وزارة التربية
Ministry of Education
State of Kuwait | دولـــة الكويـت

دليل المعلم 10.1

الصـف العاشر ■ الفصـل الدراســي الأول ■ الجزء الأول

العام الدراسي2025-2026
الطـــبعة الأولـــى

دليل المعلم 10.1

الصـــف العـــاشر

إشـــراف

أ/ منى سـالم عـوض سـالم (رئيس اللجنة)

إعـــداد

أ/ ماجدة إبراهيم مزعل الدوسري

أ/ فاطـمه خلـيف غـنيم العنزي

أ/ إيمان عبد العزيز أحمد الفارسي

أ/ سـاره ياسـين عبدالله الامـير

لجنة المواءمة

أ/ منـى مـرزوق مـخلد العازمي

أ/ منار مصطفى عبدالحميد جمال

أ/ إبراهيم عبدالله إبراهيم المياس

إخـــراج

أ/ سنيه محـمد عـلي المـؤمن

المحـــــتوى

مقدمة 6

1 الاستراتيجية التعليمية 8

2 الوحدة الأولى: الأمن السيبراني Cyber Security 18

- وصف الوحدة : الأمن السيبراني Cyber Security 19
- خطة الدروس : الأمن السيبراني Cyber Security 20
- نتاجات التعلم : الأمن السيبراني Cyber Security 21
- مصطلحات حاسوبية : الأمن السيبراني Cyber Security 23
- استراتيجيات تعليمية مقترحة : الأمن السيبراني Cyber Security 25
- أوراق العمل : الأمن السيبراني Cyber Security 26

3 الوحدة الثانية: برمجة بايثون Python 27

وصف الوحدة : برمجة بايثون Python 28

الدرس الأول : مدخل إلى البرمجة Introduction to programming 29

- خطة الدروس: مدخل إلى البرمجة 29
- نتاجات التعلم : مدخل إلى البرمجة 30
- مصطلحات حاسوبية: مدخل إلى البرمجة 31
- ملاحظات للمعلم : مدخل إلى البرمجة 33
- أمثلة :مدخل إلى البرمجة 37
- أوراق العمل :مدخل إلى البرمجة 41

الدرس الثاني : المتغيرات Variables 44

- خطة الدروس: المتغيرات Variables 44
- نتاجات التعلم: المتغيرات Variables 45
- مصطلحات حاسوبية: المتغيرات Variables 46
- استراتيجيات تعليمية مقترحة : المتغيرات Variables 47
- أوراق العمل : المتغيرات Variables 48

الدرس الثالث : السلاسل النصية Strings 52

- خطة الدروس: السلاسل النصية Strings 53
- نتاجات التعلم: السلاسل النصية Strings 54
- مصطلحات حاسوبية: السلاسل النصية Strings 55
- استراتيجيات تعليمية مقترحة : السلاسل النصية Strings 56
- أوراق العمل: السلاسل النصية Strings 57

الدرس الرابع : الشروط Conditions 64

- خطة الدروس: الشروط Conditions 64
- نتاجات التعلم: الشروط Conditions 65

- مصطلحات حاسوبية: الشروط Conditions 66
- استراتيجيات تعليمية مقترحة : الشروط Conditions 66
- أمثلة: الشروط Conditions 67
- أوراق العمل: الشروط Conditions 73

الدرس الخامس : التكرار Loop 83
- خطة الدروس: التكرار Loop 83
- نتاجات التعلم: التكرار Loop 84
- مصطلحات حاسوبية: التكرار Loop 85
- استراتيجيات تعليمية مقترحة : التكرار Loop 86
- أمثلة : التكرار Loop 87
- أوراق العمل: التكرار Loop 92

الدرس السادس : تصحيح الأخطاء والاستثناءات Debugging and Exceptions 98
- خطة الدروس: تصحيح الأخطاء والاستثناءات 99
- نتاجات التعلم: تصحيح الأخطاء والاستثناءات 100
- مصطلحات حاسوبية: تصحيح الأخطاء والاستثناءات 101
- ملاحظات للمعلم : تصحيح الأخطاء والاستثناءات 102
- أمثلة : تصحيح الأخطاء والاستثناءات 103
- أوراق العمل : تصحيح الأخطاء والاستثناءات 114

4 الوحدة الثالثة: المنتجات الرقمية 121
- المنتجات الرقمية : المشروع 122
- نتاجات التعلم: المنتجات الرقمية 122
- مراحل إعداد المشروع 123
- المهارات المقترحة للمشروع ومتابعة إعداده 125
- أمور يجب مراعاتها في وحدة المنتجات الرقمية 127

5 تحضير الدروس 128
- خطوات إعداد الدرس 132
- نموذج تحضير درس مادة الحاسوب للصف العاشر 133

6 التقويم التربوي 137
- استراتيجيات التقويم 138
- توزيع درجات مادة الحاسوب 141

7 الاختبارات 142
- الاختبارات التحصيلية 143
- الميزان النسبي/جدول مواصفات الاختبار 144
- ميزان نسبي مقترح 146

الخاتمة 148

المقـــدمة

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 والبرمجة بلغة بايثون بصورة فعالة ومبتكرة،

يتضمن هذا الدليل استراتيجيات تعليمية متنوعة تتناسب مع الفئة العمرية للمتعلمين، مثل التعلم المباشر (المحاضرة)، التعلم التعاوني، المناقشة والحوار، التعلم القائم على حل المشكلات، والتعلم القائم على المشاريع ولعب الأدوار، 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. وبعض الاستراتيجيات المقترحة لتناول محتوى محدد.

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

أما بالنسبة للبرمجة بلغة بايثون، فإن الدليل يحتوي على موارد مفيدة لشرح المفاهيم البرمجية الأساسية، من خلال أمثلة وتطبيقات إضافية، استراتيجيات مقترحة، أدوات تقييم مختلفة قد تساعدك عزيزي المعلم في الحصة الدراسية.
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، والبرمجة بلغة بايثون. نأمل أن تجد في هذا الدليل كل ما تحتاجه لتحقيق أفضل النتائج في تدريس منهج الصف العاشر.

مع أطـيب التمـــنيات

01

الاستراتيجيات التعليمية

الإستراتيجيات التعليمية:

عـزيزي المعـــــلم

في ظل التطور السريع في مجال التعليم وتكنولوجيا المعلومات، أصبح من الضروري اعتماد إستراتيجيات تعليمية مبتكرة وحديثة ومصادر تعلم متنوعة لتلبية احتياجات المتعلمين وتحفيزهم على التعلم الفعّال، مثل التعلم التعاوني والتعلم القائم على حل المشكلات، بالإضافة إلى التعلم القائم على المشاريع والعصف الذهني وغيرها من الإستراتيجيات 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 الإستراتيجيات ؛ 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، و إستراتيجياتها 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:

الإستراتيجيات التعليمية هي الخطط والأساليب والتقنيات التي يستخدمها المعلمون لتسهيل وتحسين عملية التعلم والتعليم، تهدف هذه الإستراتيجيات إلى جعل التعليم أكثر فعالية وكفاءة وجاذبية للمتعلم، يمكن أن تتضمن الإستراتيجيات 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.

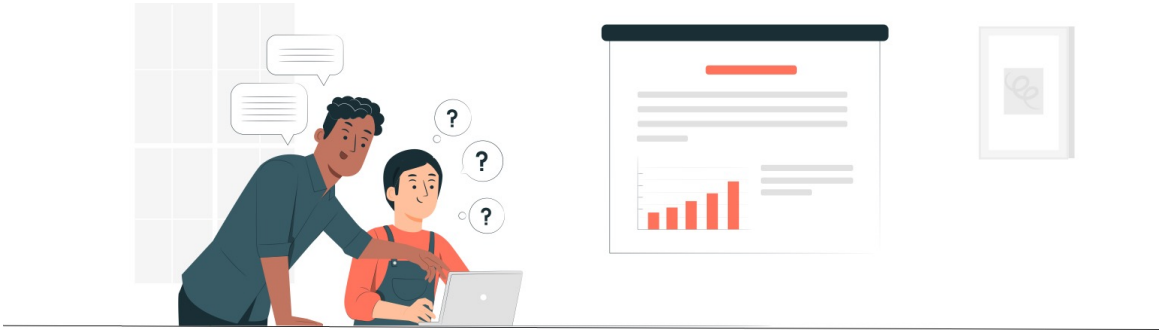
مثال:

تقديم المعلم أمثلة لبعض الوظائف المستحدثة في مجال الأمن السيبراني.

شرح مناطق نافذة محرر برنامج PyCharm.

الإستراتيجيات التعلـــيمية:

إستراتيجية المناقشة والحوار :



إستراتيجية المناقشة والحوار هي أسلوب تعليمي يهدف إلى تعزيز التعلم من خلال تبادل الأفكار والمعلومات بين المعلم والمتعلمين أو بين المتعلمين أنفسهم.

تعتبر هذه الإستراتيجية من الطرق الفاعلة التي تشجع على التفكير النقدي، وتطوير مهارات التواصل، وتعميق الفهم من خلال النقاشات المفتوحة والموجهة.

مثال :

طرح سؤال على المتعلمين

ما هي مميزات البرمجة بلغة البايثون؟ (لماذا اخترنا لغة البايثون من بين باقي اللغات البرمجية؟)

- 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.

الإستراتيجيات التعلـــيمية:

التعــلم التعــاوني:

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.

مثال:

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:

- إيجاد باقي القسمة.
- إدخال رقمين ثم طباعة ناتج الجمع.
- حساب مساحة الدائرة.

الإستراتيجيات التعلـــيمية:

التعلم القائم على حل المشكلات:

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

مثال:

برنامج تحويل وحدة قياس الزمن من الدقيقة إلى الثانية والعكس.

مشكلة البرنامج:

تحويل وحدة قياس الزمن من الدقيقة إلى الثانية والعكس.

الخوارزمية:

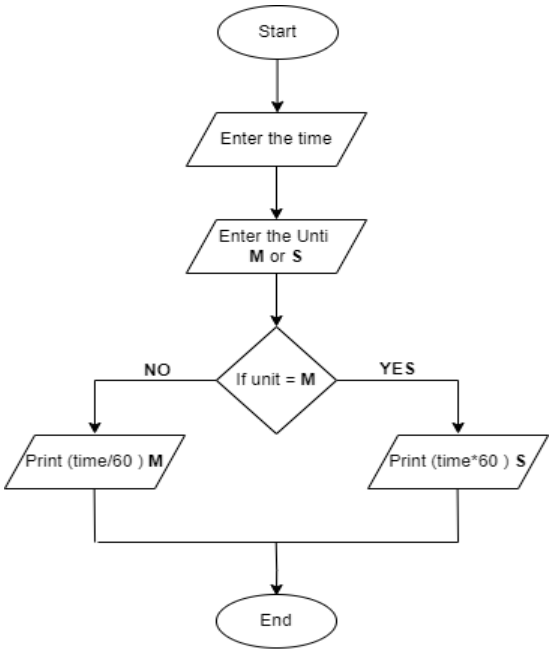
- استقبل من المستخدم قيمة الزمن.
- استقبل من المستخدم وحدة قياس الزمن المطلوب التحويل إليها سواء (دقيقة أو ثانية).
- استخدم المعادلة التالية عند استقبال "**M**" لتحويل وحدة قياس الزمن بالدقيقة إلى الثانية: S = (time * 60)
- استخدم المعادلة التالية عند استقبال رمز غير "**M**" لتحويل وحدة قياس الزمن بالثانية إلى الدقيقة:

  M = (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:

Code

Time.py

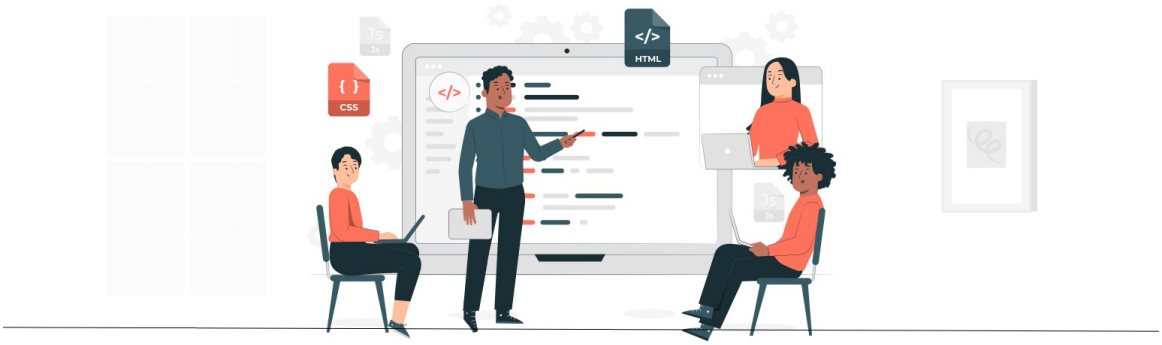
```
# Enter the time
Time = float(input("Enter the Time :  "))
# Enter The Time's Unit
unit = input ( "Enter the Time's Unit M for Minutes or S for Seconds : " )
if unit == "M":
    print (Time * 60 , "S")
else:
    print (Time /60 , "M")
```

Output

```
Run:  Time
C:\Users\ali\Desktop\python\daleel10-a1\.venv\Scripts\python.exe C
Enter the Time :  90
Enter the Time's Unit M for Minutes or S for Seconds : S
1.5 M
```

الإستراتيجيات التعلـــيمية:

العصــف الذهـــني:



العصف الذهني إستراتيجية تهدف إلى توليد أكبر عدد ممكن من الأفكار حول موضوع معين في بيئة جماعية في إطار وقت محدد، تدعو إلى الإبداع والتفكير الحر بين المتعلمين وتعويدهم على احترام الآراء المختلفة والاستفادة من أفكار الآخرين ومعلوماتهم، ودور المعلم هو التوجيه والإرشاد لتحقيق الأهداف المرجوة.

مثـــال: اكتب التعليمات البرمجية للحصول على الشكل التالي:

Code

```
# عدد الصفوف
n = 4

for i in range(1, n + 1):
  # طباعة نجمة واحدة متبوعة بمسافة
    print("* " * i)
```

Output

```
Run:  star
C:\Users\HP\PycharmProjects\pythonF
*
* *
* * *
* * * *
```

الإستراتيجيات التعلـــيمية:

التعلم القائم على المشـــاريع:

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

مـــثال:

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

التعـلم بالأقـران:

التعـلم بالأقـران إستراتيجية يكتسب فيها المتعلمين مهاراتهم ومعارفهم بمساعدة بعضهم البعض تحت إشراف المعلم، بحيث يكلف المعلم متعلم متفوق في مجال ما بدور الخبير، لتعليم متعلم آخر أقل مستوى منه أو مجموعة معينة ويقودهم نحو فهم الهدف أو إتقان المهارة.

مـــثال:

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

الإستراتيجيات التعلـــيمية:

إستراتيجـــية الصـف المقـــلوب:

إستراتيجية الصف المقلوب هي إستراتيجية تعليمية تعتمد على تزويد المتعلمين بمادة تعليمية حول الدرس كمقاطع فيديو أو ملفات صـوتية أو غيرها من الوسـائط، للاطلاع عليها في منازلهم أو في أي مكان آخر باسـتعمال حواسـيبهم أو هواتفهم الذكية أو أجهزتهم اللوحية قبل حضور الدرس. في حين يُخصص وقت الدرس للمناقشات والمشاريع والأنشطة.

- يتم تقديم مقاطع فيديو قصيرة تشرح المتغيرات وأنواع البيانات المختلفة في لغة Python.
- يُطلب من المتعلمين قراءة مصادر نصية توضح كيفية استخدام المتغيرات وتعديل قيمتها في لغة 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 تُستخدم فيه المتغيرات وتُطبق المفاهيم التي تعلموها.
- يقدم المعلم التوجيه أثناء العمل، ويُجيب على الأسئلة.
- يتم توجيه المتعلمين لتصحيح الأخطاء وتحسين أساليبهم في البرمجة.

مـــثال:

Code

```
# تعريف المتغيرات
age = 25
name = "Ahmed"
is_student = True
height = 175.5

# طباعة قيم المتغيرات
print("Name:", name)
print("Age:", age)
print("Is student:", is_student)
print("Height:", height)
```

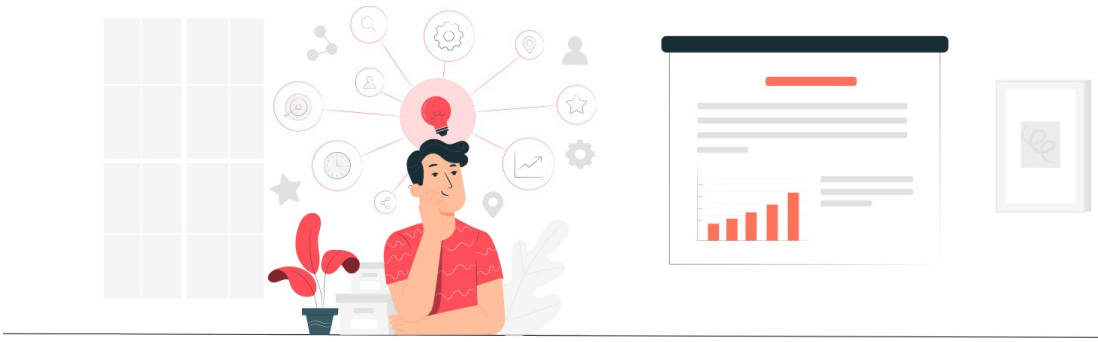
Output

```
Run:  num
Name: Ahmed
Age: 25
Is student: True
Height: 175.5

Process finished with exit code 0
```

الإستراتيجيات التعلـــيمية:

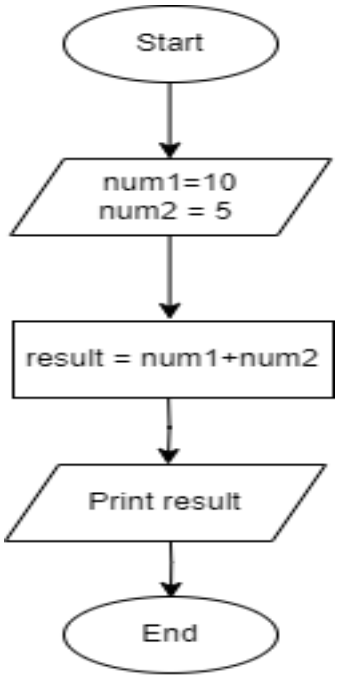
إستراتيجية الخرائط المفاهيمية:



الخرائط المفاهيمية هي إستراتيجية تعليمية فاعلة في تمثيل المعرفة عن طريق أشكال تخطيطية تربط المفاهيم مع بعضها البعض بخطوط أو أسهم يُكتب عليها كلمات تُسمى كلمات الربط لاكتشاف العلاقات بين المفاهيم وتعميق الفهم وتلخيص المعلومات وتقويم الدرس.

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

خريطة التدفق :



التعليمات البرمجية :

```
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
```

Output

```
Run:  sum
C:\Users\HP\PycharmProjects\pythonProject4\.venv\Scripts\python.exe
الناتج هو: 15

Process finished with exit code 0
```

02

الوحـدة الأولــى:

الأمن السيبراني Cyber Security

وصـف الوحـدة: الأمن السيبراني Cyber Security

في هذه الوحدة، سنتعرف على المفاهيم الأساسية للأمن السيبراني وأمن المعلومات وأهميتها في العصر الرقمي في حماية الأنظمة والمعلومات، وسنتعلم كيفية تطبيق المصادقة الثنائية واستخدم تقنيات الحماية والتشفير لتعزيز أمن البيانات، وسنميز بين الأدوار والمسؤوليات المختلفة ضمن مجالات العمل في وظائف الأمن السيبراني، وسنستكشف التحديات المختلفة التي تواجه الأمن السيبراني وكيفية التعامل معها بفعالية، مما يساعدنا على فهم الدور الحاسم للأمن السيبراني 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 و البرامج 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 التسلل ومنع التسلل.

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

	مفـــاهيم الأمـــن الســـــيبراني.	حصة دراسية	Cyber Security1
	المحاور الرئيسية لأمن المعلومات والأمن السيبراني CIA.	حصة دراسية	Cyber Security2
	الإجراءات الآمنة للمحاور الرئيسة للأمن السيبراني CIA.	حصة دراسية	Cyber Security3
	ممـــارسات الأمـــن الســـــيبراني.	حصة دراسية	Cyber Security4
	الوظـــــائف في مجال الأمـــن الســـــيبراني.		
	الجوانب الأخلاقية الرئيسية في مجال الأمـــن الســـــيبراني.		

