٩

سلطنة عمان
وزارة التربية والتعليم

رؤية عُمان 2040
OmanVision

نتقدم بثقة
Moving Forward
with Confidence

تقنية المعلومات

الصف التاسع

الفصل الدراسي الثاني

سلطنة عُمان
وزارة التربية والتعليم

٩

تقنية المعلومات

الصف التاسع - الفصل الدراسي الثاني

الطبعة الأولى ١٤٤٢هـ - ٢٠٢٠م

أُلِّف هذا الكتاب بموجب القرار الوزاري رقم ٢٠١٧/٣٨٢ م

تمَّ إدخال البيانات والتدقيق اللغوي والرسم والتصميم والإخراج
في مركز إنتاج الكتاب المدرسي والوسائل التعليمية
بالمديرية العامة لتطوير المناهج

جميع حقوق محفوظة

جميع حقوق الطبع والتأليف والنشر محفوظة لوزارة التربية والتعليم
ولا يجوز طبع الكتاب أو تصويره أو إعادة نسخه كاملاً أو مجزأً أو ترجمته أو تخزينه
في نطاق استعادة المعلومات بهدف تجاري بأي شكل من الأشكال إلا بإذن كتابي مسبق
من الوزارة، وفي حال الاقتباس القصير يجب ذكر المصدر.

المغفور له

السلطان قابوس بن سعيد -طيّب اللّه ثراه-

حضرة صاحب الجلالة

السلطان هيثم بن طارق المعظم

سلطنـــــة عُمـــــان

النَّشيدُ الْوَطَنِيُّ

يا رَبَّنا احْفَظْ لنا جَلالَةَ السُّلْطان
وَالشَّعْبَ في الأَوْطان بِالْعِزِّ والأَمان
وَلْيَدُمْ مُؤَيَّدًا عاهِلاً مُمَجَّدًا
بِالنُّفوسِ يُفْتَدى

يا عُمانُ نَحْنُ مِنْ عَهْدِ النَّبي أَوْفِياءُ مِنْ كِرامِ الْعَرَبِ
فارْتَقي هامَ السَّماء وَامْلَئي الْكَوْنَ الضِّياء
وَاسْعَدي وَانْعَمي بِالرَّخاء

بسم الله الرحمن الرحيم

التقديم

الحمد لله رب العالمين، والصلاة والسلام على أشرف المرسلين، سيدنا محمد وعلى آله وصحبه أجمعين،،،

سعت وزارة التربية والتعليم إلى تطوير المنظومة التعليمية في جوانبها المختلفة؛ لمواكبة التطورات المتسارعة في مجالي المعرفة والتقانة، وتلبية متطلبات مؤسسات التعليم العالي، واحتياجات المجتمع العماني وسوق العمل، وهي بذلك تتوافق مع أهداف رؤية عمان ٢٠٤٠ وركائزها التي أكدت أهمية رفع جودة التعليم وتطوير المناهج الدراسية والبرامج التعليمية؛ لإعداد متعلم معتز بهويته، مبدع ومبتكر، ومنافس عالميًّا في جميع المجالات.

كما جاءت المناهج الدراسية منسجمة مع فلسفة التعليم في السلطنة، والاستراتيجية الوطنية للتعليم ٢٠٤٠ في تهيئة الفرص المناسبة لبناء الشخصية المتكاملة للمتعلمين، والحرص على امتلاكهم مهارات القرن الحادي والعشرين؛ كريادة الأعمال والابتكار، وأخلاقيات العمل، والتعامل مع معطيات التكنولوجيا الحديثة وإنتاج المعرفة، وتعزيز مهارات التفكير والبحث العلمي، ورفع مستوى وعيهم بالقضايا الإنسانية، وقيم السلام والحوار، والتسامح والتقارب بين الثقافات.

ويمثل هذا الكتاب المدرسي ترجمة للمحتوى المعرفي والمهاري للمنهاج الدراسي وضع ليسترشد به المعلم والمتعلم للوصول إلى معلومات شاملة ومتنوعة، ولاكتساب مهارات تعليمية مختلفة؛ لتحقيق ما تصبو إليه الوزارة من أهداف تربوية، وغايات سامية تسهم في تقدم هذا الوطن العزيز تحت ظل القيادة الحكيمة لمولانا حضرة صاحب الجلالة السلطان هيثم بن طارق المعظم- حفظه الله ورعاه-.

د.مديحة بنت أحمد الشيبانية

وزيرة التربية والتعليم

بسم الله الرحمن الرحيم

المقدمة

عزيزي الطالب/عزيزتي الطالبة

يسرُّنا أن نضع بين يديك كتاب تقنية المعلومات للصف التاسع (الفصل الدراسي الثاني) الذي سنتناول من خلاله وحدتين دراسيتين؛ نأمل أن تُسهما في إثراء معلوماتك وتطوير مهاراتك في مجال تقنية المعلومات.

بدايةً ستتعرَّف إلى مفهوم لغة البرمجة، وكيفية توظيف الخوارزميات في كتابة برنامج، واستخدام الوحدات البرمجية في إنشاء واجهة رسومية. وفي مجال الحوسبة السحابية ستتمكن من الحفظ السحابي، واستخدام الاستبانات الإلكترونية والاختبارات الإلكترونية ونشرها باستخدام الحوسبة السحابية.

ستلاحظ أن أنشطة الكتاب جاءت متناسبة مع مرحلتك العمرية، ومرتبطة باهتماماتك؛ كي تلبي احتياجاتك العلمية والعملية، وتوجُّهَك إلى تفعيل واستخدام التقنية في كافة مجالات تعلمك. كما ستلاحظ أن تعليمات الأنشطة جاءت في خطوات واضحة ومبسطة لتساعدك على الاعتماد على نفسك من خلال التعلُّم الذاتي بحسب قدراتك والوقت المتاح لديك سواء في المدرسة أو المنزل.

إننا نوجه عنايتك عزيزي الطالب/ عزيزتي الطالبة إلى المحافظة على كتابك هذا، باعتباره دليلك ومرجعك في أثناء تعلمك للمادة وتنفيذ أنشطتها. وفقك الله في مسيرتك التعليمية.

المؤلفون

المحتويات

الوحدة الأولى البرمجة

مقدمة فى لغة البرمجة ١٧
التعبيرات والشروط ٢٩
المصفوفات وحلقات التكرار ٤١
الوحدات البرمجية ٥٩
ماذا تعلمتُ ؟ ٦٩
المراجع والمصادر ٧٢

الوحدة الثانية الحوسبة السحابية

مقدمة في الحوسبة السحابية ٧٧
خدمات الحوسبة السحابية ٨٣
التخزين السحابي ٨٥
النماذج الإلكترونية ٩٧
أمن الحوسبة السحابية ١٠٧
ماذا تعلمتُ ؟ ١١٠
المراجع والمصادر ١١٢

الوحدة الأولى

البرمجة

مقدمة

تُعدُّ لغة البايثون من اللغات البرمجية عالية المستوى، التي تَستخدمُ أسلوب البرمجة الكائنية الموجَّهة - Object Oriented Program (OOP) ، مفتوحة المصدر القابلة للتطوير. وتتميز بسهولتها وقدرتها على التعامل مع منصات وأنظمة تشغيل مختلفة مثل: Windows، ولينكس Linux، وغيرها. وتُستخدم بشكل واسع في العديد من المجالات مثل: تطوير مواقع الويب، والتطبيقات المتعلقة بالذكاء الاصطناعي، وتصميم الواجهات الرسومية وبرامج ثلاثية الأبعاد وغيرها.

