزمن بالدقيقة الى الثانية: $S = (\text{time} * 60)$
- استخدم المعادلة التالية لتحويل وحدة قياس الزمن بالثانية الى الدقيقة: $M = (\text{time} / 60)$
- اعرض قيمة الزمن بعد التحويل لوحدة القياس المطلوبة.

المطلوب:

- ارسم خريطة التدفق المناسبة.
- كتابة التعليمات البرمجية اللازمة بشكل صحيح واختبار البرنامج

خريطة التدفق





التعليمات البرمجية

input

```
Time converter.py ×
1  # استقبال الوقت من المستخدم
2  Time = float(input('Enter the time : '))
3  # استقبال مقاييس الوقت من المستخدم
4  Scale = input('Enter the scale Time M for Minute Or S for Second:')
5  # تحويل الوقت بناءً على المقاييس المدخل
6  if Scale == 'M':
7      print(Time * 60, 's')
8  else:
9      print(Time / 60, 'M')
```

Out put

```
Enter the time: 2
Enter the scale Time M for Minute or S for Second: M
120.0 S
```



الإستراتيجيات التعليمية:

العصف الذهني



استراتيجية تهدف إلى توليد أكبر عدد ممكن من الأفكار حول موضوع معين في بيئة جماعية، يدعو إلى الإبداع والتفكير الحر بين المتعلمين، و دور المعلم هو التوجيه والارشاد لتحقيق الأهداف المرجوة.

اكتب برنامج بسيط python للحصول على الشكل التالي:

مثال:

input	Out put
<pre>1 # عدد الصفوف 2 n = 4 3 4 for i in range(1, n + 1): 5 # طباعة نجمة واحدة متبوعة بمسافة 6 print("* " * i) 7</pre>	<pre>Run: star x C:\Users\HP\PycharmProjects\pythonF * * * * * * * * * * Process finished with exit code 0</pre>



الإستراتيجيات التعليمية:

التعلم القائم على المشاريع



التعلم القائم على المشاريع هو نهج تعليمي يركز على إشراك المتعلمين في مشاريع أما في صورة مستقلة أو في اطار تعاوني تتطلب منهم البحث، والتخطيط، والتنفيذ، والتقييم، يكون دور المعلم هو تقديم التوجيه والإرشاد للمتعلمين من أجل اكمال مشاريعهم بنجاح واكتساب فهم عميق للمفاهيم الأساسية .

1. يقسم المعلم المتعلمين الى مجموعات ويوضح لهم آلية العمل.
2. تقوم كل مجموعة بتعبئة نموذج التخطيط للمشروع.
3. يبدأ المتعلمين بتنفيذ المهام حسب الخطة الموضوعة.
4. يقيم المعلم مشاريع المجموعات باستخدام استمارة تقييم المشروع.

مثال :

التعلم بالأقران

استراتيجية يكتسب فيها المتعلمين مهاراتهم ومعارفهم بمساعدة بعضهم البعض تحت إشراف المعلم، بحيث يكلف المعلم متعلم متفوق في مجال ما بدور الخبير، فيقوم بتعليم متعلم آخر أقل مستوى أو مجموعة معينة ويقودهم نحو فهم الهدف أو اتقان المهارة.

- أثناء الحصة الدراسية يتم تقسيم المتعلمين الى مجموعات ثنائية.
بالرجوع للكتاب، ينفذ المتعلمين معا خطوات احدي أوراق العمل في الكتاب المدرسي.

مثال :



الإستراتيجيات التعليمية:

استراتيجية الصف المقلوب



استراتيجية تعليمية تعتمد على تزويد المتعلمين بمادة تعليمية حول الدرس كمقاطع فيديو أو ملفات صوتية أو غيرها من الوسائط، للاطلاع عليها في منازلهم أو في أي مكان آخر باستعمال حواسيبهم أو هواتفهم الذكية أو أجهزةهم اللوحية قبل حضور الدرس. في حين يُخصّص وقت الدرس للمناقشات والمشاريع والأنشطة.

مثال:

- يتم تقديم مقاطع فيديو قصيرة تشرح المتغيرات وأنواع البيانات المختلفة في برنامج Python.
- يُطلب من المتعلمين قراءة مصادر نصية توضح كيفية استخدام المتغيرات وتعديل قيمها في برنامج Python.
- في وقت الحصة الدراسية، يتم تنظيم جلسات تطبيقية تهدف إلى تطبيق ما تم مشاهدته في مقاطع الفيديو المرسلة مسبقًا.
- يُطلب من المتعلمين كتابة برامج بسيطة تستخدم المتغيرات وتطبق المفاهيم التي تعلموها.
- يقدم المعلم التوجيه أثناء العمل، ويُجيب على الأسئلة.
- يتم توجيه المتعلمين لتصحيح الأخطاء وتحسين أساليبهم في البرمجة.

input

```
1  تعريف المتغيرات
2  age = 25
3  name = "Ahmed"
4  is_student = True
5  height = 175.5
6
7  طباعة قيم المتغيرات
8  print("Name:", name)
9  print("Age:", age)
10 print("Is student:", is_student)
11 print("Height:", height)
```

Out put

```
Run: num x
Name: Ahmed
Age: 25
Is student: True
Height: 175.5
Process finished with exit code 0
```



الإستراتيجيات التعليمية:

استراتيجية الخرائط المفاهيمية



استراتيجية تعليمية فاعلة في تمثيل المعرفة عن طريق أشكال تخطيطية تربط المفاهيم وبعضها البعض بخطوط او أسهم يكتب عليها كلمات تسمى كلمات الربط لاكتشاف العلاقات بين المفاهيم وتعميق الفهم وتلخيص المعلومات وتقويم الدرس.

مثال: تطبيق استراتيجية الخرائط المفاهيمية في حل مشكلة جمع عددين في Python .

التعليمات البرمجية

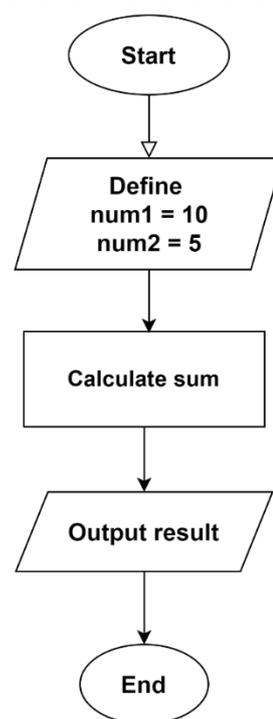
input

```
1 # تعريف المتغيرات
2 num1 = 10
3 num2 = 5
4
5 # الجمع
6 result = num1 + num2
7
8 # طباعة النتيجة
9 print("الناتج هو:", result)
10
```

Out put

```
Run: sum x
C:\Users\HP\PycharmProjects\pythonProject4\.venv\Scripts\python.exe
الناتج هو: 15
Process finished with exit code 0
```

خريطة التدفق



الوحدة الأولى:

الأمن السيبراني Cyber Security

02



وصف الوحدة: الأمن السيبراني Cyber Security

في هذه الوحدة، سنتعرف على المفاهيم الأساسية للأمن السيبراني Cybersecurity، ونتعلم كيفية تطبيق التدابير الأمنية الأساسية لحماية المعلومات والنظم، ونستكشف التحديات المختلفة التي تواجه الأمن السيبراني Cybersecurity وكيفية التعامل معها بفعالية، مما يساعدنا على فهم الدور الحاسم للأمن السيبراني Cybersecurity في عالمنا الرقمي المعاصر.

حيث مع التقدم التكنولوجي المتسارع وانتشار الإنترنت في جميع جوانب الحياة اليومية، أصبحت المعلومات والبيانات الرقمية تشكل جزءاً أساسياً من بنيتنا التحتية الحديثة، سواء كان ذلك في القطاعات الحكومية، التجارية، الصحية، أو التعليمية.

واليوم يعتمد العالم بشكل كبير على الأنظمة الرقمية لتخزين ومعالجة وتبادل المعلومات لكن، مع هذا الاعتماد المتزايد تبرز تحديات جديدة تتعلق بحماية هذه المعلومات من الهجمات السيبرانية التي يمكن أن تسبب خسائر مالية هائلة وتهدد الخصوصية والأمن القومي.

وبذلك ممكن تعريف الأمن السيبراني Cybersecurity، بأنه مجال حيوي يهدف إلى حماية الأنظمة والشبكات والبرامج من الهجمات الرقمية، ويشمل مجموعة واسعة من التدابير والتقنيات المصممة للحفاظ على سرية وسلامة وتوافر المعلومات، بالإضافة إلى ضمان عدم تعرضها للاختراق أو السرقة أو التلف باستخدام مجموعة من الأدوات والتقنيات مثل التشفير، والجدران النارية، وأنظمة الكشف عن التسلسل ومنع التسلسل.

إلى جانب الأدوات والتقنيات، يلعب الوعي الأمني والتدريب دوراً محورياً في تعزيز الأمن السيبراني Cybersecurity. من خلال تدريب الأفراد على التعرف على التهديدات والتعامل معها، وتطبيق سياسات أمنية صارمة مثل تحديثات البرمجيات وإدارة كلمات المرور والنسخ الاحتياطي للبيانات، يمكن تقليل المخاطر وتحسين الحماية بشكل كبير.



خطة الدروس: الوحدة الأولى الأمن السيبراني Cyber Security

 Cyber Security 1	حصة دراسية	مفاهيم الأمن السيبراني .	
 Cyber Security 2	حصة دراسية	المحاور الرئيسية للأمن السيبراني CIA .	
 Cyber Security 3	حصة دراسية	تطبيقات أمنية للمحاور الرئيسية للأمن السيبراني CIA .	
 Cyber Security 4	حصة دراسية	ممارسات الأمن السيبراني .	
 امسح الرمز	حصة دراسية	وظائف الأمن السيبراني .	



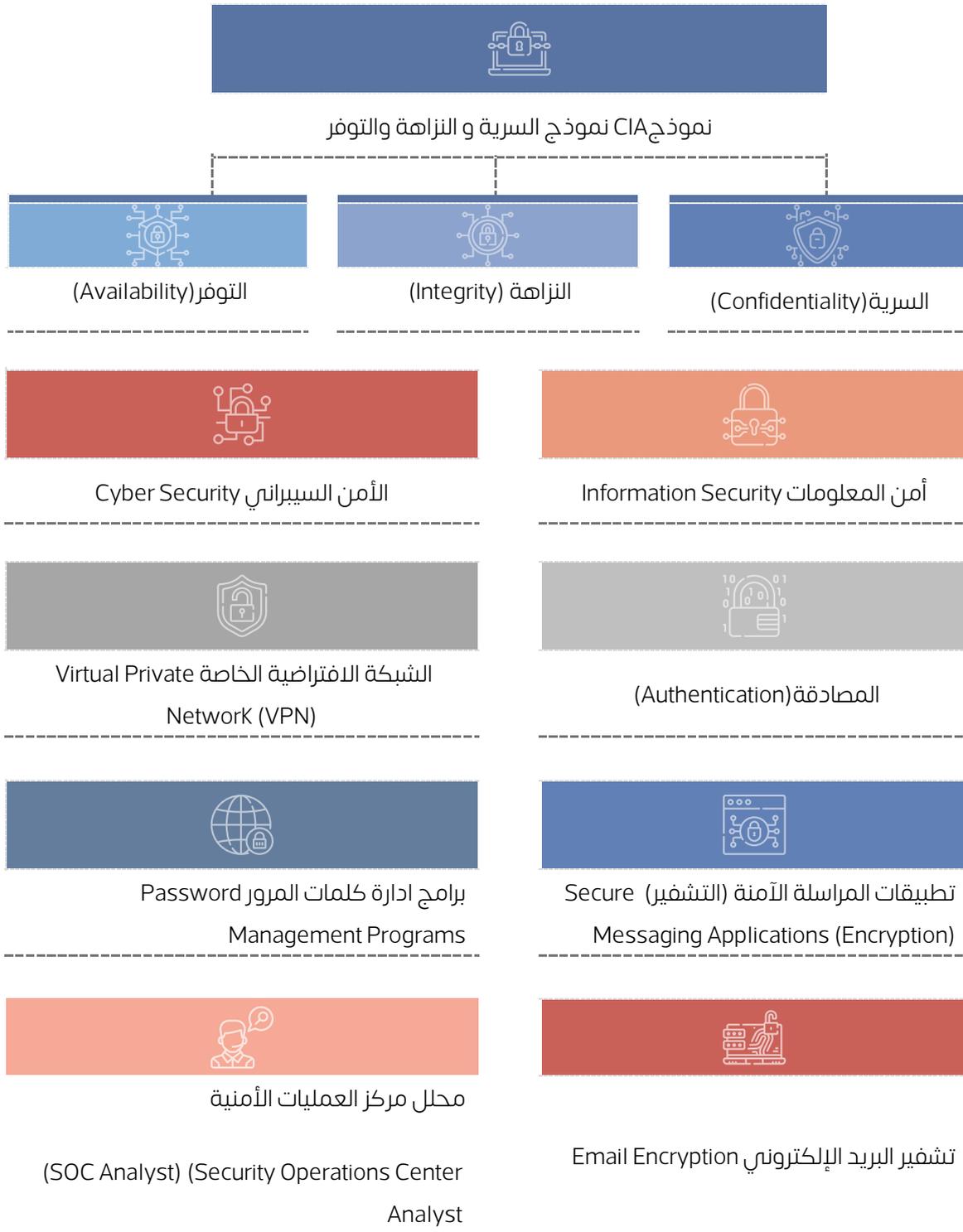
نتائج التعلم:

1	الفهم الشامل لمبادئ وممارسات أمن المعلومات.
2	التزود بالمعرفة والمهارات اللازمة لحماية الأنظمة والشبكات والمعلومات ضد الهجمات السيبرانية.
3	تمييز الفرق بين مفهوم الأمن السيبراني وأمن المعلومات.
4	التعريف بمفهوم الأمن السيبراني وأهميته لحماية الأنظمة والشبكات.
5	تعزيز الوعي الأمني لمخاطر الهجمات السيبرانية عند المتعلمين.
6	التعرف على المحاور الرئيسية لأمن المعلومات والأمن السيبراني CIA (السرية – النزاهة – التوفر).
7	التأكيد على أهمية حماية المعلومات وحفظ سريتها.
8	التعريف بمفهوم النزاهة وموثوقية المعلومات الرقمية وسلامتها.
9	التعريف بمفهوم التوفر وإتاحة المعلومات والموارد باستمرار عند الحاجة.
10	التعرف على أمثلة توضح الهجمات التي تستهدف المحاور الرئيسية لأمن المعلومات.
11	تطبيق مبادئ CIA على مثال ماكينة الصراف الآلي.
12	التعرف على الأشكال المختلفة للمصادقة التي تستخدم لحماية حسابات المستخدمين (عامل المعرفة – عامل الامتلاك – عامل المقياس الحيوي).
13	عرض أمثلة لبعض عوامل المصادقة المستخدمة.
14	التعرف على الشبكة الافتراضية الخاصة VPN ووظيفتها في إنشاء اتصال آمن.
15	التعرف على وظيفة تطبيقات التشفير لحماية خصوصية وسرية المحادثات.
16	التأكيد على أهمية إنشاء كلمات مرور قوية وفريدة لحماية حسابات المستخدمين.
17	التعرف على الخدمات الرقمية التي تتيح تشفير الرسائل الإلكترونية.
18	تطبيق مثال على تشفير الرسائل الإلكترونية من خلال أداة Flow Crypt.
19	التعرف على خاصية التصفح المتخفي للحفاظ على مستوى أعلى من الخصوصية أثناء تصفح الانترنت.

التعرف على طرق الدفع الآمنة لحماية المعلومات المالية وتقليل مخاطر الوصول اليها.	20
تطبيق بعض الإجراءات والتدابير الخاصة من أجل الحفاظ على CIA في الأمن السيبراني.	21
التعرف على الوظائف المستحدثة في مجال الأمن السيبراني لتوسيع مدارك المتعلم عن الفرص الوظيفية المستحدثة المتاحة.	22



مطلحات حاسوبية – Key Words





الخصوصية في متصفحات الانترنت

Privacy-Focused Browser Alternatives



Brave



Mozilla Firefox



المحافظ الرقمية Digital Wallets



التصفح المتخفي Incognito



ممارسات الأمن السيبراني
Practices Cybersecurity



طرق الدفع الأمنة
Methods Secure Payment



إدارة عملية مشاركة البيانات
Management "Data Sharing



التصيد الاحتيالي (Phishing)



محلل الأمن السيبراني (Cybersecurity Analyst)



الإبلاغ Reporting



مستشار أمني (Security Consultant)



مهندس أمن (Security Engineer)



المستجيب للحوادث (Incident Responder)



المخترق الأخلاقي (Ethical Hacker)



أوراق العمل.

- يتم شرح محتوى أوراق العمل من خلال اعداد فيديو يقدم أثناء الحصة الدراسية على يستفيد منها المتعلم في استخدامه لأجهزته الخاصة من باب التوعية في الأمن السيبراني.

أوراق عمل
(1-2-3) اللاصفية

- يستعرض المعلم أمثلة أخرى حياتية ليناقشها مع المتعلمين.

الوحدة الثانية: برمجة بايثون

03



وصف الوحدة الثانية: برمجة بايثون.

في عصرنا الرقمي، تلعب البرمجة دورًا حيويًا في تطوير التكنولوجيا الحديثة التي نعتمد عليها يوميًا.

في هذه الوحدة، "مدخل إلى البرمجة"، سنتعرف على أهمية مفهوم البرمجة في تحويل الأفكار إلى تطبيقات وأدوات عملية تُستخدم في شتى المجالات من خلال كتابة تعليمات وأوامر لجهاز الحاسوب لتنفيذ مهام محددة مثل تطوير التطبيقات، المواقع الإلكترونية، الألعاب، والأنظمة التكنولوجية الأخرى.

يتبعها مراحل إنشاء برنامج حاسوبي، بدءًا من تحليل المشكلة، مرورًا بكتابة الخوارزمية، وتصميم خريطة تدفق الإجراءات لحل المشكلة، وصولاً إلى كتابة الكود الفعلي، سنتعرف على لغة البرمجة بايثون (Python) التي تُعد واحدة من أكثر لغات البرمجة شيوعًا بفضل سهولتها ومرونتها من خلال محرر الأوامر (PyCharm) وهو بيئة تطوير متكاملة تساعد في كتابة وتحرير الأكواد بكفاءة، بالإضافة إلى ذلك، سنستكشف أنواع هياكل البيانات في بايثون، مثل القوائم، المجموعات، القواميس، والصفوف، وكيفية استخدامها لتنظيم وتخزين البيانات بطرق فعّالة.

من خلال هذه الوحدة، سيكتسب المتعلم فهمًا عميقًا للمفاهيم الأساسية للبرمجة، مما يمكنهم من تطبيق هذه المعرفة في تطوير حلول برمجية فعّالة، ويفتح أمامهم آفاقًا واسعة في عالم التكنولوجيا.



مدخل إلى البرمجة Introduction to programming

خطة الدروس: مدخل إلى البرمجة



Python 1

حصة دراسية

- مدخل إلى البرمجة: مفهوم البرمجة.
- خطوات انشاء برنامج لغة البرمجة (python).
- كتابة خوارزمية لحل مشكلة.
- تصميم خريطة تدفق لإجراء حل مشكله.
- استخدام محرر الأوامر (PyCharm).



python 2

حصة دراسية

- مدخل إلى البرمجة: التعرف على الدالة المدمجة (Print).
- التعرف على هياكل البيانات.
- معرفة أنواع هياكل البيانات في بايثون.