امســـح الرمـــز

نتاجات التعلم: الوحدة الأولى الأمن السيبراني Cyber Security

1	الفهم الشامل لمفهوم وأهمية الأمن السيبراني وأمن المعلومات.
2	التمييز بين مفهومي الأمن السيبراني وأمن المعلومات.
3	التعرف على عناصر نموذج CIA (السرية، السلامة، التوفر) ودوره في حماية الأنظمة والمعلومات.
4	التعريف بمفهوم السرية لحماية المعلومات التي لا يمكن الوصول إليها إلا من قبل الأفراد المصرح لهم فقط.
5	التعريف بمفهوم النزاهة وموثوقية المعلومات الرقمية وسلامتها.
6	التعريف بمفهوم التوفر وإتاحة المعلومات والموارد باستمرار عند الحاجة.
7	التزود بالمعرفة والمهارات اللازمة لحماية الأنظمة والشبكات والمعلومات ضد الهجمات السيبرانية.
8	التعرف على أمثلة توضح الهجمات التي تستهدف المحاور الرئيسية لأمن المعلومات.
9	تطبيق مبادئ CIA على مثال ماكينة الصراف الآلي.
10	توظيف أداة الشبكة الافتراضية الخاصة VPN لضمان الخصوصية عبر الإنترنت.
11	تطبيق المراسلة الآمنة من خلال التشفير من طرف إلى طرف لحماية محادثات المستخدمين.
12	استخدام برامج إدارة كلمات المرور التي تساعد على تقديم الحماية والأمان لحسابات المستخدمين.
13	تطبيق المصادقة الثنائية وتقنيات الحماية والتشفير لتعزيز أمن البيانات.
14	التعرف على الخدمات الرقمية التي تتيح تشفير الرسائل الالكترونية.
15	تطبيق مثال على تشفير الرسائل الإلكترونية من خلال أداة Flow Crypt.
16	التعرف على خاصية التصفح المتخفي للحفاظ على مستوى أعلى من الخصوصية أثناء تصفح الانترنت.
17	التعرف على طرق الدفع الآمنة لحماية المعلومات المالية وتقليل مخاطر الوصول اليها.
18	التعرف على الأشكال المختلفة للمصادقة التي تستخدم لحماية حسابات المستخدمين.

تابع - نتاجات التعلم: الوحدة الأولى الأمن السيبراني Cyber Security

19	تطبيق بعض الإجراءات والتدابير الخاصة من أجل الحفاظ على CIA في الأمن السيبراني.
20	الاستعانة بإدارة مكافحة الجرائم الإلكترونية لتقديم البلاغات الخاصة بكل ما يخالف القانون لاتخاذ الإجراءات القانونية اللازمة حيالها.
21	التعرف على الوظائف المستحدثة في مجال الأمن السيبراني لتوسيع مدارك المتعلم عن الفرص الوظيفية المستحدثة المتاحة.
22	التمييز بين الأدوار والمسؤوليات المختلفة ضمن مجالات العمل في وظائف الأمن السيبراني.
23	تعزيز الوعي الأمني لمخاطر الهجمات السيبرانية عند المتعلمين.
24	التعرف على الجوانب الأخلاقية عند التعامل مع البيانات الرقمية والمعلومات الحساسة.

مصطلحات حاسوبية – Key Words

نموذج CIA

السرية(Confidentiality)

النزاهة (Integrity)

التوفر(Availability)

أمن المعلومات

Information Security

الأمن السيبراني

Cyber Security

المصادقة

Authentication

الشبكة الافتراضية الخاصة

Virtual Private NetworK (VPN)

تطبيقات المراسلة الآمنة (التشفير)

Secure Messaging Applications (Encryption)

برامج ادارة كلمات المرور

Management Programs Password

تشفير البريد الإلكتروني

Encryption for Email

التصفح المتخفي

Incognito

تابع - مصطلحات حاسوبية — Key Words

طرق الدفع الآمنة

Secure Payment Methods

المحافظ الرقمية

Digital Wallets

ممارسات الأمن السيبراني

Cybersecurity Practices

التصيد الاحتيالي

Phishing

إدارة عملية مشاركة البيانات

Data Sharing Management

الإبلاغ

Reporting

محلل الأمن السيبراني

Cybersecurity Analyst

مهندس أمن

Security Engineer

مستشار أمني

Security Consultant

المخترق الأخلاقي

Ethical HacKer

مهندس تصميم أنظمة أمنية

Architect Security

المستجيب للحوادث

Incident Responder

محلل مركز العمليات الأمنية

Security Operations Center Analyst

(SOC Analyst)

استراتيجيات تعلـــيمية مقترحة : الأمن السيبراني .

استخدام استراتيجية المناقشة والحوار : لاستقطاب انتباه المتعلمين وربط موضوع الأمن السيبراني بموقف حياتي: هل تعرض أحدكم أو أحد معارفكم للاختراق من قبل المخترقين في الواتس آب او الإنستغرام أو أي من برامج التواصل الاجتماعية الأخرى الخاصة بكم؟ حدثنا عنها.

تهدف الاستراتيجية اثارة المواضيع التالي:

- الإجراءات الأمنية مثل المصادقة والشبكات الافتراضية (VPN) وتشفير البيانات وغيرها.
- الممارسات الأمنية السيبراني مثل كلمات السر القوية والمصادقة وحماية الأجهزة وغيرها.
- استخدام التكنولوجيا المسؤول.
- التواصل بأمان.

استخدام استراتيجية التعلم التعاوني : بتقسيم الفصل لمجموعات ثم عرض نماذج للممارسات الخاطئة في استخدام التكنولوجيا وتكليف المجموعات بتصنيفها إذا كانت آمنه أم لا، مثل عرض مجموعة من المواقع أو الإيميلات واختبار المجموعات إذا كان بإمكانها اكتشاف الآمنة منها من الخطيرة.

أوراق العمل : الوحدة الأولى الأمن السيبرانيCyber Security

أوراق عمل
(1-2-3) اللاصفية

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

03

الوحـدة الثانية:

برمجة بايثون Python

وصف الوحدة: برمجة بايثون Python.

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

في هذه الوحدة، "برمجة بايثون"، 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 التعليمات البرمجية، ثم 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 التعليمات البرمجية بكفاءة، بالإضافة إلى ذلك، سنستكشف أنواع هياكل البيانات في Python، 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.

الدرس الأول : مدخل إلى البرمجة Introduction to programming

خطة الدروس: مدخل إلى البرمجة

1	• مدخل الى البرمجة: مفهوم البرمجة. • خطوات انشاء برنامج لغة البرمجة(Python) . • كتابة خوارزمية لحل مشكلة. • تصميم خريطة تدفق لإجراء حل مشكله. • استخدام محرر الأوامر (PyCharm).	حصة دراسية	Python1
2	• مدخل الى البرمجة: التعرف على الدالة المدمجة(print). • التعرف على هياكل البيانات. • معرفة أنواع هياكل البيانات في بايثون.	حصة دراسية	Python2

امسح الرمز

نتاجات التعلم : مدخل إلى البرمجة.

1	التعرف على مفهوم البرمجة.
2	التعرف على أهمية البرمجة في عالم التكنولوجيا.
3	تحديد المراحل والإجراءات الأساسية لإنشاء برنامج حاسوبي.
4	التعرف على مفهوم الخوارزمية وكيفية كتابتها.
5	التعرف على خرائط التدفق (FlowChart) وكيفية تصميمها.
6	التمييز بين أشكال ووظائف المخططات الرسومية.
7	التعرف على مفهوم لغة البرمجة (Python) .
8	التعرف على مجالات استخدام لغة (Python).
9	التمييز بين Compiler وInterpreter.
10	التعرف على محرر الأوامر (PyCharm) لكتابة وتنفيذ التعليمات البرمجية.
11	التعرف على مكونات نافذة محرر الأوامر (PyCharm).
12	إنشاء مشروع (Python) جديد.
13	إنشاء ملف (Python) جديد.
14	استخدام الدالة المدمجة (print) لطباعة المخرجات.
15	التعرف على أنواع هياكل البيانات المختلفة في (Python).

مصطلحات حاسوبية: مدخل إلى البرمجة.

البرمجة

Programming

الخوارزمية

Algorithm

خرائط التدفق

Flowchart

عملية إدخال البيانات

Input

عملية إخراج البيانات

Output

عملية المعالجة

Process

اتخاذ قرار

Decision

لغة البرمجة

Programming Language

المترجم

Compiler

المفسر

Interpreter

محرر الأوامر

PyCharm

شجرة المشروع والملفات

Project

ملف بايثون

Python file

محرر التعليمات البرمجية Code Editor

تابع - مصطلحات حاسوبية: مدخل إلى البرمجة.

الأمر

Run

القوائم الرئيسية

Main Menu

هياكل البيانات

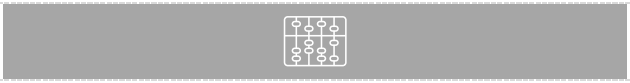
Data structure

هياكل البيانات الأولية

Primitive Data Structures

الأعداد الصحيحة

Integer



الأعداد العشرية

Float

السلسلة النصية

String

البيانات المنطقية

Boolean

هياكل البيانات غير الأولية

Non-primitive Data Structures

هياكل البيانات الخطية

Linear Data Structures

القائمة

List

المجموعة

Set

القاموس

Dictionary

الصف

Tuple

ملاحظات للمعلم:

شـرح واجهة المسـتخدم :عند شـرح نافذة المحرر يجب التطرق الى مكونات واجهة المسـتخدم في PyCharm بما في ذلك نافذة المشـروع، نافذة الإخراج (console) ، نافذة التوجيه، ونوافذ الأدوات الأخرى.

حفظ الملف: التأكيد على خطوات حفظ الملف وامتداده .py ، مثلاً pract1.py.

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

إغلاق المشـروع : التأكيد على أهمية اغلاق المشـروع لكل متعلم بعد انتهاء جلسـة العمل لضـمان استمرار كل متعلم في تنفيذ أوراق العمل الخاصة به فقط.

تابع- ملاحظات للمعلم:

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

1	اختيار الأمر settings من قائمة file.
2	تحديد الاختيار Editor من القائمة الجانبية لـ appearance and behavior من نافذة Settings .
3	اختيار الأمر الفرعي Font من Editor ثم تحديد الخيارات المطلوبة .
4	تطبيق التغييرات بالضغط على Apply .

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

1	اختيار الأمر settings من قائمة file.
2	تحديد الاختيار Appearance من القائمة الجانبية لـ appearance and behavior من نافذة Settings.
3	يتم اختيار الـ theme المرغوب.
4	تطبيق التغييرات بالضغط على Apply .

تابع- ملاحظات للمعلم:

عزيزي المعلم، في المراحل التعليمية السابقة تعرف المتعلمين على مبادئ التفكير المنطقي وتصميم خرائط التدفق ، والآن حان الوقت لصقل هذه المبادئ وتوظيفها ، فالتخطيط الجيد والمنظم يعتبر من الممارسات والعادات السليمة للمبرمجين والتي تساعد المتعلم على تصور عملية سير الأوامر البرمجية بصورة مرئية، وهذا الشيء مهم جداً في ضوء افتقار محرر الأوامر (PyCharm) للواجهة الرسومية. فلابد من التركيز على مهارتين أساسيتين:

1. حل مشكلة ما باستخدام الخوارزميات وترجمتها لخريطة تدفق تدريجيا من البسيط إلى المركب.
2. تدريب المتعلمين على الكتابة على لوحة المفاتيح بصورة جيدة، فكلما زادت طلاقة الكتابة ازداد معها سرعة الحل مما يتيح مساحة أكثر لمزيد من التمارين.

نموذج مقترح لشرح الخوارزميات وخرائط التدفق:

الخطوة الأولى: استقطاب انتباه المتعلمين.
الهدف: استعراض مشكلة وتحليلها و إبداء حلول واقعية لها.

باستخدام استراتيجية المناقشة والحوار أو العصف الذهني، قم بتقديم مشكلة ضعف مجموعة من المتعلمين في العمليات الرياضية وتحديداً عمليات الضرب.
الهدف: الوصول إلى فكرة تطوير برنامج يساعد المتعلمين على الطلاقة الرياضية في عمليات الضرب البسيطة.

الخطوة الثانية: بناء الخوارزمية.
الهدف: بناء الخطوات المنطقية لحل المشكلة.

باستخدام استراتيجية التعلم التعاوني قم بتقسيم المتعلمين إلى مجموعات للبدء بمناقشة وكتابة الخطوات اللازمة التي يحتاجها المتعلم (المستخدم) لحل مسألة رياضية بسيطة واحدة (ملاحظة: يخصص وقت محدد للمجموعات لأداء المهمة).

يقوم المعلم بعرض ومناقشة الخوارزميات المقترحة منهم والتأكد من احتوائها على التالي:

1. عرض السؤال للمتعلم.
2. استقبال الإجابة من المتعلم.
3. اختبار الإجابة إذا كانت صحيحة أم خاطئة.
4. عرض النتيجة للمتعلم.

تابع- ملاحظات للمعلم:

الخطوة الثالثة: تصميم خريطة التدفق.

الهدف: تحويل خطوات الحل (الخوارزمية) لخريطة تدفق.

باستخدام استراتيجية المناقشة والحوار واستراتيجية التعليم المباشر قم بتحويل خوارزمية الحل إلى خريطة تدفق، ثم قم بتكليف كل فريق بتصميم خريطة التدفق الخاصة بهم (ملاحظة: يخصص وقت محدد للمجموعات لأداء المهمة).

التقويم:

استعرض مشكلة ما واطلب من المجموعات التالي:

1. تحليل المشكلة ووضع الحلول المناسبة.
2. تصميم خوارزمية الحل.
3. تصميم خريطة التدفق.

نموذج مقترح لتدريب المتعلمين على الكتابة باستخدام لوحة المفاتيح:

باستخدام استراتيجية التعلم التعاوني قم بتقسيم المتعلمين إلى مجموعات واطلب منهم - باستخدام محرر الأوامر (PyCharm) - تصميم برنامج يعرض للمستخدم عند التشغيل الجدول المدرسي الخاص في الفصل كما يلي:

D/P	1	2	3	4	5	6	7
Sunday	subject	subject	subject	subject	subject	subject	Subject
Monday	subject	subject	subject	subject	subject	subject	Subject
Tuesday	subject	subject	subject	subject	subject	subject	Subject
Wednesday	subject	subject	subject	subject	subject	subject	Subject
Thursday	subject	subject	subject	subject	subject	subject	Subject

ولخلق روح المنافسة، حوّل التمرين إلى مسابقة بين المجموعات وخصص لوحة ثابتة في الفصل تستعرض كل الفصول و أسماء المتعلمين الذين ينتمون للمجموعة الفائزة في المسابقة.

قوانين المسابقة:

1. يقوم كل عضو في الفريق بتصميم مشروعه الخاص.
2. يمنع الكتابة عن أي عضو و إلا تُستبعد المجموعة.
3. تعتبر المجموعة قد نفذت المهمة عندما ينتهي جميع الأعضاء من تصميم مشاريعهم.
4. ترتيب المجموعات الفائزة يكون حسب سرعة إنجاز المهمة.

حل أمثلة: مدخل إلى البرمجة.

مثال (1)

المشكلة: **عمل برنامج بسيط يحسب مساحة المستطيل.**

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

1. مقدمة عن العمليات الحسابية.
2. مفهوم المدخلات والمخرجات: حيث إنه في أي برنامج يطلب من المستخدم المدخلات (الأرقام ونوع العملية) ويعطي المخرجات (النتائج).
3. مفهوم المعالجة: وهي الخطوات أو العمليات التي يجريها البرنامج على البيانات المدخلة للحصول على النتيجة.

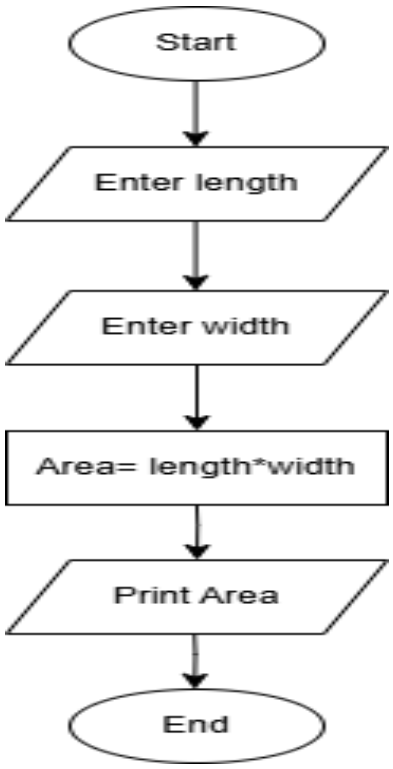
شرح خطوات الخوارزمية:

1. ادخال الرقم الأول (طول المستطيل) من قبل المستخدم.
2. ادخال الرقم الثاني (عرض المستطيل) من قبل المستخدم.
3. حساب (مساحة المستطيل) = الطول * العرض .
4. عرض النتيجة على الشاشة.

تابع - حل أمثلة: مدخل إلى البرمجة.

شرح خريطة التدفق (Flowchart) :

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



البداية (Start)	←	بداية البرنامج.
ادخال المدخلات	←	- ادخال الرقم الأول (طول المستطيل) من قبل المستخدم. - ادخال الرقم الثاني (عرض المستطيل) من قبل المستخدم.
المعالجة	←	حساب مساحة المستطيل
المخرجات	←	عرض ناتج مساحة المستطيل على الشاشة.
النهاية (End)	←	نهاية البرنامج

ترجمة الخريطة وتحويلها إلى تعليمات برمجية يتبعها تشغيل و اختبار البرنامج.

تابع - حل أمثلة: مدخل إلى البرمجة.

مثـــال (2)

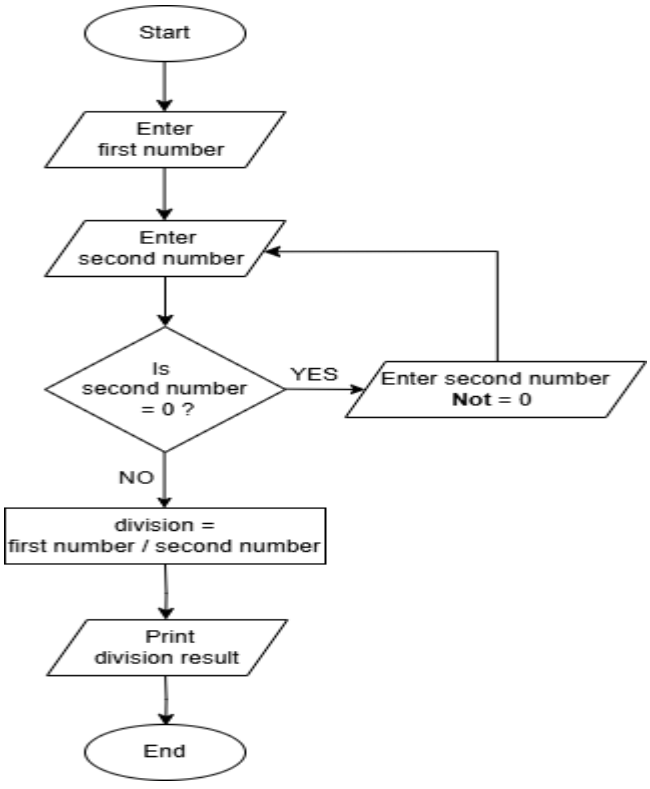
المشكلة:

كتابة برنامج بسيط يحسب ناتج القسمة لرقمين مع التأكيد على ألا يكون الرقم المقسوم عليه يساوي صفر.

الخوارزمية:

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

شرح خريطة التدفق (Flowchart) :



البداية (Start)	←	بداية البرنامج.
ادخال المدخلات	←	- ادخال الرقم الأول(البسط) من قبل المستخدم. - ادخال الرقم الثاني (المقام) من قبل المستخدم.
اتخاذ قرار	←	التحقق من المقام
	↳	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

تابع - حل أمثلة: مدخل إلى البرمجة.

مثـــال (3)

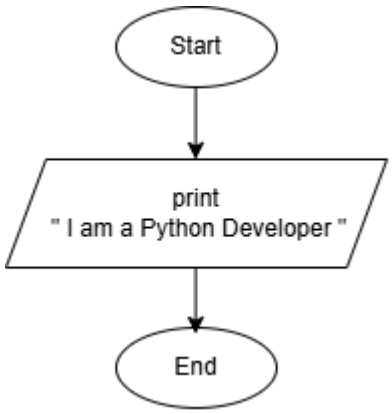
المشكلة:

كتابة برنامج بسيط لطباعة جملة "I am a Python Developer" على الشاشة.