تهدف هذه الوحدة إلى تعريفك مفهوم لغة البايثون، وكيفية التخطيط لإنشاء برنامج ما، بالإضافة إلى استخدام الدوال والجمل الشرطية، وحلقات التكرار في التحكم بالبرنامج الذي تنشئه، كما ستتعرف إلى كيفية استخدام البايثون في إنشاء رسومات متجهة.

يتوقَّع منك عزيزي الطالب / عزيزتي الطالبة بعد دراسة هذه الوحدة أن:

- تتعرف إلى مفهوم البرمجة.
- تتعرف إلى أنواع البيانات وكيفية تحويلها.
- تستخدم الخوارزميات للتخطيط وكتابة البرامج.
- تنفذ مهام محددة باستخدام الجمل الشرطية.
- توظف حلقات التكرار لتكرار تنفيذ مهام برمجية.
- تستورد وحدات برمجية لتصميم واجهة رسومية.

الدرس الأول

مقدمة في لغة البرمجة

لغـة البرمجة Programming Language عبارة عن مجموعة أوامر وتعليمات تُستخدم فـي إنشـاء برنامج بهـدف تنفيذ مهام محـددة. ويعمـل المبرمجون على كتابـة التعليمات باستخدام لغات برمجية عالية المستوى، صُمِّمت لتكون سهلة في القراءة والفهم. ومن الأمثلة عليها: Java، وC++، و Python.

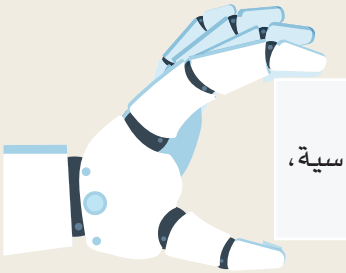
نشاط ١: لماذا نتعلم لغة البايثون؟

شـاهد مقطع الفيديو (مقدمة إلى البرمجة)، واسـتعن بالملف المرفـق (Programming.pdf)، لمناقشة ما يلي:

١. أهمية تعلم البرمجة.
٢. مفهوم برمجة البايثون.
٣. مزايا استخدام لغة بايثون.

ابحث

الفرق بين لغات البرمجة عالية المستوى High Level Languages ، ولغات البرمجة منخفضة المستوى Low Level Languages .



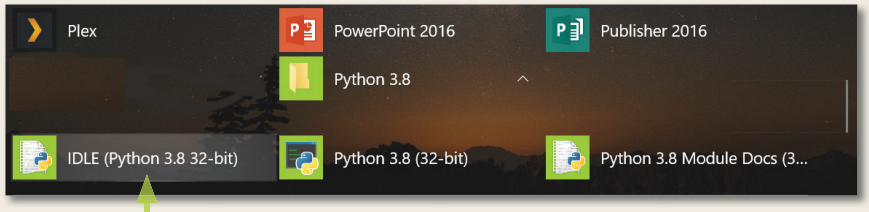
تذكر!
جميع لغات البرمجة تشترك في المفاهيم والمبادئ الأساسية، إن أتقنتها فمن السهل عليك الانتقال إلى لغة أخرى.

استكشاف بيئة الأوامر في بايثون

تتضمن البايثون عدة تطبيقات، أهمها بيئة التطوير والتعليم المتكاملة IDLE وهي اختصار لـ (Integrated-Development and Learning Environment)، المستخدمة في إنشاء البرامج وتطويرها.

ولفتح بيئة التطوير IDLE، اتبع ما يلي:

- من قائمة ابدأ Start، افتح مجلد Python، لتظهر لك عدة خيارات.



انقر على
IDLE Python

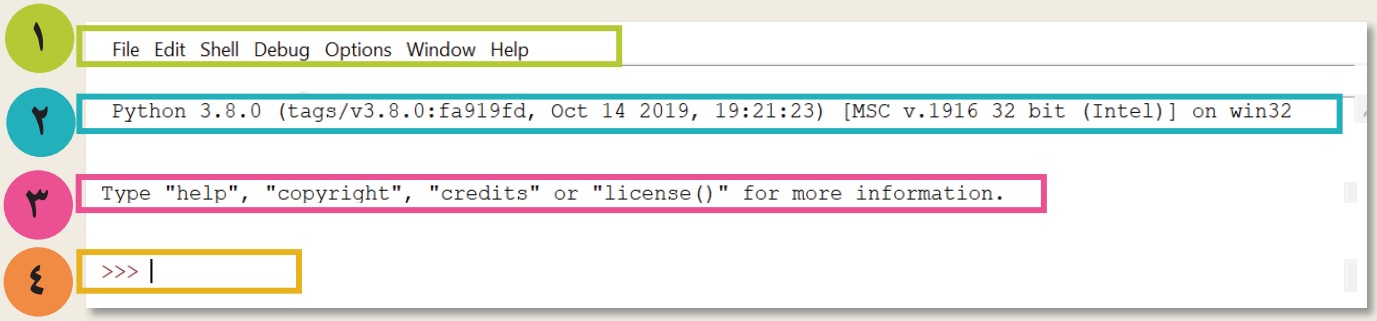
يمكنك تحميل برنامج البايثون من الموقع www.python.io

ستلاحظ أن بيئة التطوير IDLE تحتوي على نوعين من النوافذ، هما:

أولا: نافذة الأوامر Python Shell:

وهي النافذة الافتراضية التي تظهر لك بمجرد النقر على التطبيق، وتُعدُّ نافذة تفاعلية؛ لأنه يتم فيها كتابة أمر برمجي وعرض الناتج مباشرة في السطر التالي عند الضغط على مفتاح الإدخال Enter، وتتكون مما يلي:

عليك التأكد من أنك تستخدم أحدث إصدار للبايثون، لوجود اختلاف في الأوامر بين إصداراته، كالاختلاف بين الإصدار الثاني والثالث مثلًا، مما قد يصعب عليك تنفيذ بعض أوامر الإصدارات السابقة في الإصدارات الحديثة.



١ شريط القوائم Menu bar: يتضمن مجموعة قوائم لتنفيذ الأوامر، مثل: حفظ البرنامج، والطباعة، وتتبع الأخطاء، وإعادة ضبط البيئة البرمجية وغيرها.

٢ سطر المعلومات Information line: يحتوي على معلومات تتعلق برقم الإصدار للبرنامج المثبت على الجهاز، وتاريخ بدء النشر، ومواصفات الجهاز.

٣ سطر الأوامر البرمجية Command line: عبارة عن أوامر برمجية، عند كتابتها ستعرض لك المساعدة وحقوق النشر وشروط استخدام البرنامج.

٤ موجِّه الإدخال Input prompt (>>>): يدل على استعداد البرنامج لتلقي الأوامر، ويتم كتابة الأوامر البرمجية بعد علامة موجه الإدخال.