امسح الرمز



نتائج التعلم مدخل إلى البرمجة:

1	التعرف على مفهوم البرمجة.
2	تعداد بعض فوائد البرمجة.
3	التعرف على خطوات إنشاء برنامج حاسوبي.
4	التعرف على مفهوم الخوارزمية.
5	التعرف على خرائط التدفق (Flow Chart).
6	التمييز بين أشكال ووظائف المخططات الرسومية.
7	التعرف على مفهوم لغة البرمجة (Python).
8	تعداد بعض استخدامات لغة python في المجالات المختلفة.
9	تعداد مميزات لغة (python).
10	التعرف على بعض مواقع تحرير التعليمات البرمجية على الانترنت.
11	التعرف على برنامج (PyCharm) وكيفية تحميله.
12	التعرف على مكونات نافذة المحرر.
13	إنشاء مشروع جديد.
14	إنشاء ملف بايثون جديد.
15	كتابة تعليمة برمجية باستخدام دالة (print).
16	التعرف على طرق تشغيل برنامج (python).
17	التعرف على مفهوم هياكل البيانات.
18	التعرف على أنواع هياكل البيانات.



مصطلحات حاسوبية مدخل إلى البرمجة



الخوارزمية (Algorithm)



البرمجة (Programming)



عملية إدخال البيانات: (Input)



خرائط التدفق (Flow Chart)



عملية المعالج (Process)



عملية إخراج البيانات (Output)



لغة البرمجة (Programming Language)



اتخاذ قرار (Decision)



محرر التعليمات البرمجية (Code Editor)



محرر الأوامر (PyCharm)



ملف بايثون (Python file)



مجلد أو مشروع الذي يحوي ملفات البايثون
(Project)



المترجم (Compiler)



الامر Run

مصطلحات حاسوبية مدخل إلى البرمجة



القوائم الرئيسية (Main Menu)



المفسر (Interpreter)



هياكل البيانات الأولية
(Primitive Data Structures)



هياكل البيانات (Data structure)



الأعداد العشرية (Float)



الأعداد الصحيحة (Integer)



البيانات المنطقية (Boolean)



السلسلة النصية (String)



هياكل البيانات الخطية
(Linear Data Structures)



هياكل البيانات غير الأولية
(Non-primitive Data Structures)



المجموعة (Set)



قائمة (List)



الصف (Tuple)



القاموس (Dictionary)



الشجرة (Tree)



هياكل البيانات غير الخطية
(Non-linear Data Structures)



الرسم البياني (Graph)



ما هو الفرق بين المترجم (Compiler) والمفسر (Interpreter)؟

المقارنة	المترجم (Compiler)	المفسر (Interpreter)
التنفيذ 	ترجمة التعليمات البرمجية بالكامل دفعة واحدة وتنفيذه كبرنامج مستقل.	ترجمة التعليمات سطرًا بسطرًا أو فقرة من البرنامج.
السرعة 	- بطيء في وقت الترجمة. - سريع في وقت التشغيل.	- سريع في وقت الترجمة. - بطيء في وقت التشغيل.
اكتشاف الأخطاء 	يكتشف الأخطاء دفعة واحدة قبل التشغيل.	يكتشف الأخطاء سطرًا بسطرًا أثناء التنفيذ.
حجم البرنامج 	ينتج ملفًا تنفيذيًا منفصلاً.	لا ينتج ملفًا تنفيذيًا مستقلًا.

ملاحظات للمعلم:

شرح واجهة المستخدم: عند شرح نافذة المحرر يجب التطرق الى مكونات واجهة المستخدم في PyCharm بما في ذلك نافذة المشروع، نافذة الإخراج (console) ، نافذة التوجيه، ونوافذ الأدوات الأخرى.



التأكيد على خطوات حفظ الملف وامتداده .py ، مثلًا pract1.py.



تخصيص الواجهة: يمكن تخصيص واجهة البرنامج بناءً على احتياجات المتعلمين، مثل تغيير حجم الخط أو تخصيص الألوان لتسهيل القراءة.



التأكيد على أهمية إغلاق المشروع لكل متعلم بعد انتهاء جلسة العمل لضمان استمرار كل متعلم في تنفيذ أوراق العمل الخاصة به فقط.



لتغيير نوع وحجم الخط في PyCharm، يمكنك اتباع الخطوات التالية:



1 اختيار الامر settings من قائمة file.

2 تحديد الاختيار Editor من القائمة الجانبية ل appearance and behavior من نافذة Settings.

3 اختيار الامر الفرعي Font من Editor ثم تحديد الخيارات المطلوبة.

4 تطبيق التغييرات الضغط Apply.

لتغيير لون خلفية الشاشة في PyCharm، يمكنك اتباع الخطوات التالية:



1 اختيار الامر settings من قائمة file.

2 تحديد الاختيار Appearance من القائمة الجانبية ل appearance and behavior من نافذة Settings.

3 يتم اختيار ال theme المرغوب.

4 تطبيق التغييرات الضغط Apply.



أوراق العمل: مدخل إلى البرمجة.

مثال 1:

المشكلة : عمل برنامج بسيط يحسب ناتج القسمة لرقمين مع التأكيد على الا يكون الرقم المقسوم عليه يساوي صفر.

لشرح هذا المثال لأول مرة للمتعلمين، من المهم تقسيم الشرح إلى خطوات واضحة وتقديم مفاهيم أساسية

1. مقدمة عن العمليات الحسابية
2. مفهوم المدخلات والمخرجات وذلك أن أي برنامج يتطلب مدخلات من المستخدم (الأرقام ونوع العملية) ويعطي مخرجات (النتائج).

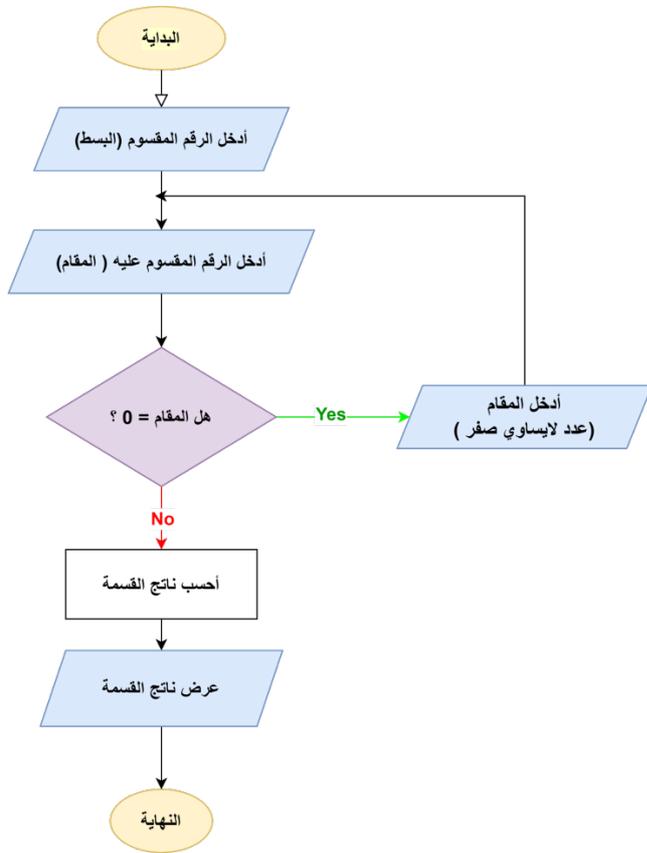
شرح خطوات الخوارزمية

1. ادخال الرقم الأول (البسط) من قبل المستخدم.
2. ادخال الرقم الثاني (المقام) من قبل المستخدم.
3. اختبار اذا ما كان المقام = صفر، يعرض البرنامج رسالة تنبيه .
(لا يجوز القسمة على صفر , أدخل الرقم الصحيح) .
4. اختبار اذا ما كان المقام لا يساوي صفر يجري عملية القسمة.
5. عرض النتيجة على الشاشة.

أوراق العمل: مدخل إلى البرمجة.

شرح خريطة التدفق (Flowchart)

تفسير الخوارزمية واسقاطها في خريطة التدفق على السبورة أو ورق موزع على مجاميع المتعلمين أو استخدام شرائح لعرضها لشرح كل جزء حسب الشكل المرسوم



البداية (Start)	←	بداية البرنامج.
ادخال المدخلات	←	- ادخال الرقم الأول (البسط) من قبل المستخدم. - ادخال الرقم الثاني (المقام) من قبل المستخدم.
اتخاذ قرار	←	التحقق من المقسوم عليه
	↙	في حال المقام = صفر عرض رسالة تنبيه (أدخل المقام عدد لا يساوي صفر)
	↙	في حالة لا يساوي صفر يجري عملية القسمة
المخرجات	←	عرض ناتج القسمة على الشاشة.
النهاية (End)	←	نهاية البرنامج

ترجمة الخريطة وتحويلها الى تعليمات برمجية يتبعها تشغيل واختبار البرنامج

أوراق العمل: مدخل إلى البرمجة

مثال (2) : المشكلة:

كتابة برنامج بسيط لطباعة جملة "Hello, Kuwait!" على الشاشة.

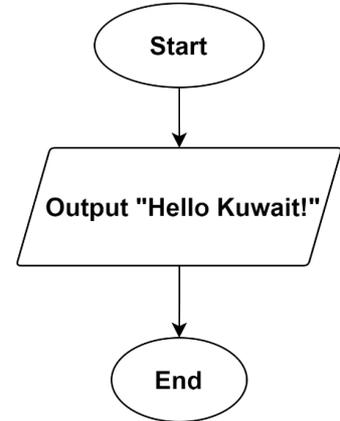
الخوارزمية:

استخدم الدالة `print ()` لطباعة النص "Hello, Kuwait!" على الشاشة.

التعليمات البرمجية

```
input
1 # طباعة الجملة على الشاشة
2 print("Hello, Kuwait!")
```

خريطة التدفق



Out put

```
C:\Users\HP\PycharmProjects\pythonProject
Hello, Kuwait!
```

مثال (3) : المشكلة:

كتابة برنامج بسيط لحساب مساحة مستطيل بمعلومية طوله وعرضه.

الخوارزمية:

1. استقبل قيمة الطول من المستخدم (بالمتر).
2. استقبل قيمة العرض من المستخدم (بالمتر).
3. احسب المساحة باستخدام المعادلة : المساحة=الطول×العرض.
4. اطبع قيمة المساحة على الشاشة.

أوراق العمل: مدخل إلى البرمجة.

تابع مثال (3) : المشكلة:

التعليمات البرمجية

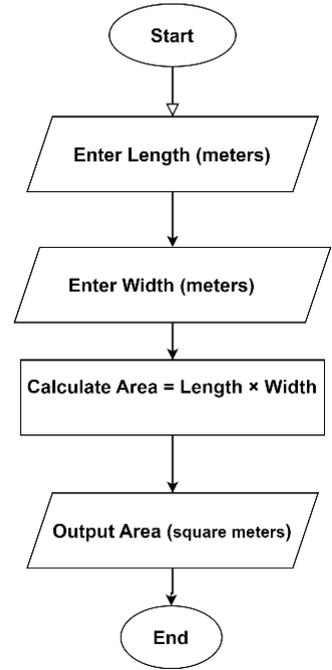
input

```
1 # استقبال القيم من المستخدم
2 length = float(input("Enter the length in meters: "))
3 width = float(input("Enter the width in meters: "))
4
5 # حساب المساحة باستخدام المعادلة
6 area = length * width
7
8 # عرض المساحة
9 print("Area = ", area, "m^2")
```

Out put

```
C:\Users\HP\PycharmProjects\pythonProject
Enter the length in meters: 12
Enter the width in meters: 9
Area = 108.0 m^2
```

الخوارزمية



حل ورقة عمل (1-1)

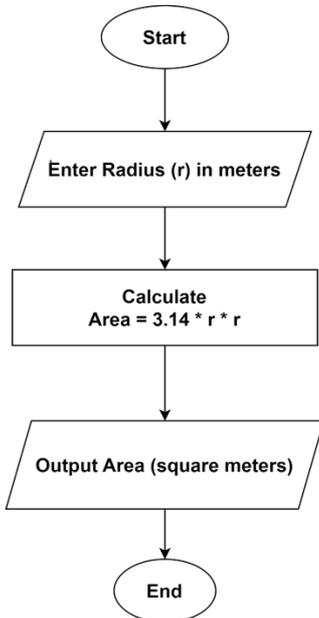
المشكلة:

من خلال دراستك لقوانين مساحة الأشكال الهندسية، احسب مساحة الدائرة بمعلومية نصف القطر.

الخوارزمية

1. استقبال من المستخدم نصف القطر (r).
2. احسب مساحة الدائرة باستخدام المعادلة πr^2 حيث $\pi = 3.14$
3. أعرض المساحة على الشاشة.

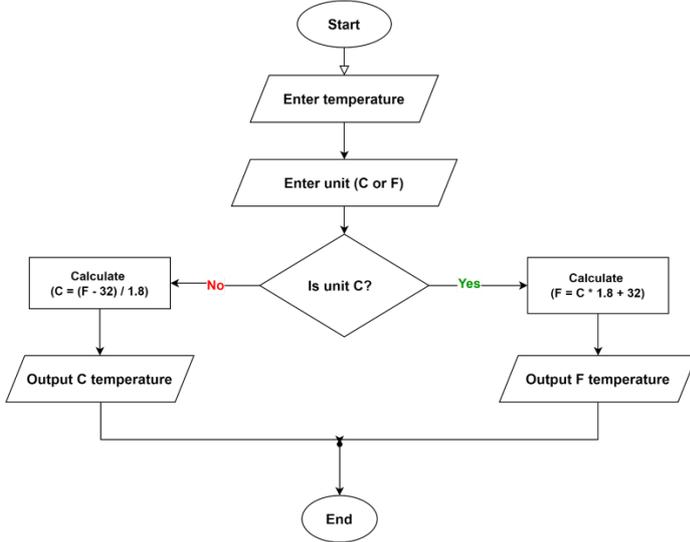
خريطة التدفق



أوراق العمل: مدخل إلى البرمجة.

حل ورقة عمل (2-1)

خريطة التدفق



المشكلة:

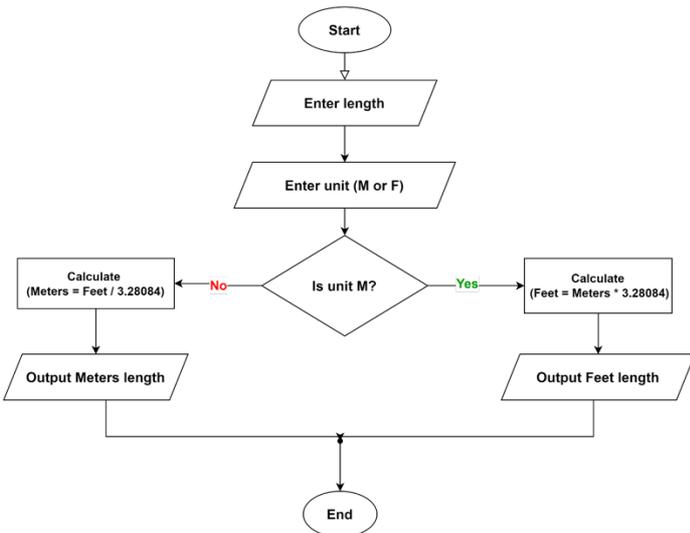
التحويل ما بين وحدات القياس المختلفة لدرجة الحرارة.

الخوارزمية:

1. استقبل من المستخدم درجة الحرارة الطقس.
2. استقبل من المستخدم نوع وحدة القياس (سيليزي أو فهرنهايت).
3. استخدم المعادلة التالية للتحويل درجة الحرارة من فهرنهايت الى سيليزية $C = (F - 32) \div 1.8$
4. استخدم المعادلة التالية للتحويل درجة الحرارة من سيليزية الى فهرنهايت $F = (C \times 1.8) + 32$
5. اعرض درجة الحرارة بعد التحويل لوحدة القياس المطلوبة.

مقترح أمثلة أخرى:

خريطة التدفق



المشكلة:

التحويل ما بين وحدات القياس المختلفة للأطوال

الخوارزمية:

1. تحويل الأطوال بين الأمتار والقدم.
2. استقبل من المستخدم الطول.
3. استقبل من المستخدم نوع الطول المدخل (أمتار أم قدم).
4. استخدم المعادلة إذا كان الطول بالأمتار: $Feet = Meters \times 3.28084$
5. استخدم المعادلة إذا كان الطول بالقدم: $Meters = Feet \div 3.28084$
6. اعرض الأطوال بعد التحويل لوحدة القياس المطلوبة.



المتغيرات Variables

البرمجة باستخدام برنامج بايثون تتيح لنا القدرة على إنشاء برامج متنوعة باستخدام تعليمات واضحة وبسيطة، في بداية تعلم البرمجة من المهم التعرف على بعض المفاهيم الأساسية مثل المتغيرات وأنواعها، وكيفية تخصيص القيم لها، حيث تعد المتغيرات هي أماكن في الذاكرة يمكن تخزين البيانات فيها، ويمكن أن تكون هذه البيانات من أنواع مختلفة مثل الأعداد الصحيحة `integer`، الأعداد العشرية `float`، النصوص `string`، والقيم المنطقية `Boolean` أيضاً أهمية تسمية المتغيرات بطريقة صحيحة وفقاً لقواعد التسمية في بايثون.

سنتعلم في هذه الوحدة كيفية استخدام دالة `print()` لعرض المعلومات على الشاشة، ودالة `input()` لاستقبال المدخلات من المستخدم، سنتعرف أيضاً على رموز العمليات الحسابية في بايثون وأولويات تنفيذها وأخيراً، سنتعلم كيفية تحويل البيانات بين الأنواع المختلفة باستخدام الدوال المناسبة مثل `int()`، `float()`، و `str()` هذه المفاهيم الأساسية تشكل الأساس لكتابة برامج فعالة ومفيدة في بايثون.

خطة الدروس: المتغيرات Variables.



python3



حصة دراسية

- المتغيرات: الإعلان عن متغير.
- تحديد أنواع المتغيرات.
- قواعد تسمية المتغيرات.
- استخدام المتغيرات.
- استخدام دالة `Print()`.
- استخدام دالة `Input()`.
- التحويل بين أنواع البيانات.





نتائج التعلم: المتغيرات.

1	التعرف على مفهوم المتغير.
2	التعرف على أنواع المتغيرات.
3	تخصيص قيمة لمتغير.
4	تسمية متغير بطريقة صحيحة مراعيًا قواعد التسمية.
5	التعرف على وظيفة دالة <code>print()</code> .
6	التعرف على رموز العمليات الحسابية في بايثون.
7	تمييز أولويات تنفيذ العمليات الحسابية.
8	التعرف على وظيفة دالة <code>input()</code> .
9	التحويل بين أنواع البيانات باستخدام الدوال.

مصطلحات حاسوبية: المتغيرات Variables.



الإعلان عن المتغيرات (Variable Declaration)



متغير: (Variable)



بيانات نصية (string)



دالة: (Function)



بيانات عددية عشرية (Float)



بيانات عددية صحيحة (integer)



دالة الإخراج - طباعة (print)



بيانات منطقية (Boolean)



دالة نوع المتغير (Type)



دالة الإدخال (Input)

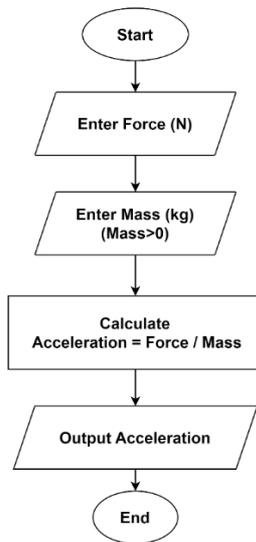


أوراق العمل: المتغيرات Variables.

ورقة عمل
(I-I)

المشكلة: حساب العجلة بمعلومية محصلة القوى والكتلة، وفقاً لقانون نيوتن الثاني.
الشروط: - القوة (Force) يجب أن تكون بالنيوتن.
- الكتلة (Mass) يجب أن تكون بالكيلو جرام وأكبر من الصفر.

خريطة التدفق:



الخوارزمية:

1. استقبل من المستخدم القوة المؤثرة (Force) بالنيوتن.
2. استقبل من المستخدم الكتلة (Mass) بالكيلو جرام (ملاحظة للمستخدم أن تكون أكبر من الصفر).
3. احسب العجلة (Acceleration) باستخدام القانون: العجلة = محصلة القوى / الكتلة.
4. اطبع قيمة العجلة على الشاشة.