الخوارزمية:

1. بدء البرنامج.
2. استخدام الدالة print () لطباعة النص "I am a Python Developer" على الشاشة.
3. إنهاء البرنامج.

خريطة التدفق (Flowchart):



التعليمات البرمجية:

Code

```
hello.py
1  #طباعة الجملة على الشاشة
2  print ( "I am a Python Developer")
```

Output

```
Run:  hello
C:\Users\ali\Desktop\python\dale
I am a Python Developer
```

أوراق العمل: مدخل إلى البرمجة.

حل ورقة عمل (4): برنامج حساب مساحة الدائرة.

المشكلة:

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

المدخلات: نصف القطر r.

المعالجة: حساب مساحة الدائرة باستخدام المعادلة.

مساحة الدائرة = πr^2 حيث $\pi = 3.14$

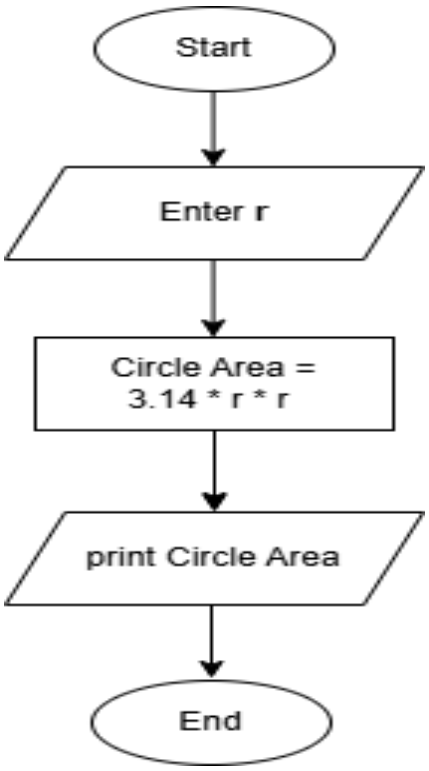
المخرجات: طباعة مساحة الدائرة.

الخوارزمية:

1. بداية البرنامج.
2. استقبال من المستخدم قيمة نصف قطر الدائرة(r) .
3. حساب مساحة الدائرة باستخدام المعادلة:

   مساحة الدائرة πr^2 حيث $\pi = 3.14$
4. عرض المساحة على الشاشة.
5. إنهاء البرنامج.

خريطة التدفق Flowchart :



تابع - أوراق العمل: مدخل إلى البرمجة.

حل ورقة عمل (5): برنامج تحويل وحدات قياس درجات الحرارة

المشكلة:

التحويل ما بين وحدات القياس المختلفة لدرجة الحرارة.

المدخلات: درجة الحرارة temp ووحدة قياس درجة الحرارة type.

المعالجة: اتخاذ القرار ثم:

- حساب درجة الحرارة باستخدام المعادلة التالية للتحويل درجة الحرارة من فهرنهايت الى سيليزية $C = (F - 32) \div 1.8$
- استخدام المعادلة التالية للتحويل درجة الحرارة من سيليزية الى فهرنهايت $F = (C \times 32) + (1.8$.

المخرجات: عرض درجة الحرارة بعد التحويل لوحدة القياس المطلوبة.

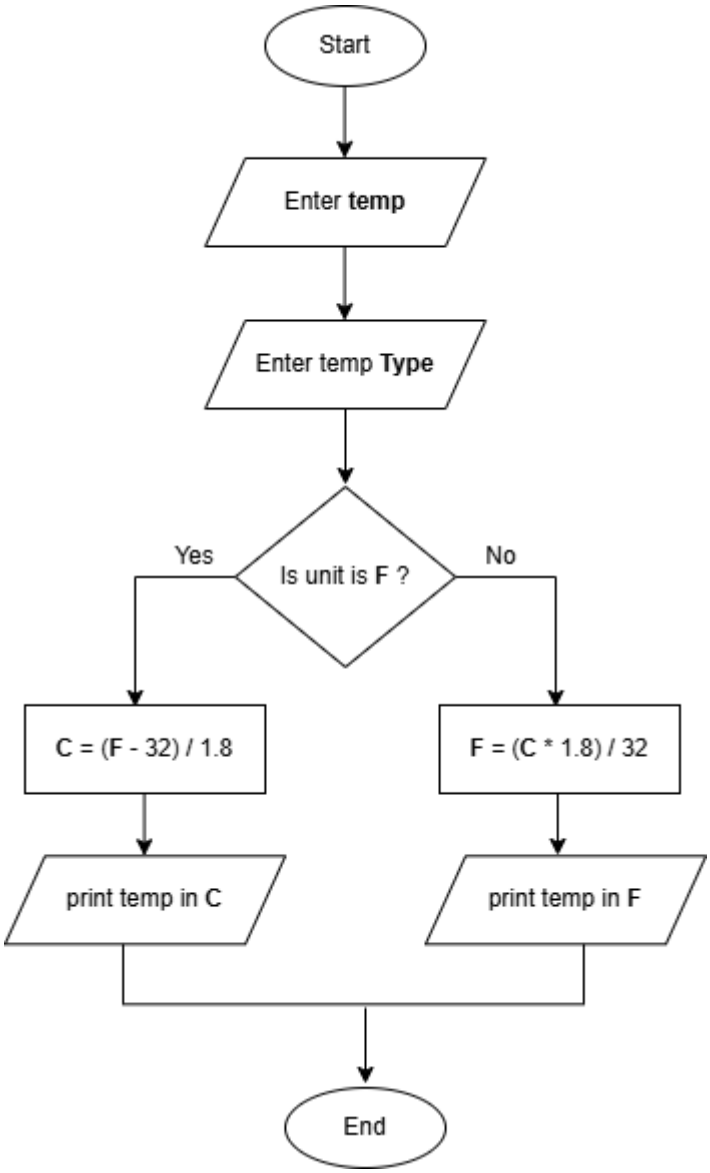
الخوارزمية:

1. بداية البرنامج.
2. استقبال من المستخدم درجة الحرارة الطقس.
3. استقبال من المستخدم نوع وحدة القياس. (سيليزي أو فهرنهايت).
4. استخدام المعادلة التالية للتحويل درجة الحرارة من فهرنهايت إلى سيليزية:

   $C = (F - 32) \div 1.8$
5. استخدام المعادلة التالية لغير ذلك:

   $F = (C \times 1.8) + 32$.
6. عرض درجة الحرارة بعد التحويل لوحدة القياس المطلوبة.
7. نهاية البرنامج.

خريطة التدفق Flowchart



مقترح أمثلة أخرى:

المشكلة:

التحويل ما بين وحدات القياس المختلفة للأطوال

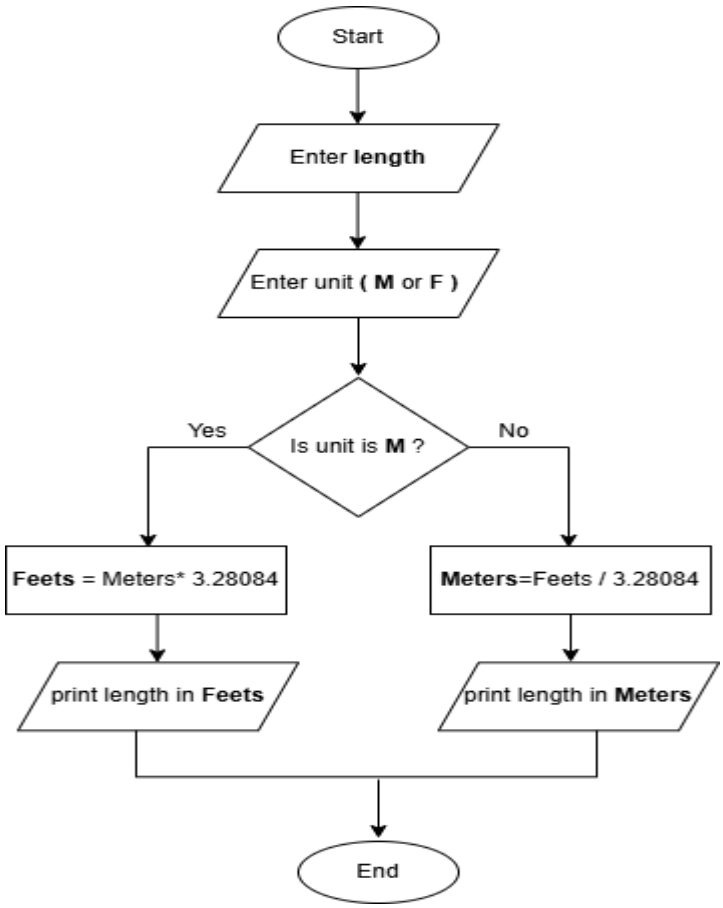
الخوارزمية

1. تحويل الأطوال بين الأمتار والقدم.
2. استقبل من المستخدم الطول.
3. استقبل من المستخدم نوع الطول المدخل.
4. استخدم المعادلة إذا كان الطول بالأمتار:

   Feet=Meters×3.28084
5. استخدم المعادلة إذا كان غير ذلك:

   Meters=Feet÷3.28084
6. اعرض الأطوال بعد التحويل لوحدة القياس المطلوبة.

خريطة التدفق Flowchart :



التعليمات البرمجية :

Code

```
# استقبل القيم من المستخدم
length = float (input("Enter the length: "))
unit =  input("Enter the unit F or M: ")

if unit == "M":
    feet = length * 3.28084
    print (feet)
else:
    meters = length/3.28084
    print (meters)
```

Output

```
"/Users/mirror/PycharmProjects/Grade 10/.ve
Enter the length: 10
Enter the unit F or M: F
3.047999902464003

Process finished with exit code 0
```

الدرس الثاني: المتغيرات Variables

البرمجة باستخدام لغة 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 أن 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 ومن المهم الحرص على تسمية المتغيرات بطريقة صحيحة وفقًا لقواعد التسمية في لغة البايثون.

سنتعلم في هذه الوحدة كيفية استخدام الدالة type() للتعرف على نوع المتغير ، والدالة print() لعرض المعلومات على الشاشة، ودالة input() لاستقبال المدخلات من المستخدم، سنتعرف أيضًا على رموز العمليات الحسابية في لغة البايثون وأولويات تنفيذها ، وأخيرا سنتعلم على كيفية تحويل البيانات باستخدام الدوال المناسبة مثل int(), float(), و str() حيث أن هذه المفاهيم الأساسية تشكل الأساس لكتابة برامج فعالة ومفيدة في لغة البايثون.

خطة الدروس: المتغيرات Variables.

	• مفهوم المتغيرات و الإعلان عن متغير. • تحديد أنواع المتغيرات. • قواعد تسمية المتغيرات. • استخدام المتغيرات. • استخدام دالة type(). • استخدام دالة print(). • استخدام دالة input(). • التحويل بين أنواع البيانات.	حصة دراسية	 Variables1

نتاجات التعلم: المتغيرات Variables.

1	التعرف على مفهوم المتغيرات.
2	التعرف على أنواع المتغيرات.
3	تخصيص قيمة للمتغير.
4	تسمية متغير بطريقة صحيحة مراعيا قواعد التسمية.
5	التعرف على وظيفة دالة type().
6	التعرف على وظيفة دالة print().
7	التعرف على رموز العمليات الحسابية في بايثون.
8	تمييز أولويات تنفيذ العمليات الحسابية.
9	التعرف على وظيفة دالة input ().
10	التحويل بين أنواع البيانات باستخدام الدوال.

مصطلحات حاسوبية: المتغيرات Variables.

متغير

Variable

الإعلان عن المتغيرات

Variable Declaration

دالة

Function

بيانات نصية

string

بيانات عددية صحيحة

Integer

بيانات عددية عشرية

Float

بيانات منطقية

Boolean

دالة الإخراج

print الطباعة

دالة الادخال

Input

دالة نوع المتغير

Type

استراتيجيات تعلـــيمية مقترحة : المتغيرات Variables .

استخدام استراتيجية المناقشة والحوار : لمناقشة كيف تعلم البرامج من منظور برمجي للتعرف على أساسيات البرمجة الرئيسية، فيمكن القول:

- أن البرامج (عادةً) تتكون من حلقات تكرارية.

مثال: عند تشغيل أي برنامج مثل محرر الأوامر (PyCharm) يضل البرنامج في حلقة تكرارية مستمرة لكي تضل الواجهة ظاهرة يمكن التفاعل معها.

- وللإنتقال من حلقة تكرارية إلى أخرى نحتاج إلى شرط.
- ولتكوين شرط (عادةً) نقارن بين متغير وقيمة ما باستخدام أحد علامات المقارنة.
- وللإعلان عن متغير يتوجب علينا معرفة أنواع المتغيرات.

تهدف الاستراتيجية التعليمية لاستعراض الخطوات التي سيتبعا المتعلم لتعلم أساسيات البرمجة وهي التعرف على:

- أنواع المتغيرات.
- كيفية الإعلان عن المتغيرات.
- كيفية إنشاء الشروط.
- كيفية إنشاء الحلقات التكرار.

باستخدام استراتيجية التعلم التعاوني : للتعرف على أنواع البيانات بتقسيم المتعلمين لمجموعات ثم عرض صورة وهمية للبطاقة المدنية وتكليفهم باستخراج البيانات وتدوين أنواعها.

أوراق العمل: المتغيرات Variables.

حل ورقة عمل (6): حساب تكلفة رحلة مدرسية لأبراج الكويت.

ورقة عمل (6)

المشكلة: حساب تكلفة رحلة مدرسية لأبراج الكويت بمعلومية تكلفة المواصلات، دخول أبراج الكويت والوجبة.

المدخلات: سعر دخول أبراج الكويت (ticKet) و سعر الوجبة (meal)و سعر المواصلات(transportation)

المعالجة: حساب إجمالي تكلفة الرحلة total_cost باستخدام المعادلة:

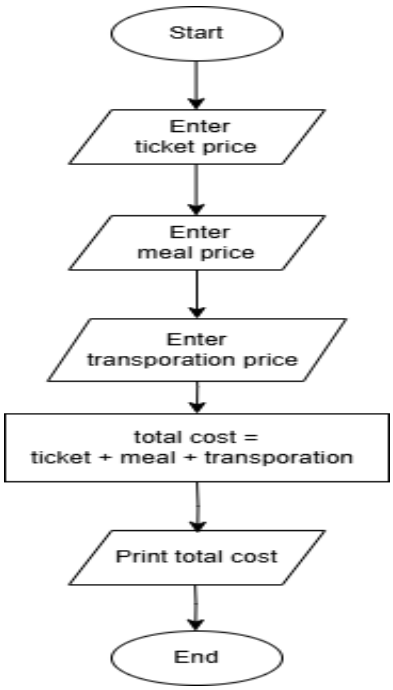
total_cost = ticKet + meal + transportation

المخرجات: عرض إجمالي تكلفة الرحلة total_cost.

الخوارزمية :

1. بداية البرنامج.
2. استقبال من المستخدم:
   أ. سعر دخول أبراج الكويت.
   ب. سعر الوجبة.
   ج. سعر المواصلات.
3. حساب إجمالي تكلفة الرحلة.
4. طباعة إجمالي تكلفة الرحلة.
5. نهاية البرنامج.

خريطة التدفق FlowChart :



التعليمات البرمجية:

Code

Kuwait_Tower_Trip.py

```python
# Enter ticket , meal , transportation price:

ticket_price = float(input("Enter ticket price: "))
meal_price = float(input("Enter meal price: "))
trans_price = float(input("Enter transportation price: "))

# Calculate the total cost
total_cost = ticket_price + meal_price + trans_price

# print the total cost on the screen
print (total_cost)
```

Output

Run Kuwait_Tower_Trip

```
"/Users/mirror/PycharmProjects/Grade 10/.venv/bin/python"
Enter ticket price: 3
Enter meal price: 1.500
Enter transportation price: 2
6.5

Process finished with exit code 0
```

تابع - أوراق العمل: المتغيرات Variables.

ورقة عمل (7): حساب الشغل المبذول

ورقة عمل (7)

المشـكلة: حساب الشغل المبذول بمعلومية القوى المؤثرة، والإزاحة في اتجاهها.

المدخلات: القوة المؤثرة Force بوحدة (النيوتن)، الإزاحة Displacement بوحدة (المتر)

(ملاحظة للمستخدم يجب أن تكون قيمة الإزاحة أكبر من الصفر).

المعالجة: حساب قيمة الشغل WorK باستخدام المعادلة:

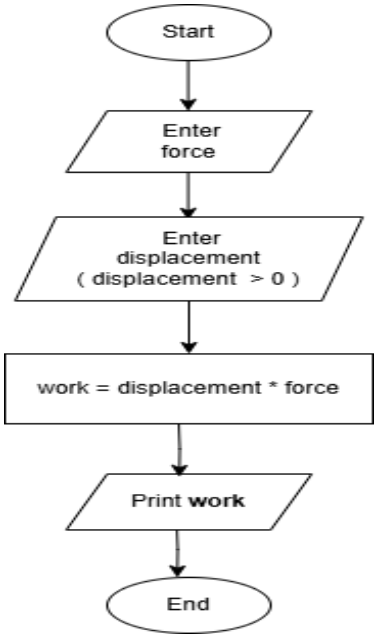
الشغل W = القوة F x الإزاحة D.

المخرجات: الشغل المبذول WorK بوحدة (الجول).

الخوارزمية:

1. بداية البرنامج.
2. استقبال من المستخدم القوة المؤثرة Force بوحدة (النيوتن).
3. استقبال من المستخدم الإزاحة Displacement بوحدة (المتر). (أكبر من الصفر).
4. حساب الشغل WorK بوحدة (الجول) باستخدام المعادلة $W = F \times D$.
5. طباعة الشغل WorK على الشاشة.
6. نهاية البرنامج.