استكشف

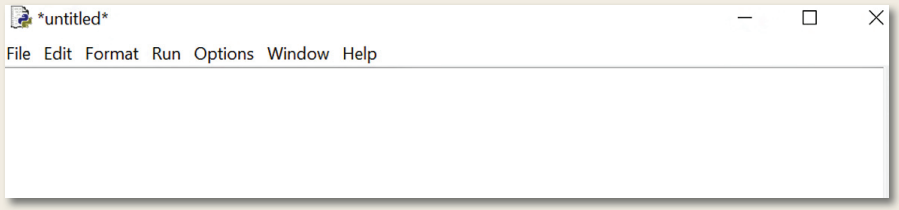
حقوق لغة البايثون من خلال كتابة الأمر **copyright** عند موجِّه الإدخال.

ثانيًا: نافذة المحرِّر Editor

يتم فيها إنشاء ملفات جديدة وكتابة الأوامر البرمجية بداخلها، وحفظه ثم تشغيله بالضغط على أمر Run Module، من قائمة Run، ليظهر الناتج بالكامل ككتلة واحدة في شاشة الأوامر Python Shell.

ولفتح نافذة المحرر Editor، اتبع الآتي:

- اختر أمر **ملف جديد New File، من قائمة ملف File**، لتظهر لك نافذة المحرر كما يلي:



استكشف

- الفرق بين القوائم الموجودة في نافذة الأوامر **Python Shell**، والموجودة في نافذة المحرر **Editor**.
- قائمة المساعدة **Help** وما تحويه من تعليمات للبايثون.

إدخال البيانات

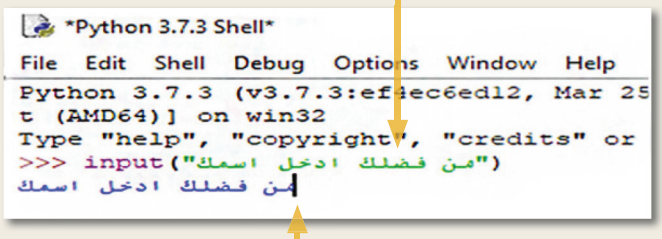
تُستخدم دالة إدخال **input ( )** لإدخال أنواع مختلفة من البيانات التي تتعامل معها لغة البايثون حيث تظهر للمستخدم رسالة تطلب منه إدخال بيانات محددة، ويتم كتابة الأمر كالآتالي:

```
input ("نص الرسالة الموجهة")
```

ولاستخدام دالة إدخال( ) input في كتابة رسالة تطلب من المستخدم إدخال اسمه، اتبع الآتي:

- تأكد من فتح نافذة الأوامر Python Shell.
- اكتب دالة إدخال **input( )**، ثم:

١ اكتب النص "**من فضلك أدخل اسمك**" بداخل الأقواس، ثم اضغط على مفتاح الإدخال Enter



ستلاحظ تنفيذ الرسالة في السطر التالي للأمر مباشرة، مع ظهور مؤشر الفأرة في انتظار لاستقبال إدخال البيانات

- أدخل اسمك، ثم اضغط على مفتاح **الإدخال Enter**.

ستلاحظ أنه يتم التمييز بين الأمر، والنص المكتوب بداخله، والناتج المخرج بألوان مختلفة.

- احفظ الملف الذي أنشأته بامتداد (.py).

فكر

كيفية كتابة أمر يُظهر للمستخدم رسالة تطلب منه إدخال اسمه، ثم رسالة أخرى لإدخال عمره في نفس السطر البرمجي .

معلومة تهمك

- يتم كتابة الأوامر البرمجية باستخدام الحروف الإنجليزية الصغيرة Small letters دائمًا؛ لأن البرنامج لا يتعرف على الأوامر عند كتابتها بحروف كبيرة Capital letter.
- يتم كتابة النصوص بين علامتي التنصيص المزدوجة "**نص**"، أو علامتي التنصيص المفردة '**نص**'.

إخراج البيانات

تُستخدم دالة طباعة **print ( )** لإظهار رسالة معينة وطباعتها على الشاشة، أو طباعة ناتج لعملية ما وغيرها بعد معالجتها، ويتم كتابتها كالآتي:

```
print("البيانات المطلوب إظهارها")
```

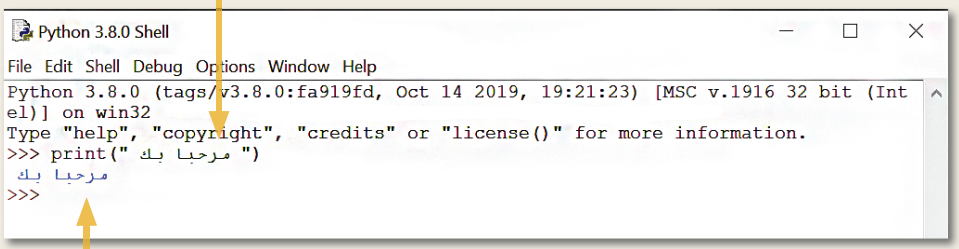
أو

```
print( اسم المتغيِّر)
```

لإنشاء برنامج يعمل على طباعة عبارة ترحيبية، اتبع الآتي:

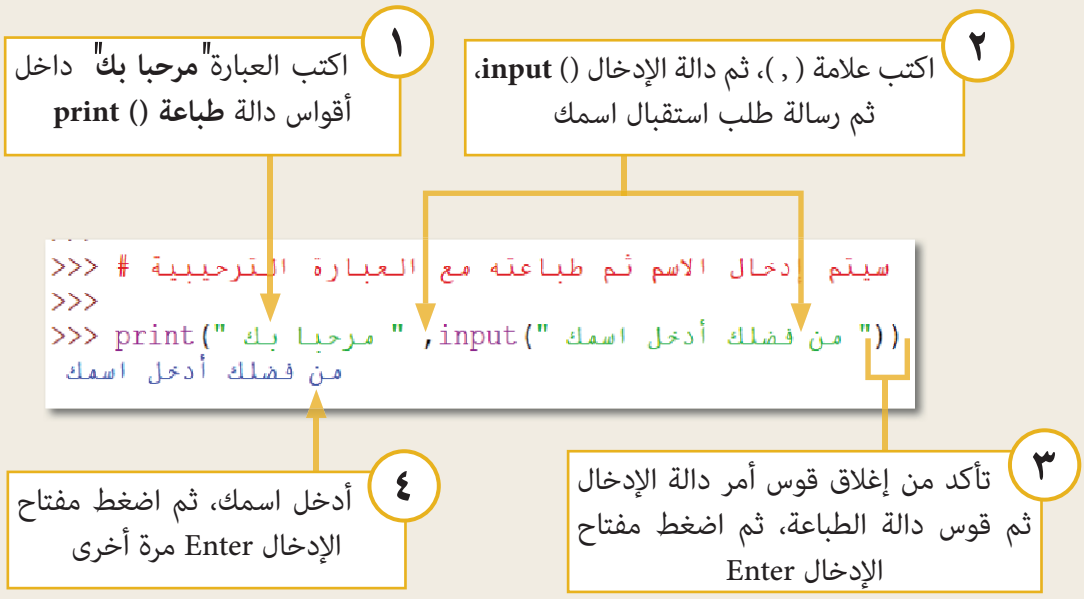
- تأكد من فتح نافذة الأوامر Python Shell.
- اكتب دالة طباعة **print( )**، ثم:

١ اكتب العبارة الترحيبية "**مرحبا بك**" داخل أقواس دالة طباعة () **print**، ثم اضغط مفتاح الإدخال Enter