التعليمات البرمجية

Input

```
1 # استقبال القوة والكتلة من المستخدم
2 var_force = float(input("Enter the force in Newtens: "))
3 # التأكد من أن الكتلة أكبر من الصفر
4 var_mass = float(input("Enter the mass in kg (must be greater than zero): "))
5 # حساب العجلة باستخدام القانون
6 Acceleration = var_force / var_mass
7 # عرض العجلة
8 print("The Acceleration is:", Acceleration, "m/s²")
9
```

Out put

```
"C:\Users\HP\PycharmProjects\pythonProject 10-1\.venv\
Enter the force in Newtens: 45
Enter the mass in kg (must be greater than zero): 17
The Acceleration is: 2.6470588235294117 m/s²

Process finished with exit code 0
```



أوراق العمل: المتغيرات Variables.

ورقة عمل

المشكلة: حساب الشغل المبذول بمعلومية القوى المؤثرة والإزاحة في اتجاهها.

المدخلات: القوة Force المؤثرة بوحدة (بالنيوتن).

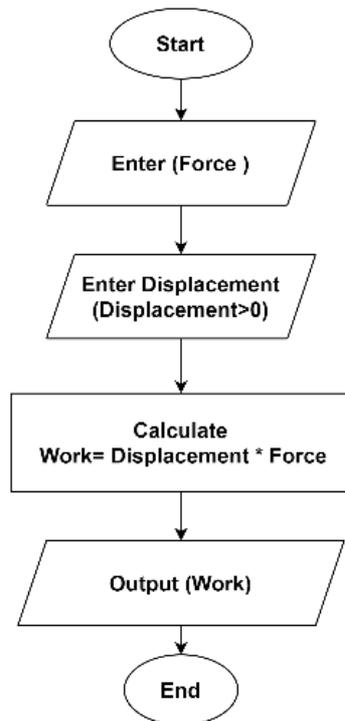
(٢-١)

الإزاحة في Displacement بوحدة (بالمتر) (ملاحظة للمستخدم أن تكون أكبر من الصفر).

المخرجات: الشغل المبذول Work بوحدة (بالجول).

خريطة التدفق:

الخوارزمية:



1. استقبل من المستخدم القوة المؤثرة Force بوحدة (بالنيوتن).
2. استقبل من المستخدم الإزاحة Displacement بوحدة (بالمتر). (أكبر من الصفر).
3. يحسب البرنامج الشغل Work بوحدة (الجول) باستخدام المعادلة $W = F \times D$.
4. يطبع البرنامج الشغل على الشاشة.

أوراق العمل: المتغيرات Variables.

تابع حل ورقة عمل (1-2)

التعليمات البرمجية

Input

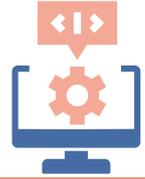
```
1 # استقبال القوة والإزاحة من المستخدم
2 var_force = float(input("Enter the force in Newtens: "))
3 # التأكيد من أن الإزاحة أكبر من الصفر
4 var_Displacement = float(input("Enter the Displacement in kg (must be greater than zero): "))
5 # حساب الشغل باستخدام القانون
6 var_Work = var_force * var_Displacement
7 # عرض الشغل
8 print("The Work is:", var_Work, "Joules")
```

Out put

```
"C:\Users\HP\PycharmProjects\pythonProject 10-1\.venv\Scripts
Enter the force in Newtens: 65
Enter the Displacement in kg (must be greater than zero): 14
The Work is: 910.0 Joules
Process finished with exit code 0
```

ملاحظة:

1. تم استخدام دالة float () تحويل النص إلى عدد عشري.

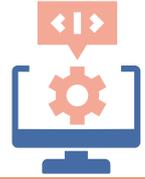


الشروط Condition

تعتبر عمليات المقارنة والدوال الشرطية في لغة بايثون، أدوات أساسية لاتخاذ القرارات. عمليات المقارنة مثل '=', '<', '>', '<=', '>=' تسمح لنا بمقارنة القيم، بينما تتيح لنا عمليات 'and'-'or' دمج الشروط المنطقية. باستخدام الدوال الشرطية 'if', 'elif', 'else' يمكننا تنفيذ أكواد مختلفة بناءً على الشروط المحددة، مما يجعل البرامج أكثر تفاعلية وديناميكية. على سبيل المثال، يمكننا تحديد قيمة معينة واستخدام الشروط لطباعة رسائل مختلفة بناءً على هذه القيمة، مما يساعد في التحكم في تدفق البرنامج بمرونة وفعالية.

 python 4	حصة دراسية	• الشروط : يميز بين الإسناد والعمليات المنطقية.	
 python 5	حصة دراسية	• الشروط : يوظف العمليات الشرطية -elif- else -if لاتخاذ القرار.	





نتائج التعلم: الشروط Condition.

التعرف على رموز عمليات المقارنة المنطقية.	1
التعرف على عمليات المقارنة المنطقية (and, or) .	2
التعرف على الدوال الشرطية (else,if,elif) لاتخاذ القرارات.	3

مصطلحات حاسوبية الشروط Condition:



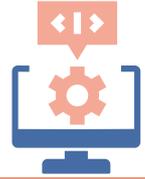
التعبير (Expression)



شروط: (Condition)



العمليات الشرطية (if , elif , else)



أوراق العمل: الشروط. Conditions

شرح مثال عملي شامل لتوظيف عمليات المقارنة بين متغيرين x , y

التعليمات البرمجية

input	Out put
<pre>1 x = 10 2 y = 5 3 print(x == y) # False 4 print(x != y) # True 5 print(x > y) # True 6 print(x < y) # False 7 print(x >= y) # True 8 print(x <= y) # False</pre>	<pre>C:\Users\HP\PycharmProjects\ False True True False True False</pre>

ما يجب على المعلم التأكيد عليه عند تعليم عمليات المقارنة المنطقية

أولاً:	التعرف على رموز عمليات المقارنة المنطقية:
←	التأكد من فهم المتعلمين أن هذا الرمز يستخدم للتحقق مما إذا كانت القيمتين متساويتين. على سبيل المثال، $5 == 5$ الناتج True، بينما $5 == 6$ الناتج False.
←	المساواة: ($==$)
←	يجب توضيح أن هذا الرمز يتحقق مما إذا كانت القيمتين غير متساويتين. على سبيل المثال، $5 != 6$ الناتج True، بينما $5 != 5$ الناتج False.
←	عدم المساواة ($!=$)
←	ينبغي التأكيد على أن هذا الرمز يستخدم للتحقق مما إذا كانت القيمة الأولى أكبر من الثانية على سبيل المثال، $7 > 5$ الناتج True، بينما $5 > 7$ الناتج False.
←	أكبر من ($>$)
←	يجب توضيح أن هذا الرمز يتحقق مما إذا كانت القيمة الأولى أقل من الثانية. على سبيل المثال، $3 < 5$ الناتج True، بينما $5 < 3$ الناتج False.
←	أقل من ($<$)
←	يساوي ($>=$): التأكيد على أن هذا الرمز يتحقق مما إذا كانت القيمة الأولى أكبر من أو تساوي القيمة الثانية. على سبيل المثال، $5 >= 5$ الناتج True، بينما $4 >= 5$ الناتج False.
←	أكبر من أو يساوي: ($>=$)
←	ينبغي توضيح أن هذا الرمز يتحقق مما إذا كانت القيمة الأولى أقل من أو تساوي القيمة الثانية. على سبيل المثال، $5 <= 5$ الناتج True، بينما $6 <= 5$ الناتج False.
←	أقل من أو يساوي: ($<=$)



ما يجب على المعلم التأكيد عليه عند تعليم عمليات المقارنة المنطقية.

التعرف على عمليات المقارنة المنطقية (and, or)		ثانياً:
عمليات المقارنة المنطقية تستخدم للجمع بين شرطين أو أكثر هذه العمليات تشمل:		
And: الناتج True إذا كانت كل الشروط صحيحة.		←
Or: الناتج True إذا كان على الأقل واحد من الشروط صحيحاً.		←
1	<code>a = 5</code>	
2	<code>b = 6</code>	
3	<code>c = 7</code>	
4	<code>print(a < b and b < c) # True</code>	
5	<code>print(a > b or b < c) # True</code>	
6	<code>print(a > b and b < c) # False</code>	
7	<code>print(a > b or b > c) # False</code>	

أمثلة لشرح الدوال الشرطية:

في هذا المثال، سنطلب من المستخدم إدخال قيمة لـ `x`، ثم نستخدم الدوال الشرطية `if`, `elif`, `else` لتحديد الرسالة المناسبة بناءً على القيمة المدخلة.

مثال توضيحي خارجي

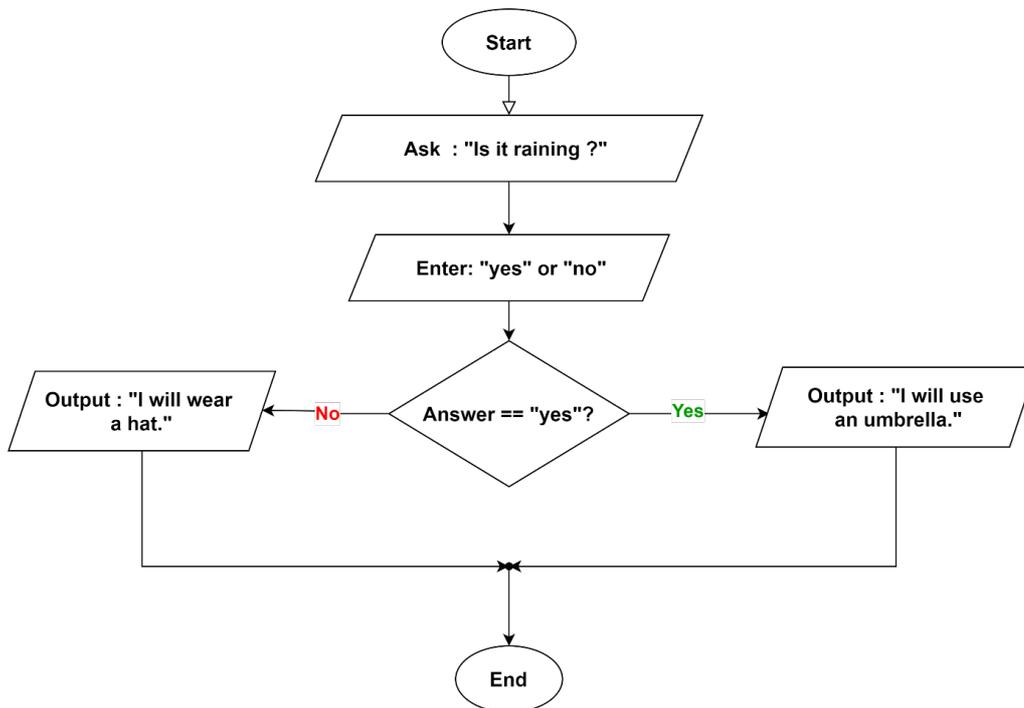
المشكلة:

كتابة برنامج بسيط: إذا كانت السماء تمطر عندها سأستخدم المظلة والا سوف ارتدي قبعه.

كتابة الخوارزمية:

1. اسأل (هل السماء تمطر ؟).
2. اقرأ الإجابة.
3. إذا كانت الإجابة نعم اطبع (سأستخدم مظلة).
4. إذا كانت الإجابة لا اطبع (سأرتدي قبعه).

خريطة التدفق



التعليمات البرمجية

Input

```

1  برنامج يحدد استخدام المظلة أو القبعة بناءً على حالة الطقس #
2  print(" is it raining ? please answer yes or no ?")
3  weather = input()
4  if weather == "yes":
5      print("I will use an umbrella.")
6  else:
7      print("I will wear a hat.")
8

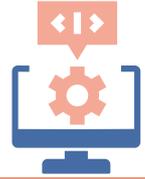
```

Out put

```

C:\Users\HP\PycharmProjects\pythonProject4\
is it raining ? please answer yes or no ?
no
I will wear a hat.
Process finished with exit code 0

```



حل مثال تطبيقي (11-1)

استكمل البرنامج الذي يظهر العدد الأقصى للإلكترونات حسب رمز مستوى الطاقة بناء على خريطة التدفق الموضحة في كتاب المتعلم .

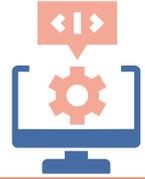
التعليمات البرمجية

Input

```
1 energylevel=input("أدخل رمز مستوى الطاقة:")
2 if energylevel=="K":
3     print(2)
4 elif energylevel == "L":
5     print(8)
6 elif energylevel == "M":
7     print(18)
8 elif energylevel == "N":
9     print(32)
10 else:
11     print("المدخلات خطأ")
```

Out put

```
C:\Users\HP\PycharmProjects\pythonProject4
أدخل رمز مستوى الطاقة:N
32
Process finished with exit code 0
```



حل أوراق العمل: الشروط Conditions.

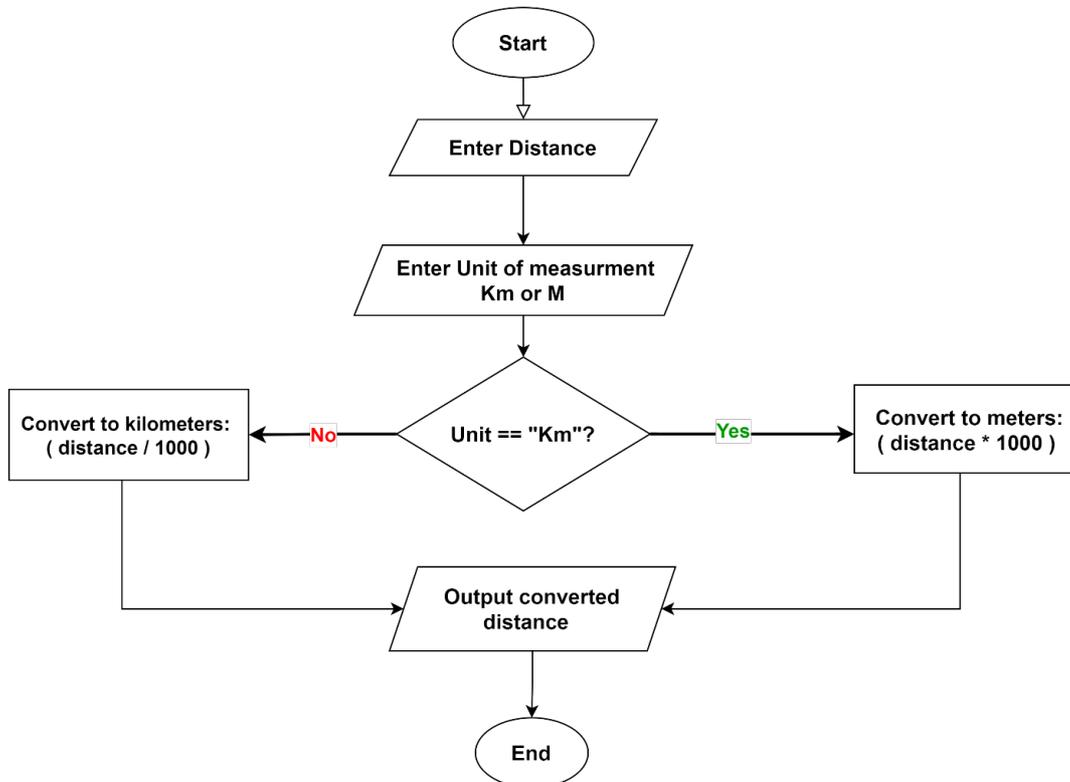
ورقة عمل

(1-1)

المشكلة: برنامج تحويل وحدة قياس المسافة من الكيلومتر إلى المتر و العكس .
الخوارزمية:

1. استقبل من المستخدم المسافة Distance .
2. استقبل من المستخدم وحدة القياس Unit .
3. استخدم المعادلة المناسبة وفقا لوحدة القياس المدخلة:
 - المسافة بالكيلومتر = المسافة بالمتر * 1000 (للتحويل من متر إلى كيلومتر) .
 - المسافة بالمتر = المسافة بالكيلومتر / 1000 (للتحويل من كيلومتر إلى متر) .
4. اطبع المسافة بوحدة القياس المطلوبة.

خريطة التدفق



تابع حل أوراق العمل: الشروط Conditions.

التعليمات البرمجية:

Input

```
1  # استقبال المسافة من المستخدم
2  Distance = float(input("Enter Distance: "))
3  # استقبال وحدة القياس من المستخدم
4  Distance_Unit = input("Enter the units of measurement(km-kilometers, m-metres):")
5  # تحويل وحدة القياس المدخلة إلى وحدة القياس المطلوبة
6  if Distance_Unit == "km" :
7      Distance_m = Distance * 1000
8      print(Distance, "km =", Distance_m, "meters.")
9  else:
10     Distance_km = Distance / 1000
11     print(Distance, "m=", Distance_km, "kilometers")
12
```

Out put

```
C:\Users\HP\PycharmProjects\pythonProject4\.venv\Scripts\python.exe
Enter Distance: 56
Enter the units of measurement(km-kilometers, m-metres):m
56.0 m= 0.056 kilometers
Process finished with exit code 0
```

تابع حل أوراق العمل: الشروط Conditions.

ورقة عمل

المشكلة: التعرف على حالة الأعداد المدخلة.

الخوارزمية:

(2-1)

1. استقبال من المستخدم العدد `number`.

2. استخدم الشروط `Condition`:

- إذا كان العدد المدخل أكبر من صفر، يطبع العدد موجب.
- إذا كان العدد المدخل أصغر من صفر، يطبع العدد سالب.
- إذا كان العدد المدخل يساوي صفر، يطبع (صفر).