خريطة التدفق FlowChart:



تابع حل : ورقة عمل (7):

التعليمات البرمجية:

Code

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

Output

```
"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:

الدالة المدمجة (input) ترجع دائماً قيمة نصية، فيتوجب تحويل هذه القيمة النصية إلى قيمة عددية لإجراء العمليات الحسابية.

الدرس الثالث: السلاسل النصية Strings.

السلاسل النصية في البرمجة تمثل تسلسلًا 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 مرنًا وسهلًا، يمكن الوصول إلى أحرف محددة داخل السلسلة النصية باستخدام الفهرسة (Indexing) والقطع (Slicing) لاستخراج أجزاء محددة من النص أيضا عمليات جمع السلاسل النصية (Concatenation)تسمح بدمج نصين أو أكثر و تكرار السلاسل النصية لأكثر من مره ، أيضًا مقارنة السلاسل النصية لمعرفة مدى تطابقها أو ترتيبها، توفر دوال مفيدة مثل len() لمعرفة طول السلسلة النصية، capitalize() لجعل الحرف الأول كبيرًا، upper() -lower() لتحويل النص إلى أحرف كبيرة أو صغيرة، find() للبحث عن نص معين داخل النص، replace() لاستبدال جزء من النص و لإمكانية تنسيق النصوص تستخدم f-string، بينما تسمح دالتي int() -float() 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 Strings.

1	• تعريف السلاسل النصية وخصائصها عمليات السلاسل النصية • الفهرسة indexing. • القطع Slicing. • جمع السلاسل النصية Concatenation. • تكرار السلاسل النصية . • المقارنة Comparison .	حصتان دراسية	Strings1
2	دوال السلاسل : • طول السلسة النصية len(). • capitalize (). • upper(). • lower(). • find(). • replace(). • f-string. • int(). • float(). • count(). • center().	حصتان دراسية	Strings2

امسح الرمز

نتاجات التعلم: السلاسل النصية Strings.

1. التعرف على مفهوم السلاسل النصية و خصائصها .
2. انشاء سلسلة نصية باستخدام علامات اقتباس فردية أو مزدوجة .
3. التعرف على العمليات الأساسية على السلاسل النصية .
4. الوصول إلى مواقع الأحرف في السلسلة النصية (الفهرسة) .
5. استخدام عملية التقطيع Slicing للحصول على جزء من السلسلة النصية للوصول إلى بيانات محددة.
6. جمع سلسلتين نصيتين للحصول على سلسلة نصية جديدة باستخدام معامل الجمع +.
7. تكرار السلاسل النصية باستخدام معامل الضرب *.
8. مقارنة السلاسل النصية باستخدام معامل المقارنة == , != .
9. التعرف على بعض الدوال الجاهزة لتنفيذ مهمة محددة عند استخدامها .
10. معرفة طول السلسلة النصية باستخدام دالة len().
11. التمييز بين الدوال المدمجة و الأساليب المدمجة .
12. ينشئ برامج بسيطة تتضمن مدخلات و مخرجات تعتمد على معالجة السلاسل النصية .

مصطلحات حاسوبية: السلاسل النصية Strings.

الفهرسة Indexing

جمع السلاسل النصية Concatenation

الأساليب المدمجة Built-in Methods

دالة Len

دالة Lower

دالة Find

دالة f-string

دالة Count

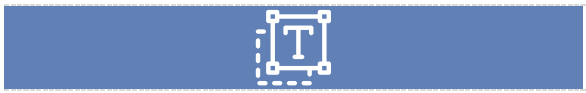
السلاسل النصية string

القطع Slicing

الدوال المدمجة Built-in Functions

المقارنة Comparison

دالة Upper



دالة Capitalize

دالة Replace

دالة Center

استراتيجيات تعلـــيمية مقترحة: السلاسل النصية Strings.

استخدام استراتيجية التعلم التعاوني : للتعرف على مفهوم الفهرسة والاستقطاع بحيث يقوم المعلم بتقسيم المتعلمين إلى مجموعات لهم فقرة محددة مثل : أبيات شعر ، أو سورة كريمة على أن تكون الحروف مرقمة (مفهرسة) ثم يقوم بتكليف كل مجموعة باستخراج كلمات محددة بتحديد رقم الحرف الأول والأخير، مثال:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14					
ق	ل		هـ	و		ا	لـ	ـلـ	ه		أ	حـ	د						
15	16	17	18	19	20	21	22	23	24	25									
ا	لـ	ـلـ	ه		ا	لـ	صـ	مـ	د										
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42			
لـ	م		يـ	لـ	د		و		لـ	م		يـ	و	لـ	د				
43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62
و		لـ	م		يـ	كـ	ن		لـ	ه		كـ	فـ	و	أ		أ	حـ	حـ

استخرج من السورة الكريمة ما يلي:

- كم عدد مرات تكرار لفظ الجلاله "اللّه" ؟ (شرح مفهوم count).
- ابحث عن أول كلمة "أحد" موضحاً موضع الكلمة بدلالة رقم أول حرف. (شرح مفهوم find).
- كم عدد حروف السورة الكريمة ؟ (شرح مفهوم len).

حل أوراق العمل: السلاسل النصية Strings.

ورقة عمل (8): تأمين كلمة المرور

ورقة عمل (8)

المشكلة:

عكس ترتيب الأحرف في السلسلة النصية لتصبح مؤمنة (كلمة السر).

المدخلات:

كلمة المرور (**password**)

المعالجة:

استخدام التقطيع **Slicing** لعكس الكلمة وتخزينها في (**reversed_password**).

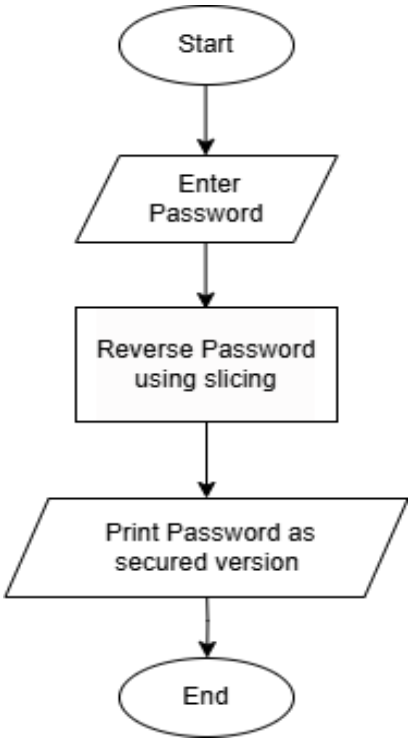
المخرجات:

طباعة الكلمة المعكوسة (**reversed_password**).

الخوارزمية:

1. بداية البرنامج.
2. طلب من المستخدم إدخال كلمة المرور (**password**).
3. تخزين الكلمة في متغير (**password**)
4. إنشاء متغيراً جديداً (**reversed_password**) لتخزين الكلمة المعكوسة.
5. استخدام التقطيع **Slicing** لعكس الكلمة [::-1].
6. طباعة الكلمة المعكوسة على أنها كلمة مرور محمية.
7. نهاية البرنامج.

خريطة التدفق FlowChart:



تابع حل ورقة عمل (8):

التعليمات البرمجية:

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

Code

reverse_string.py

```python
# Ask the user to enter the password
password = input("Enter Password: ")

# Store the reversed word using slicing
reversed_password = password[::-1]

# Print the reversed password as a secured version
print("Secured password: ", reversed_password)
```

Output

Run reverse_string

```
"/Users/mirror/PycharmProjects/Grade 10/.venv/bin/python" /
Enter Password: Kuwait
Secured password:  tiawuK

Process finished with exit code 0
```

ممكن إضافة حل آخر لنفس ورقة العمل بعد درس الloops.

حل أوراق العمل: السلاسل النصية Strings.

ورقة عمل (9) : تقطيع السلسلة النصية

ورقة عمل (9)

المشكلة:

استخراج الجزء الصحيح من السلسلة(دينار)، والجزء العشري (فلس) من مبلغ مالي عددي مُدخل من المستخدم.

المدخلات: المبلغ المالي **amount**.

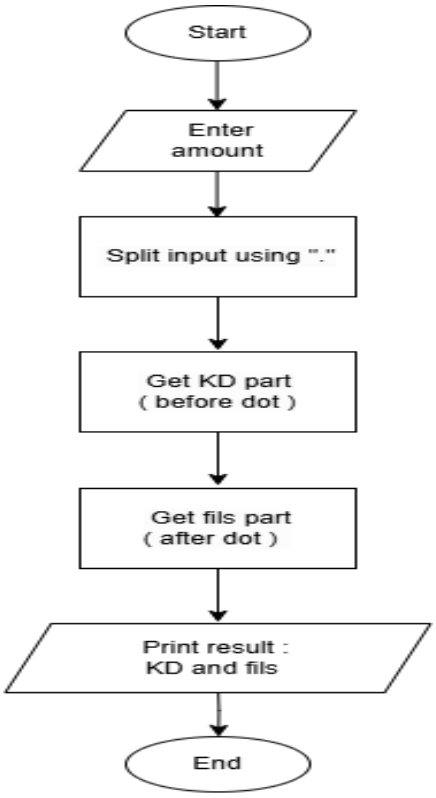
المعالجة: تحديد موضع الفاصلة العشرية للسلسلة النصية في المتغير **position**، وتحديد جزء قبل الفاصلة المبلغ بالدينار (**Kd**) ، تحديد جزء بعد الفاصلة المبلغ بالفلس (**fils**).

المخرجات: طباعة المبلغ بالتنسيق المطلوب (**KD** و **fils**).

الخوارزمية :

1. بداية البرنامج.
2. استقبال قيمة المبلغ المالي مثال (500.750) من المستخدم.
3. تحديد موضع الفاصلة العشرية للسلسلة النصية المُدخلة.
4. تحديد جزء قبل الفاصلة المبلغ بالدينار.
5. تحديد جزء بعد الفاصلة المبلغ بالفلس.
6. طباعة المبلغ بكتابة بالتنسيق التالي: **750 KD 500 and fils**.
7. نهاية البرنامج.

خريطة التدفق FlowChart :



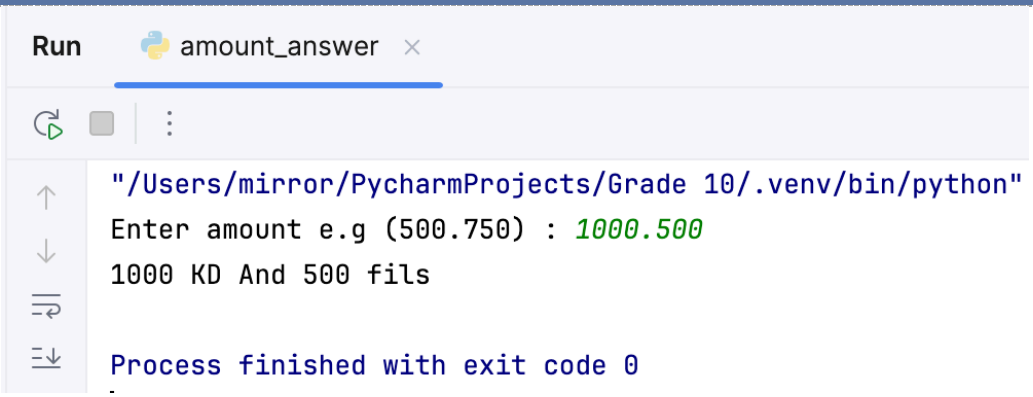
تابع حل : ورقة عمل (9):

التعليمات البرمجية:

Code

```
amount = input("Enter amount e.g (500.750) : ")
position= amount.split (".")
kd = position[0]
fils = position [-1]
print(f"{kd} KD And {fils} fils")
```

Output



تابع - حل أوراق العمل: السلاسل النصية Strings.

ورقة عمل (10) : تقطيع السلسلة النصية

ورقة عمل (10)

المشكلة:

تقسيم الاسم الكامل **full name** (باللغة الانجليزية) للحصول على الاسم الأول والأوسط والأخير.

المدخلات: الاسم الكامل (ثلاثي باللغة الانجليزية) **full_name**.

المعالجة:

- استخدام دالة **find()** لـ:
  - إيجاد موضع المسافة الأولى بين الاسم الأول والثاني وحفظه في المتغير **position_space1**.
  - إيجاد موضع المسافة الثانية بين الاسم الثاني والثالث وحفظه في المتغير **position_space2**.
- استخدام الفهرسة والتقطيع للحصول على الاسم الأول والثاني والثالث وحفظهم في المتغيرات **first_name** و **second_name** و **third_name**.

المخرجات:

- طباعة عدد أحرف الاسم الكامل باستخدام دالة **len()**.
- طباعة الاسم الأول **first_name** مع استخدام الدالة **title ()** لتحويل أول حرف من الكلمة إلى **Capital letter**.
- طباعة الاسم الثاني **second_name** مع استخدام الدالة **title ()** لتحويل أول حرف من الكلمة إلى **Capital letter**.
- طباعة الاسم الثالث **third_name** مع استخدام الدالة **title ()** لتحويل أول حرف من الكلمة إلى **Capital letter**.

تابع حل : ورقة عمل (10):

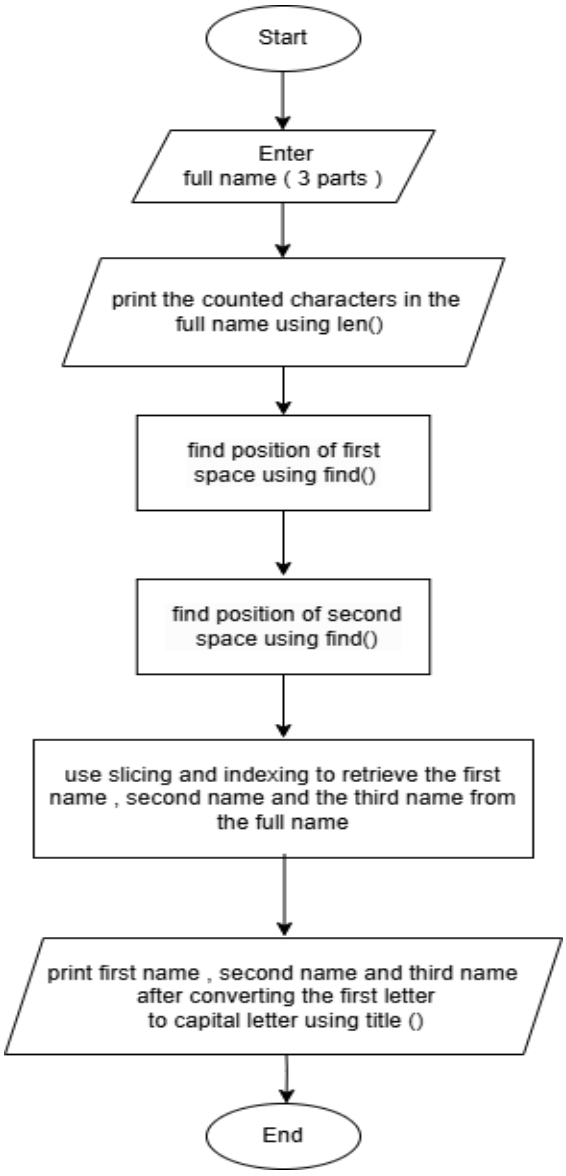
الخوارزمية :

1. بداية البرنامج.
2. استقبال الاسم الكامل (ثلاثي) باللغة الإنجليزية من المستخدم، وحفظه في متغير **full_name**.
3. طباعة عدد أحرف الاسم الكامل باستخدام دالة **len()**.
4. استخدام دالة **find()** لإيجاد موضع (**Position**) المسافة الأولى بين الاسم الأول والثاني، وحفظه في متغير **position_space1**.
5. استخدام دالة **find()** لإيجاد موضع (**Position**) المسافة الثانية بين الاسم الثاني والثالث، وحفظه في متغير **position_space2**.
6. استخدام الفهرسة والتقطيع للحصول على الاسم الأول ، الثاني، والثالث وحفظهم في متغيرات **first_name** و **second_name** و **third_name**.
7. طباعة:

الاسم الأول والاسم الثاني والاسم الثالث مع تحويل أول حرف فقط من كل جزء في الاسم الكامل إلى حرف كبير **Capital letter**.

8. نهاية البرنامج.

خريطة التدفق FlowChart :



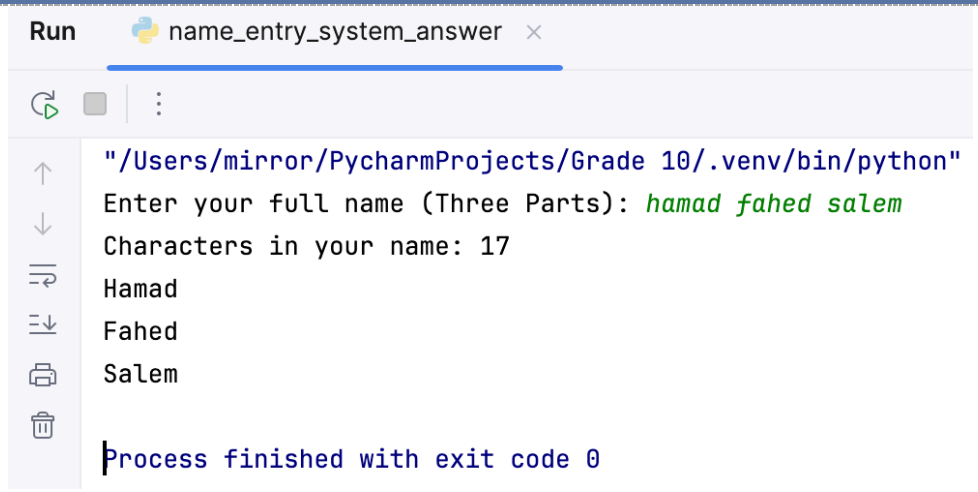
تابع حل : ورقة عمل (10):

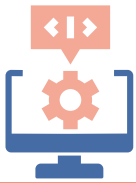
التعليمات البرمجية :

Code

```
full_name = input("Enter your full name (Three Parts): ")
print("Characters in your name:", len(full_name))                    # Step 3 in the algorithm.
position_space1 = full_name.find(" ")                                # Step 4 in the algorithm.
position_space2 = full_name.find(" ", position_space1 + 1)           # Step 5 in the algorithm.
first_name = full_name [:position_space1]                            # Step 6 in the algorithm.
second_name = full_name[position_space1 + 1:position_space2]         # Step 6 in the algorithm.
third_name =full_name [position_space2+1:]                           # Step 6 in the algorithm.
print(first_name.title())                                            # Step 7 in the algorithm.
print (second_name.title())
print (third_name.title())
```

Output





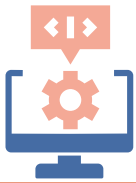
الدرس الرابع : الشروط Conditions.

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

خطة الدروس: الشروط Conditions.

	• الشروط : يميز بين عمليات المقارنة والعمليات المنطقية.	حصة دراسية	Conditions1
	• الشروط : يوظف العمليات الشرطية -if - else -elif لاتخاذ القرار.	حصة دراسية	Conditions2

امسح الرمز



نتاجات التعلم: الشروط Conditions.

1. التعرف على مفهوم الشروط و دورها في اتخاذ القرار و التحكم في مسار البرنامج .
2. التعرف على رموز عمليات المقارنة والعمليات المنطقية .
3. توظيف التعليمات الشرطية(else,if,elif) في كتابة برامج تتخذ قرارات مختلفة حسب المدخلات .
4. تصميم برامج تعتمد على شروط متعددة لتنفيذ أوامر مختلفة .
5. استخدام الشروط المنطقية المتداخلة في مواقف تتطلب أكثر من شرط .
6. التحقق من صحة الشروط و تنفيذ التصحيح اللازم عند ظهور نتائج غير متوقعة .
7. تقدير أهمية التفكير المنطقي في تصميم البرامج و اتخاذ القرارات البرمجية الصحيحة .

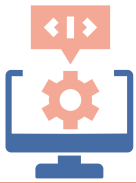
مصطلحات حاسوبية : الشروط Conditions

عمليات المقارنة Comparison Operators	الشرط Condition
التعليمات الشرطية Condition Statements	العمليات المنطقية Logical Operators

استراتيجيات تعلـــيمية مقترحة: الشروط Conditions

استخدام استراتيجية **العصف الذهني** : للتعرف على مفهوم الشروط وكيفية استخدامها، عزيزي المعلم ناقش مع المتعلمين فكرة تصميم برنامج يحسب الوقت الذي يستغرقه المتعلم للوصول إلى المدرسة إذا علمت أن المتعلم يستغرق 15 دقيقة من مغادرة المنزل إلى الوصول للمدرسة من دون أي عوائق قد تؤدي لتأخيره.

- قم بتوليد مجموعة من العوائق التي قد تؤدي لتأخير المتعلم مع إعطاء قيمة بالدقيقة لمقدار التأخير.
- مستعيناً بما سبق، قم بتصميم برنامج يطرح على المستخدم مجموعة من العوائق التي قد يتعرض لها المتعلم على أن تكون الإجابة بنعم أو لا، ثم إظهار الوقت الكلي الذي يستغرقه المتعلم للوصول إلى المدرسة بناءً على تلك الإجابات.



حل أمثلة: الشروط Conditions.

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

التعليمات البرمجية :

Code

```
x = 10
y = 5
print(x == y)   # False
print(x != y)   # True
print(x > y)    # True
print(x < y)    # False
print(x >= y)   # True
print(x <= y)   # False
```

Output

```
C:\Users\HP\PycharmProjects\
False
True
True
False
True
False
```

تابع - حل أمثلة: الشروط Conditions.

ما يجب على المعلم التأكيد عليه عند تدريس عمليات المقارنة.

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

تابع - حل أمثلة: الشروط Conditions.

ما يجب على المعلم التأكيد عليه عند تدريس العمليات المنطقية.

ثانياً:

التعرف على العمليات المنطقية(and, or) :

العمليات المنطقية تُستخدم للجمع بين شرطين أو أكثر و هذه العمليات تشمل:

- And: الناتج True إذا كانت كل الشروط صحيحة.
- Or: الناتج True إذا كان على الأقل واحد من الشروط صحيحًا.