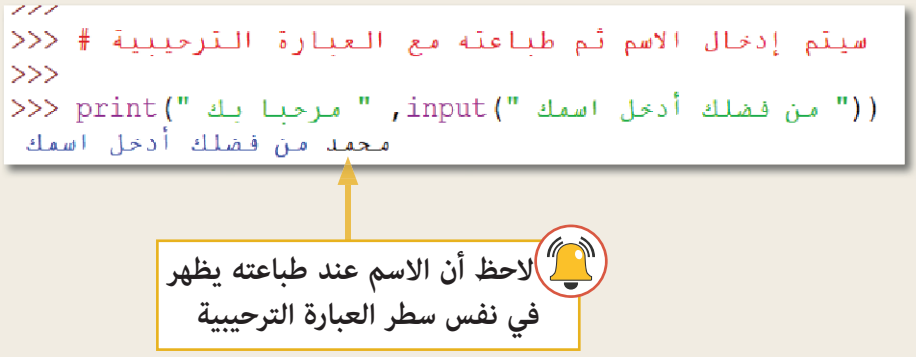


لاحظ نتيجة تنفيذ البرنامج

ولطباعة العبارة الترحيبية، بعد إدخال اسم المستخدم، اتبع الآتي:



ستلاحظ أن تنفيذ البرنامج للأوامر قد بدأ من الأقواس الداخلية، ثم الأقواس الخارجية.

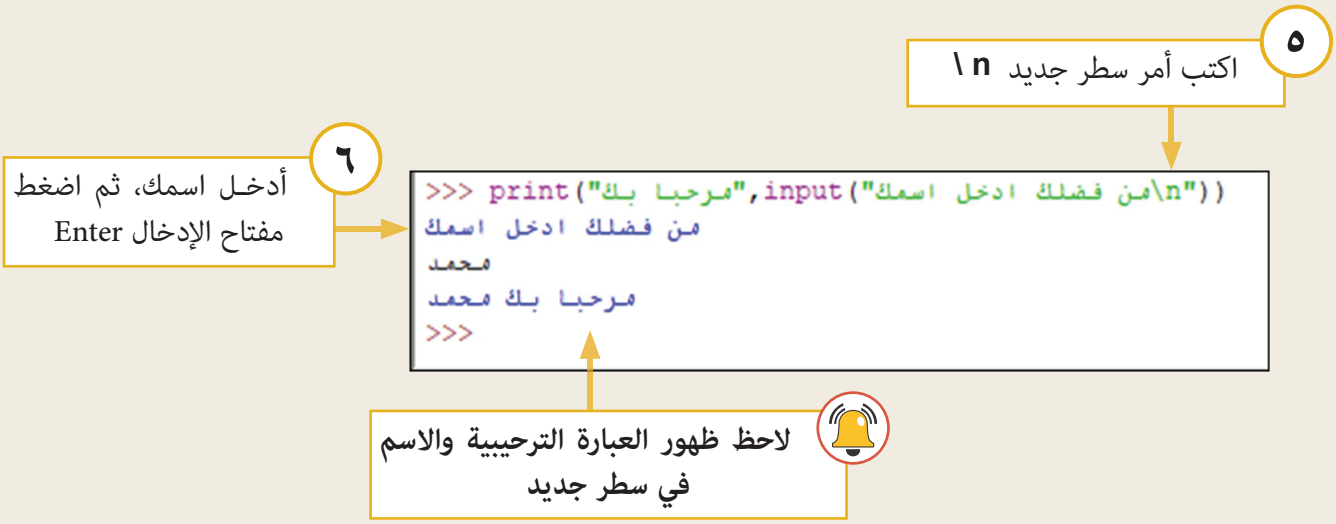


معلومة تهمك

- تُستخدم علامة **(,)** للربط بين أكثر من كلمة أو أكثر من جملة بجانب بعضها.
- **تُستخدم علامة (#) في حالتين:**
  - لإضافة توضيح في البرنامج، حيث يتم تجاهله عند تنفيذ البرنامج.
  - لإيقاف تنفيذ أمر برمجي عند إضافتها قبل الأمر كما يلي: **#print(name)**

لوضع الاسم المُدخَل في سطر جديد new line باستخدام أمر (\n)، اتبع ما يلي:

- أعد كتابة الأمر البرمجي السابق كالآتي:



المتغيِّرات Variables

المتغيِّر عبارة عن مكان في الذاكرة يُستخدم لتخزين بيانات وقيم قابلة للتغير أثناء تنفيذ البرنامج، حيث يأخذ كل متغير اسمًا خاصًا به يتم استدعاؤه من خلاله لمعرفة ما بداخله من بيانات، واستخدامها لاحقًا في البرنامج.

عند إنشاء متغير، لابد أن يتكون مما يلي:

- **اسم المتغير Variable name**: يجب أن يكون لكل متغير اسم فريد غير مكرَّر.
- **قيمة المتغير Variable value**: وهي القيمة التي تُخزَّن في المتغير.
- **نوع البيانات Data type**: وتكون البيانات المخزنة إما سلاسل نصية **String** أو رقمية (عددًا صحيحًا **Integer** أو عددًا عشريًا **Float**) وغيرها، وتحديد نوع البيانات يساعد على تحديد حجم مساحة التخزين اللازمة لحفظ المتغير.

معلومة تهمك

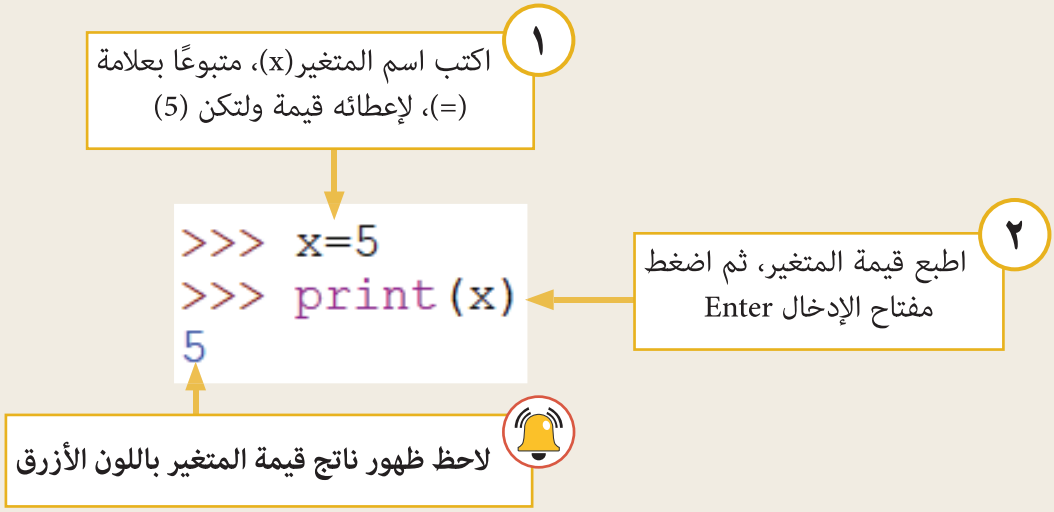
- **عند تعريف المتغير، لابد من مراعاة الشروط الآتية:**
  - أن يبدأ بحرف أو علامة (_).
  - أن لا يبدأ برقم.
  - أن لا يحتوي على رموز أو علامات خاصة مثل: (=، ؟، %، $، ....).
  - أن لايحتوي على مسافات.
  - أن لا يكون أحد الكلمات المحجوزة في لغة البرنامج، مثل: and أو True وغيرها.
  - أسماء المتغيرات تُعدُّ حساسة لحالة الأحرف، على سبيل المثال: name، و Name ،و NAME هي ثلاثة متغيرات مختلفة.