التعليمات البرمجية

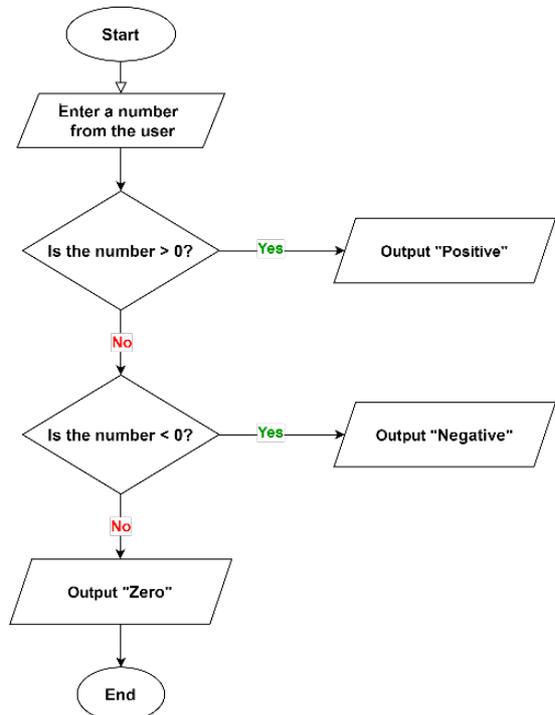
خريطة التدفق

Input

```
1 # استقبال رقم من المستخدم
2 number = float(input("Enter a number: "))
3 # التحقق من حالة العدد المدخل
4 if number > 0:
5     print("The number is positive")
6 elif number < 0:
7     print("The number is negative")
8 else:
9     print("The number is zero")
```

Out put

```
"C:\Users\HP\PycharmProjects\pythc
Enter a number: -65
The number is negative
Process finished with exit code 0
```



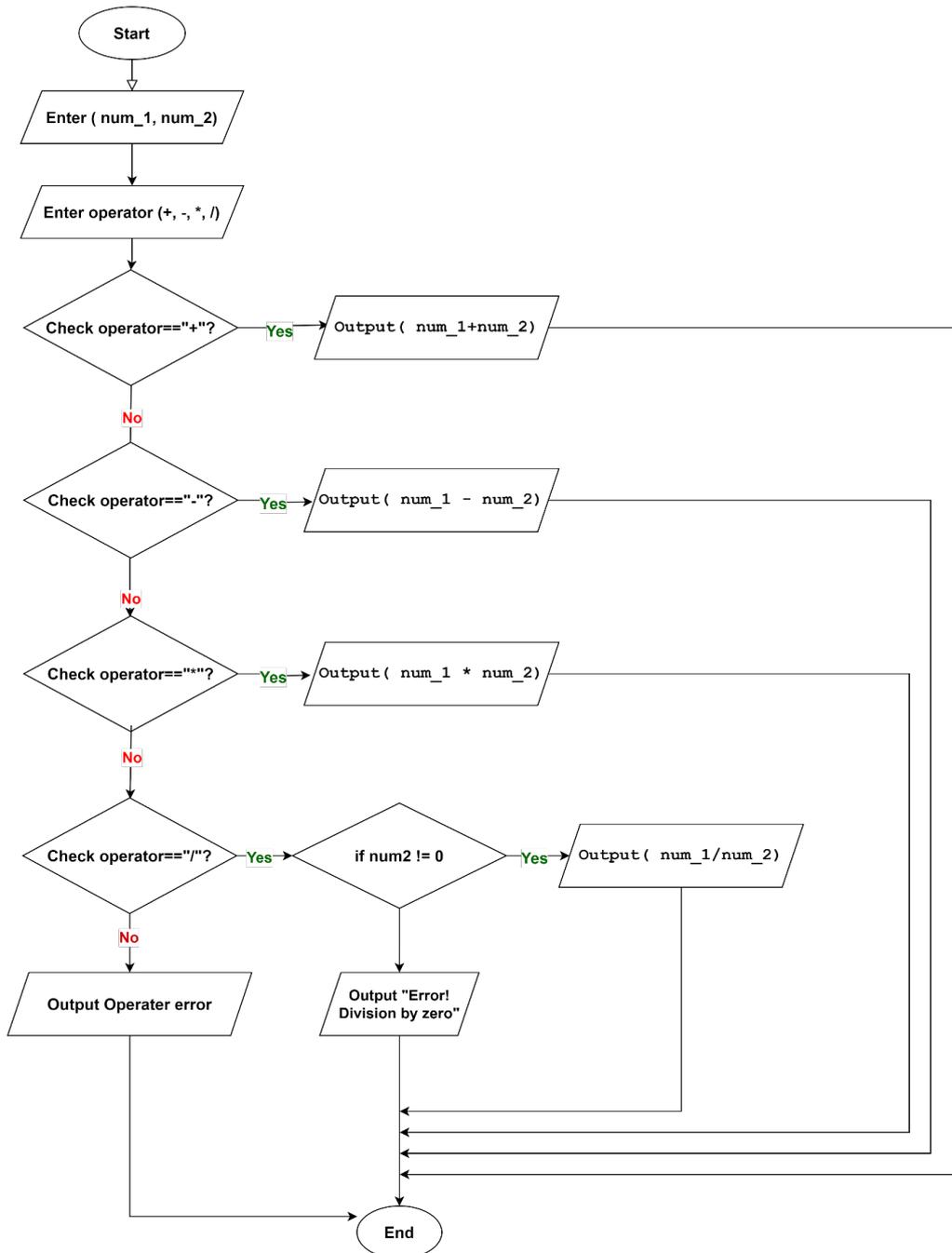
تابع حل أوراق العمل: الشروط Conditions.

ادرس خريطة التدفق التالية ، والتي صممت لعمل آلة حاسبة بسيطة للعمليات الأساسية ، ثم أضف ما يلزم من التعليمات لي يعمل البرنامج بشكل صحيح طبقا لخريطة التدفق .

ورقة عمل

(3-1)

خريطة التدفق



تابع حل أوراق العمل: الشروط Conditions.

تابع حل ورقة عمل (3-1)

التعليقات البرمجية

Input

```
1 # استقبال الرقم الأول من المستخدم
2 num_1 = float(input("Enter num1: "))
3 # استقبال الرقم الثاني من المستخدم
4 num_2 = float(input("Enter num2:"))
5 # استقبال العملية المطلوبة من المستخدم
6 operator = input("choose operator (+, -, *, /): ")
7 # التحقق من نوع العملية وتنفيذ العملية الحسابية
8 if operator == "+":
9     print("result:", num_1 + num_2)
10 elif operator == "-":
11     print("result:", num_1 - num_2)
12 elif operator == "*":
13     print("result:", num_1 * num_2)
14 elif operator == "/":
15     if num_2 != 0:
16         print("result:", num_1 / num_2)
17     else:
18         print("Error: Division by zero")
19 else:
20     print("Operator Error")
```

Out put

```
"C:\Users\HP\PycharmProjects\pythonF
Enter num1: 9
Enter num2:7
choose operator (+, -, *, /): *
result: 63.0
Process finished with exit code 0
```

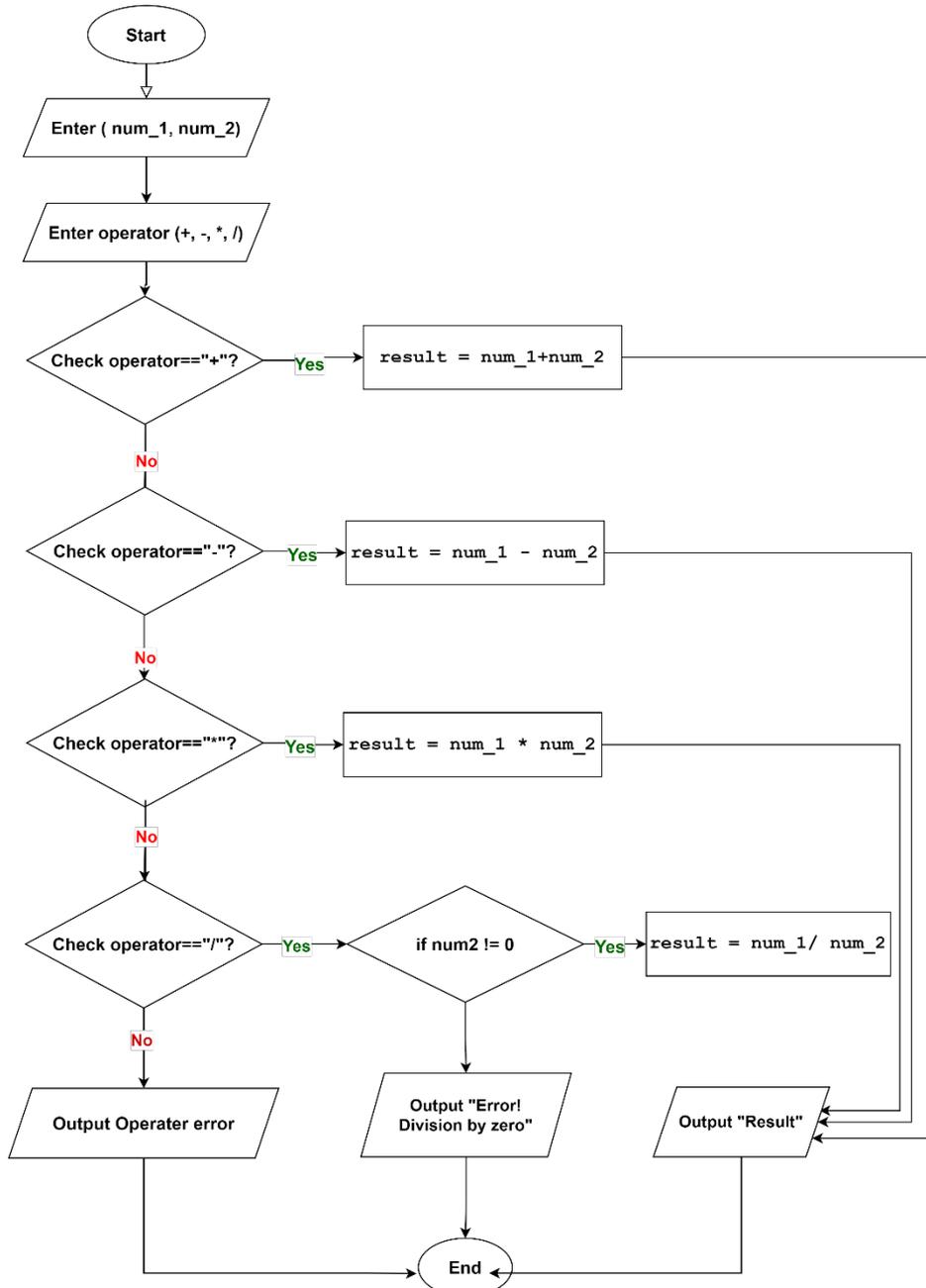
تابع حل أوراق العمل: الشروط Conditions.

- إضافة جزء للمعالجة (اجراء العملية الحسابية).
- تليها خطوة الطباعة لنتيجة العملية.
- في المثال التالي إضافة متغير result يتم تخزين ناتج العملية الحسابية.

حل آخر

ورقة عمل (3-1)

خريطة التدفق



تابع حل أوراق العمل: الشروط Conditions.

التعليمات البرمجية

Input

```
1 # استقبال الرقم الأول من المستخدم
2 num_1 = float(input("Enter num1: "))
3 # استقبال الرقم الثاني من المستخدم
4 num_2 = float(input("Enter num2:"))
5 # استقبال العملية المطلوبة من المستخدم
6 operator = input("choose operator (+, -, *, /): ")
7 # التحقق من نوع العملية وتنفيذ العملية الحسابية
8 if operator == "+":
9     result = num_1 + num_2
10    print("result:", result)
11 elif operator == "-":
12    result = num_1 - num_2
13    print("result:", result)
14 elif operator == "*":
15    result = num_1 * num_2
16    print("result:", result)
17 elif operator == "/":
18    if num_2 != 0:
19        result = num_1 / num_2
20        print("result:", result)
21    else:
22        print("Error: Division by zero")
23 else:
24    print("Operator Error")
```

Out put

```
C:\Users\HP\PycharmProjects\pythonProject
Enter num1: 9
Enter num2:8
choose operator (+, -, *, /): *
result: 72.0

Process finished with exit code 0
```



التكرار Loop

تُعد الحلقات التكرارية أدوات قوية وفعّالة في البرمجة، وتتيح لك تكرار تنفيذ مجموعة من التعليمات بسهولة، باستخدام `for` ، `while` يمكنك تنفيذ التكرار على مجموعة متنوعة من التعليمات واستخدام الحلقات المتداخلة تزيد من تمكين عمليات متعددة المستويات، ودوال التحكم مثل `break` - `continue` تضيف مزيدًا من التحكم والديناميكية لكيفية تنفيذ التكرار.

خطة الدروس: التكرار Loop.

 python 6	حصة دراسية	الحلقات التكرارية (while loop) - (while ... else).	
 python 7	حصة دراسية	الحلقة التكرارية (for loop).	
 python 8	حصة دراسية	الحلقات المتداخلة (Nested Loops).	
 python 9	حصة دراسية	تابع: الحلقات المتداخلة (Nested Loops) • دالة (break). • دالة (continue).	



امسح الرمز



نتائج التعلم: التكرار Loop.

1	التعرف على مفهوم التكرار في البرمجة.
2	استخدام الدالة (while).
3	استخدام الدالة (for).
4	تكرار تعليمات برمجية عددا محددا من المرات باستخدام دالة (for).
5	استخدام الدالة (While) لتكرار نص برمجي طالما تحقق شرط معين.
6	استخدام الدالة while true لتكرار تنفيذ تعليمات برمجية الى مالا نهاية.
7	التعرف على استخدامات الدالة (Range).
8	التعرف على دالة (Break).
9	التعرف على دالة (Continue).

مصطلحات حاسوبية: التكرار Loop.



دالة (While)



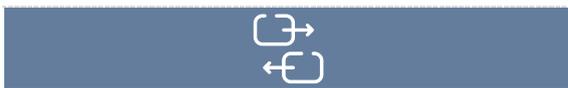
حلقة تكرارية (Loop)



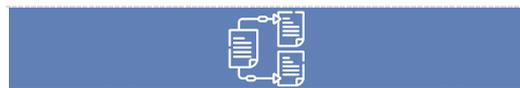
كائن: (Object)



دالة (for)



التكرارات المتداخلة (nested loops)



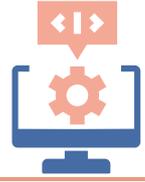
التكرار اللانهائي (loop infinity)



دالة (Continue)



دالة (Break)



حل أمثلة: التكرار Loop.

مثال (1-1)

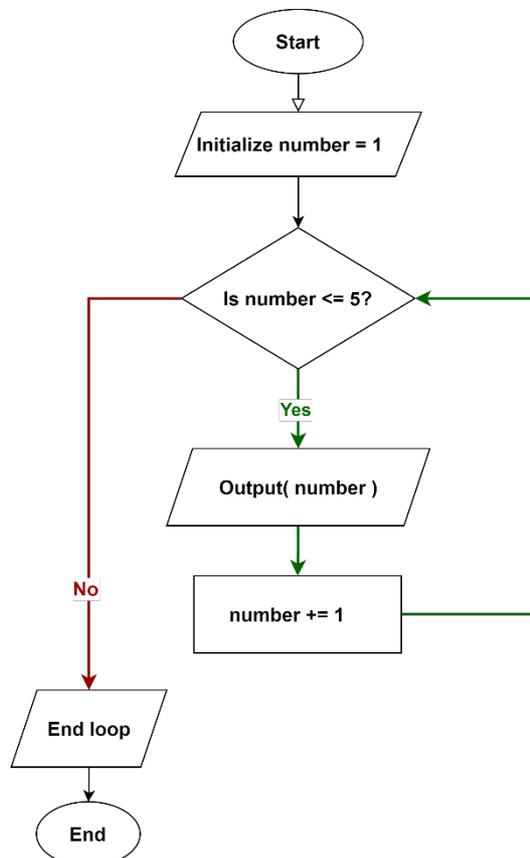
المشكلة: طباعة أعداد من 1-5 وتطبيق نقاط مهمة.
التأكيد للمتعلمين عند استخدام الدالة while لا بد من شرط ينهي التكرار لتجنب الحلقات اللانهائية.

كتابة الخوارزمية: التأكيد على شرح المكونات الرئيسية الثلاثة في الحلقات (Loops).

من خلال المثال:

1. **تحديد المتغير:** يتم تحديد قيمة ابتدائية للمتغير قبل بدء الحلقة المتمثل في التعليمة البرمجية (number = 1)
2. **شرط التوقف:** الشرط المستخدم في الحلقة هو (number <= 5) ، مما يعني أن الحلقة ستتوقف عندما تصبح قيمة number أكبر من 5 نهاية التكرار .
3. **معادلة المتغير:** داخل الحلقة (number += 1).

خريطة التدفق



تابع حل مثال (1-1).

التعليمات البرمجية

```
Input
1 # تحديد المتغير
2 number = 1
3 # while مازال العدد أقل أو يساوي 5 شرط التحقق لتنفيذ الحلقة
4 while number <= 5:
5     print("Number is:", number)
6     # معادلة المتغير لضمان الوصول إلى شرط التوقف
7     number += 1
8 print("End Loop")

Out put
C:\Users\HP\PycharmProjects\pythonProject
Number is: 1
Number is: 2
Number is: 3
Number is: 4
Number is: 5
End Loop

Process finished with exit code 0
```

يطبق نفس المثال باستخدام الدالة For

1. يتم تحديد المتغير (`number`) ونطاق التكرار باستخدام `range`.
2. نستخدم دالة `range(start, stop)`.
3. نستخدم دالة `range(1, 6)` لإنشاء تسلسل من الأرقام من 1 إلى 5.

التأكيد على الدالة `range` وكيفية التعامل حسب القيمة المدرجة.

- `range(stop)`
- `range(start, stop)`
- `range(start, stop, step)`

التعليمات البرمجية

```
Input
1 # استخدام الحلقة for لتحديد المتغير ونطاق التكرار
2 for number in range(1, 6):
3     # طباعة الرقم
4     print("Number is :", number)
5     print("End Loop ")

Out put
C:\Users\HP\PycharmProjects\pythonProject
Number is: 1
Number is: 2
Number is: 3
Number is: 4
Number is: 5
End Loop

Process finished with exit code 0
```

حل أوراق العمل: التكرار Loop.

مثال (6-1)

استخدام الدالة while.

التأكيد على أن التكرار (loop) في البرمجة هو عملية تنفيذ مجموعة من التعليمات بشكل متكرر حتى يتحقق شرط معين.

التعليمات البرمجية

Input

```
1  # تحديد كلمة المرور الصحيحة
2  correctpassword = "1234"
3  #تهيئة المتغير password سلسلة فارغة يتم تخزين كلمة المرور المدخلة من قبل المستخدم
4  password = ""
5  #التحقق من مطابقة كلمة المرور
6  while (password != correctpassword):
7      # إذا كانت كلمة المرور غير صحيحة، اطلب من المستخدم الإدخال مرة أخرى
8      password = input("Enter password : ")
9      # إذا تمت المطابقة، اطبع رسالة النجاح
10 print("Password Correct.")
```

Out put

```
"C:\Users\HP\PycharmProjects\pytho
Enter password: 111
Enter password: 121
Enter password: 123
All attempts finish.
Process finished with exit code 0
```

تابع حل مثال (6-1): إراثي.

تطوير البرنامج يتم تحديد عدد محاولات الإدخال بثلاث مرات:

التعليمات البرمجية

Input

```
1 # تحديد كلمة المرور الصحيحة
2 correct_password = "1234"
3 # عدد المحاولات المسموح بها
4 attempts = 3
5 # تهيئة عدد المحاولات
6 attempt_count = 0
7 # التحقق من مطابقة كلمة المرور
8 while attempt_count < attempts:
9     # طلب إدخال كلمة المرور من المستخدم
10    password = input("Enter password: ")
11    # زيادة عداد المحاولات
12    attempt_count += 1
13    # التحقق من كلمة المرور
14    if password == correct_password:
15        print("Password Correct.")
16        break
17    else:
18        print("All attempts finish.")
```

Out put

```
C:\Users\HP\PycharmProjects\pythonProject4\.venv\Scr:
Enter passwords: 111
The password is incorrect. You have 2 attempts left.
Enter passwords: 21
The password is incorrect. You have 1 attempts left.
Enter passwords: 12
All attempts Finish, Please try again |

Process finished with exit code 0
```

تابع حل مثال (6-1) حل اخر باستخدام دالة for

التعليمات البرمجية

Out put

```
1  # الكلمة الصحيحة التي يجب على المستخدم إدخالها
2  correct_password= "1234"
3  # عدد المحاولات المسموح بها
4  for x in range(3):
5      password = input("Enter passwords: ")
6      if password == correct_password:
7          print("passwords correct")
8          break
9      else:
10         print("The password is incorrect.")
```



حل أوراق العمل: التكرار Loop.

ورقة عمل (1-1) المشكلة: تصميم برنامج يخمن المستخدم عددا صحيحا ثم يقارنه البرنامج مع العدد المعلن عنه مسبقا .

كتابة الخوارزمية:

1. الاعلان عن متغير يساوي العدد المطلوب تخمينه.
2. استقبال عدد صحيح من المستخدم.
3. مطابقة العدد المدخل مع العدد المحدد في الحالات التالية: -
 - إذا كان أكبر يطبع رسالة العدد أكبر ويستمر المستخدم في عملية الادخال.
 - إذا كان أصغر يطبع رسالة العدد أصغر ويستمر المستخدم في عملية الادخال.
 - طباعة رسالة العدد المدخل مطابق للعدد المعرف وينتهي البرنامج.

التعليمات البرمجية

Input

```
1 # الإعلان عن العدد المطلوب تخمينه
2 correct_number = 8
3 player_number = 0
4 # مطابقة العدد المدخل مع العدد المحدد
5 while player_number != correct_number:
6     # استقبال عدد صحيح من المستخدم
7     player_number = int(input("Guess !what the number in the mind : "))
8     if player_number > correct_number:
9         print("player_number is higher ")
10    elif player_number < correct_number:
11        print("player_number is lower.")
12    else:
13        print("Correct well done .")
```

Out put

```
"C:\Users\HP\PycharmProjects\pythonProject
Guess !what the number in the mind : 9
player_number is higher
Guess !what the number in the mind : 7
player_number is lower.
Guess !what the number in the mind : 8
Correct well done .

Process finished with exit code 0
```

تابع حل أوراق العمل: التكرار Loop.