```
1  a = 5
2  b = 6
3  c = 7
4  print(a < b and b < c)  # True
5  print(a > b or b < c)   # True
6  print(a > b and b < c)  # False
7  print(a > b or b > c)   # False
```

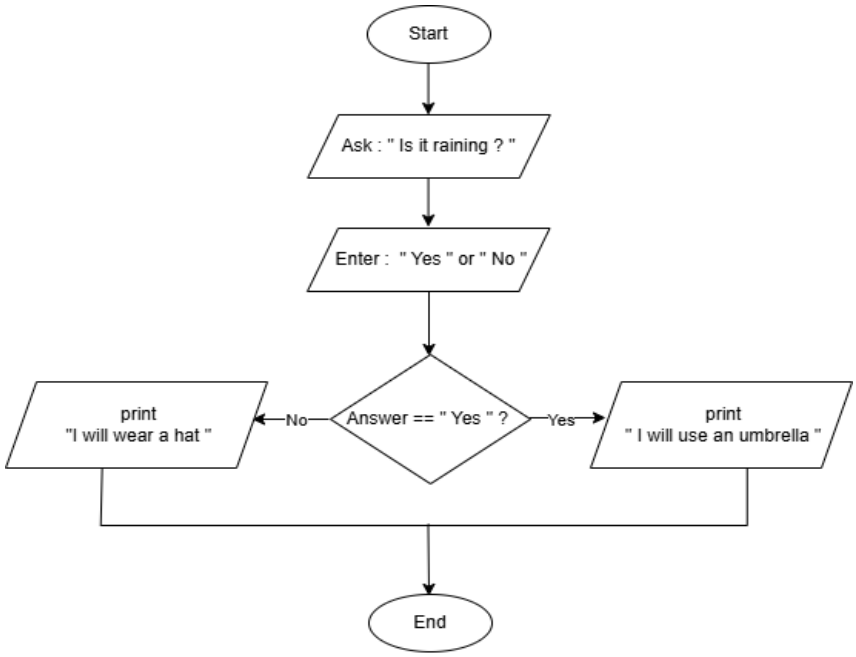
تابع - حل أمثلة: الشروط Conditions.

أمثلة لشرح الدوال الشرطية:

مثال توضيحي خارجي	في هذا المثال، يطلب من المستخدم إدخال قيمة لـ weather، ثم استخدام الدوال الشرطية if, else لتحديد الرسالة المناسبة بناءً على القيمة المدخلة. **المشكلة:** كتابة برنامج بسيط: إذا كانت السماء تمطر عندها سأستخدم المظلة وإلا سوف أرتدي قبعة. **المدخلات:** المتغير weather و قيمته yes أو no. **المعالجة:** اتخاذ قرار في حال كان weather مساوياً yes أو no. **المخرجات:** - طباعة "I will use an umbrella." إذا كان weather مساوياً yes. - طباعة "I will wear a hat." إذا كان weather مساوياً no. **كتابة الخوارزمية:** 1. بداية البرنامج. 2. سؤال المستخدم (is it raining? Please answer yes or no). 3. قراءة الإجابة: - اذا كانت الإجابة نعم (yes) اطبع (.I will use an umbrella). - اذا كانت الإجابة لا (No) اطبع (I will wear a hat.). 4. نهاية البرنامج.

تابع حل المثال :

خريطة التدفق FlowChart :



التعليمات البرمجية:

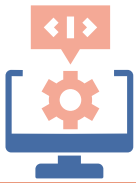
Code

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

Output

```
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
```



تابع - حل أمثلة: الشروط Conditions

حل مثال تطبيقي (15)

استكمل البرنامج الذي يظهر العدد الأقصى للإلكترونيات حسب رمز مستوى الطاقة بناء على خريطة التدفق الموضحة في كتاب المتعلم. ووضح مدخلات ومخرجات البرنامج.

التعليمات البرمجية :

Code

```
energylevel=input("Enter energy level code: " )
if energylevel=="K":
    print("Energy level K ->",2)
elif energylevel=="L":
    print("Energy level L ->", 8)
elif energylevel == "M":
    print("Energy level M ->",18)
elif energylevel == "N":
    print("Energy level N ->", 32)
else:
    print("Invalid input.")
```

Output

```
Run    energylevel_Answer
"/Users/mirror/PycharmProjects/Grade 10/.venv/bin/python"
Enter energy level code: N
Energy level N -> 32

Process finished with exit code 0
```

حل أوراق العمل: الشروط Conditions.

ورقة عمل(11) : برنامج حساب الخصومات على المنتجات

ورقة عمل (11)

المشكلة: تصميم برنامج يحسب التكلفة النهائية لشراء منتجات، وفقاً لقائمة الخصومات المعلنة.

المدخلات: السعر الإجمالي للمنتجات **total_price**.

المعالجة:

- التحقق من قيمة السعر الإجمالي وتخزين قيمة الخصم بناءً على سعر المنتج في المتغير **discount**.
- حساب السعر النهائي بعد تطبيق الخصم في المتغير **final_price** كما في المعادلة:

final_price = total_price — (total_price x discount)

المخرجات: طباعة السعر النهائي **final_price**.

الخوارزمية:

1. بداية البرنامج.
2. استقبال من المستخدم السعر الإجمالي للمنتجات.
3. التحقق من قيمة السعر الإجمالي:
   - إذا كان سعر المُنتج أكبر من 1000:
     - احسب الخصم بنسبة 25%.
   - إذا كان السعر الإجمالي يتراوح ما بين 500 و 1000:
     - احسب الخصم بنسبة 15%.
   - إذا كان السعر الإجمالي أقل من 500:
     - لا يوجد خصم.
4. حساب السعر النهائي بعد تطبيق الخصم.
5. طباعة السعر النهائي.
6. نهاية البرنامج.

تابع حل ورقة عمل (11) :

خريطة التدفق FlowChart:

تابع حل ورقة عمل (11) :

التعليمات البرمجية:

Code

calculate_final_price.py

```python
#Get total price from the user
total_price = float(input("Enter the total price of the products: "))

#Determine the discount percentage
if total_price > 1000:
    discount = 0.25
elif total_price >= 500:
    discount = 0.15
else:
    discount = 0.0

#Calculate the final price after discount
final_price = total_price - (total_price * discount)

# Print the final price
print("Final price after discount:", final_price)
```

Output

Run calculate_final_price

```
"/Users/mirror/PycharmProjects/Grade 10/.venv/bin/python"
Enter the total price of the products: 1500
Final price after discount: 1125.0

Process finished with exit code 0
```

تابع-حل أوراق العمل: الشروط Conditions.

ورقة عمل(12): برنامج لمعرفة حالة الرصيد الائتماني

ورقة عمل (12)

المشكلة: تصميم برنامج يتحقق من حالة رصيد حساب المستخدم.

المدخلات: استقبال قيمة الرصيد **balance** من المستخدم.

المعالجة: التحقق من قيمة الرصيد إذا كان مساوياً (صفر) أو أكبر من (صفر) أو أصغر من (صفر).

المخرجات:

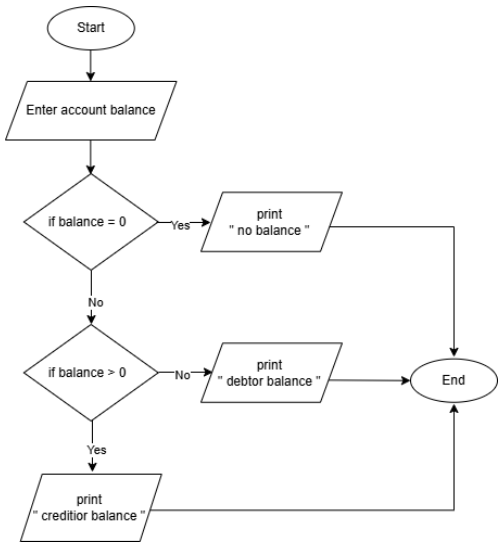
- طباعة (**no balance**) في حال كانت قيمة الرصيد مساوية (صفر).
- طباعة (**Creditor balance [balance]**) في حال كانت قيمة الرصيد أكبر من (صفر).
- طباعة (**Debtor balance [balance]**) في حال كانت قيمة الرصيد أصغر من (صفر).

الخوارزمية:

1. بداية البرنامج.
2. استقبال من المستخدم قيمة الرصيد كرقم عشري.
3. التحقق من قيمة الرصيد:
   - إذا كان الرصيد يساوي صفراً: طباعة "لا يوجد رصيد."
   - إذا كان الرصيد أكبر من صفر: طباعة "رصيد دائن [الرصيد]."
   - إذا كان الرصيد أقل من صفر:طباعة "رصيد مدين [الرصيد]."
4. نهاية البرنامج.

تابع حل ورقة عمل (12) :

خريطة التدفق FlowChart :



التعليمات البرمجية:

Code

checks_balance.py

```python
#Ask the user to enter their account balance
balance = float(input("Enter your account balance: "))

#Check the balance status
if balance == 0:
    print("No balance.")
elif balance > 0:
    print("Creditor balance: ", balance)
else:
    print("Debtor balance: ", balance)
```

Output

Run checks_balance

```
"/Users/mirror/PycharmProjects/Grade 10/.venv/bin/python"
Enter your account balance: 0
No balance.

Process finished with exit code 0
```

تابع-حل أوراق العمل: الشروط Conditions.

ورقة عمل (13): برنامج تصميم آلة حاسبة

ورقة عمل (13)

المشكلة: استخدام المعاملات الحسابية (+ ، - ، * ، /) في تنفيذ العمليات الحسابية (الجمع، الطرح، الضرب، القسمة).

المدخلات:

- استقبال من المستخدم العدد الأول والثاني.
- استقبال من المستخدم المعامل الحسابي المطلوب.

المعالجة: استخدام الشروط لتنفيذ العملية الحسابية المطلوبة.

المخرجات:

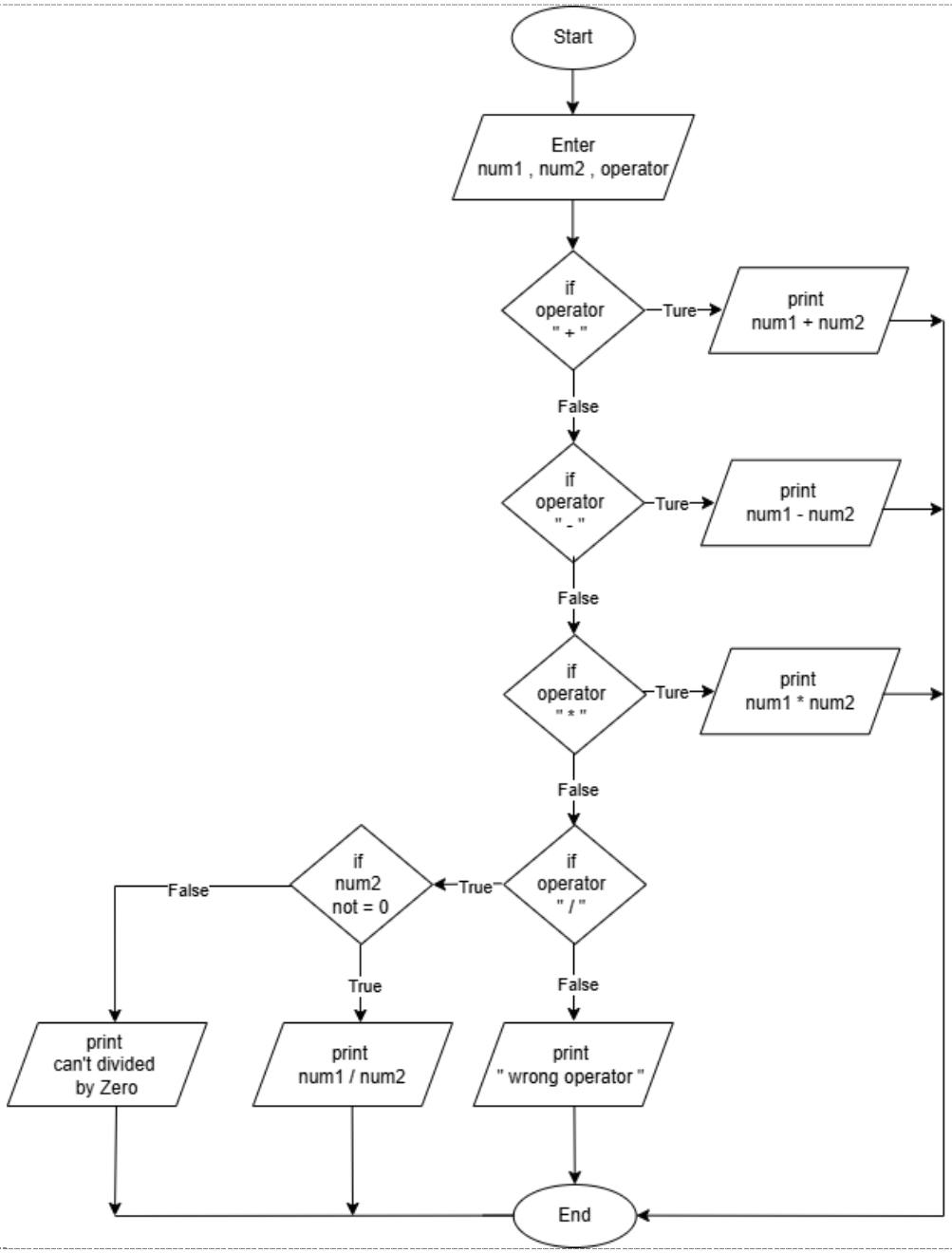
- طباعة ناتج العملية الحسابية المطلوبة.
- طباعة "Error Division by zero is not allowed". في حال كانت العملية الحسابية / والعدد الثاني مساوياً (صفر).
- طباعة "Invalid operator" في حال ادخال رمز مختلف عن المعاملات الحسابية (+ ، - ، * ، /).

الخوارزمية:

1. بداية البرنامج.
2. استقبال من المستخدم العدد الأول، ثم العدد الثاني.
3. استقبال من المستخدم المعامل الحسابي المطلوب.
4. استخدام الشروط Conditions:
   - إذا كان المعامل المُدخل +، يجمع العددين الأول والثاني.
   - إذا كان المعامل المُدخل -، يطرح العدد الثاني من العدد الأول.
   - إذا كان المعامل المُدخل *، يضرب العددين الأول والثاني.
   - إذا كان المعامل المُدخل /، يقسم العدد الأول من العدد الثاني. (التأكد من العدد الثاني لا يساوي صفر، وطباعة رسالة تفيد ذلك).
5. طباعة ناتج العملية الحسابية.
6. نهاية البرنامج.

تابع حل ورقة عمل (13):

خريطة التدفق FlowChart :



تابع حل ورقة عمل (13):

التعليمات البرمجية:

Code

Calculator_Answer.py

```python
#Conditions and math. operator

num1 = float(input("Enter First Number: "))
num2 = float(input("Enter Second Number: "))
operator = input("Enter Math. Operator(+, -, *, /): ")
if operator == '+':
    print(num1+ num2)
elif operator == '-':
    print (num1-num2)
elif operator == '*':
    print(num1 * num2)
elif operator == '/':
    if num2 != 0:
        print(num1/num2)
    else:
        print ("Error Division by zero is not allowed.")

else:
    print("Invalid operator")
```

Output

Run Calculator_Answer

```
"/Users/mirror/PycharmProjects/Grade 10/.venv/bin/python"
Enter First Number: 100
Enter Second Number: 2
Enter Math. Operator(+, -, *, /): /
50.0

Process finished with exit code 0
```

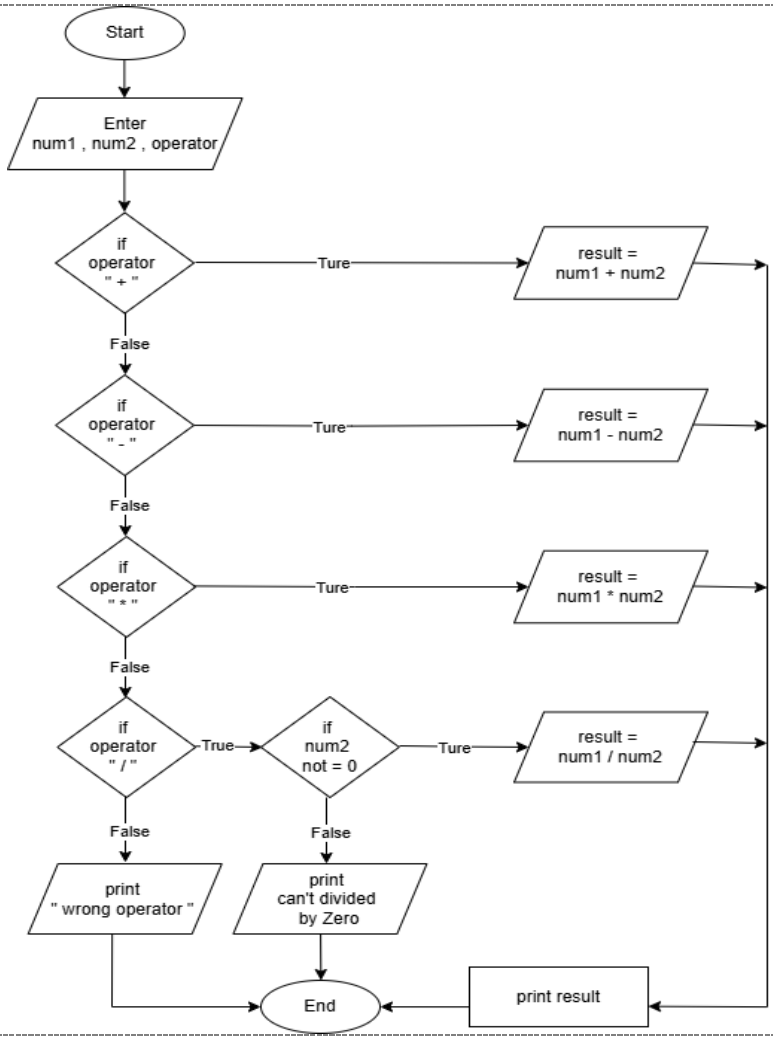
تابع-حل أوراق العمل: الشروط Conditions.

حـــــل آخـــر :

ورقــة عمـــل(13)
برنامج تصميم آلة حاسبة

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

خريطة التدفق FlowChart :



تابع – حل آخر لورقة العمل (13) :

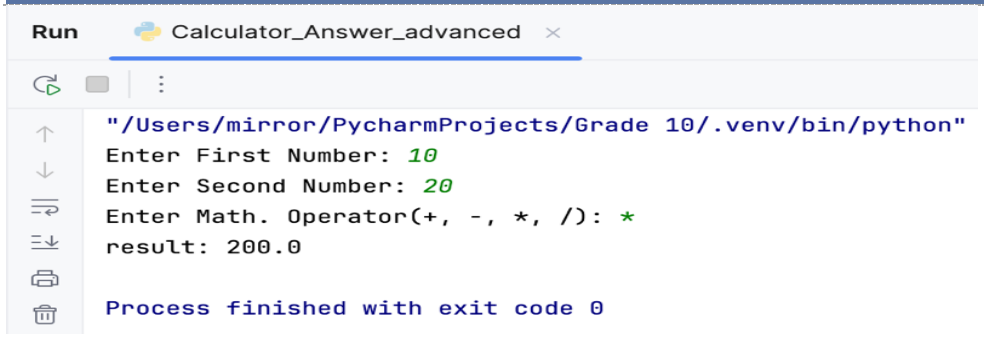
التعليمات البرمجية:

Code

```python
#Conditions and math. operator

num1 = float(input("Enter First Number: "))
num2 = float(input("Enter Second Number: "))
operator = input("Enter Math. Operator(+, -, *, /): ")
if operator == '+':
    result =num1+ num2
    print("result:",result)
elif operator == '-':
    result =num1- num2
    print("result:",result)
elif operator == '*':
    result =num1* num2
    print("result:",result)
elif operator == '/':
    if num2 != 0:
        result = num1/num2
        print("result:", result)
    else:
        print ("Error Division by zero is not allowed.")
```

Output



الدرس الخامس: التكرار Loop

تُعد الحلقات التكرارية أدوات قوية وفعّالة في البرمجة، وتتيح لك تكرار تنفيذ مجموعة من التعليمات بسهولة، باستخدام while، for 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.

	الحلقات التكرارية (while loop) — (while ... else).	حصة دراسية	Loop1
	الحلقة التكرارية (for loop).	حصة دراسية	Loop2
	الحلقات المتداخلة (Nested Loops).	حصة دراسية	Loop3
	تابع: الحلقات المتداخلة (Nested Loops) • دالة(breaK). • دالة (continue).	حصة دراسية	Loop4

نتاجات التعلم: التكرار Loop.

1	التعرف على مفهوم التكرار في البرمجة.
2	استخدام الحلقة التكرارية (while) لتنفيذ أوامر ما دامت الشروط متحققة.
3	استخدام الدالة (for) للتكرار على عناصر مثل القوائم والنطاقات الرقمية.
4	التعرف على استخدامات الدالة (Range).
5	ينفذ تكرارات متداخلة Nested Loops لحل مشكلات معقدة تتطلب أكثر من مستوى من التكرار.
6	استخدام الدالة (while true) لتكرار تنفيذ تعليمات برمجية الى مالا نهاية.
7	استخدام التعليمة البرمجية (BreaK) للخروج من الحلقة التكرارية عند تحقق شرط معين.
8	توظيف التعليمة (Continue) لتخطي تكرار معين دون إنهاء الحلقة بالكامل.
9	استخدام التعليمات البرمجية (Continue) و (BreaK) للتحكم في عملية التكرار .
10	يشرح ويستخدم البنية العامة (while...else) لفهم كيفية تنفيذ التعليمات بعد انتهاء التكرار.

مصطلحات حاسوبية: التكرار Loop.

التكرار Loop

الكائن Object

الحلقة التكرارية For

الحلقة التكرارية While

التكرارات اللانهائية Infinite Loop

التكرارات المتداخلة Nested Loops

القيمة الابتدائية Initial Value

قابل للتكرار Iterable

التعليمة البرمجية BreaK

التعليمة البرمجية Continue

الدالة Range

استراتيجيات تعلـــيمية مقترحة: التكرار Loop.

استخدام **استراتيجية المناقشة والحوار** للتعرف على مفهوم التكرار (while)، يمكن مناقشة النقاط التالية بعد استعراض جملة شرطية بسيطة مثل:

```
1    x=1
2    if x < 5:
3        print(x)
4    print("program ended")
```

- ما هو ناتج التعليمات البرمجية السابقة؟
- ما هو ناتج التعليمات البرمجية إذا كانت قيمة المتغير x تساوي 7 ؟
- كم عدد المرات الذي ستنفذ التعليمة البرمجية (print(x))؟
- ماذا لو استبدلنا (if) بـ (while) ماذا سيحدث؟
  - شرح وظيفة while
- لماذا لم يتم طباعة الرقم (1) بشكل مستمر دون توقف؟
  - الاشارة إلى أن الشرط يتحقق دائما.
- كيف نتغلب على هذه المشكلة؟
  - أن شرط التكرار ينتهي في مرحلة ما، مثل زيادة مقدار المتغير x بمقدار1، فينهي تحقق شرط التكرار عندما يكون قيمة المتغير 5.
  - استخدام التعليمة breaK.

استخدام **استراتيجية المناقشة والحوار** : للتعرف على مفهوم التكرار (for)، حيث يمكن استخدام كشف حضور الطلبة كمثال عملي، ناقش مع الطلبة ما هي الخطوات المتبعة لأخذ الغياب اليومي؟

- احضار دفتر الغياب الخاص في الفصل.
- قراءة الأسماء بالتسلسل.
- انتظار الرد من عبارة تبين وجوده في الفصل.
- احتساب المتعلم غياب في حين عدم وجوده.

يهدف التمرير بيان الفرق بين عملية التكرار (while) و (for) حيث أن التكرار (while) يستمر باستمرار تحقق الشرط، أما التكرار (for) يستخدم مجموعة من العناصر (مثل أسماء المتعلمين في كشف الغياب) لإجراء عملية التكرار.

حل أمثلة: التكرار Loop.

مثـــال (1) : استخدام الحلقة التكرارية while لطباعة الأعداد من 1 إلى 5 .

مثـــال (1)

المشكلة: استخدام الحلقة التكرارية while لطباعة الأعداد من 1 إلى 5.

التأكيد على المتعلمين عند استخدام الدالة while لا بد من شرط ينهي التكرار لتجنب الحلقات اللانهائية.

المدخلات: التأكيد على المتعلمين تحديد مدخلات البرنامج.

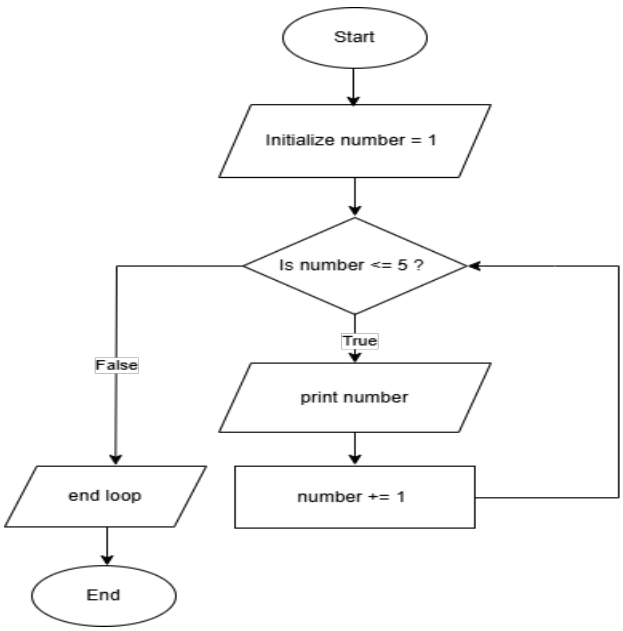
المخرجات: التأكيد على المتعلمين تحديد مخرجات البرنامج.

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 FlowChart :



تابع حل مثال (1) :

التعليمات البرمجية :

Code

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

Output

```
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 الحلقة التكرارية 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 :

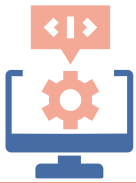
Code

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

Output

```
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
```



تابع - حل أمثلة: التكرار Loop.

مثـــال (7) : إدخال كلمة المرور و مطابقتها

مثـــال (7)

استخدام الحلقة التكرارية while.

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

التعليمات البرمجية :

Code

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

Output

```
"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 مثـــال (7) حل آخر باستخدام الحلقة التكرارية for

التعليمات البرمجية:

Code

loop.py

```
# الكلمة الصحيحة التي يجب على المستخدم إدخالها
correct_password = "1234"
# عدد المحاولات المسموح بها
for x in range(3):
    password = input(" enter password : ")
    if password == correct_password :
        print (" password is correct ")
    else :
        print (" password is incorrect ")

```

Output

Run: loop

```
enter password : 555
password is incorrect
enter password : 1234
password is correct
enter password : 127
password is incorrect

Process finished with exit code 0
```

تابع - حل أمثلة: التكرار Loop.

مثـــال (12) : برنامج يظهر رسالة خطأ عند محاولة إدخال المستخدم كلمة المرور ثلاث مرات بصورة خاطئة.

التعليمات البرمجية:

Code

attempt.py

```
# تحديد كلمة المرور الصحيحة
correct_password = "1234"
# عدد المحاولات المسموح بها
attempts = 3
# تهيئة عدد المحاولات
attempt_count = 0
#التحقق من مطابقة كلمة المرور
while attempt_count < attempts :
    #طلب ادخال كلمة المرور من المستخدم
    password = input(" enter password : ")
    #زياده عدد المحاولات
    attempt_count +=1
    # التحقق من كلمة المرور
    if password == correct_password :
        print (" password is correct ")
        break
else :
        print (" All Ateempts finish ")
```

Output

Run: attempt

```
C:\Users\ali\Desktop\python\daleel10-a1\.venv
 enter password : 112
 enter password : 668
 enter password : 920
 All Ateempts finish
```

حل أوراق العمل: التكرار Loop.

ورقة عمل (14) : برنامج تخمين عدد .

ورقة عمل (14)

المشكلة: تصميم برنامج يخمن المستخدم عددا صحيحا ثم يقارنه البرنامج مع العدد المعلن عنه مسبقا.

المدخلات: يستقبل من المستخدم عدد (player_number).

المعالجة:

1. الإعلان عن عدد (correct_number).
2. مقارنة العدد المدخل (player_number) بالعدد الصحيح (correct_number) المخزن في البرنامج وتكرار المقارنة حيث يتوقف عند تخمين المستخدم للعدد الصحيح.

المخرجات: طباعة العبارة المناسبة لكل حالة بعد المقارنة.

كتابة الخوارزمية:

- بداية قيمة مؤشر كتلة الجسم (BMI) بدقة خانتين عشريتين.
- وصف الحالة الصحية بناءً على قيمة الـBMI.
- رسائل الخطأ في حال وجود خطأ في الإدخال (مثلًا نص بدل رقم أو قسمة على صفر).

1. بداية البرنامج.
2. الاعلان عن متغير يساوي العدد المطلوب تخمينه.
3. استقبال عدد صحيح من المستخدم.
4. 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.
5. نهاية البرنامج

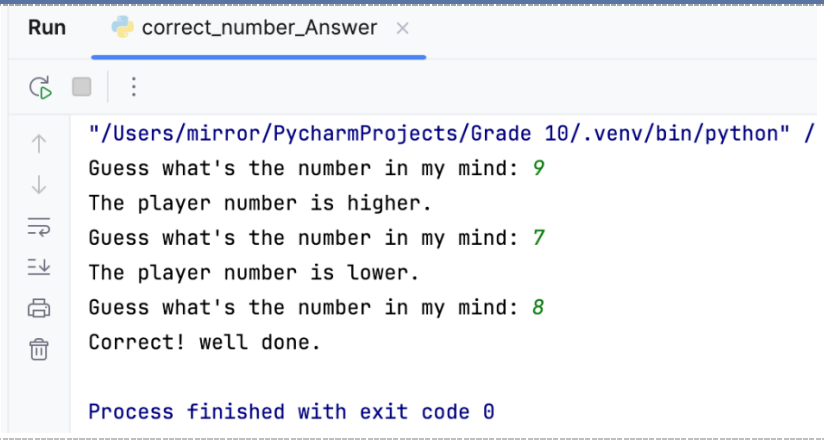
تابع حل ورقة العمل (14) :

التعليمات البرمجية :

Code

```
correct_number = 8
player_number = 0
while player_number!= correct_number:
    player_number = int (input("Guess what's the number in my mind: "))
    if player_number > correct_number:
        print("The player number is higher.")
    elif player_number < correct_number:  #if the number less
        print("The player number is lower.")
print("Correct! well done.")
```

Output



تطوير البرنامج لورقة عمل (14) :

إثرائي : تطوير البرنامج السابق (ورقة عمل 14) - لتوليد رقم عشوائي:

Code

```
# لتوليد رقم عشوائي من 1 - 30
import random
correct_number = random.randint(1,30)
# ----------------------
player_number = 0
while player_number!= correct_number:
    player_number = int (input("Enter a number between 1 and 30: "))
    if player_number > 30 or player_number <1:
        print("Please enter a number between 1 and 30")
        continue
    if player_number < correct_number:
        print("The player number is low, Try again.")
    elif player_number > correct_number:
        print("The player number is high, Try again.")
    else:
        print("Congratulations! You guessed the correct number.")
```

Out put

```
"/Users/mirror/PycharmProjects/Grade 10/.venv/bin/python"
Enter a number between 1 and 30: 15
The player number is high, Try again.
Enter a number between 1 and 30: 10
The player number is low, Try again.
Enter a number between 1 and 30: 13
The player number is low, Try again.
Enter a number between 1 and 30: 14
Congratulations! You guessed the correct number.

Process finished with exit code 0
```

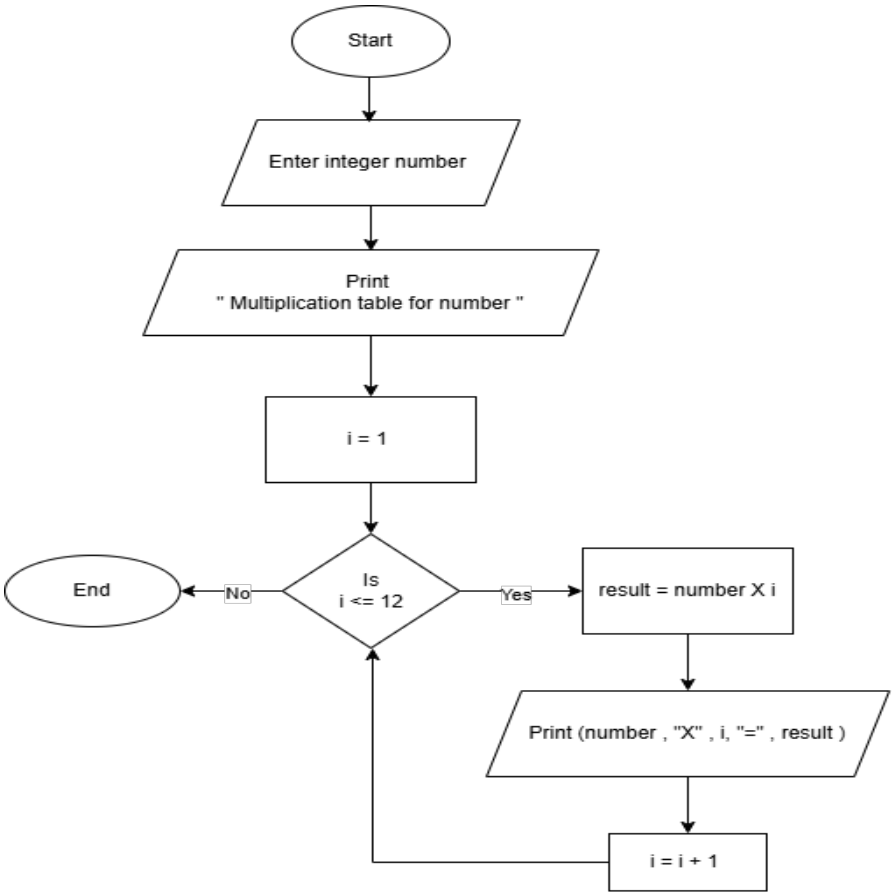
تابع حل أوراق العمل: التكرار Loop.

ورقة عمل (15): برنامج جدول الضرب.

ورقة عـــمل (15)	**المشكلة:** تصميم برنامج يطبع جدول الضرب لعدد محدد. **المدخلات:** استقبال العدد المطلوب طباعة جدول الضرب الخاص به. **المعالجة:** إنشاء حلقة تكرارية لحساب ناتج الضرب باستخدام المعادلة: الناتج= العدد المدخل * قيمة متغير التكرار. **المخرجات:** - طباعة رسالة (Multiplication table for) - طباعة جدول الضرب الخاص بالعدد المدخل. **كتابة الخوارزمية:** 1. بداية البرنامج. 2. استقبال من المستخدم العدد المطلوب طباعة جدول الضرب الخاص به. 3. طباعة رسالة (جدول ضرب العدد المُدخل). 4. إنشاء حلقة تكرارية لطباعة ضرب العدد المدخل في الأعداد من 1 الى 12. - كتابة التعليمات البرمجية لحساب: الناتج = العدد المدخل * قيمة متغير التكرار. - طباعة جدول الضرب. 5. نهاية البرنامج.

تابع حل ورقة عمل (15) :

خريطة التدفق FlowChart :



تابع حل ورقة عمل (15):

التعليمات البرمجية:

Code

multiplication_table.py

```python
#Ask the user to enter an integer number
number = int(input("Please enter an integer number: "))
#Print a message showing the table heading
print("Multiplication table for ", number)

#Use a loop to print the multiplication table from 1 to 12
for i in range(1, 13):
    result = number * i
    print(number, "x", i, "=", result)
```

Output

```
"/Users/mirror/PycharmProjects/Grade 10/.venv/bin/python"
Please enter an integer number: 6
Multiplication table for  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
```

الدرس السادس:
تصحيح الأخطاء والاستثناءات Debugging and Exceptions

تصحيح الأخطاء (Debugging) والاستثناءات (Exceptions) هما مفهومان أساسيان في البرمجة بلغة بايثون، حيث يمثلان جزءًا حيويًا من عملية تطوير البرامج. تصحيح الأخطاء هي عملية تحديد وإزالة الأخطاء (الأخطاء أو العيوب) من التعليمات البرمجية، بينما الاستثناءات هي أحداث غير متوقعة تحدث أثناء تنفيذ البرنامج وتتسبب في توقف البرنامج عن العمل بشكل طبيعي .

سنتعرف في هذا الدرس على الأخطاء الشائعة مثل : الأخطاء اللغوية (Syntax Errors)، وأخطاء وقت التشغيل (Runtime Errors)، وأخطاء منطقية (Logic Errors).

أيضا سنستخدم عبارات (try) و (except) للتعامل مع الاستثناءات، حيث يتم وضع التعليمات البرمجية التي من المحتمل أن تستثنى داخل كتلة (try)، ويتم كتابة التعليمات البرمجية التي ستتعامل مع الاستثناء داخل كتلة (except).

باختصار، تصحيح الأخطاء والاستثناءات يتطلب فهمًا جيدًا لكيفية عملهما لتجنب المشاكل وضمان تشغيل البرنامج بشكل صحيح.

سنتعلم مهارات تصحيح الأخطاء (Debugging) في تتبع مصدر الخطأ وتحليله ومعالجته بكفاءة، يُعَد منقّح الأخطاء أداةً مفيدة لتعقّب الأخطاء حيث يشغّل سطرًا واحد من الشيفرة البرمجية ثم ينتظر حتى تخبره بالمتابعة. يمكنك أن تأخذ الوقت الذي تريده لفحص القيم الموجودة في المتغيرات في أي مرحلة معينة من عمر البرنامج من خلال تشغيل برنامجك مع منقّح الأخطاء باستخدام هذه الطريقة.

يمكنك تشغيل البرنامج مع منقّح أخطاء المحرّر من خلال النقر على زر تنقيح الأخطاء "Debug" في الصف العلوي من الأزرار بجانب زر التشغيل"Run".

يضيف وضع تنقيح الأخطاء أيضًا أزرارًا جديدة إلى أعلى المحرّر وهي: زر المتابعة "Continue" وزر "Step Over" وزر "Step In" وزر "Step Out"، ويوجد زر التوقف "Stop" أيضًا .

خطة الدروس: تصحيح الأخطاء و الاستثناءات .

1	مفهوم الأخطاء (Errors) والاستثناءات (Exceptions) • أنواع الأخطاء الشائعة Errors . - أخطاء نحوية (Syntax Errors). - أخطاء وقت التشغيل (Runtime Errors). - أخطاء المنطق (Logic Errors).	حصة دراسية	Debugging and Exceptions1
2	تابع /تصحيح الأخطاء و الاستثناءات • التعامل مع الاستثناءات Exceptions. • التعليمة البرمجية try / except • تصحيح الأخطاء Debugging	حصتان دراسيتان	Debugging and Exceptions2

امسح الرمز

نتاجات التعلم: تصحيح الأخطاء والاستثناءات.

1	التعرف على مفهوم الأخطاء (**Errors**) والاستثناءات (**Exceptions**) في البرمجة.
2	التمييز بين أنواع الأخطاء الشائعة مثل: أخطاء نحوية (**Syntax Errors**)، وأخطاء وقت التشغيل (**Runtime Errors**)، وأخطاء المنطق (**Logic Errors**).
3	تمييز الأخطاء اللغوية (**Syntax Errors**) التي تظهر بسبب عدم اتباع قواعد اللغة البرمجية.
4	تمييز الأخطاء الدلالية (**Runtime Errors**) التي تظهر أثناء تشغيل البرنامج نتيجة ظروف غير متوقعة .
5	التمييز بين أنواع أخطاء وقت التشغيل (**Runtime Errors**)
6	تمييز الأخطاء المنطقية (**Logic Errors**) التي تُظهر نتائج غير صحيحة أو غير متوقعة التي يتم اكتشافها قبل المبرمج.
7	التعرف على أنواع الاستثناءات ومنها (**ZeroDivisionError** - **NameError** - **TypeError** - **TypeError**)
8	استخدام الاستثناءات (**Exceptions**) لضمان استقرار البرنامج ومنع توقفه المفاجئ.
9	استخدام التعليمة البرمجية '**try/except**' للتعامل مع الأخطاء المتوقعة أثناء تنفيذ البرنامج واستمراريته دون توقف.
10	التعامل مع أدوات تصحيح الأخطاء لمتابعة تنفيذ التعليمات البرمجية خطوة بخطوة .
11	تطبيق مهارات تصحيح الأخطاء (**Debugging**) في تتبع مصدر الخطأ وتحليله ومعالجته بكفاءة.

المصطلحات الحاسوبية: تصحيح الأخطاء والاستثناءات.

Debugging تصحيح الأخطاء

Errors الأخطاء

Rumtime Errors أخطاء وقت التشغيل

Syntax Errors الأخطاء اللغوية

ZeroDivision Error القسمة على صفر

Logic Errors الأخطاء المنطقية

Type Error نوع بيانات غير صحيحة

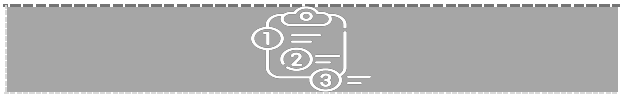
Name Error متغير غير معرّف

Try التعليمة البرمجية

Value Error قيمة غير مناسبة لنوع البيانات

Finally الكتلة البرمجية

Except التعليمة البرمجية



Step Over/Into تنفيذ سطر تلو الآخر

BreaKPoint نقطة توقف

Catch-all Exceptions استثناءات شاملة

Specific Exceptions استثناءات محددة

الاطلاع على تقنيات تصحيح الأخطاء الشائعة Python وسيتم عرض أبسطهم و التي تتناسب مع الفئة العمرية للمتعلمين .

الطباعة باستخدام()print

الأسلوب الأبسط ، حيث يمكنك وضع أمر ()print بعد تعليمات برمجية لرؤية قيم المتغيرات أو التحقق من الخطوات.

نقاط التوقف(BreaKpoints)

هي ميزة في PyCharm تتيح لك إيقاف التعليمة البرمجية مؤقتًا عند سطر معين، للتمكن من فحص قيم المتغيرات أو التحقق من التعليمات البرمجية خطوة بخطوة.

قراءة رسائل الخطأ فيPyCharm

PyCharm يعرض رسائل خطأ مفصلة تساعدك على معرفة نوع الخطأ وموقعه.

كتابة comment قبل التعليمات البرمجية

لشرح الهدف من التعليمات البرمجية أو توضيح الأجزاء (المقاطع) البرمجية .

التعامل مع الأخطاء باستخدام try/except

في حالة التوقع حدوث خطأ، يمكن استخدام **try/except** لمنع توقف البرنامج فجأة يساعد على التعامل مع الأخطاء وتوفير رسائل توضيحية للمستخدم.

وغيرها من التقنيات لتصحيح الأخطاء.

أمثلة على الأخطاء اللغوية (Syntax Errors):

مثـــال(1)

الخطأ:

نسيان إغلاق القوس في دالة **print()** .

التعليمات البرمجية :

Code