ويتم إنشاء متغير بطريقتين:

أولا: إسناد مباشر:

يتم عن طريق كتابة اسم المتغير، وإدخال قيمة مباشرة له. ولتعريف متغير وليكن (x) يحتوي على قيمة معينة وطباعته على الشاشة، اتبع الآتي:

- تأكد من فتح نافذة الأوامر Python Shell.



جرّب تعريف متغيرين بنفس القيمة كما يلي: x=y=5 ، ثم استخدم أمر print لطباعة الناتج. **ماذا تلاحظ؟**

استكشف

كيفية تعريف متغيِّرَيْن وإسناد قيمتين مختلفتين لكل منهما في آن واحد وطباعة ناتجهما.

ثانيا: إسناد باستخدام دالة إدخال ( ) input:

تُستخدم لاستقبال قيمة متغيرة يدخلها المستخدم عند تشغيل البرنامج وتخزينها في متغير، ولتعريف متغير وليكن (a) يطلب منك إدخال عدد صحيح، اتبع الآتي:

١ اكتب اسم المتغير(a) متبوعًا بعلامة (=)، ثم استخدم دالة **الإدخال** ( ) **input** لإظهار رسالة تطلب منك إدخال عدد صحيح، ثم اضغط مفتاح الإدخال Enter

```
>>> a=input('أدخل عدد صحيح ' )
أدخل عدد صحيح 5
>>> print(a)
5
```

٢ أدخل العدد الصحيح، ثم اضغط مفتاح الإدخال Enter

٣ اطبع قيمة المتغير (a)

تحويل البيانات

تُستخدم **دالة** ( )**type** لمعرفة نوع البيانات المخزنة في المتغيرات، ولعمل ذلك اتبع الآتي:

في نافذة الأوامر **Python Shell**:

١ اكتب اسم دالة **type(a)**، ثم اضغط مفتاح الإدخال Enter

```
>>> a=input('أدخل عدد صحيح ' )
أدخل عدد صحيح 5
>>> print(a)
5
>>>
>>> type(a)
<class 'str'>
```

لاحظ أن البيانات المخزنة في المتغير (a) من نوع سلاسل نصية String

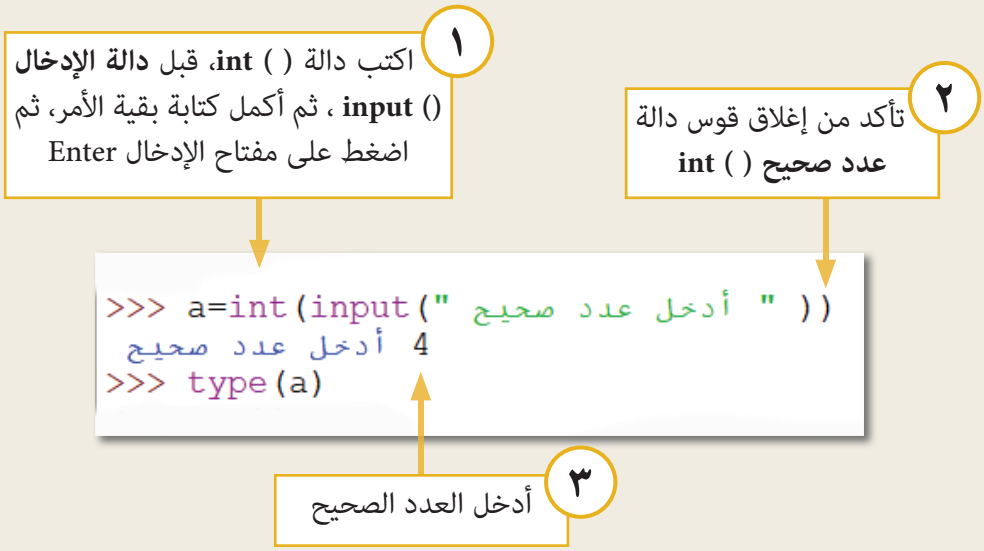
فكر

لماذا تكون البيانات المدخلة باستخدام دالة إدخال ( ) **input** من نوع سلاسل نصية **String**؟

لتحويل نوع البيانات من سلاسل نصية إلى عدد صحيح باستخدام دالة ( ) int، اتبع الآتي:

في نافذة الأوامر Python Shell:

- أعد كتابة الأمر البرمجي السابق كالتالي:



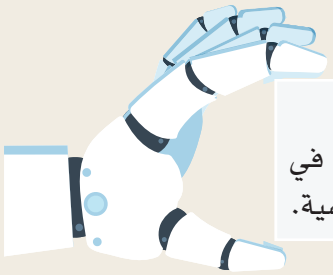
- تأكد من كتابة دالة type (a)، واضغط على مفتاح الإدخال Enter مرة أخرى، و**لاحظ ما نوع البيانات المدخلة؟**

معلومة تهمك

- **يتم تغيير نوع المتغيرات في برنامج بايثون، باستخدام الدوال الآتية:**
  - دالة ( ) str: لتحويل نوع المتغير إلى سلسلة نصية string.
  - دالة ( ) int: لتحويل نوع المتغير إلى عدد صحيح integer.
  - دالة ( ) float: لتحويل نوع المتغير إلى عدد عشري float.

نشاط ٢: تسجيل دخول المستخدم

- **أنشئ برنامجًا في نافذة الأوامر** Python Shell، **يطلب من المستخدم إسناد قيمة للمتغير باستخدام دالة إدخال** input ( )، **بحيث ينفذ المهام الآتية:**
  - **المتغير الأول:** يظهر رسالة تطلب من المستخدم إدخال اسم المستخدم Username.
  - **المتغير الثاني:** يظهر رسالة تطلب من المستخدم إدخال رقم سري Password، على أن يكون قيمة الرقم المدخل من نوع عدد عشري Float.
- **تأكد من نوع البيانات المخزنة في متغير الرقم السري باستخدام دالة** type ( ).



تذكر!
تَعَلُّمُ لغة البرمجة يساعدك على رفع مستوى ذكائك في التفكير، وسرعتك في اتخاذ القرارات في حياتك الشخصية.

الدرس الثاني

التعبيرات والشروط

التعبيـرات عبارة عن مزيج بين متغيـرات وثوابت يتم استخدامها إما لحساب قيم وتخزينها، أو للتحقق من صحة البيانات، أو لدمج أكثر من كلمة في جملة واحدة. أما الشروط فهي تحدِّد طريقة عمل البرنامج، وآلية تنفيذه للأوامر باستخدام ما يسمى بالجمل الشرطية.