ورقة عمل

المشكلة:

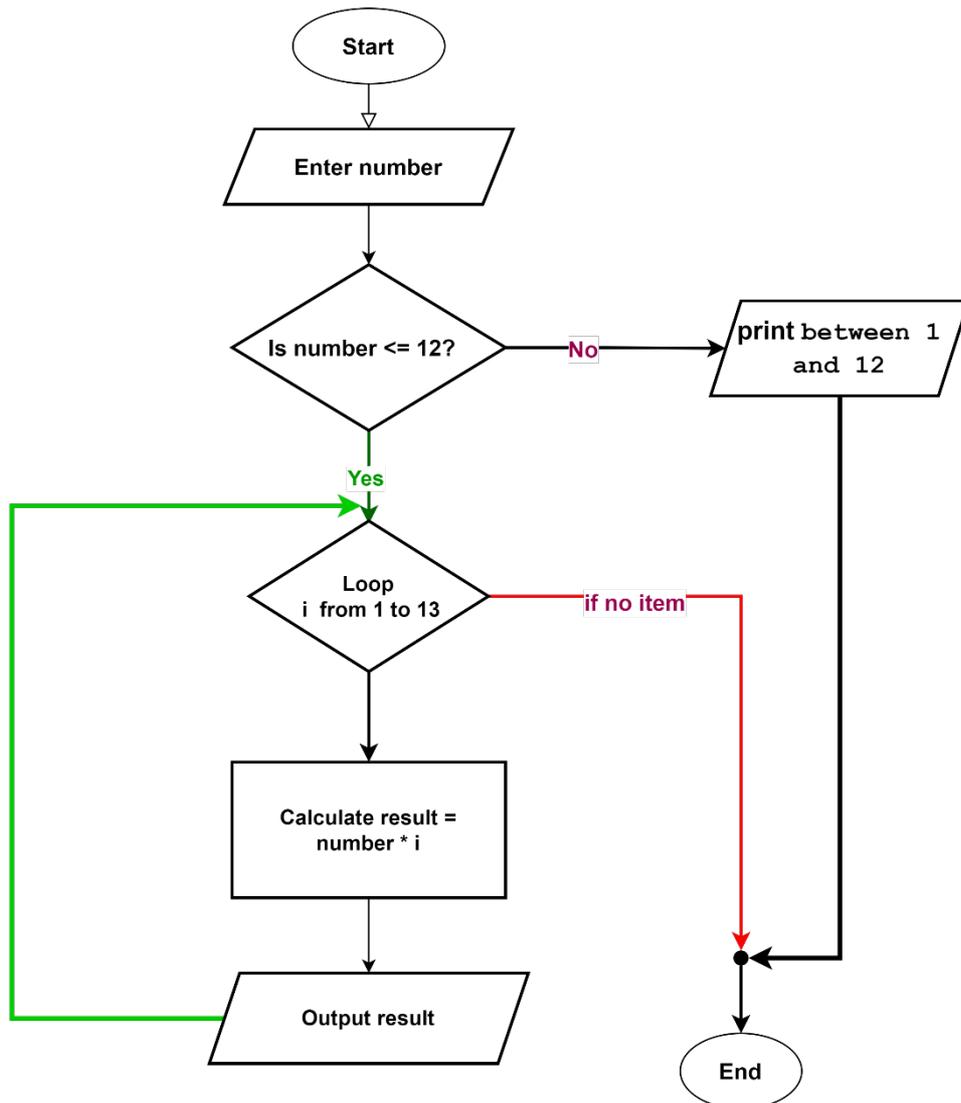
تصميم برنامج ضرب بسيط يظهر عمليات الضرب من 1 وحتى 12.

(2-1)

الخوارزمية:

1. استقبال من المستخدم العدد المطلوب طباعة جدول الضرب الخاص به.
2. إذا كان العدد أصغر من أو يساوي 12.
 - انشاء حلقة تكرارية لطباعة ضرب العدد المدخل في الأعداد من 1 إلى 12.
 - استخدام معادلة الناتج = العدد المدخل * قيمة متغير التكرار.
3. إذا كان العدد أكبر من 12، يطبع رسالة تنبيه. (ملاحظة التأكيد أنه سينتهي البرنامج)

خريطة التدفق



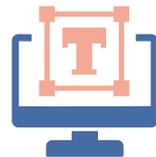
Input

```
1 # استقبال الرقم من المستخدم
2 number = int(input("Enter a number:"))
3 # التحقق من صحة الرقم
4 if number <= 12:
5     # إنشاء وطباعة جدول الضرب
6     for i in range(1, 13):
7         result = number * i
8         print(number, "x", i, "=", result)
9 else:
10    # عرض رسالة تنبيه
11    print("Please enter a number between 1 and 12 .")
```

Out put

```
"C:\Users\HP\PycharmProjects\pythonProjec
Enter a number:6
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60
6 x 11 = 66
6 x 12 = 72

Process finished with exit code 0
```



السلاسل النصية Strings.

السلاسل النصية في البرمجة تمثل تسلسلاً من الأحرف أو الرموز المتتالية داخل علامة اقتباس (سواء كانت فردية أو مزدوجة) في لغة البرمجة، هذه السلاسل تُستخدم لتمثيل النصوص والبيانات التي تكون من الأحرف مثل الأرقام والرموز الخاصة، تتميز بعدة خصائص تجعل التعامل معها مرناً وسهلاً، يمكن الوصول إلى أحرف محددة داخل السلسلة النصية باستخدام الفهرسة (Indexing) والقطع (Slicing) لاستخراج أجزاء محددة من النص أيضاً عمليات جمع السلاسل النصية (Concatenation) تسمح بدمج نصين أو أكثر وتكرار السلاسل النصية لأكثر من مرة، أيضاً مقارنة السلاسل النصية لمعرفة مدى تطابقها أو ترتيبها، توفر دوال مفيدة مثل `len()` لمعرفة طول السلسلة النصية، `capitalize()` لجعل الحرف الأول كبيراً، `upper()` لتحويل النص إلى أحرف كبيرة أو صغيرة، `find()` للبحث عن نص معين داخل النص، `replace()` لاستبدال جزء من النص إمكانية التنسيق للنصوص تستخدم `f-string`، بينما تسمح دالتى `float()` - `int()` بتحويل النصوص إلى أعداد صحيحة وعشرية و دالة `count()` تُستخدم لعد تكرار جزء معين من النص، و `center()` لمحاذاة النص في المنتصف، وغيرها من الدوال المفيدة.



خطة الدروس: السلاسل النصية.



py 10

حصتان دراسية

- تعريف السلاسل النصية وخصائصها
- عمليات السلاسل النصية
- الفهرسة indexing.
- القطع Slicing.
- جمع السلاسل النصية Concatenation.
- تكرار السلاسل النصية.
- المقارنة Comparison.



python 11

حصتان دراسية

- دوال السلاسل :
- طول السلسلة النصية Len().
- Capitalize ()
- Upper()
- Lower()
- Find()
- Replace()
- F-string
- int()
- Float()
- count()
- center()





نتائج التعلم: السلاسل النصية Strings.

1	التعرف على مفهوم السلاسل النصية.
2	انشاء سلسلة نصية باستخدام علامات اقتباس فردية أو مزدوجة " " .
3	التعرف على فهرسة السلسلة النصية.
4	التعرف على عملية القطع Slicing للحصول على جزء من السلسلة النصية.
5	جمع سلسلتين نصيتين للحصول على سلسلة نصية جديدة.
6	تكرار السلاسل النصية باستخدام عامل الضرب *.
7	مقارنة السلاسل النصية باستخدام معاملي المقارنة ==, != .
8	معرفة طول السلسلة النصية باستخدام دالة len().

مصطلحات حاسوبية: السلاسل النصية Strings.



الفهرسة Indexing



السلاسل النصية string



جمع السلاسل النصية Concatenation



القطع Slicing



دالة Capitalize



دالة Len



دالة Lower



دالة Upper

مصطلحات حاسوبية: السلاسل النصية Strings.



دالة Replace



دالة find



دالة Count



دالة Center

بعض النقاط الخاصة التي يجب على المعلم التأكيد عليها.

السلسلة النصية هي مجموعة من الحروف والكلمات التي يمكن استخدامها لتمثيل النصوص والبيانات في البرمجة

الفهرسة والوصول للحروف التأكيد أن الفهرسة تبدأ من صفر، مثال للحصول على الحرف الأول .

استخدام القطع (Slicing) لاستخراج جزء معين من السلسلة لتحليل النصوص ومعالجتها.

Concatenation يمكن دمج سلاسل نصية باستخدام عامل الجمع '+' .

يمكن تكرار سلسلة نصية باستخدام عامل الضرب '*', مما يساعد في إنشاء أنماط نصوص متكررة.

تأكيد أهمية عمليات المقارنة لمقارنة السلاسل النصية باستخدام عوامل المقارنة مثل '==', '!=', '>', '<', '>=', '<='.

يجب فهم كيفية تنسيق النص باستخدام دوال مثل upper()-lower()-capitalize() center() لتغيير حالة الأحرف في السلسلة، واستخدام دالة f-strings لتسهيل التنسيق.

يجب فهم كيفية استخدام دوال مثل find()-replace() لبحث عن نصوص محددة داخل السلسلة.

تحويل قيم الى أعداد صحيحة أو عشريه باستخدام الدالتين float()-int() .



حل أوراق العمل: السلاسل النصية Strings.

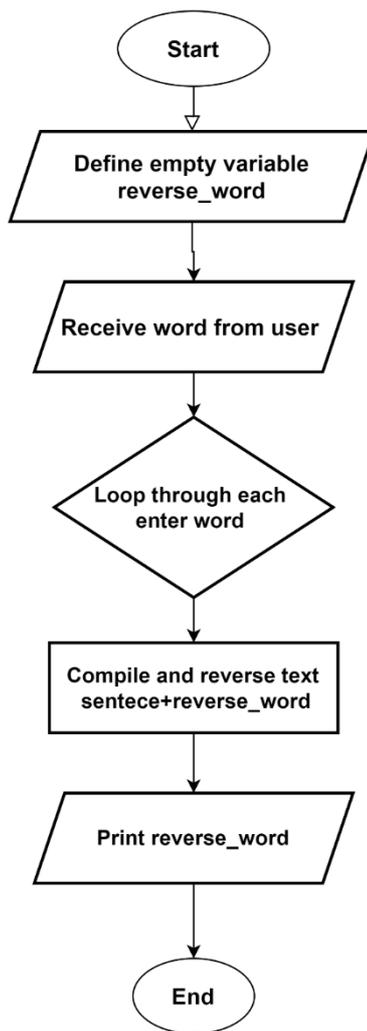
المشكلة:

ورقة عمل (1-1)

انشاء سلسلة نصية بحيث تبدأ من النهاية وحتى البداية.

خريطة التدفق

الخوارزمية:



1. عرف متغير بقيمة ابتدائية فارغة.
2. استقبل من المستخدم العبارة النصية المطلوب عكس أحرفها.
3. أنشئ حلقة تكرارية وفقا لعدد أحرف العبارة النصية المدخلة.
4. استخدم معادلة تجميع النصوص التالية:
المتغير الخاص بتجميع أحرف السلسلة النصية =
حرف السلسلة وفق التكرار المتغير الخاص
بتجميع أحرف السلسلة النصية.
5. اطبع البرنامج المتغير الخاص بتجميع أحرف السلسلة النصية.

input

```

1  # تعريف المتغير الخاص بتجميع أحرف السلسلة النصية بقيمة ابتدائية فارغة
2  reversed_word = ""
3  # استقبال العبارة النصية من المستخدم
4  user_word = input("iEnter sentece or word to reserve : ")
5  # إنشاء حلقة تكرارية وفقاً لعدد الأحرف
6  for sentence in user_word:
7      # استخدام معادلة تجميع النصوص لعكس الأحرف
8      reversed_word = sentence + reversed_word
9      # طباعة السلسلة النصية المعكوسة
10 print("The reversed word is:", reversed_word)

```

Out put

```

C:\Users\HP\PycharmProjects\pythonProject
iEnter sentece or word to reserve : kuwait
The reversed word is: tiawuk
Process finished with exit code 0

```

حل آخر: استخدام خاصية (Slicing).

هذه الطريقة تعتمد على [start: end: step] للوصول إلى أجزاء معينة من السلسلة النصية. مع تعيين قيمة step بقيمة -1 لعكس النص بأكمله.

input

```

1  # استقبال السلسلة النصية من المستخدم
2  user_word = input("Enter the sentence or word to reverse: ")
3  # استخدام التقطيع لعكس السلسلة النصية
4  reverse_word = user_word[::-1]
5  # طباعة السلسلة النصية المعكوسة
6  print("The reverse word is:", reverse_word)

```

Out put

```

C:\Users\HP\PycharmProjects\pythonProject 10.
Enter the sentence or word to reverse: kuwait
The reverse word is: tiawuk
Process finished with exit code 0

```

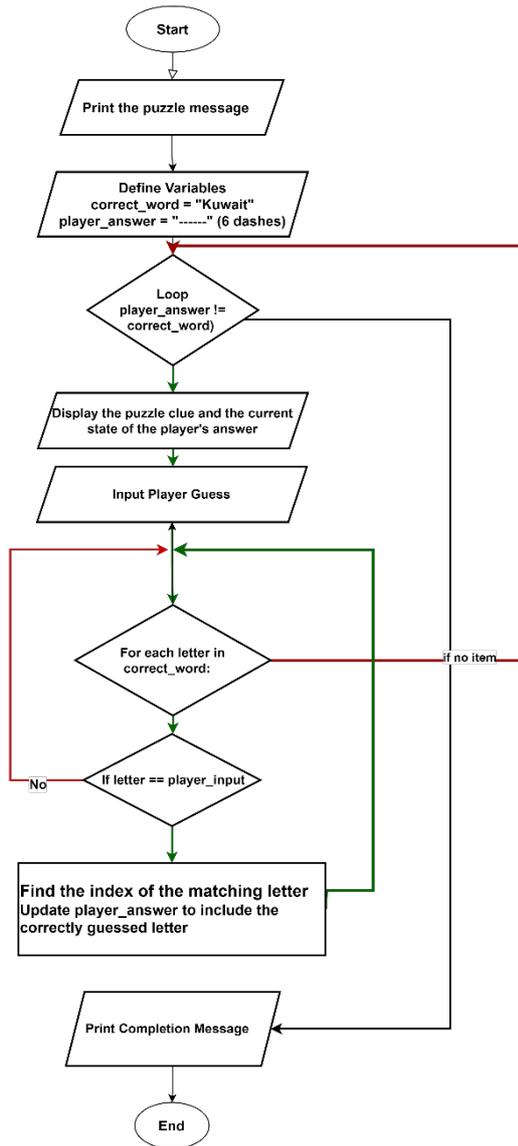
حل ورقة العمل الإثرائية (2-1).

برنامج لغز التخمين: المشكلة:

يستنتج المستخدم الكلمة الصحيحة للغز وفق رسالة توضيحية تظهر له من البرنامج .

الخوارزمية:

خريطة التدفق:



1. طباعة رسالة توضيحية حول اللغز.
2. الاعلان عن متغير يمثل كلمة اللغز.
3. تظهر رسالة توضح عدد أحرف اجابة اللغز ممثلة بالرمز (-) ،
مضروبة بعدد أحرف الكلمة.
4. انشاء حلقة تكرارية، تستمر حتى تتوافق إجابة المستخدم
مع الإجابة الصحيحة.
 - استقبال من المستخدم إحدى حروف كلمة اللغز.
 - إذا كان الحرف المدخل يطابق إحدى أحرف كلمة اللغز.
 - تحديد موضع (indexing) الحرف المدخل من الكلمة الصحيحة
للغز.
 - تعديل قيمة المتغير إجابة المستخدمة ليصبح كالتالي:
إجابة المستخدم = الأحرف من بداية قيمة متغير إجابة
المستخدم، وحتى موقع الحرف الصحيح + الحرف الصحيح المدخل
+ الأحرف من بعد موقع الحرف الصحيح المدخل حتى نهاية قيمة
متغير إجابة المستخدم.
 - طباعة رسالة بالأحرف الصحيحة المدخلة، والرمز (-) بالأحرف
الناقصة .
5. اظهار رسالة تشجيعية للمستخدم

Input

```

1 # الكلمة الصحيحة التي يجب على اللاعب تخمينها
2 correct_word = "Kuwait"
3 # رسالة اللغز التي تظهر للمستخدم
4 puzzle = "I am a country in the middle east, my capital is where my name shines. What am I?"
5 # إنشاء سلسلة من الرموز "-" بعدد أحرف الكلمة الصحيحة لتمثيل التقدم في التخمين
6 player_answer = "-" * len(correct_word)
7 # حلقة تستمر حتى يتمكن اللاعب من تخمين الكلمة الصحيحة بالكامل
8 while player_answer != correct_word:
9     # عرض اللغز والتقدم الحالي في التخمين
10    print(puzzle)
11    print(player_answer)
12    # طلب إدخال حرف من المستخدم
13    player_input = input("Guess a letter: ")
14    # فحص كل حرف في الكلمة الصحيحة لمعرفة ما إذا كان يتطابق مع الحرف المدخل
15    for letter in correct_word:
16        if letter == player_input:
17            # إيجاد موقع الحرف الصحيح في الكلمة
18            letter_index = correct_word.find(letter)
19            # تحديث إجابة اللاعب بإضافة الحرف الصحيح في الموقع المناسب
20            player_answer = player_answer[:letter_index] + letter + player_answer[letter_index + 1:]
21            # عند التخمين الصحيح للكلمة، طباعة رسالة تهنئة
22    print("Well done! The correct answer is: " + correct_word)

```

Out put

```

"C:\Users\HP\PycharmProjects\pythonProject 10-1\.venv\Scripts\python.exe" "C:\Users\HP\
I am a country in the middle east, my capital is where my name shines. What am I?
-----
Guess a letter: l
I am a country in the middle east, my capital is where my name shines. What am I?
-----
Guess a letter: K
I am a country in the middle east, my capital is where my name shines. What am I?
K-----
Guess a letter: t
I am a country in the middle east, my capital is where my name shines. What am I?
K---t
Guess a letter: u
I am a country in the middle east, my capital is where my name shines. What am I?
Ku---t
Guess a letter: w
I am a country in the middle east, my capital is where my name shines. What am I?
Kuw--t
Guess a letter: a
I am a country in the middle east, my capital is where my name shines. What am I?
Kuwa-t
Guess a letter: i
Well done! The correct answer is: Kuwait

```

Input

```

1 correct_word = "Mecca"
2 puzzle = "I am a country in the Kingdom of Saudi Arabia. I am the most important country for Muslims in the world. what am I!?"
3 player_answer = "-" * len(correct_word)
4 while player_answer != correct_word:
5     print(puzzle)
6     print(player_answer)
7     player_input = input("Guess a letter: ")
8     # المرور عبر كل حرف في الكلمة الصحيحة للتحقق من وجود الحرف المنحل
9     for x in range(len(correct_word)):
10        if correct_word[x] == player_input:
11            # تحديث تعيين اللاعب بتبادل الحرف الصحيح في الموقع المناسب
12            player_answer = player_answer[:x] + player_input + player_answer[x + 1:]
13        print("Well done! the correct answer is: " + correct_word)

```

والتعامل مع مشكلة أن تكون كلمة اللغز بها أحرف مكررة.

شرح للمتعلم التعامل مع الأحرف المكررة من خلال:

- `range()` تُستخدم لتوليد سلسلة من الأعداد تبدأ من 0 وإذا طبق الكلمة "Mecca" (طولها 5)، فإن `range(len(correct_word))` سينتج سلسلة الأعداد

```

M   e   c   c   a
[0] [1] [2] [3] [4]

```

`for x in range(len(correct_word))`

- **التأكيد X** يمثل فهرسًا (أو موضعًا) في الكلمة وبتراوح من 0 إلى طول الكلمة - 1. يتم استخدامه للوصول إلى كل حرف في الكلمة باستخدام الفهرس.
- في حال تم إدخال من المستخدم للحرف "c"، يتم تحديث في كل موضع 2-3 يتطابق فيه الحرف "c"



القوائم List

القوائم في Python تعد من الهياكل البيانية الأساسية التي تسمح بتخزين وتنظيم مجموعات من البيانات بطريقة مرنة وفعالة، تُستخدم القوائم لتخزين مجموعة من العناصر، مثل الأرقام والنصوص، والأشياء المعقدة، وتُعتبر جزءاً أساسياً لقدرتها على تخزين البيانات بترتيب محدد وإمكانية تعديلها والتعامل معها بسهولة.