```
1  x = 5
2  y = 10
3  print(x + y
```

Output

```
  File "C:\Users\HP\Documents\pythonpart 1\1-1.py", line 3
    print(x + y
          ^
SyntaxError: '(' was never closed
```

مثـــال(2)

الخطأ:

نسيان النقطتين **(:)** في الجمل الشرطية أو الحلقات **(Loops)**.

التعليمات البرمجية:

Code

```
1  if x > 5
2      print("x is greater than 5")
```

Output

```
    if x > 5
            ^
SyntaxError: expected ':'
```

تابع - أمثلة إضافية على الأخطاء اللغوية :

مثـــال(3)

الخطأ في التعليمة البرمجية هو **Indentation Error**، ويحدث عندما يكون هناك مشكلة في المسافة البادئة (الإزاحة) في التعليمية البرمجية في**Python**.

التعليمات البرمجية :

Code

```
1    if x > 5:
2    print("x is greater than 5")
```

Output

```
    print("x is greater than 5")
    ^
IndentationError: expected an indented block after 'if' statement on line 1
```

أمثلة على الأخطاء وقت التشغيل :(Runtime Errors)

الأخطاء وقت التشغيل (Runtime Errors) (أو الأخطاء التنفيذية) تظهر هذه الأخطاء في وقت تشغيل البرنامج في Python ، إما بسبب مدخلات أو عمليات رغم أن كتابته النحوية سليمة بالرغم من أنه تم التحقق من بناء الجملة بشكل صحيح، وغالبًا ما تحدث عندما يواجه المترجم (أو المفسر في حالة بايثون) ظروفًا غير متوقعة أثناء تشغيل التعليمات البرمجية.

مثـــال(1) Zero Division Error

الخطأ وقت التشغيل : عندما يقوم المستخدم بإدخال قيمة عدد صحيح للمتغير الأول والثاني تتم عملية القسمة بشكل صحيح ولكن عند ادخال قيمة 0 للمقسوم عليه . **(var_y)** يحاول البرنامج تنفيذ عملية القسمة **var_x / var_y**، فيؤدي إلى خطأ في البرنامج بسبب القسمة على الصفر، وهو أمر غير مقبول رياضيًا .

النتيجة المتوقعة : إذا لم يتم التعامل المستخدم مع هذا الخطأ بشكل مناسب بإدخال اعداد صحيحة **(سيقوم البرنامج بإصدار ZeroDivisionError)** وسيتوقف عن العمل .

التعليمات البرمجية :

Code

```
var_x = int(input('Enter first number: '))
var_y = int(input('Enter second number: '))
var_z = var_x / var_y
print(var_z)
```

Output

```
Enter first number: 6
Enter second number: 0
Traceback (most recent call last): Explain with AI
  File "C:\Users\HP\Documents\pythonpart 1\main.py", line 3, in <module>
    var_z = var_x / var_y
            ~~~~~~^~~~~~~
ZeroDivisionError: division by zero
```

تابع - أمثلة إضافية على الأخطاء وقت التشغيل:

مثـــال(2)
Name Error

الخطأ وقت التشغيل: استخدام متغير x و هو متغير غير مُعرف.
النتيجة المتوقعة: هنا سيحدث الخطأ وقت التشغيل عندما يكتشف المفسّر أن المتغير لا يحتوي على قيمة.
يجب أن يكون هناك تعريف مسبق للمتغير x.

التعليمات البرمجية:

Code

```
1    print (x)
```

Output

```
Run:  1
Traceback (most recent call last):
  File "C:\Users\ali\Desktop\python\daleel10-a1\.venv\1
    print (x)
          ^
NameError: name 'x' is not defined
```

مثـــال(3)
Type Error

الخطأ وقت التشغيل: المتغير x هو سلسلة نصية (string)، والمتغير y هو عدد صحيح (integer). عملية الجمع + غير مدعومة بين هذين النوعين من البيانات، مما يؤدي إلى حدوث خطأ النوع (TypeError).
النتيجة المتوقعة: عملية الجمع + غير مدعومة بين هذين النوعين من البيانات، مما يؤدي إلى حدوث خطأ النوع (TypeError).
يجب عليك التأكد من أنك تستخدم العمليات المدعومة مع أنواع البيانات الصحيحة. في المثال أعلاه، يمكنك تحويل العدد الصحيح إلى سلسلة نصية قبل الجمع، أو يمكنك استخدام طريقة أخرى لمعالجة البيانات حسب الحاجة.

تابع - أمثلة إضافية على الأخطاء وقت التشغيل:

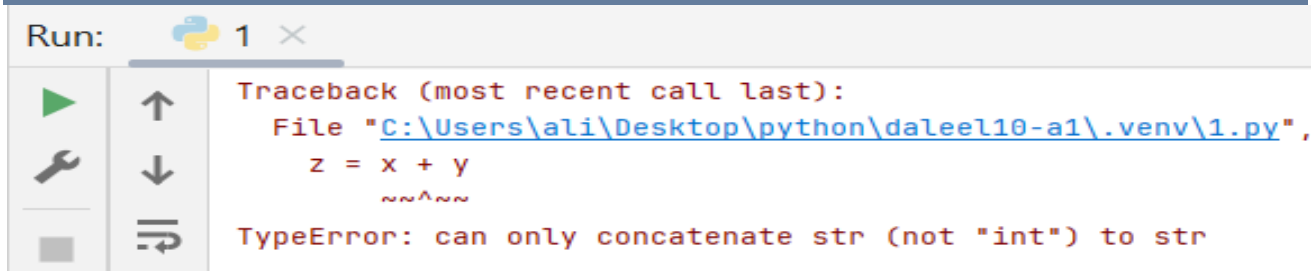
تابع - مثـــال(3) Type Error

التعليمات البرمجية:

Code

```
1  x = "مرحبا"
2  y = 5
3  z = x + y
4  print(z)
```

Output



مثـــال(4) Value Error

الخطأ وقت التشغيل : "أ" ليست عددًا صحيحًا

النتيجة المتوقعة : سيؤدي إلى ValueError لأن المستخدم أدخل المستخدم سلسلة نصية غير قابلة للتحويل إلى عدد صحيح.

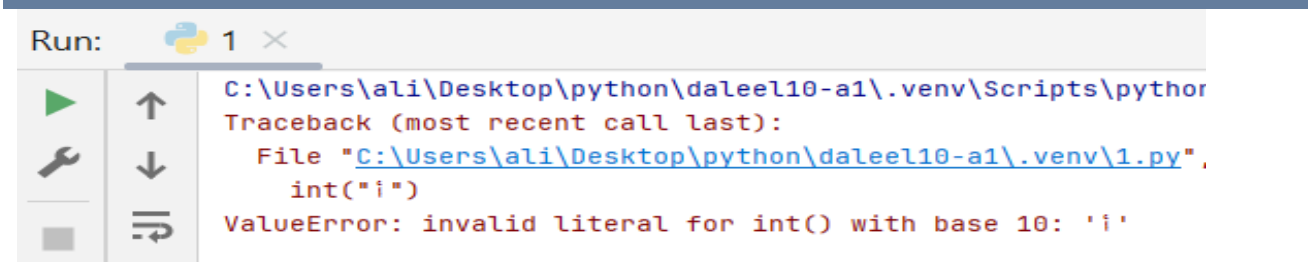
يجب كتابة قيمة قابلة للتحويل إلى عدد صحيح.

التعليمات البرمجية :

Code

```
1  int("i")
2
```

Output



أمثلة على الأخطاء المنطقية:(Logic Errors)

هي أخطاء لا يستطيع المفسر **(Python Interpreter)** اكتشافها وتكتشف من قبل المبرمج، مما يعني أن البرنامج يعمل دون مشاكل واضحة أو توقف، ولكنه يظهر نتائج غير متوقعة أو غير صحيحة. هذه الأخطاء تحدث بسبب نتيجة لخلل في منطق البرنامج أو في كيفية تنفيذ العمليات.

مثـــال (1) استخدام شرط غير صحيح

الخطأ المنطقي: الشرط غير صحيح x<5 (وحسب البرنامج قيمة المتغير X يساوي 10 أي اكبر من 5 المذكورة في الشرط)، والبرنامج يطبع "x is greater than or equal to 5" وهو صحيح.

النتيجة المتوقعة: إذا كان الهدف التحقق مما إذا كان x أقل من 10، يجب تعديل الشرط ليصبح if x<10: .

إذا كان الهدف هو التحقق مما إذا كان x أقل من 10، فيجب تعديل الشرط.

التعليمات البرمجية :

Code

```
x = 10
if x < 5:
    print("x is less than 5")
else:
    print("x is greater than or equal to 5")
```

Output

```
x is greater than or equal to 5

Process finished with exit code 0
```

تابع - أمثلة على الأخطاء المنطقية:(Logic Errors)

مثـــال(2)

الخطأ المنطقي :يفترض البرنامج يقوم بطباعة القيمة الأولية للمتغير count بشكل مستمر بدون توقف لأن المتغير لا يتم تحديثه، مما يؤدي إلى حلقة لا نهائية.

النتيجة المتوقعة :كان يجب إضافة المتغير على أن يزيد بمعدل 1 في كل مره في الحلقة التكرارية count += 1 ويتم تحديث المتغير في كل تكرار للحلقة.

التعليمات البرمجية:

Code

```
count = 0
while count < 5:
    print("Counting:", count)
    # count += 1  # تم نسيان من قبل المبرمج تحديث المتغير
```

Output

```
Counting: 0
Counting: 0
Counting: 0
Counting: 0
Counting: 0
Counting: 0
```

بعد الضغط على أداة Stop أو Ctrl+F2 تظهر الشاشة التالية:

Out put

```
  File "C:\Users\HP\Documents\pythonpart 1\main.py", line 3, in <module>
    print("Counting:", count)
KeyboardInterrupt
```

تابع - أمثلة على الأخطاء المنطقية:(Logic Errors)

مثـــال(3)
مقارنة
منطقية
خاطئة

الخطأ المنطقي (Logic Error) :

البرنامج يعمل بشكل صحيح ، ولا يوجد أخطاء أثناء التنفيذ ولكنه لا يعطي النتيجة المتوقعة بسبب خطأ في المنطق المستخدم في هذه الحالة، تم استخدام صيغة لحساب محيط المستطيل بدلاً من مساحة المستطيل .

النتيجة المتوقعة: الهدف هو حساب المساحة، فيجب تعديل الصيغة لتصبح :

(var_area = var_l * var_w)

التعليمات البرمجية :

Code

```
var_l = 5
var_w = 6
# المطلوب حساب المساحة و والعملية المحسوبه لقياس المحيط
var_area = 2* ( var_l + var_w)
# طباعة النتيجة
print('Area = ', var_area)
```

Output

```
Area =  22

Process finished with exit code 0
```

تصحيح الأخطاء :

مثـال :

Input

```
x = input('Enter first number: ')     # تخزين المدخل في المتغير x
if x == 0                             # مقارنة x مع الصفر
print('x is equal to 0.')
else:
    print('x is not equal to 0.')
```

الأخطاء المنطقية Logic Errors

- x = input('Enter first number: ')

لم يتم تحويل المدخل في المتغير X من نص الى قيمة عددية باستخدام أحد الدوال (int, float,..).

- في السطر الذي يحتوي على جملة الشرطية if x == 0، يتم مقارنة المتغير x مع الرقم 0 المشكلة هنا هي أن المدخل x يتم استدعاؤه باستخدام دالة input()، والتي تقوم بإرجاع القيمة ك سلسلة نصية(String) ، حتى لو أدخل المستخدم رقمًا.
- عدم وجود : في عبارة if.
- عدم وجود مسافة بادئة في عبارة print الأولى.

الاستثناءات (Exceptions)

الاستثناءات في البرمجة هي أخطاء قد تحدث أثناء تشغيل البرنامج نتيجة لأحداث غير متوقعة تساعدك الاستثناءات على التعامل مع الأخطاء بطريقة منظمة بحيث يستمر البرنامج في العمل دون توقف مفاجئ.

تسمح التعليمة البرمجية (try/except) بتحديد الكود الذي قد يسبب خطأ في كتلة (try) و تنفيذ كود بديل في كتلة (except) للتعامل مع هذا الخطأ إذا حدث، بدلاً من ترك البرنامج يتوقف بشكل غير متوقع.

أنه يمكن استخدام التعليمات البرمجية **try/except** ، لتحديد الجزء الذي قد يحدث فيه خطأ، وللتأكد من استمرار عمل البرنامج.

كيفية استخدام try/except :

- التعليمة البرمجية **try** تحتوي على التعليمات البرمجية الذي قد يحدث فيه خطأ.
- التعليمة البرمجية **except** تحتوي على التعليمات البرمجية الذي سيتم تنفيذه إذا حدث خطأ في الجزء الموجود داخل **try**.

مثـــال توضيحي :

Input

```
try:
    x = 10 / 0  # ZeroDivisionError هذا سيؤدي إلى خطأ
    print(x)
except ZeroDivisionError:
    print("لا يمكن القسمة على صفر")
except:
    print("حدث خطأ غير متوقع")
```

Out put

```
Run:  1
C:\Users\ali\Desktop\python\daleel10-a1\.venv\Scripts\python.exe
لا يمكن القسمة على صفر

Process finished with exit code 0
```

شرح المثال:

سيتم تنفيذ كتلة except ZeroDivisionError لأن قسمة 10 على 0 تسبب في خطأ ZeroDivisionError .

إذا تم تغيير الكود ليصبح x = 10 / 2 ، فلن يتم تنفيذ أي من كتل except لأنه لن يحدث أي خطأ.

تصحيح الأخطاء في بايثون Debugging :

تصحيح الأخطاء في بايثون (Debugging) هو عملية تحديد وإزالة الأخطاء من برنامج بايثون.

خطوات تصحيح الأخطاء في بايثون:

1. **تحديد نوع الخطأ:** أول خطوة هي تحديد نوع الخطأ في البرنامج.
2. **استخدام مصحح الأخطاء مثل:**
   - **PDB**: يعد مصحح الأخطاء المدمج في بايثون (PDB) أداة قوية لتصحيح الأخطاء.
3. **تتبع الأخطاء:**
   - **الخطوة الأولى** : مثلا استخدام PDB لتحديد مكان الخطأ في الكود.
   - **الخطوة الثانية** :فحص قيم المتغيرات في نقاط مختلفة من الكود لتحديد سبب الخطأ.
   - **الخطوة الثالثة** :إجراء التعديلات اللازمة على الكود لإصلاح الخطأ.
4. **اختبار الأخطاء:**
   - **الخطوة الأولى** :تشغيل الكود بعد إجراء التعديلات.
   - **الخطوة الثانية** :تأكيد أن الكود يعمل بشكل صحيح و أنه لا توجد أخطاء أخرى.

نصائح لتصحيح الأخطاء:

- **اجعل الكود بسيطًا** : كلما كان الكود أبسط، كان من الأسهل تحديد الأخطاء.
- **استخدم التعليقات** : التعليقات في الكود تساعد على فهمه وتسهيل عملية التصحيح.
- **اختبر الكود بشكل منتظم**: لا تنتظر حتى يكتمل الكود لتصحيحه، اختبره بشكل دوري.
- **استخدم المصححات**: استخدم PDB أو IDE لتصحيح الأخطاء، فهي أداة قوية.
- **ابحث عن الأخطاء الشائعة**:تعرف على الأخطاء الشائعة في بايثون وحاول تجنبها.

باستخدام هذه الأدوات والتقنيات، يمكنك أن تصبح مصحح أخطاء فعالًا في بايثون وتضمن تشغيل برامجك بسلاسة.

حل أوراق العمل : تصحيح الأخطاء والاستثناءات.

ورقة عمل (16) : حساب قيمة فاتورة المشتريات بعد الخصم.

المشكلة : حساب المستخدم السعر بعد تطبيق خصم الفاتورة، من خلال:

- إدخال قيمة الفاتورة (أكبر من صفر).
- إدخال نسبة خصم (تتراوح بين 0و100).

المدخلات: السعر الأصلي (price) و نسبة الخصم (discount).

المعالجة: حساب السعر النهائي باستخدام المعادلة:

net_price = price - (price * discount / 100)

المخرجات:

- طباعة السعر بعد الخصم (net_price).
- طباعة (Error: Please enter valid numeric values) في حال وجود خطأ من النوع (ValueError).
- طباعة (Program has ended).

الخوارزمية:

1. بداية البرنامج.
2. استخدام كتلة try لمحاولة تنفيذ الخطوات التالية:
   a. طلب من المستخدم إدخال price (السعر الأصلي).
   b. طلب من المستخدم إدخال discount (نسبة الخصم).
   c. حساب السعر النهائي باستخدام:
      net_price = price - (price * discount / 100)
   d. طباعة "Net price after discount is: net_price"
3. استخدام كتلة except بحيث إذا حدث خطأ في نوع البيانات (ValueError):
   طباعة : ". "Error: Please enter valid numeric values
4. في جميع الأحوال، طباعة الرسالة النهائية:
   "Program has ended" داخل finally.
5. نهاية البرنامج.

تابع حل ورقة العمل (16) :

التعليمات البرمجية :

Code

discount.py

```python
try:
    #Get the original price from the user
    price = float(input("Enter the original price: "))

    #Get the discount percentage from the user
    discount = float(input("Enter the discount percentage (0 - 100): "))

    #Calculate the net price
    net_price = price - (price * discount / 100)

    #Print the result
    print("Net price after discount is:", net_price)

except ValueError:
    #Handle invalid input
    print("Error: Please enter valid numeric values.")

finally:
    #Always print this message
    print("Program has ended.")
```

Output

```
"/Users/mirror/PycharmProjects/Grade 10/.venv/bin/python"
Enter the original price: 600g
Error: Please enter valid numeric values.
Program has ended.

Process finished with exit code 0
```

تابع حل ورقة العمل (16) :

تطوير البرنامج :

Code

discount_upgrade.py

```python
try:
    #Get the original price from the user
    price = float(input("Enter the original price: "))
    #Check if price is valid
    if price <= 0:
        print ("Error: Price must be greater than zero!")
    else:
        #Get the discount percentage from the user
        discount = float(input("Enter the discount percentage (0 - 100): "))
        #Check if discount in valid
        if discount < 0 or discount > 100:
            print("Error: Discount must be between 0 and 100!")
        else:
            #Calculate the net price
            net_price = price - (price * discount / 100)
            #Print the result
            print("Net price after discount is:", net_price)

except ValueError:
    #Handle invalid input
    print("Error: Please enter valid numeric values.")

finally:
    #Always print this message
    print("Program has ended.")
```

Output

```
"/Users/mirror/PycharmProjects/Grade 10/.venv/bin/python"
Enter the original price: 1500
Enter the discount percentage (0 - 100): 105
Error: Discount must be between 0 and 100!
Program has ended.