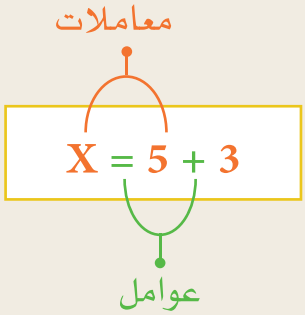
خطوات إنشاء برنامج بلغة البرمجة

لكتابة أي برنامج بلغة البرمجة، عليك اتباع الخطوات الآتية:

1. **التحليل**
   تعريف المشكلة وتحديد الحل المناسب لها.
2. **التخطيط**
   كتابة خطوات الحل مرتبة بشكل منطقي في مخطط انسيابي.
3. **كتابة الأمر البرمجي**
   تحويل الخطوات إلى أوامر برمجية مكتوبة بلغة البرمجة.
4. **تنفيذ البرنامج**
   تشغيل البرنامج والتأكد من أنه يعمل بشكل صحيح.

التعبيرات Expressions

تتكون التعبيرات من قيم ثابتة أو متغيرة تسمى **معاملات Operands**، يتم الربط بينها عن طريق رموز أو كلمات خاصة تسمى **عوامل Operators**، تُستخدم حسب الغرض من التعبير أو نوع العملية المراد إجراؤها، على سبيل المثال:



وتتعامل لغة البايثون مع عدة أنواع من العوامل، مثل:

العوامل الحسابية Artihmetic Operators

تُستخدم لإجراء عمليات حسابية باستخدام العوامل الحسابية المختلفة، كما يوضحها الجدول الآتية:

العملية	العامل	الأمثلة
الضرب	*	4*5=20
القسمة	/	6 / 3 = 2
الجمع	+	4 + 3 = 7
الطرح	–	7 -2 = 5
باقي القسمة	%	5 % 2 = 1

ولاستخدام العامل الحسابي في إنشاء برنامج يعمل على إيجاد المتوسط الحسابي لطولي شخصين، طول الأول = ١٦٨ سم، وطول الثاني =١٧٥ سم، اتبع الآتي:

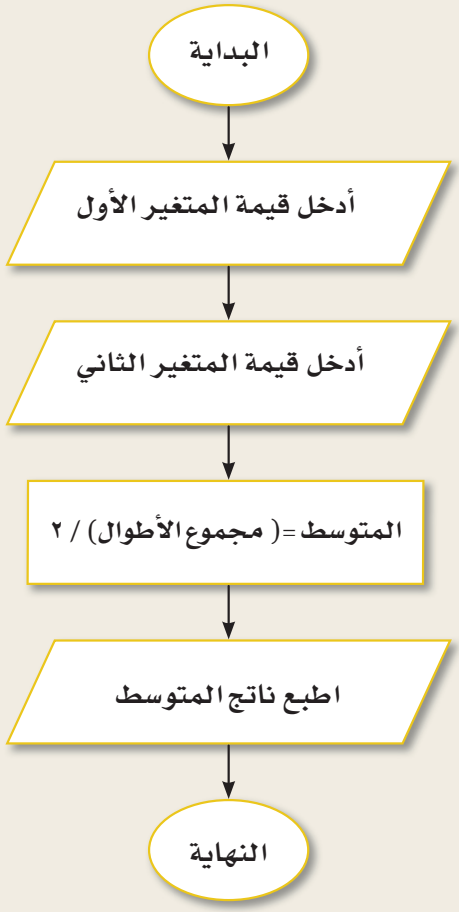
- اتبع خطوات إنشاء برنامج، لتحديد الآتي:

أولاً: التحليل

المعطيات: طول الشخص الأول = ١٦٨سم ،طول الشخص الثاني= ١٧٥سم

المطلوب: إيجاد متوسط الطولين.

ثانيا: التخطيط



ثالثا: كتابة الأوامر البرمجية

من نافذة الأوامر Python Shell:

- اختر أمر **ملف جديد New File**، لفتح نافذة جديدة في **IDLE المحرّر**، وأحفظ الملف باسم.
- عرّف المتغيرات اللازمة لتخزين القيم الآتية:
  - x: وأسند لها قيمة طول الشخص الأول.
  - y: وأسند لها قيمة طول الشخص الثاني.
  - average: لتخزين ناتج المتوسط الحسابي.

- أطبع ناتج العملية باستخدام دالة **print ( )** ، مع كتابة عبارة (متوسط الأطوال) ، كما هو موضح لك:

```
x=168
y=175
average=(x+y)/2
# دالة الإخراج ستطبع قيمة المتغير average، مع عبارة متوسط الأطوال
print("متوسط الأطوال =" , average)
```

رابعًا : تنفيذ البرنامج

- شغّل المقطع البرمجي باستخدام أمر **Run Module** من قائمة **Run**، أو اضغط على مفتاح F5 من لوحة المفاتيح.

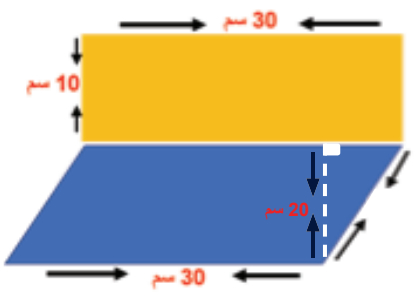
يمكنك معرفة اسم ملف البرنامج الذي تشغله باستخدام أمر Run Module، من خلال السطر الموجود أعلاه الذي يحتوي على اسم ملف البرنامج المُنَفِّذْ

```
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/hp/AppData/Local/Programs/Python/Python38-32/average ==========
متوسط الأطوال = 171.5
>>>
```

لاحظ أن تنفيذ البرنامج وظهور ناتج المتوسط الحسابي يكون في نافذة الأوامر Python Shell

نشاط ٣: حساب مساحة الأشكال

- **تأمل الشكل التالي المكوَّن من مستطيل ومتوازي أضلاع، ثم اتبع خطوات إنشاء برنامج يعمل على حساب المساحة الكلية للشكلين.**



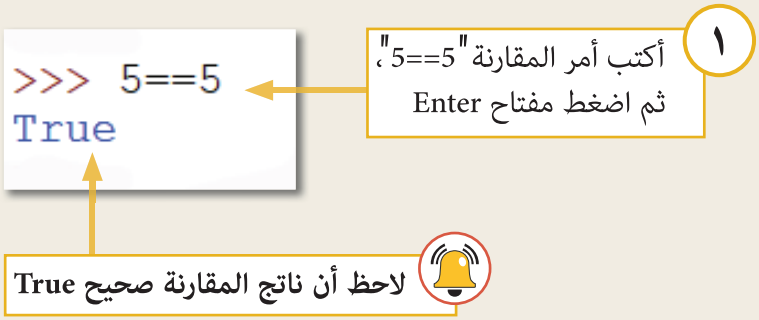
٢ عوامل المقارنة Comparison Operators

تُستخدم لإجراء مقارنة بين قيمتين، والحصول على إحدى النتيجتين، إما صحيح True إذا كانت الإجابة نعم، أو خطأ False إذا كانت الإجابة لا، باستخدام رموز المقارنة الآتية:

العملية	العامل	أمثلة	توضيح الأمثلة
يساوي	==	a==b	هل قيمة a تساوي قيمة b؟
لا يساوي	!=	a!=b	هل قيمة a لا تساوي قيمة b؟
أصغر من	<	a<b	هل قيمة a أصغر من قيمة b؟
أصغر من أو يساوي	<=	a<=b	هل قيمة a أصغر أو تساوي قيمة b؟
أكبر من	>	a>b	هل قيمة a أكبر من قيمة b؟
أكبر من أو يساوي	>=	a>=b	هل قيمة a أكبر أو تساوي قيمة b؟

ولاستخدام عوامل المقارنة في كتابة برنامج يتحقق من تساوي قيمتين، اتبع الآتي:

- تأكد من فتح نافذة الأوامر Python Shell.