خطة الدروس: القوائم List

 python 12	حصة دراسية	<p>التعرف على القوائم Lists وكيفية انشاؤها . البيانات غير الأولية. القوائم.</p> <ul style="list-style-type: none">• انشاء القوائم.• العمليات على القوائم.• فهرسة Indexing عناصر القائمة List .• تعديل عناصر القائمة List .• إضافة وتحرير عناصر للقائمة List .• حذف عناصر من القائمة List.• الحصول على طول القائمة len().	
 python 13	حصتان دراسيتان	<p>تابع/القوائم Lists:</p> <ul style="list-style-type: none">• التأكد من وجود عنصر في القائمة if in .• تعبئة القائمة من المستخدم• التعامل مع عناصر القائمة من خلال عملية التكرار• عمليات الفرز على القوائم	



امسح الرمز



نتائج التعلم: القوائم List

التعرف على مفهوم القوائم.	1
انشاء قائمة.	2
انشاء قائمة تحتوي على قوائم مختلفة.	3
التعرف على فهرسة عناصر القائمة.	4
تعديل عناصر القائمة.	5
اضافة وتحرير عناصر القائمة.	6
حذف عناصر من القائم.	7
الحصول على طول القائم.	8
التأكد من وجود عنصر في القائمة.	9
تعبئة قائمة من المستخدم.	10
طباعة عناصر القائمة.	11
نسخ قائمة من المستخدم.	12
ترتيب القائمة.	13
تقسيم السلسلة النصية الى قائمة.	14
تجميع قائمة من السلاسل النصية.	15



مصطلحات حاسوبية: القوائم List .



فهرسة القوائم Indexing



القوائم Lists



الداالة Remove



الداالة pop



الداالة Len



الداالة Clear



الداالة Sort



الداالة Copy



الداالة Split



الداالة Join



ملاحظات للمعلم

يمكن إنشاء قائمة من العديد من أنواع البيانات مثل الأعداد الصحيحة والعشرية والسلاسل النصية وأيضا ممكن أن تحتوي القائمة على عدة قوائم مختلفة :

التعليمات البرمجية

Input

```
1 # قائمة الأعداد الصحيحة
2 numbers = [1,2,3,4,5,6]
3 # قائمة الأعداد العشرية
4 decimals = [1.0, 2.5, 3.14]
5 # قائمة السلاسل النصية
6 string = ["hello ", " world", "I am", "student"]
7 # قائمة متعددة الأنواع
8 multi = ["hello ", " world", "I am", 16, "years", "old"]
9 # قائمة تحتوي على قوائم أخرى (Nested Lists):
10 name_degree = [{"Fahad", "Ahmed", "Ali"}, [13,14,15]]
```

التأكيد على العمليات على القوائم وعرض أمثله لكل عملية

الوصول إلى العناصر باستخدام الفهرس (الفهرس يبدأ من 0).
إمكانية التعديل للقيمة المخزنة باستخدام الفهرس.
إضافة عناصر جديدة إلى نهاية القائمة باستخدام الدالة `append()`.
حذف عناصر من القائمة باستخدام `pop()` - `remove()`.
حذف كافة العناصر استخدام الدالة `clear()`.
الحصول على طول القائمة باستخدام الدالة `len()`.
إمكانية فرز القوائم `sort()`.
التحقق من وجود عنصر معين في القائمة باستخدام `in` في التعبير الشرطي.





حل أوراق العمل: القوائم List.

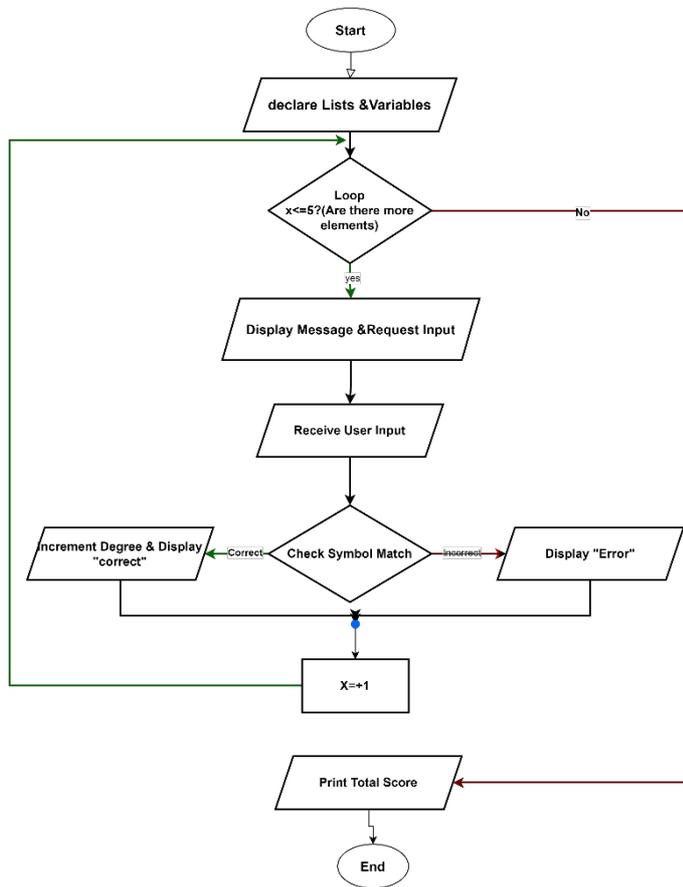
ورقة عمل (1-1)

المشكلة:

البرنامج يجب أن يعرض أسماء العناصر الكيميائية من قائمة ويطلب من المستخدم إدخال الرمز الكيميائي لكل عنصر، ثم يحسب الدرجة بناءً على صحة الإدخالات.

كتابة الخوارزمية:

رسم خريطة التدفق:



1. الإعلان عن قائمة بأسماء العناصر الكيميائية
element _ name
2. الإعلان عن قائمة برموز العناصر
element _ symbole بنفس ترتيب القائمة الأولى.
3. الإعلان عن المتغير degree ، وقيمته الابتدائية تساوي صفر .
4. الإعلان عن المتغير es لاستقبال الرموز العناصر من المستخدم وتكون قيمته الابتدائية " " .
5. إنشاء حلقة تكرارية بعدد العناصر القائمة .
 - يظهر عنصر من القائمة element_name ويطلب من المستخدم ادخال رمزه .
 - يختبر تطابق الرمز المدخل مع نظيره في القائمة element_symbol ثم :
 - يظهر عبارة "Correct" اذا كانت الإجابة صحيحة ، ويضيف 1 الى قيمة المتغير degree .
 - يظهر عبارة "Error" اذا كانت الإجابة خاطئة بدون تغيير في قيمة degree .
6. يظهر الدرجة الكلية.

Input

```

1  # قائمة بأسماء العناصر الكيميائية
2  element_name = ['Sodium', 'Potassium', 'Copper', 'Iron', 'Mercury', 'Lead']
3  # قائمة برموز العناصر الكيميائية المقابلة
4  element_symbol = ["Na", 'K', 'Cu', 'Fe', 'Hg', 'Pb']
5  # إعلان المتغير x لتتبع موضع العنصر في القائمة
6  x = 0
7  # إعلان المتغير es لإستقبال إدخال المستخدم (رمز العنصر)
8  es = ""
9  # إعلان المتغير degree لحساب درجة المستخدم
10 degree = 0
11 # حلقة تكرارية تستمر حتى يتم المرور على جميع العناصر في القائمة
12 while x <= 5:
13     # طباعة رسالة تطلب من المستخدم إدخال رمز العنصر الحالي
14     print('Enter element symbol for', "(" , element_name[x], ")")
15     es = input()
16     # التحقق مما إذا كان الرمز المدخل يتطابق مع الرمز الصحيح
17     if es == element_symbol[x]:
18         # إذا كان الرمز صحيحاً، زيادة درجة المستخدم بمقدار 1
19         degree += 1
20         print("Correct")
21     else:
22         # إذا كان الرمز خاطئاً، عرض رسالة خطأ
23         print("Error")
24         # للانتقال إلى العنصر التالي في القائمة X زيادة قيمة
25         x += 1
26 # طباعة الدرجة الكلية التي حصل عليها المستخدم مع عدد الأسئلة الكلية
27 print('Degree:', degree, "/6")

```

Out put

```

C:\Users\HP\PycharmProjects\pythonProject
What is the symbol for Sodium? ,
Error
What is the symbol for Potassium? K
Correct
What is the symbol for Copper? Cu
Correct
What is the symbol for Iron? Fe
Correct
What is the symbol for Mercury? M
Error
What is the symbol for Lead? L
Error
Your total score is: 3/6

```

تابع حل ورقة عمل (1-1).

حل آخر باستخدام دالة For إضافي يمكن مناقشته مع المتعلمين.

التعليمات البرمجية

Input

```
1 # الإعلان عن قائمة بأسماء العناصر الكيميائية
2 element_name = ['Sodium', 'Potassium', 'Copper', 'Iron', 'Mercury', 'Lead']
3 # الإعلان عن قائمة برموز العناصر بنفس ترتيب القائمة الأولى
4 element_symbol = ["Na", "K", "Cu", "Fe", "Hg", "Pb"]
5 # قيمته الابتدائية تساوي صفر degree الإعلان عن المتغير
6 degree = 0
7 # إنشاء حلقة تكرارية بعدد عناصر القائمة
8 for i in range(len(element_name)):
9     # يطلب من المستخدم إدخال رمزه element_name يُطوّر عنصر من القائمة
10    es = input("What is the symbol for " + element_name[i] + "? ")
11    # يختبر تطابق الرمز المدخل مع نظيره في القائمة
12    if es == element_symbol[i]:
13        print("Correct")
14        # يضيف 1 إلى قيمة المتغير degree
15        degree += 1
16    else:
17        print("Error")
18    # يُطوّر الدرجة الكلية
19 print("Your total score is: " + str(degree) + "/" + str(len(element_name)))
```

ورقة عمل (1-2)

المشكلة:

إنشاء قائمة فارغة، وإدخال عناصرها العددية من المستخدم، ثم حساب المتوسط الحسابي للعناصر.

الخوارزمية:

1. الإعلان عن قائمة فارغة.
2. الإعلان عن متغير فارغ sum لجمع عناصر القائمة .
3. إنشاء حلقة تكرارية لا نهائية لتنفيذ التالي:
 - استقبال من المستخدم عدد.
 - إذا كان عملية الإدخال تساوي (تم). يخرج من الحلقة التكرارية.
 - إذا لم يتحقق الشرط السابق: يحول المدخل إلى رقم عشري، ويضيفه للقائمة .
 - جمع العنصر المدخل في المتغير sum.
4. حساب المتوسط الحسابي الذي يساوي (مجموع عناصر القائمة مقسوما على طول القائمة) .
5. طباعة عبارة (الأعداد التي قمت بإدخالها)
6. طباعة القائمة.
7. طباعة عبارة (المتوسط الحسابي) متبوعة بالمتوسط الحسابي الذي تم حسابه.

- التأكد من تحويل المدخل إلى نوع رقمي (عشري) دالة float.
 - فهم مفهوم الحلقة اللانهائية (while True) وكيفية الخروج (break) .
 - استيعاب كيفية حساب المتوسط الحسابي
 - تجميع الأرقام في قائمة numbers.
 - len(numbers) عدد الأرقام التي تم إدخالها.
- المتوسط الحسابي يُحسب بقسمة مجموع الأرقام (sum) على عددها (len(numbers)) .**



ملاحظات للمتعلم

التعليمات البرمجية

خريطة التدفق

input

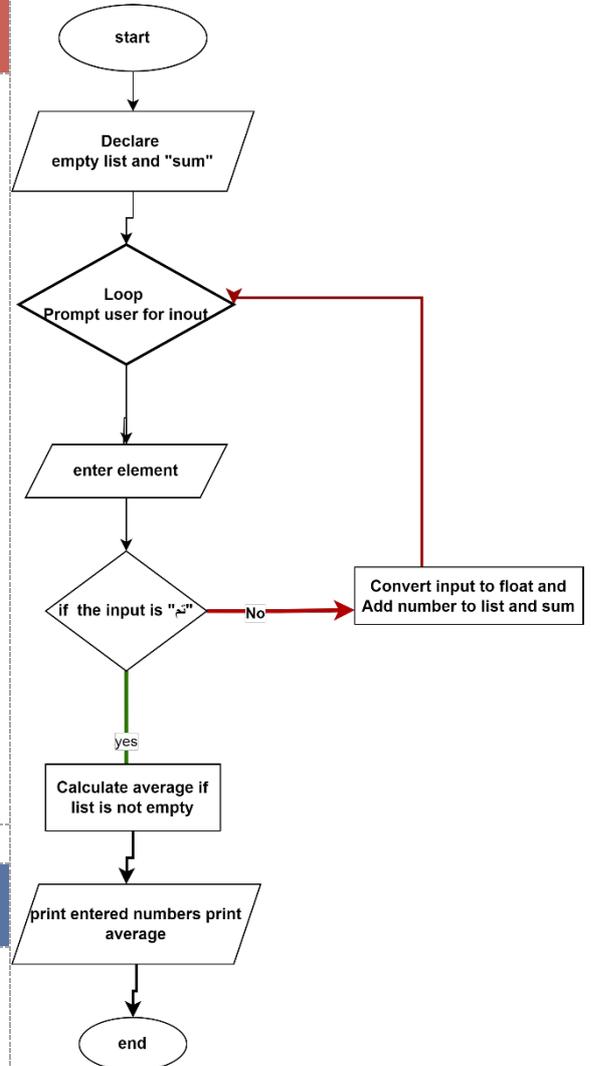
```

1  # الإعلان عن قائمة فارغة لتخزين الأعداد
2  numbers_list = []
3  # الإعلان عن متغير لجمع الأعداد
4  sum = 0
5  # استخدام حيلة للسماح للمستخدم بإدخال الأعداد
6  while True:
7      # للحصول على إدخال المستخدم استخدم دالة input()
8      input_number = input("الرجاء إدخال عدد (أدخل 'تم' لالنتهاء): ")
9      # التحقق مما إذا كان المستخدم يريد الإنهاء
10     if input_number == 'تم':
11         break
12     else:
13         # تحويل النص الذي أدخله المستخدم إلى عدد عشري
14         input_number = float(input_number)
15         # إضافة الحد إلى القائمة
16         numbers_list.append(input_number)
17         # إضافة الحد إلى المتغير 'sum'
18         sum = sum + input_number
19     # حساب متوسط الأعداد
20     if len(numbers_list) > 0:
21         average = sum / len(numbers_list)
22     else:
23         average = 0
24     # طباعة عناصر القائمة
25     print("الأعداد التي قمت بإدخالها:")
26     print(numbers_list)
27     # طباعة متوسط الأعداد
28     print("المتوسط الحسابي:", average)
    
```

Out put

```

"C:\Users\HP\PycharmProjects\pythonProject 10-
الرجاء إدخال عدد (أدخل 'تم' لالنتهاء): 25
الرجاء إدخال عدد (أدخل 'تم' لالنتهاء): 33
الرجاء إدخال عدد (أدخل 'تم' لالنتهاء): 43
الرجاء إدخال عدد (أدخل 'تم' لالنتهاء): 34
الرجاء إدخال عدد (أدخل 'تم' لالنتهاء): 12
الرجاء إدخال عدد (أدخل 'تم' لالنتهاء): تم
الأعداد التي قمت بإدخالها:
[25.0, 33.0, 43.0, 34.0, 12.0]
المتوسط الحسابي: 29.4
    
```



ملاحظة : تذكر أن في لغات البرمجة حلول مختلفة

تابع حل أوراق العمل: القوائم List .

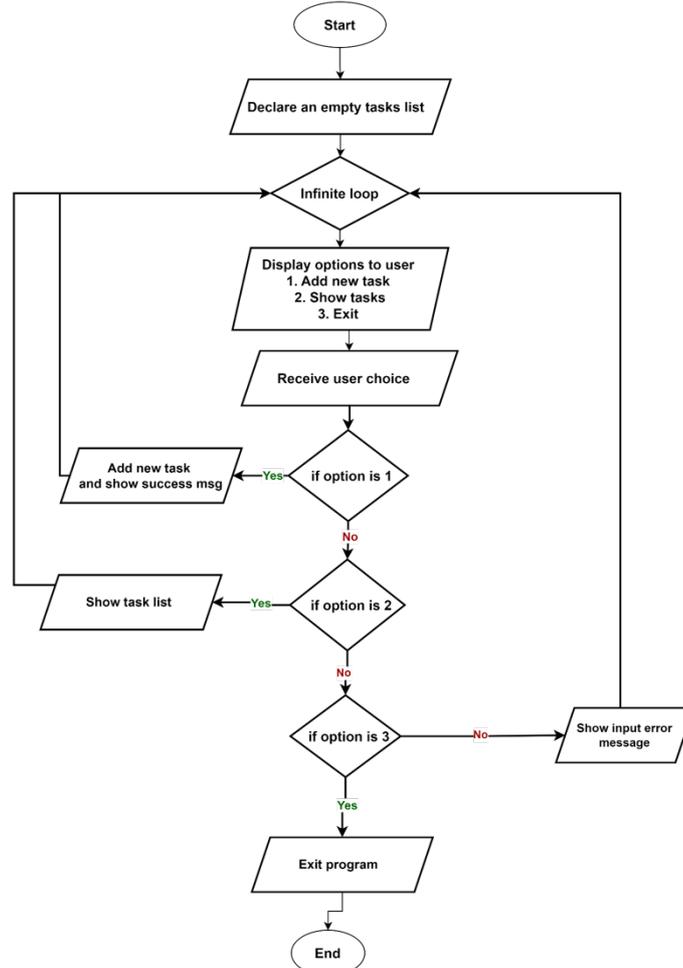
ورقة عمل (3-1) المشكلة:

توظيف بعض الدوال المتعلقة بالقوائم، (الطباعة ، والإضافة)

الخوارزمية:

- يظهر للمستخدم ثلاث خيارات:
 1. إدخال مهمة جديدة.
 2. عرض المهام.
 3. خروج من البرنامج
- يستقبل من المستخدم الخيار المناسب:
 1. إذا اختار المستخدم 1: (يستقبل من المستخدم المهمة الجديدة، ويضيفها إلى القائمة) ، ويظهر رسالة " تمت إضافة المهمة بنجاح " ، ثم يستمر التكرار .
 2. إذا اختار المستخدم 2: (يطبع محتوى القائمة على الشاشة)، يظهر عبارة " مهامك هي " ، ثم يعرض القائمة ، ثم يستمر التكرار.
 3. إذا اختار المستخدم 3: (يخرج من البرنامج).
 4. إذا أدخل المستخدم عدد مختلف عن الخيارات السابقة، يظهر رسالة تحذير خطأ في الإدخال.

خريطة التدفق



تابع حل أوراق العمل: القوائم List.