Process finished with exit code 0
```

تابع – حل أوراق العمل: تصحيح الأخطاء و الاستثناءات

ورقة عمل (17): مؤشر كتلة الجسم BMI (Body mass index)

المشكلة: اكتشاف الأخطاء وتجاوزها (استكمال البرنامج ليعمل بصورة صحيحة).

المدخلات: الوزن بالكيلوجرام (weight) و الطول بالمتر (height).

المعالجة:

- حساب مؤشر كتلة الجسم BMI باستخدام المعادلة:

قانون كتلة الجسم:(BMI) = $\frac{\text{الوزن بالكيلوجرام}}{\text{الطول بالمتر المربع}}$

- تصنيف الحالة الصحية حسب نتيجة BMI :
  - إذا كان BMI أقل من 18.5.
  - إذا كان BMI بين 18.5 و 24.9.
  - إذا كان BMI بين 25 و 29.9.
  - إذا كان BMI غير ذلك.
- معالجة الأخطاء:
  - إذا أدخل المستخدم نص أو قيمة غير رقمية (ValueError).
  - إذا أدخل الطول = 0 (ZeroDivisionError).

المخرجات:

- طباعة قيمة مؤشر كتلة الجسم (BMI) بدقة خانتين عشريتين.
- وصف الحالة الصحية بناءً على قيمة الBMI بطباعة الرسالة المناسبة.
- طباعة رسائل الخطأ في حال وجود خطأ في الإدخال (ValueError ، ZeroDivisionError).

الخوارزمية:

1. بداية البرنامج.
2. بداية التكرار اللانهائي وتنفيذ التالي:

- استخدام كتلة try لمحاولة تنفيذ الخطوات التالية:
  - طلب من المستخدم إدخال الوزن بالكيلوغرام وحفظه كعدد عشري.(float)
  - طلب من المستخدم إدخال الطول بالمتر وحفظه كعدد عشري.(float)
  - حساب مؤشر كتلة الجسم BMI باستخدام المعادلة:

قانون كتلة الجسم:(BMI) = $\frac{\text{الوزن بالكيلوجرام}}{\text{الطول بالمتر المربع}}$

  - طباعة قيمة الـ BMI بتنسيق عشري من خانتين.
  - تحديد الفئة الصحية بناءً على قيمة الBMI :
    a) إذا كان BMI أقل من 18.5، طباعة "You are underweight." .
    b) إذا كان BMI بين 18.5 و 24.9، طباعة"You have a normal weight." .
    c) إذا كان BMI بين 25 و 29.9، طباعة "You are overweight." .
    d) إذا كان BMI غير ذلك طباعة "You are obese.".
  - استخدام كتلة except بحيث إذا ادخل المستخدم قيمة غير رقمية (ValueError):
    a) طباعة رسالة: "Error: Please enter valid numeric values for weight and height.".
  - استخدام كتلة except بحيث إذا حدث خطأ وأدخل المستخدم الطول صفراً (لايمكن القسمة على صفر) (ZeroDivisionError):
    a) طباعة رسالة: "Error: You can't divide by zero!" .

3. الرجوع إلى الخطوة 2.
4. نهاية البرنامج.

تابع حل ورقة العمل (17) :

التعليمات البرمجية :

Code

BMI_Answer.py

```python
# Program to calculate BMI with error handling
while True:
    try:
        # Get user input: weight and height
        weight = float(input("Enter your weight in kilograms: "))
        height = float(input("Enter your height in meters: "))
        # Calculate BMI
        bmi = weight / (height ** 2)
        # Display the BMI
        print(f"Your BMI is: {bmi:.2f}")
        # Determine BMI category
        if bmi < 18.5:
            print("You are underweight.")
        elif 18.5 <= bmi < 24.9:
            print("You have a normal weight.")
        elif 25 <= bmi < 29.9:
            print("You are overweight.")
        else:
            print("You are obese.")

    except ValueError:
        print("Error: Please enter valid numeric values for weight and height.")
    except ZeroDivisionError:
        print("Error: You can't divide by zero!")
```

Output

```
"/Users/mirror/PycharmProjects/Grade 10/.venv/bin/python"
Enter your weight in kilograms: 70
Enter your height in meters: 150
Your BMI is: 0.00
You are underweight.
Enter your weight in kilograms: 80
Enter your height in meters: 0
Error: You can't divide by zero!
Enter your weight in kilograms: 60
Enter your height in meters: 190
Your BMI is: 0.00
You are underweight.
Enter your weight in kilograms:
```

مثال إضافي

لبرنامج آلة حاسبة بسيط يستقبل من المستخدم رقمين ثم يحدد نوع العملية الحسابية ثم يطبع النتيجة بعد التنفيذ, يحتوي هذا البرنامج على نوعين من الأخطاء:

- الأخطاء النحوية (Syntax Errors) .
- الأخطاء المنطقية (Logical Errors) .

التعليمات البرمجية:

Code

```
# آلة حاسبة بسيطة تجمع، تطرح، تضرب، وتقسم رقمين

# الحصول على الأرقام من المستخدم
num1 = input(" :أدخل الرقم الأول")
num2 = input(" :أدخل الرقم الثاني")

# اختيار العملية الحسابية
operation = input(" :(/ ,* ,- ,+) أدخل العملية التي تريد القيام بها")

# تحويل الأرقام إلى float مع معالجة الأخطاء
try:
    num1 = float(num1)
    num2 = float(num2)
except ValueError
    print(".خطأ: يرجى إدخال أرقام صحيحة")
    exit()

# إجراء العملية الحسابية وطباعة النتيجة
if operation == "+":
    result = num1 - num2  # خطأ دلالي: استخدام الطرح بدلاً من الجمع
elif operation == "-":
    result = num1 + num2  # خطأ دلالي: استخدام الجمع بدلاً من الطرح
elif operation == "*":
    result = num1 * 2  # خطأ دلالي: يجب أن يكون num2 هو العدد المضروب
elif operation == "/":
    if num2 == 0:
        print("خطأ: لا يمكن القسمة على الصفر")
    else:
        result = num1 / num1  # خطأ دلالي: القسمة على نفس العدد بدلاً من num2
else:
    print("خطأ: عملية غير صالحة")

# طباعة النتيجة
print(":النتيجة", result
```

حل المثال الإضافي :

رقم السطر	الخطأ	نوع الخطأ	التصحيح	ملاحظات
14	except نسيان النقطتين :بعد ValueError	الأخطاء النحوية (Syntax Errors)	except ValueError:	عدم كتابة النقطتين : except بعد ValueError
34	نسيان إغلاق القوس في نهاية جملة الطباعة print("النتيجة:", result)	الأخطاء النحوية (Syntax Errors)	print("النتيجة:", result)	
25	القسمة على num2 بدون التحقق مما إذا كان يساوي صفرًا elif operation:"/" == result = num1 / num2	الأخطاء المنطقية (Logical Errors)	Elif operation == "/": if num2 == 0: print (" خطأ: لا يمكن القسمة على الصفر") else: result = num1 / num2	إضافة شرط للتحقق مما إذا كان num2 يساوي صفرًا قبل إجراء القسمة
31	إذا كانت العملية المدخلة (operation)صالحة إذا أدخل المستخدم رمزًا غير صالح else: print("خطأ: عملية غير صالحة")		اضافة شرط if إضافي قبل الطباعة للتحقق مما إذا كانت العملية صالحة if operation not in:["/" ,"*" ,"-" ,"+"] print("خطأ: عملية غير صالحة")	
24	في عملية الضرب، يتم ضرب num2 بدلاً من num1في elif operation == "*": result = num1 * 2		elif operation == "*": result = num1 * num2	استبدال 2 بـ num2 لإتمام عملية الضرب بشكل صحيح

04

الوحـــدة الثـــالثة:
المنتجات الرقمية

المنتجات الرقمية : المشروع .

تختلف وحدة (المنتجات الرقمية) عن باقي وحدات الكتاب كونها لا تعتمد على تنفيذ أوراق عمل محددة ومعدة مسبقا للمتعلم، وإنما تعتمد على ابداع المتعلم سواء كان منفردا 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.

امسح الرمز

Project

نتاجات التعلم: المنتجات الرقمية.

1. توظيف المهارات البرمجية المكتسبة في Python بشكل فعّال لإنتاج برامج متنوعة.
2. تنمية مهارات التعلم الذاتي والبحث باستخدام مصادر متنوعة للتعلم لتطوير مشروعه.
3. تنمية القدرات على الابتكار والإبداع من خلال تحليل المشكلات وتصميم حلول برمجية مناسبة.
4. تطبيق المهارات البرمجية في تصميم وتطوير برامج تعالج احتياجات من واقع الحياة اليومية.
5. المشاركة بفاعلية ضمن فرق العمل البرمجية، و إظهار روح التعاون وتبادل المعرفة.
6. عرض الأفكار البرمجية أمام الآخرين، و مناقشة الحلول المختلفة و تقبّل وجهات النظر المتنوعة.
7. التفاعل مع التغذية الراجعة لتحسين جودة المشروع، وتطوير الأداء البرمجي.

مراحل إعداد المشروع:

1-أعداد فريق العمل

تحديد فريق العمل لإعداد المشروع من 4 إلى 5 متعلمين (أو حسب ما يراه المعلم مناسباً) في الأسابيع الأولى من الفصل الدراسي بعد استعراض فكرة عامة عن طبيعة المنتجات الرقمية.

2-تحديد الفكرة

تحديد موضوع المشروع والهدف منه ضمن ضوابط وشروط قبول فكرة المنتج الرقمي ومن أهمها توظيف أحد تقنيات الذكاء الاصطناعي في البرنامج ولا يقبل المنتج الرقمي ويعتبر مرفوضاً من دونه، والحرص على بساطة المنتج الرقمي لتجنب أي تعقيدات واستغراق وقت أكثر.

3-إعداد خطة الحل (الخوارزمية)

إعداد خوارزمية الحل لتحقيق فكرة المشروع ومراجعتها من قبل المعلم حيث يقتصر دور المعلم على إرشاد المتعلمين نحو الطريق الصحيح.

4-تصميم خريطة التدفق

تمثيل خوازمية الحل بخريطة تدفق (Flowchart).

5-كتابة التعليمات البرمجية

تحويل خريطة التدفق الى مجموعة من التعليمات البرمجية باستخدام لغة بايثون

تابع - مراحل اعداد المشروع:

6-اختبار البرنامج ، تصحيح البرنامج إن وجد.

تشغيل البرنامج وإدخال بيانات ومقارنة النتائج لاختبار صحة وعمل البرنامج بصورة سليمة، وتصحيح الأخطاء ان وجدت.

7-عرض المشروع على الزملاء و معلم الحاسوب

مناقشة المشروع مع زملائه بعد عرض تقديمي يقوم به الفريق في الحصة المخصصة لتعزيز الثقة في النفس والتدرب على الإلقاء ومواجهة الجمهور وتقبل النقد من الطرف الآخر.

المهارات المقترحة للمشروع:

من المهم أن يتضمن مشروع المتعلم معظم المهارات التي اكتسبها خلال الفصل الدراسي ومنها:

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()

متابعة مراحل إعداد المشروع:

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

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

أمور يجب مراعاتها في وحدة المنتجات الرقمية:

1. 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 أعدها المعلم أو المتعلمون 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 ستة أعضاء مع تحديد القائد ليكون مسئولاً عن التواصل مع المعلم وتنسيق الفريق.

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

6. لتسهيل عملية اجتماع الفرق والعمل الجماعي، يمكن استخدام قنوات الفصل في برنامج Microsoft Teams للاجتماعات وتنسيق العمل.

05

تحضـــير الـــدروس

تحضـــير الـــدروس:

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 سجلا مكتوبا لمحتوى التعلم يرجع إليه المعلم إذا نسي شيئا 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	يعد إعداد الدرس مهما من الناحية التربوية، إذ يعتمد عليه مدير المدرسة أو الموجه الفني لمتابعة المعلم أثناء زيارة الحصة.

سجـل التحضـــير

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.

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

يتم تسجيلها لكل حصة دراسية

مصادر التعلم والوسائل التعليمية	القيمة التربوية
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	تحديد القيمة التربوية حسب الجدول الزمني المقترح مع وصف زمني لكيفية تفعيلها خلال الحصة بما يتلاءم مع عرض المفاهيم الدرس

	النشاط الاستهلالي	الزمن
المقدمة والتمهيد	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	5-3 دقائق

م	نواتج التعلم	محتوى التعلم (الأنشطة والاستراتيجيات)	التقويم	الزمن
	عبارة تتضمن ما الذي يتوقع من المعلم تحقيقه نتيجة لأنشطة التعلم ويعبر عنها على شكل معارف أو مهارات او اتجاهات ويجب أن تكون واضحة تصف ما الذي يتوقع ان يتعلمه المتعلمين نتيجة اكتمال عملية التعلم	محتوى التعلم يجب أن يجيب 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 كنشاطات 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		

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

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 ومترابطا لطريقة سير الدرس قم بتسجيلها على شكل خطوات واضحة ومحددة، شاملةً عناصر الإعداد، مراعيا في كل خطوة عامل الوقت وارتباطها بأهداف الدرس.
- الإعداد الكتابي بعد تكوين تصور كامل ومترابط لطريقة سير الدرس، يتم تسجيله كتابيًا على شكل خطوات واضحة ومحددة، تتضمن جميع عناصر الإعداد، مع مراعاة عامل الوقت وارتباط كل خطوة 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.

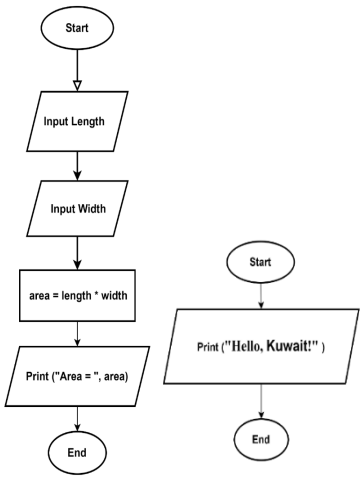
نموذج تحضير درس مادة الحاسوب للصف العاشر

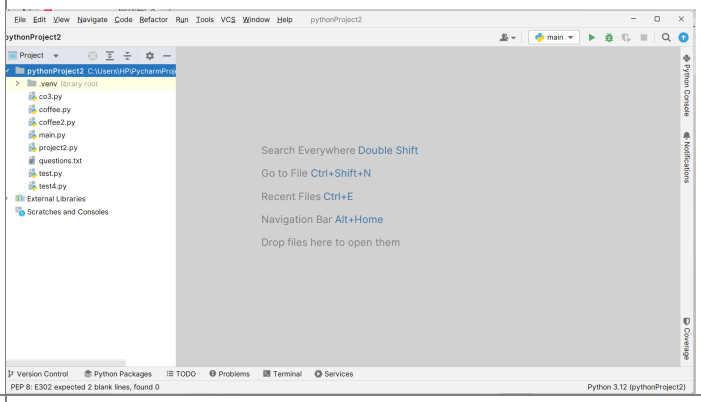
عنوان الدرس	الأمن السيبراني Cyber Security - مدخل الى البرمجة			
الأسبوع	الوحدة	اليوم	التاريخ	الفصل
الأول	الأمن السيبراني برمجة Python	الأربعاء	2025-9-18	1/10
		الخميس	2025-9-19	2/10

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

	النشاط الاستهلالي	الزمن
المقدمة والتمهيد	- الترحيب بالمتعلمين في بداية العام الدراسي - عرض قوانين المختبر - عرض محتوى المنهج الدراسي والية توزيع الدرجات	5د

م	نواتج التعلم	محتوى التعلم (الأنشطة والاستراتيجيات)	التقويم	الزمن
1	- 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.	- 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	- 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 لألعاب الفيديو، التطبيقات، وحتى المواقع الإلكترونية أن تعمل وتؤدي وظائفها بشكل دقيق؟ - -من خلال استراتيجية التشبيه الواقعي: - طرح مثال يشبه الحاسوب هو طاهٍ في مطبخ - والبرمجة هي كتابة وصفة لطبق معين - كل خطوة في الوصفة هي تعليمات - يجب أن يتبعها الطاهي لتحضير وجبة لذيذة أي انجاز البرنامج النهائي	- أتوقع الإجابة من المتعلمين - كل هذا يتم بفضل البرمجة - سؤال: ما مفهوم البرمجة؟	5
3	- خطوات انشاء برنامج لغة البرمجةpython	- 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) - استقبل قيمة الطول من المستخدم (بالمتر). - استقبل 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 سبورات 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 منهج 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 جهازالمعلم	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 احدى الطرق ويتم تطبيقها لمشاهدة النتيجة على أجهزتهم	5
11	- هياكل البيانات	- من خلال PwerPoint: يتم استعراض هياكل البيانات	- يلاحظ المتعلم هياكل البيانات ويميز أقسامها	5
	التقويم الختامي	حل ورقة عمل (1-1) ورقة عمل (1-2)		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
			☺	😐	☹	

06

التقـــويم الـــــتربوي

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 الصفي- 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 التربوي :

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 .

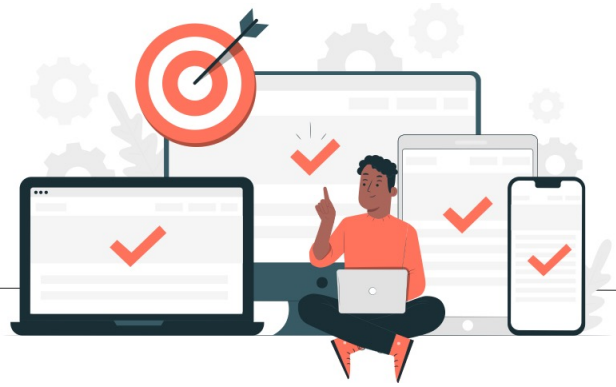
أ	الملاحظات الصفية والمناقشات	تشمل هذه الاستراتيجية ملاحظة أداء المتعلمين أثناء الأنشطة الصفية والمناقشات.
ب	المشاريع	تتضمن المشاريع تكليف المتعلمين بمهمة تتطلب منهم تطبيق المعرفة والمهارات التي تعلموها.
ج	العروض التقديمية	يتضمن هذا النوع من التقويم إعداد المتعلمين وتقديمهم لعروض حول موضوعات معينة.
د	حل المشكلات:	تتضمن تقديم مشكلة للمتعلمين لحلها باستخدام مهاراتهم ومعرفتهم.

تابع - استراتيجيات التقويم التربوي :

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

تابع - استراتيجيات التقويم التربوي :

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 لاحقا. وتُعد هذه الاستراتيجية مكونا أساسيا 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)

للصف (العاشر -الحادي عشر علمي- الحادي عشر أدبي – الثاني عشر علمي – الثاني عشر أدبي)

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

07

الاختبارات

الاختبارات التحصيلية:

الاختبارات التحصيلية: هي الأداة التي تستخدم في قياس المعرفة والفهم والمهارة لمادة دراسية معينه قياسا كميا بهدف قياس نواتج التعلم لدي 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

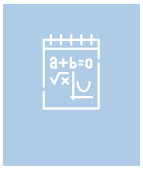
تحديد موضوع الاختبار	إعداد نموذج الإجابة وتوزيع الدرجات
تحديد أهداف الاختبار	وضع تعليمات الاختبار
تحديد الأوزان النسبية لكل هدف.(جدول المواصفات)	مراجعة وتدقيق الاختبار
تحديد شكل الاختبار	إجراء وتنفيذ الاختبار
صياغة فقرات الاختبار	تصحيح الاختبار
ترتيب الأسئلة	تحليل نتيجة الاختبار
	تقويم الاختبار

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 شاملا لكل المواضيع ومتوازنا. 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 رسميا 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 لاعتماده.
- 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 من 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 منصفا في تقييم عمل المتعلمين.
4. التغلب على العامل الشخصي عند تقدير الدرجات وعدم توقف علامة الُمتعلم على من يصحح ورقته.
5. التركيز على إجابة الُمتعلم 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 السلاسل النصية Strings	• عمليات المقارنة • العمليات الشرطية If , elif ,else	2
6		الأسبوع السادس الشروط Conditions	• الدالة while • الدالة for • الدالة Range	1
7		الأسبوع السابع + الثامن التكرار Loops	• 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() • تعليمية f-string • Int() • Float() • Center()	2

تابع -الميزان النسبي المقترح: لدرجات الامتحان العملي للفترة الدراسية الأولى

لمادة تقنية المعلومات - الصف العاشر

م		الأسـبوع	المهـــارات	الدرجة
8		الأسـبوع التاسع +العاشر تصحيح الأخطاء والاستثناءات Debugging and Exceptions	مفهوم الأخطاء (Errors) والاستثناءات (Exceptions) • أنواع الأخطاء الشائعة Errors . - أخطاء نحوية (Syntax Errors). - أخطاء وقت التشغيل (Runtime Errors). - أخطاء المنطق (Logic Errors). تابع /تصحيح الأخطاء و الاستثناءات • التعامل مع الاستثناءات Exceptions. • التعليمة البرمجية try / except • تصحيح الأخطاء Debugging	2
الإجمـــالي=10				

الخاتمة

نختتم هذا الدليل بتفاؤل وامل في ان يكون له الأثر الإيجابي على رحلتكم التعليمية في تدريس منهج الصف العاشر وأن يكون معينا للمعلم أثناء تقديم المحتوى العلمي , فقد تم تصميم هذا الدليل لمساعدة المعلم في تقديم محتوى الامن السيبراني من خلال فيديوهات 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 فيديوهات .

وندعو جميع المعلمين للاستفادة القصوى من الموارد والمعلومات المقدمة في هذا الدليل بهدف اثراء مهاراتهم وتعزيز معلوماتهم وخبراتهم , ونتطلع 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 مهاراتكم والاطلاع على كل ما هو جديد في مجال التكنلوجيا 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 نتاجات 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.