جرّب كتابة أمر المقارنة: "hello"=="Hello"، ثم اضغط مفتاح الإدخال، ماذا تلاحظ؟ فسّر ذلك.

٣ العوامل المنطقية Logical Operators

تُستخدم لفحص عدة شروط يتم تحديدها لتنفيذ أوامر معينة، فإذا كان ناتج تحقق الشرط صحيح True سيتم تنفيذ الأمر، وإذا كان ناتج تحقق الشرط خطأ False، فلن يتم تنفيذ الأمر أو يتم تنفيذ أمر آخر، كما يلي:

العملية	أمثلة	ناتج الفحص
and	6>4 and 5<6	يتحقق الشرط إذا كان كلا الشرطين صحيحين
or	8==8 or 6!=6	يتحقق الشرط إذا كان أحد الشرطين صحيحًا
not	not 3 >1	يعكس النتيجة، أي إذا كانت النتيجة صحيحة True فإنها ستحولها إلى خطأ False والعكس صحيح.

لاستخدام العوامل المنطقية في كتابة برنامج يقوم بالتحقق من تساوي قيمتين، اتبع الآتي:

١ اكتب شرط المقارنة
1==1 and 4!=4،
ثم اضغط مفتاح ضبط الإدخال Enter

```
>>> 1==1 and 4!=4
False
```

لاحظ أن ناتج المقارنة خطأ False،
لأن أحد الشرطين خطأ

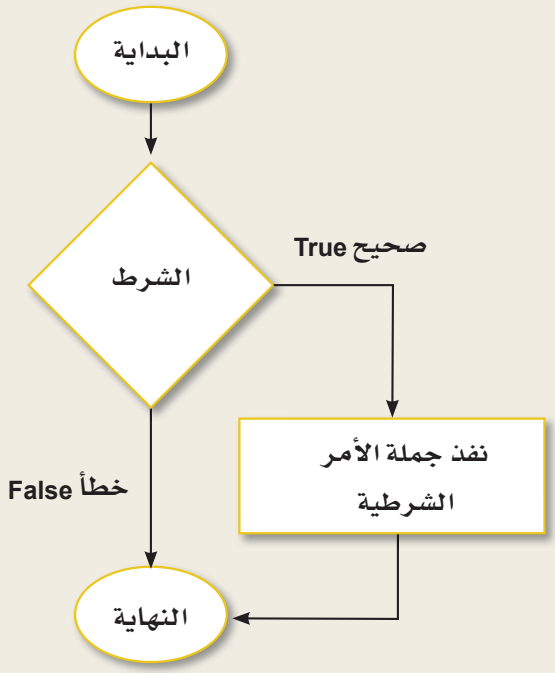
جرّب استخدم عامل (or)، في تنفيذ التعبير المنطقي السابق، ولاحظ الناتج.

الجمل الشرطية

تُستخدم لتنفيذ جزء محدَّد من البرنامج في حال تحقق شرط معين، بحيث تكون نتيجة التحقق قيمة منطقية معينة، إما صحيحًا True أو خطأ False، وتدعم لغة بايثون الأنواع التالية من الجمل الشرطية:

١ جملة if الشرطية:

يتم فيها التحقق من صحة الشرط المحدَّد، فإذا كان الشرط صحيحًا True ينفذ جملة الأمر الشرطية، وإذا كان خطأ False يتم تجاهله، كما يوضحه المخطط الآتية:



ولإنشاء برنامج يقارن بين قيمة متغيرين (a ، b)، ويطبع عبارة ("b is greater than a") عند تحقق شرط أن قيمة b أكبر من قيمة a، اتبع ما يلي:

- افتح (**ملف جديد New File**) في **IDLE المحرّر**، واحفظ الملف باسم.
- عرِّف المتغيِّرَيْن a، b، وأسند لهما قيمة:

```
a,b= 33 , 200
if b>a:   # الجملة الشرطية للمقارنة بين القيمتين
    print(" b is greater than a" )   # الأمر الذي سيتم تنفيذه في حالة تحقق الشرط
```

لاحظ وجود مسافة فارغة بشكل افتراضي عند الضغط على مفتاح الإدخال Enter

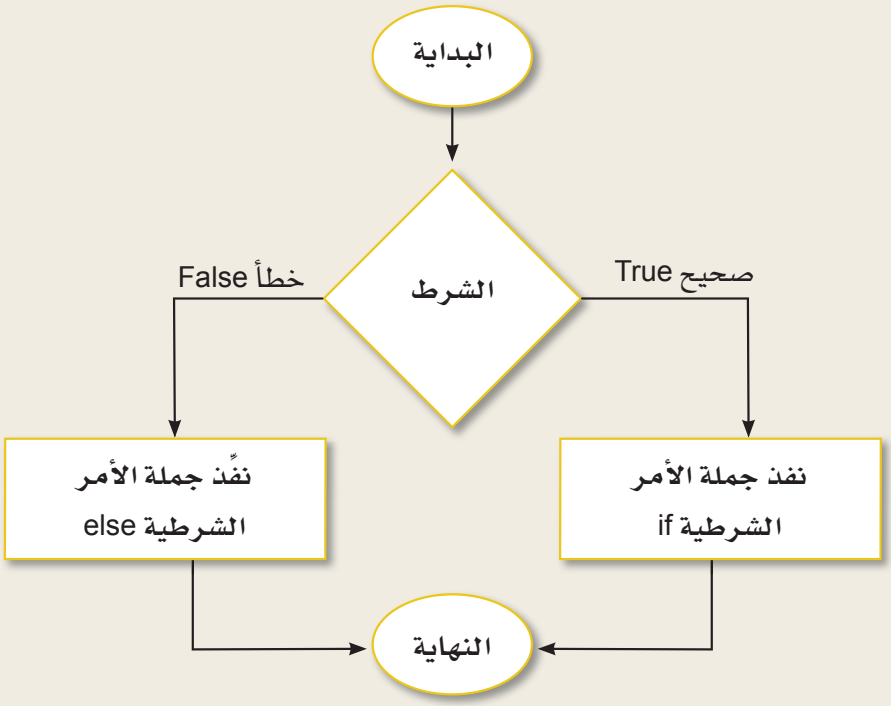
- شغِّل المقطع البرمجي، ولاحظ النتيجة التي ستظهر لك.
- **جرّب** تغيير عامل أكبر من (>) إلى أصغر من (<)، ثم نفِّذ المقطع البرمجي. **ماذا تلاحظ؟ فسّر ذلك.**

معلومة تهمك

- تُعدُّ (if) من الكلمات المحجوزة، أي أنها مخصصة لأداء وظيفة معينة في البرنامج، ويتم تمييز الكلمات المحجوزة في لغة البايثون باللون البرتقالي.
- تعتمد لغة البايثون على تحديد المسافة الفارغة بشكل تلقائي في بداية السطر البرمجي الذي يلي دالة الجملة الشرطية لتحديد الأوامر البرمجية التي تتبع الدالة، فإذا كانت العبارة بدون هذه المسافة الفارغة فإنه لا يمكن تنفيذ البرنامج وتظهر رسالة وجود خطأ.