ورقة عمل (3-1)

التعليمات البرمجية

input

```
1 # إعلان قائمة المهام الفارغة
2 tasks_list = []
3 while True:
4     # عرض خيارات المستخدم
5     print("مدير المهام:")
6     print("1. إضافة مهمة")
7     print("2. عرض المهام")
8     print("3. الخروج")
9     # استقبال اختيار المستخدم
10    choice = input("(1/2/3) الرجاء اختيار الخيار ")
11    if choice == '1':
12        # طلب إدخال مهمة جديدة
13        task = input("الرجاء إدخال مهمة جديدة: ")
14        # إضافة المهمة إلى القائمة
15        tasks_list.append(task)
16        # عرض رسالة نجاح
17        print("تمت إضافة المهمة بنجاح")
18    elif choice == '2':
19        # عرض قائمة المهام
20        print("قائمة المهام:")
21        print(tasks_list)
22    elif choice == '3':
23        # إنهاء البرنامج
24        print("تم الخروج من مدير المهام")
25        break
26    else:
27        # عرض رسالة خطأ
28        print("الرجاء اختيار خيار صحيح")
```

Out put

```
تمت إضافة المهمة بنجاح
مدير المهام:
1. إضافة مهمة
2. عرض المهام
3. الخروج
الرجاء اختيار الخيار (1/2/3): 1
الرجاء إدخال مهمة جديدة: قراءه
تمت إضافة المهمة بنجاح
مدير المهام:
1. إضافة مهمة
2. عرض المهام
3. الخروج
الرجاء اختيار الخيار (1/2/3): 2
الرجاء إدخال مهمة جديدة: بحث
تمت إضافة المهمة بنجاح
مدير المهام:
1. إضافة مهمة
2. عرض المهام
3. الخروج
الرجاء اختيار الخيار (1/2/3): 3
تم الخروج من مدير المهام
```

Process finished with exit code 0

- الدالة \n تستخدم داخل سلسلة نصية تعمل على نقل المؤشر إلى السطر التالي

استخدم التعليمة التالية:

```
print("\nمدير المهام:\n1. إضافة مهمة.\n2. عرض المهام.\n3. الخروج")
```

- حلقة while True: لإنشاء تكرار مستمر.

دالة break تستخدم في كسر حلقة التكرار سواء حلقات for و while لوقف التكرار والخروج من الحلقة.



ملاحظات للمتعلم

- برمجة بايثون Python
المنتجات الرقمية.

04



برمجة بايثون Python: المنتجات الرقمية.

تختلف وحدة (المنتجات الرقمية) عن باقي وحدات الكتاب كونها لا تعتمد على تنفيذ أوراق عمل محددة ومعدة مسبقاً للمتعلم، وإنما تعتمد على ابداع المتعلم سواء كان منفرداً أو ضمن مجموعة في توظيف ما تم تعلمه من معارف ومهارات حاسوبية في انتاج مشروع بسيط يخدم موضوع معين أو يقدم خدمة معينة أو يعرض معلومة بشكل مختلف. يقدم المتعلم في وحدة المنتجات الرقمية مشروعاً من خلال ما تعلمه من مهارات ومعارف في لغة البايثون والمعلومات التي تم تطبيقها من خلال الدروس وتطبيق أوراق العمل والأمثلة خلال الفصل الدراسي.



project

نتائج التعلم



امسح الرمز

نتائج التعلم: المنتجات الرقمية.

1. اظهار الابداع والابتكار لدى المتعلم.
2. المشاركة بإيجابية والتعاون مع زملائه في الفريق لإنتاج المشروع.
3. تنمية العمل الجماعي ومشاركة المعلومات والاستماع الى أفكار الآخرين.
4. توظيف المهارات التي اكتسبها المتعلم والاستفادة منها.
5. تعزيز مهارات التفكير العليا من تحليل للمشكلات وحلها والربط بين المعلومات.
6. تطبيق معارف المتعلمين ومكتسباتهم المعلوماتية في مواقف حقيقية.
7. تنمية مهارات التعلم الذاتي والبحث باستخدام مصادر متنوعة للتعلم لتطوير مشروع.
8. زيادة تحليل وتركيب وتقويم المعلومات بطريقة عقلانية.



مراحل اعداد المشروع:

1-تحديد الفكرة

تحديد موضوع المشروع والهدف منه



2-اعداد خطة الحل (الخوارزمية)

تعرف الخوارزمية على أنها مجموعة من الخطوات المحددة والمتسلسلة من أجل تنفيذ مهمة معينة أو حل مشكلة ما أو مجموعة من التعليمات البرمجية لتنفيذ مهمة بحيث يتم إدخال المدخلات (inputs) وتنفيذ المهمة (process) لإنتاج المخرجات (outputs).



3-اعداد خريطة التدفق

المخطط الانسيابي (Flowchart) هو تمثيل مرئي لعملية أو سير عمل. يتكون عادةً من سلسلة من الأشكال، يمثل كل منها خطوة مختلفة في العملية، متصلة بواسطة أسهم توضح تدفق العملية من خطوة إلى أخرى. وقد تم شرح هذه الأشكال مع وظائفها في كتاب المتعلم.



4-كتابة البرنامج

يتم تحويل خريطة التدفق الى مجموعة من التعليمات البرمجية باستخدام لغة بايثون



5-اختبار البرنامج

يتم تشغيل البرنامج وإدخال بيانات ومقارنة النتائج للاختبار صحة وعمل البرنامج بصورة سليمة، وتصحيح الأخطاء ان وجدت في هذه المرحلة.





المهارات المقترحة للمشروع:

من المهم أن يتضمن مشروع المتعلم معظم المهارات التي اكتسبها المتعلم خلال الفصل الدراسي ويمكنك استخدام المقترح التالي للمهارات المتضمنة في المشروع:

م	البند	مثال
1	انشاء مشروع جديد	File ____ > new project
2	انشاء ملف بايثون جديد	New ---- → python file
3	الإعلان عن المتغيرات (نصية , عددية , منطقية)	Str () , int() , float() , booi()
4	دالة الإخراج	Print ()
5	استخدام العمليات الحسابية	** , % , // , / , * , - , +
6	دالة استقبال البيانات	Input()
7	استخدام العمليات المنطقية	And , or
8	استخدام العمليات الشرطية	If , elif , else
9	استخدام الدوال التكرارية	While , for
10	انشاء السلاسل النصية	
11	فهرسة السلاسل النصية	Indexing []
12	تقسيم السلاسل النصية slicing	
13	جمع السلاسل النصية concatenation	+
14	تكرار عناصر السلسلة النصية	*
15	استخدام دوال السلاسل النصية	Capitalize() - Upper() - Lower() - Find() - Replace() - f-string - Int() - Float() - Center() -



تابع: المهارات المقترحة للمشروع:

		
مثال	البند	م
	انشاء القوائم	16
<ul style="list-style-type: none">- فهرسة indexing- تعديل عناصر القائمة- إضافة وتحرير عناصر للقائمة- حذف عناصر من القائمة (حسب الرقم-pop-remove)- استخراج طول القائمة len()- التأكد من وجود عنصر بالقائمة- تعبئة قائمة من المستخدم- طباعة عناصر القائمة- عمليات الفرز على القوائم (حسب الأعداد العشرية, حسب السلاسل النصية)	العمليات على القوائم	17



تابع: المهارات المقترحة للمشروع.

يبدأ المتعلم العمل بالمشروع بدءاً من الاسبوع السابع، ويُستكمل تدريجياً خلال الأسابيع اللاحقة ويمكن للمعلم متابعة مشاريع المتعلمين من خلال الجدول المقترح التالي:

		متابعة مراحل اعداد المشروع						
		الأسبوع 12	الأسبوع 11	الأسبوع 10	الأسبوع 9	الأسبوع 8	الأسبوع 7	
ملاحظات	التقييم النهائي	عرض ومناقشة المشروع	اختبار البرنامج وتصحيح الأخطاء	كتابة البرنامج	اعداد خريطة التدفق	اعداد خطة الحل (الخوارزمية)	تحديد الفكرة	اسم المشروع وأسماء أعضاء المشروع
								مشروع:
								-1
								-2
								-3
								-4
								مشروع:
								-1
								-2
								-3
								-4

يسجل المعلم في هذا الجدول ما أنجزه كل فريق في كل أسبوع من أسابيع تنفيذ المشروع ليسهل عليه متابعة انجاز المتعلمين ورصد الدرجات في نهاية الفصل.



أمور يجب مراعاتها في وحدة المنتجات الرقمية:

1. يجب تقديم فكرة عن فصل المنتجات الرقمية (وحدة المشاريع) قبل البدء فيها بوقت كافي حتى يستعد المتعلمون نفسيا وذهنيا لها مع بداية الفصل الدراسي.



2. الحرص على تقديم نماذج من مشاريع جيدة تم انجازها مسبقا أو أعتها المعلم حتى يتضح مفهوم المشروع في أذهان المتعلمين.



3. الحرص على متابعة فرق العمل والتأكيد على العمل التعاوني وقيام كل متعلم بدوره، وتكثيف المتابعة للفرق التي تحتاج وحل المشاكل الفنية التي تواجه المتعلمين أولا بأول .



4. ترك الحرية للمتعلمين في اختيار أعضاء الفريق للمشروع مع الالتزام بعدد الأعضاء بحيث لا يقل عن اثنين ولا يزيد عن أربعة مع تحديد القائد ليكون مسؤولا عن التواصل مع المعلم وتنسيق الفريق .



5. ضرورة تجميع مشاريع المتعلمين على وسيط تخزين مناسب وتسليمها لرئيس القسم مع الدرجات الخاصة بها.



تحضير الدروس

05



تحضير الدروس:

أهمية تحضير الدرس للمعلم

1	يساعد المعلم على تنظيم أفكاره وترتيبها، فيتوقع المواقف التعليمية التي قد تظهر له.
2	يعين المعلم على اختيار الأنشطة والاستراتيجيات المناسبة للدرس.
3	يعتبر إعداد الدرس سجلاً مكتوباً لمحتوى التعلم يرجع إليه المعلم إذا نسي شيئاً أثناء الحصة.
4	يستطيع المعلم من خلال التحضير اليومي أن يعرف ماذا شرح للمتعلم، من مفاهيم ومعلومات، وماذا تبقى له من مفاهيم لم يتمكن من شرحها وتم تأجيلها للحصة القادمة بسبب ظرف طارئ.
5	بعد إعداد الدرس مهما من الناحية التربوية، إذ يعتمد عليه مدير المدرسة أو الموجه الفني لمتابعة المعلم أثناء زيارة الحصة.

سجل التحضير

يجب أن يتضمن السجل الصفحات التالية:

1	بيانات المعلم الأساسية. (الاسم الرباعي، الرقم المدني، الجنسية، رقم الملف، تاريخ التعيين، مركز العمل السابق، المؤهل).
2	توقيت اليوم الدراسي.
3	الجدول الدراسي للمعلم والمواد التي يقوم بتدريسها، ثم الأعمال المسندة إليه سواء أنشطة أو مهام إدارية.
4	خطة توزيع المقرر للفصل الدراسي الحالي.
5	الأهداف التربوية. (الهدف الشامل للتربية بالكويت، أهداف المرحلة التعليمية، أهداف تدريس الحاسوب العامة)
6	التحضير اليومي للدرس حسب خطة توزيع المقرر.
7	يمكن أن يضم سجل متابعة المتعلمين اليومي لسجل التحضير أو أن يرفق للحصة بسجل مستقل.

خطة درس مادة الحاسوب للصف

بصاغ بحسب خطة المقرر الدراسي			
الوحدة	اليوم	التاريخ	الفصل
بصاغ من الكتاب المدرسي			

يتم تسجيلها لكل حصة دراسية

القيمة التربوية	مصادر التعلم والوسائل التعليمية
تحديد القيمة التربوية حسب الجدول الزمني المقترح مع وصف زمني لكيفية تفعيلها خلال الحصة بما يتلاءم مع عرض المفاهيم الدرس	تسجل المصادر التي يستخدمها أثناء عرض الدرس لإثراء الموقف التعليمي (مواد، أدوات رقمية) لا يوجد عدد معين للمصادر التي يمكن استخدامها في الدرس الواحد فهذا يتوقف على طبيعة الدروس

الزمن	النشاط الاستهلاكي	المقدمة والتمهيد
5-3 دقائق	تهيئة عقول المتعلمين للمعلومات الجديدة وتشويقهم لها وتختلف حسب موضوع الدرس، كأمثلة للمقدمة: - عرض الناتج النهائي للوحدة ليكون المتعلم فكرة عن المفاهيم التي ستعرض في الدروس القادمة - مراجعة جزء سبق دراسته له علاقة بالدرس - عرض فيديو متعلق بموضوع الدرس او استعراض مواقع وصفحات على النت. *يراعى في المقدمة وضوحها ومناسبتها للفئة العمرية للمتعلمين وارتباطها بالدرس وعدم الإطالة بالمقدمة والتأثير على الزمن المخصص لشرح الدرس	

م	نواتج التعلم	محتوى التعلم (الأنشطة والاستراتيجيات)	التقويم	الزمن
	عبارة تتضمن ما الذي يتوقع من المعلم تحقيقه نتيجة للأنشطة التعلم ويعبر عنها على شكل معارف أو مهارات او اتجاهات ويجب أن تكون واضحة تصف ما الذي يتوقع ان يتعلمه المتعلمين نتيجة اكمال عملية التعلم	محتوى التعلم يجب أن يجب عن السؤال التالي: ما الذي يجب تعلمه؟ ويص الأنشطة والاستراتيجيات المستخدمة مع وصف مختصر للدور المناط بكل من المعلم والتعلم	صياغة لغوية واضحة للأسئلة والتعليمات التي توجه نشاط المتعلم منعا للارتجال وتحسينا للتواصل اللفظي مع المتعلم باستخدام أدوات ووسائل التقويم، وتعكس هذه الأدوات إنجازات المتعلم وقياسها في مواقف حقيقية، فيبدو كمنشآت تعلم يمارس فيه المتعلمين مهارات التفكير العليا وحل المشكلات الحياتية الحقيقية	تقدير الزمن المناسب لتنفيذ النشاط لاستثمار وقت الحصة
	التقويم الختامي	حل ورقة عمل: ص: أو ورقة عمل بديلة من تصميم المعلم بحيث تقيس النواتج بشكل شامل		

التأمل الذاتي			الحصة	الفصل	اليوم والتاريخ
مقترحات لتحسين الأداء	تحديات واجهتني	الرضا عن تحقق الأهداف			
يقود المعلم الى تحسين أداءه واجراء التغييرات المناسبة التي تنعكس ايجابياً على التعليم (استبدال نشاط بآخر والتي تساعد على تحسين الأداء	التعرف على الأسباب والمشكلات ووضع خطط حلولها (من التحديات توقف النت – ضعف أحد الأنشطة لتوصيل المهارة المطلوبة)	☹️ ☺️ 😊			



خطوات اعداد الدرس:

- . القراءة الدقيقة في موضوع الدرس من خلال: الكتاب المدرسي، دليل المعلم أو أيا من مختلف المراجع.
- . تحديد أهداف الدرس / نواتج التعلم بدقة ووضوح، وصياغتها صياغة صحيحة.
- . الإعداد الذهني وهو رسم الخطة لتحقيق تلك الأهداف/ نواتج التعلم. وقبل أن تبدأ في الكتابة يجب أن تكون فكرة خطة التدريس قد تبلورت في ذهنك.
- . الإعداد الكتابي بعد أن تكون تصورت تصورا كاملا ومترابيا لطريقة سير الدرس قم بتسجيلها على شكل خطوات واضحة ومحددة، شاملة عناصر الإعداد، مراعيًا في كل خطوة عامل الوقت وارتباطها بأهداف الدرس.
- . إعداد متطلبات الدرس، غالبا ما يحتاج المعلم في الشرح لبعض الأدوات، وينبغي عليه الاهتمام بتحضيرها والتأكد من صلاحيتها وإمكانية استخدامها في المكان الذي ستستخدم فيه.

أخطاء شائعة يقع فيها المعلم عند إعداد الدرس

- . الإهمال في كتابة التاريخ أو لفصل واحد فقط.
- . تجاهل استخدام وسيلة تعليمية أثناء الشرح، تم الإشارة إليها في بند مصادر التعلم والوسائل التعليمية.
- . في المقدمة والنشاط الاستهلالي لا يصح كتابة) مراجعة قصيرة للدرس السابق وإنما تحدد بنود المراجعة).
- . عدم ذكر حل التمارين صفحة . . . أو تترك بدون حل.
- . عدم مناسبة أسلوب العرض من حيث عدد النقاط فلا تكون قصيرة جداً ولا طويلة جداً.
- . عدم تحديد دور المعلم أو دور المتعلم بشكل واضح.
- . اختيار أسئلة تقويم بصياغة خاطئة. (الأسئلة المربكة التي تطرح أكثر من فكرة، الأسئلة التي تكون إجابتها بنعم / لا، الأسئلة الغامضة غير محددة الهدف، الأسئلة الموحية بالإجابة).
- . إهمال المعلم لتسجيل ملاحظاته في جزء التأمل الذاتي.

نموذج تحضير درس مادة الحاسوب للصف العاشر

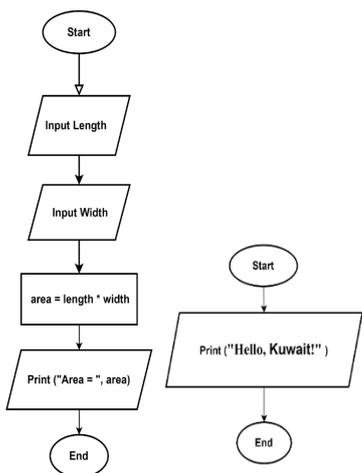
الأمن السيبراني Cyber Security - مدخل الى البرمجة				عنوان الدرس
الأسبوع	الوحدة	اليوم	التاريخ	الفصل
الأول	الأمن السيبراني برمجة Python	الأربعاء	2024-9-18	1/10
		الخميس	2024-9-19	2/10

القيمة التربوية	مصادر التعلم والوسائل التعليمية
توضيح مفهوم الأمن السيبراني ونشر التوعية بمخاطر الهجمات السيبرانية وطرق الوقاية والتخلص منها	السبورة والاقلام — جهاز Data show — بطاقات — سبورة متعلم — فيديو خارجي — عرض تقديمي

الزمن	النشاط الاستهلاكي	المقدمة والتمهيد
5د	<ul style="list-style-type: none"> - الترحيب بالمعلمين في بداية العام الدراسي - عرض قوانين المختبر - عرض محتوى المنهج الدراسي والية توزيع الدرجات 	

م	نواتج التعلم	محتوى التعلم (الأنشطة والاستراتيجيات)	التقويم	الزمن
1	<ul style="list-style-type: none"> - الفهم الشامل لمبادئ وممارسات أمن المعلومات. - التزود بالمعرفة والمهارات اللازمة لحماية الأنظمة والشبكات والمعلومات ضد الهجمات السيبرانية. - التعريف بمفهوم الأمن السيبراني وأهميته لحماية الأنظمة والشبكات. - التمييز بين مفهومي أمن المعلومات والأمن السيبراني . - تعزيز الوعي الأمني لمخاطر الهجمات السيبرانية عند المتعلمين. 	<ul style="list-style-type: none"> - من خلال استراتيجية سرد قصة: - يعرض قصة هجوم "WannaCry" السيبراني الذي أصاب عشرات الآلاف من الأجهزة في عدد كبير من الدول واستخدام برمجيات خبيثة تسمى "الفدية" وكيفية تأثيره على الأفراد والمؤسسات . - من خلال استراتيجية المناقشة: عرض المخاطر وتحديد التهديدات الأمنية وكيفية التصدي واتخاذ إجراءات لحماية معلوماتهم. - من خلال العرض التقديمي: يتم عرض الفيديو التوعوي عن الأمن السيبراني المعد من قبل التوجيه العام 	<ul style="list-style-type: none"> - مناقشة المتعلمين عن طريق السؤال ما هو تأثير هذه البرامج على المجتمع والافراد؟ - مناقشة المتعلمين عن طريق السؤال كيف نحمي بياناتنا الشخصية من الاختراق؟ - توزيع بطاقات تحوي أسئلة وعدة عبارات - من خلال ما تم عرضه وضح مفهوم الأمن السيبراني ؟ - مفهوم أمن المعلومات؟ - عرض برنامج أسئلة وأجوبة تفاعلية لقياس فهم المتعلمين للأهمية الحيوية للأمن السيبراني. - عدد أهمية اتخاذ إجراءات وقائية لحماية الأنظمة والمعلومات. 	5 5 5

م	نواتج التعلم	محتوى التعلم (الأنشطة والاستراتيجيات)	التقويم	الزمن
2	- مدخل الى البرمجة: مفهوم البرمجة	- من خلال استراتيجية العصف الذهني: - يتم طرح سؤال كيف يمكن للألعاب الفيديو، التطبيقات، وحتى المواقع الإلكترونية أن تعمل وتؤدي وظائفها بشكل دقيق؟ - من خلال استراتيجية التشبيه الواقعي: - طرح مثال يشبه الحاسوب هو طاه في مطبخ - والبرمجة هي كتابة وصفة لطبق معين - كل خطوة في الوصفة هي تعليمات - يجب أن يتبعها الطاهي لتحضير وجبة لذیذة أي انجاز البرنامج النهائي	- أتوقع الإجابة من المتعلمين - كل هذا يتم بفضل البرمجة - سؤال: ما مفهوم البرمجة؟	5
3	- خطوات انشاء برنامج لغة البرمجة python	- من خلال عرض تقديمي: - يتم عرض ومناقشة خطوات انشاء البرنامج (تحديد المشكلة – اعداد الخوارزمية – اعداد خريطة التدفق – كتابة البرنامج – اختبار البرنامج - اصدار البرنامج)	- مناقشة المتعلمين حول المفاهيم التالية: - ماهي الخوارزمية Algorithm ؟ - ما هي خريطة التدفق Flowchart ؟	5
4	- كتابة خوارزمية لحل مشكلة	من خلال استراتيجية التعلم القائم على المشاريع: تحديد المشكلة (إيجاد مساحة مستطيل) - استقبال قيمة الطول من المستخدم (بالمتر). - استقبال قيمة العرض من المستخدم (بالمتر). - العملية احسب المساحة باستخدام المعادلة : المساحة= الطول× العرض - اطبع قيمة المساحة على الشاشة.	- مناقشة المتعلمين: - ما هي المعطيات المتاحة؟ أطوال اضلاع المستطيل : طول و عرض المستطيل - ما هي العمليات والإجراءات المطلوبة؟ - قانون المساحة المستطيل = الطول * العرض - توزيع المتعلمين على مجموعات ثم توزيع سورات لتنفيذ التالي: - كتابة الخوارزمية قبل تطبيقها برمجيا	10
5	- تصميم خريطة تدفق Flowchart لإجراء حل مشكلة	- باستخدام عرض تقديمي يتم عرض الأشكال ووظيفة كل شكل وتطبيقه على المثال السابق - باستخدام استراتيجية التعلم التعاوني يتم عرض مثال آخر - كتابة برنامج بسيط لطباعة جملة "Hello, Kuwait" على الشاشة. - كتابة الخوارزمية - استخدم الدالة print () لطباعة النص "Hello, Kuwait" على الشاشة. - رسم خريطة التدفق	- توزيع بطاقات على المجاميع لرسم الخريطة	10



م	نواتج التعلم	محتوى التعلم (الأنشطة والاستراتيجيات)	التقويم	الزمن
6	- استخدام محرر الأوامر PyCharm	- باستخدام استراتيجية المحاضرة: - يتم التوضيح للطلبة أنه يوجد العديد من مواقع تحرير التعليمات البرمجية التي توفر الأدوات لكتابة التعليمات وتحريرها واختبارها وتصحيح أخطائها في مكان واحد. - وسيتم استخدام محرر PyCharm في منحج الحاسوب لتعلم وتطبيق تعليمات لغة البايثون . - عن طريق جهاز المعلم يتم فتح برنامج PyCharm واستعراضه على Data Show واستعراض أجزاء نافذة المحرر	- مناقشة المتعلمين حول معلوماتهم السابقة عن مناطق الشاشة : - ما اسم الشريط العلوي في البرنامج وماذا يعرض ؟	5
7	- انشاء مشروع جديد	- باستخدام التطبيق العملي على جهاز المعلم يتم شرح خطوات انشاء مشروع جديد وتحديد اسمه ومكان الحفظ	- بمشاركة المتعلمين يتم إعادة تطبيق العملية على جهاز المعلم	5
8	- انشاء ملف بايثون جديد	- باستخدام استراتيجية العصف الذهني يطلب من المتعلمين استنتاج طريقة انشاء ملف بايثون جديد - ثم يتم شرح الخطوات عمليا على جهاز المعلم مع توضيح مكان تبويب اسم الملف و أرقام أسطر التعليمات البرمجية	- يتم الطلب من أحد المتعلمين انشاء ملف بايثون جديد على جهاز المعلم أين ظهر تبويب اسم الملف الجديد ؟	5
9	- كتابة برنامج بسيط	- باستخدام السبورة يتم شرح كيفية كتابة تعليمة بسيطة بلغة البايثون - Print (" I'm a Python developer ")	- يتم التطبيق عمليا على أجهزة المتعلمين.	5
1	- تشغيل البرنامج	- من خلال نافذة البرنامج في جهاز المعلم يتم استعراض طرق تشغيل البرنامج (مفاتيح Ctrl+Shift+F10 , الامر Run من القائمة المختصرة لمنطقة كتابة التعليمات البرمجية , الأداة Run من شريط الأدوات)	- يختار المتعلمين احدهم الطرق ويتم تطبيقها لمشاهدة النتيجة على أجهزتهم	5
11	- هياكل البيانات	- من خلال power point يتم استعراض هياكل البيانات	- يلاحظ المتعلم هياكل البيانات ويميز أقسامها	5
التقويم الختامي		حل ورقة عمل (1-1) ورقة عمل (2-1)		20

** تذكير المتعلمين بضرورة إغلاق الملف و البرنامج وإعادة تنظيم المختبر .

اليوم والتاريخ	الفصل	الحصة	التأمل الذاتي		
			الرضا عن تحقق الأهداف	تحديات واجهتني	مقترحات لتحسين الأداء
			☹️ ☺️		

التقويم التربوي

06



التقويم التربوي

التقويم التربوي هو عملية شاملة تهدف إلى تقييم وتطوير أداء المتعلمين على مدار فترة زمنية محددة، سواء كانت فصلًا دراسيًا أو سنة دراسية كاملة.

هذه العملية ليست فقط لقياس مدى فهم المتعلمين للمواد الدراسية، ولكنها أيضًا أداة لتحسين التعليم من خلال تقديم تغذية راجعة مستمرة وتعديل استراتيجيات التدريس وفقًا للبيانات التي يتم جمعها.

أنواع التقويم حسب الفترة الزمنية

التقويم التشخيصي: يهدف إلى تحديد نقاط القوة والضعف لدى المتعلمين، والفجوات في معرفتهم ومهاراتهم، يتم تطبيقه قبل بداية الوحدة أو الدرس وعليه يمكن المعلمين تعديل خططهم التعليمية لتلبية احتياجات متعلميهم الفردية وضمان تحقيق النجاح.



التقويم التكويني: التقويم المستمر يتم تطبيقه أثناء سير الحصة لتقديم تغذية راجعة مستمرة لكل من المعلمين والمتعلمين بهدف تحسين عملية التعلم والأداء (المناقشة - الملاحظة - التقويم الصفّي - الواجبات).



التقويم الختامي: هو التقويم النهائي للجهد المبذول يتم تطبيقه في نهاية وحدة دراسية أو فصل دراسي أو سنة دراسية يهدف هذا التقويم إلى قياس مدى استيعاب المتعلمين للمهارات التعليمية المحددة وتقييم أدائهم الشامل.





استراتيجيات التقويم



1- استراتيجيات التقويم المعتمد على الأداء

التقويم المعتمد على الأداء يركز على تقييم قدرة المتعلمين على تطبيق المعرفة والمهارات في مواقف واقعية. وهو نوع يشجع على التعلم النشط والتفاعل ويعطي صورة دقيقة عن مدى تمكن المتعلمين من المواد الدراسية .

تشمل هذه الاستراتيجية ملاحظة أداء المتعلمين أثناء الأنشطة الصفية والمناقشات.

الملاحظات الصفية
والمناقشات

أ

تتضمن المشاريع تكليف المتعلمين بمهمة تتطلب منهم تطبيق المعرفة والمهارات التي تعلموها.

المشاريع

ب

يتضمن هذا النوع من التقويم إعداد المتعلمين وتقديمهم لعروض حول موضوعات معينة.

العروض التقديمية

ج

تتضمن تقديم مشكلة للمتعلمين لحلها باستخدام مهاراتهم ومعرفتهم.

حل المشكلات:

د



استراتيجيات التقويم



2- استراتيجية التقويم بالتواصل

هو نوع من التقويم يعتمد على الحوار والتفاعل بين المعلم والمتعلمين، أو بين المتعلمين أنفسهم، كوسيلة لتقييم وفهم مدى استيعابهم للمفاهيم والمعارف.

- المناقشات الصفية



3- التقويم المعتمد على الملاحظة

تعد استراتيجية التقويم المعتمد على الملاحظة من أنواع التقويم النوعي الذي يدون فيه سلوك المتعلمين بهدف التعرف على اهتماماتهم وميولهم واتجاهاتهم وتفاعلهم مع بعضهم البعض، بقصد الحصول على معلومات تفيد في الحكم على أدائهم وفي تقويم مهاراتهم وأخلاقياتهم وطريقة التفكير التي ينتهجونها.

- تفعيل سجل المتابعة اليومية



استراتيجيات التقويم



4- استراتيجية مراجعة الذات

تقوم هذه الاستراتيجية على تحويل التعلم السابق إلى تعلم جديد، وذلك بتقييم ما تعلمه المتعلم من خلال تأمله الخبرة السابقة، وتحديد نقاط القوة والنقاط التي بحاجة إلى تحسين، وتحديد ما سيتم تعلمه لاحقاً. وتُعد هذه الاستراتيجية مكوناً أساسياً للتعلم الذاتي، و حل المشكلات مما يمكن المتعلم من تشخيص نقاط القوة في أدائه، وتحديد حاجاته وتقييم اتجاهاته.



5- استراتيجية تقويم الأقران

استراتيجية تعليمية يقوم فيها مجموعة من المتعلمين أو أقرانهم الذين يشاركون في العملية التعليمية بتقييم مهام ونشاطات زملائهم. وفقاً لمعايير الجودة التي يضعها المشرف أو المعلم.



توزيع درجات مادة الحاسوب للمرحلة الثانوية (التعليم العام)
للف (العاشر - الحادي عشر علمي - الحادي عشر أدبي - الثاني عشر علمي - الثاني عشر أدبي)

الدرجة الكلية للنهاية العام	الفترة الدراسية الثانية						الفترة الدراسية الأولى					
	الاجمالي	الاختبار	المشروع	توزيع درجة الأعمال			الاجمالي	الاختبار	المشروع	توزيع درجة الأعمال		
20		10	4	3	2	1		10	4	3	2	1
20=2/(20+20)	20	الاختبار العملي	المشروع الحاسوبي	التطبيق العملي لأوراق العمل	أنشطة صفية	الالتزام والتعاون وسلوك المتعلم	20	الاختبار العملي	المشروع الحاسوبي	التطبيق العملي لأوراق العمل	أنشطة صفية	الالتزام والتعاون وسلوك المتعلم

الاختبارات

07



الاختبارات التحصيلية:

الاختبارات التحصيلية: هي الأداة التي تستخدم في قياس المعرفة والفهم والمهارة لمادة دراسية معينة قياسا كميًا بهدف قياس نواتج التعلم لدى المتعلم في شكل اختبار، بحيث يراعي فيه قواعد القياس والتقنين من ثبات وصدق.

أغراض الاختبارات التحصيلية

1. التعرف على ما يمتلكه المتعلم من المهارات اللازمة لتعلم محتوى دراسي جديد.
2. الوقوف على مواطن القوة والضعف لدى المتعلم.
3. تصنيف المتعلمين في مجموعات وقياس مستوى تقدمهم بالمادة.
4. تقدير المستوى العام لتحصيل المتعلمين في مادة دراسية معينة.
5. تنشيط دافعية التعلم والنقل من صف لآخر ومنح الدرجات والشهادات.

كيفية التخطيط لبناء الاختبارات التحصيلية

إعداد نموذج الإجابة وتوزيع الدرجات		تحديد موضوع الاختبار	
وضع تعليمات الاختبار		تحديد أهداف الاختبار	
مراجعة وتدقيق الاختبار		تحديد الأوزان النسبية لكل هدف. (جدول المواصفات)	
إجراء وتنفيذ الاختبار		تحديد شكل الاختبار	
تصحيح الاختبار		صياغة فقرات الاختبار	
تحليل نتيجة الاختبار		ترتيب الأسئلة	
تقويم الاختبار			



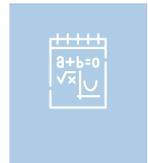
الميزان النسبي / جدول المواصفات للاختبار

هو جدول يوضع لبيان الوزن النسبي لكل موضوع بالنسبة للمقرر ومن ثم توزيع درجات الاختبار على المواضيع مقارنة بأوزانها النسبية.

وضع جدول المواصفات لضمان أن يكون الاختبار شاملاً لكل المواضيع ومتوازناً. فيعد بالشكل التالي:

- تحديد موضوعات المادة التي يراد قياس تحصيل المتعلم فيها.
- تحديد عدد الحصص اللازمة لتدريس كل موضوع.
- تحديد الوزن النسبي لموضوعات المادة الدراسية ويمكن الاستفادة بإعداده من خلال المعادلة التالية:

$$\text{الوزن النسبي لأهمية الموضوع} = \frac{\text{عدد الحصص اللازمة لتدريس الموضوع} \times 100}{\text{عدد الحصص اللازمة لتدريس المادة}}$$



خطوات إعداد الاختبار

- يتم تكليف كل معلم رسمياً بعمل اختبار معين وتحدد له فترة زمنية لتسليمه لرئيس القسم مع نموذج الإجابة حسب الميزان النسبي المتفق عليه.
- يتسلم رئيس القسم الاختبار ويراجعه ويغير ما هو ضروري.
- يعرض رئيس القسم الاختبار على الموجه الفني للاعتماد.
- يوضع الاختبار في ظرف مغلق ومختوم في مكان أمين بالمدرسة بمعرفة رئيس القسم.
- تسحب الأسئلة وتدبس وتوضع في أظرف مغلقة ومختومة في مكان أمين بالمدرسة تحت إشراف رئيس القسم.
- تحفظ الأسئلة.
- تتلف الأوراق الزائدة.



ملاحظات الواجب إتباعها عند وضع الأسئلة

- ينبغي دراسة محتوى المنهج وأهدافه دراسة وافية حتى تأتي الأسئلة محققة للأهداف التي وضع المنهج وأهدافه الواجب تحقيقها.
- يراعى في الأسئلة أن تكون شاملة لموضوعات المنهج وموزعة على أجزائه المختلفة.
- يراعى التنوع في ما تقيسه الأسئلة (مهارات) الحفظ، التذكر، التحليل، الاستنتاج.
- يراعى أن تكون الأسئلة السهلة والصعبة قليلة، أما أغلب الأسئلة متوسطة.
- ينبغي أن تكون الأسئلة واضحة العبارة خالية من اللبس والغموض.
- ينبغي أن تكون الأسئلة لها إجابة واحدة.
- توضع درجة كل بند في مربع واضح أمامه.

تصحيح الاختبار

هناك إرشادات تساعد المعلم على القيام بهذه المهمة على أسس علمية وبطريقة سريعة منها:

1. وجود نموذج للإجابات لأسئلة الاختبار، محدد فيها توزيع الدرجات على جزئيات السؤال.
2. مراعاة الدقة أثناء التصحيح.
3. الالتزام بنظام معين لتقدير الدرجات ليكون منصفاً في تقييم عمل المتعلمين.
4. التغلب على العامل الشخصي عند تقدير الدرجات وعدم توقف علامة المُتعلّم على من يصحح ورقته.
5. التركيز على إجابة المُتعلّم وليس على الخلفية السابقة لمستواه العلمي.
6. ضرورة توثيق ملفات اختبارات المعلمين بأخذ نسخ منها على وسيط تخزيني خارجي، ثم توضع نسخه منه على قرص في سجل الاختبارات بالقسم.
7. المراجعة العلمية والفنية لجميع أوراق المتعلمين بعد انتهاء التصحيح، ويقصد بالمراجعة إعادة تصحيح وليس فقط التأكد من جمع الدرجات.



الميزان النسبي المقترح: لدرجات الامتحان العملي للفترة الدراسية الأولى لمادة تقنية المعلومات - الصف العاشر

م	الأسبوع	المهارات	الدرجة
1	الأسبوع الأول الأمن السيبراني	● نظري	-
2	الأسبوع الثاني برمجة python مدخل إلى البرمجة	● فتح البرنامج ● فتح /حفظ الملف ● انشاء ملف بايثون جديد ● اعداد خطة الحل الخوارزمية Algorithm ● انشاء خريطة التدفق Flowchart	1
3	الأسبوع الثالث المتغيرات Variables	● الإعلان عن متغير ● دالة الإخراج print() ● العمليات الحسابية ● دالة input()	2
5	الأسبوع الرابع+الخامس الشروط Conditions	● عمليات المقارنة ● العمليات الشرطية if , elif ,else	2
6	الأسبوع السادس التكرار Loops	● الدالة while ● الدالة for ● الدالة Range	1
7	الأسبوع السابع + الثامن وحدة السلاسل النصية Strings	● انشاء سلسلة نصية ● الفهرسة [] Indexing ● القطع Slicing ● جمع السلاسل النصية ● تكرار عناصر السلسلة النصية ● دوال السلاسل النصية: ● طول السلسلة النصية len() ● Capitalize() ● Upper() ● Lower() ● Find() ● Replace() ● تعليمة f-string ● Int() ● Float() ● Center()	2

تابع -الميزان النسبي المقترح: لدرجات الامتحان العملي للفترة الدراسية الأولى
لمادة تقنية المعلومات - الصف العاشر

الدرجة	المهارات	الأسبوع	م
2	<ul style="list-style-type: none"> ● إنشاء القوائم ● العمليات على القوائم: ● فهرسة indexing ● تعديل عناصر القائمة ● إضافة وتحرير عناصر للقائمة ● حذف عناصر من القائمة (حسب الرقم-pop-remove) ● استخراج طول القائمة (len()) ● التأكد من وجود عنصر بالقائمة ● تعبئة قائمة من المستخدم ● طباعة عناصر القائمة ● عمليات الفرز على القوائم (حسب الأعداد العشرية, حسب السلاسل النصية) 	<p>الأسبوع التاسع+العاشر</p> <p>القوائم Lists</p> 	8
الإجمالي=10			

الخاتمة

نختتم هذا الدليل بتفاؤل وامل في ان يكون له الأثر الإيجابي على رحلتكم التعليمية في تدريس منهج الصف العاشر وأن يكون معيناً للمعلم أثناء تقديم المحتوى العلمي , فقد تم تصميم هذا الدليل لمساعدة المعلم في تقديم محتوى الامن السيبراني من خلال فيديوات معدة وجاهزة لاستخدام المعلم يمكن أن يستعين بها لتقديم المحتوى العلمي وعرضه للمتعلمين كوسيلة مساعدة أثناء الشرح , وهي غير ملزمة له , ويترك للمعلم حرية الاختيار في اعداد وتصميم ما يناسبه من فيديوات .

وندعو جميع المعلمين للاستفادة القصوى من الموارد والمعلومات المقدمة في هذا الدليل بهدف اثراء مهاراتهم وتعزيز معلوماتهم وخبراتهم , وتطلع بشغف الى رؤية تفاعلاتهم ومساهماتهم في هذا المجال . نحن ملتزمون بدعمكم في رحلتكم التعليمية التي ستستمر بفضل جهودكم المخلصة في تقديم الأفضل لمتعلميكم وحرصكم على استكشاف وتطوير مهاراتهم والاطلاع على كل ما هو جديد في مجال التكنولوجيا والبرمجة , حيث تحتم علينا ظروف العالم المتسارعة والتطور السريع في مجال مادة تقنية المعلومات الى ضرورة مواكبة كل ما هو جديد على الساحة .

نتطلع الى رؤية نتائجكم المذهلة في مجال البرمجة متمثلة في نتائج المتعلمين من مشاريع جيدة تعكس مدى استفادة المتعلمين من معلومات أثناء الفصل الدراسي , وندعوكم الى المشاركة النشطة في المسابقات والتحديات التقنية لتطوير المهارات البرمجية لكل من المعلم والمتعلم .

نتمنى لكم كل التوفيق في رحلتكم التعليمية والمهنية ونأمل أن يكون هذا الدليل بمثابة مورد ومرجع ملهم لكم عند الحاجة .