٢ جملة if else الشرطية:

يتم فيها التحقق من صحة الشرط المحدَّد، فإذا كان الشرط صحيحًا True يتم تنفيذ جملة الأمر الشرطية الأولى، وإذا كان خطأ False يتم تنفيذ جملة الأمر الشرطية الثانية، كما يوضِّحه المخطط الآتي:



ولإنشاء برنامج يقارن بين قيمة متغيرين (a ، b)، ويطبع عبارة ("b is greater than a") عند تحقق شرط أن قيمة b أكبر من قيمة a أي أن النتيجة (True)، أو عبارة ("a is greater than b") في حالة عدم تحقق الشرط أي أن النتيجة (False)، اتبع الآتي:

في المقطع البرمجي السابق:

- تأكد من كتابة عامل أكبر من (>) ، في جملة الشرط **if**.

١ أضف تعليمة else إلى جملة الشرط، وتأكد من إلغاء المسافة البادئة للتعليمة.

```
a,b=33,200
if b>a:
    print("b is greater than a")
else:
      print("a is greater than b") #الأمر الذي سيتم تنفيذه في حالة عدم تحقق الشرط
```

لاحظ المسافة الفارغة التي تبدأ عند كتابة دالة ( )print التي تتبع جملة الشرط if، والمسافة الفارغة التي تبدأ عند كتابة دالة ( )print التي تتبع تعليمة else.

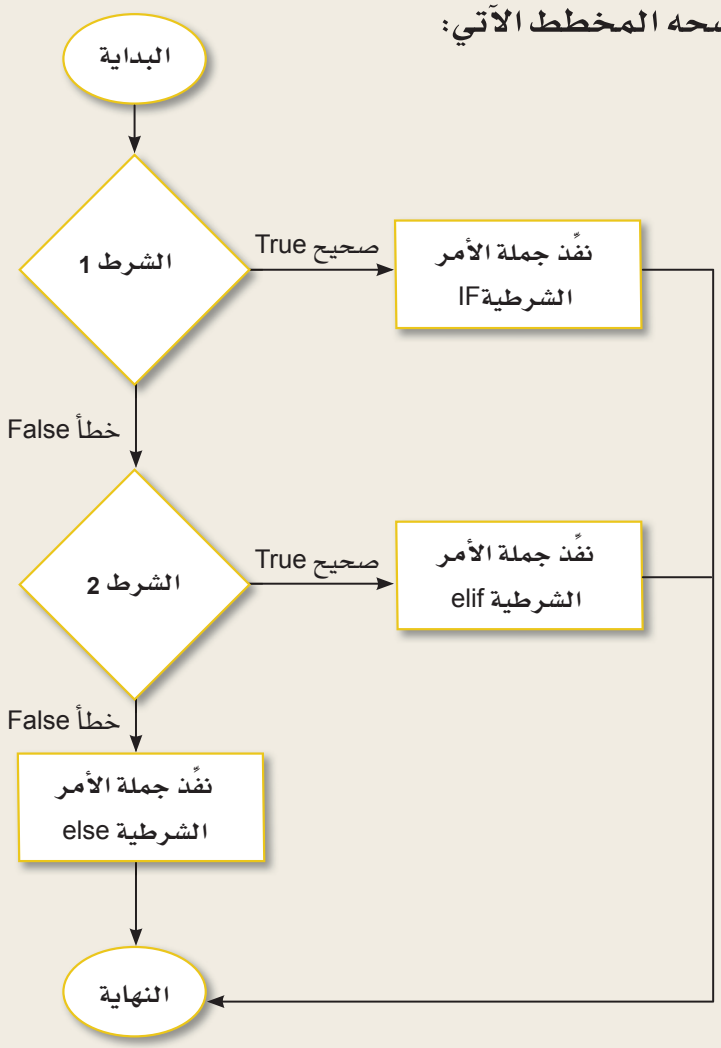
- شغّل المقطع البرمجي، ولاحظ النتيجة.

نشاط ٤: المقارنة بين القيم

- **اتبع خطوات إنشاء برنامج يقارن بين قيم لثلاث متغيرات، مع مراعاة الآتي:**
  - عرّف المتغيرات، وأسند لها قيمة مباشرة.
  - وظّف العملية المنطقية (or) لاختبار إذا كانت قيمة المتغير الأول أكبر من قيمة المتغير الثاني، أو إذا كانت قيمة المتغير الأول أكبر من قيمة المتغير الثالث.
  - إذا كان أحد الشرطين صحيحًا True، نفّذ أمر طباعة العبارة:
    ("one of the condition is true")
  - وإذا كان كُلٌّ من الشرطين خطأ نفّذ أمر طباعة العبارة:
    ("both conditions are false").

٣ جملة الشروط المتعددة (elif):

يتم فيها التحقق من صحة أكثر من شرط، فإذا كان الشرط الأول صحيحًا True ينفِّذ جملة الأمر الشرطية الأولى، وإذا كان خطأ False يتم التحقق من صحة شرط آخر لتنفيذ أوامر أخرى، وهكذا، كما يوضحه المخطط الآتي:



ولإنشاء برنامج يتحقق من صحة أكثر من شرط، اتبع الآتي:

- أضف تعليمة elif، وشرط التحقق إلى المقطع البرمجي السابق، ليبدو كالآتي:

```
a,b=300,200
if b>a:
    print("b is greater than a")
elif b==a: #جملة الشرط المستخدمة للتحقق من الشرط الثاني في حالة عدم تحقق الشرط الأول
      print("a and b are equal")
else:
    print("a is greater than b")
```

- شغّل المقطع البرمجي، ولاحظ النتيجة التي تظهر لك.

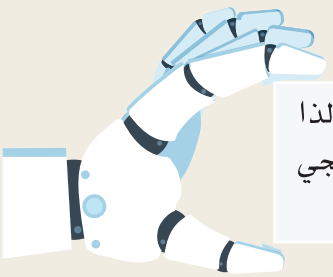
جرّب تغيير عامل المقارنة (>) لجملة الشرط if بعامل (==) ، و عامل المقارنة (==) لجملة الشرط elif بعامل (=!)

جرّب تغيير عامل المقارنة لجملة الشرط if من (>) إلى (==) ، وعامل المقارنة لجملة الشرط elif من (==) إلى (=!) ، ثم نفِّذ البرنامج. **ماذا تلاحظ؟**

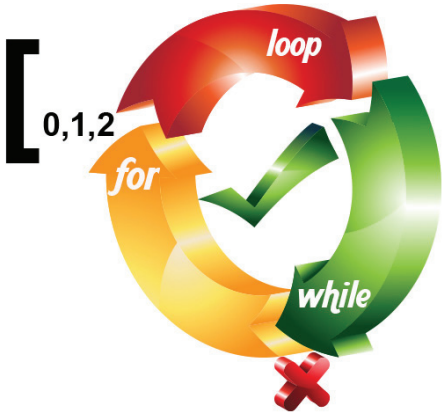
نشاط ٥: التحقق من نتيجة مادة

- **اتبع خطوات إنشاء برنامج يتحقق من نتيجتك في مادة تقنية المعلومات، مع مراعاة ما يلي:**
  - عرّف متغيرًا باسم (mark) ، وأسند له قيمة باستخدام دالة ( ) input.
  - وظّف العامل المنطقي (and) للتحقق من صحة الشروط، ومن ثَمَّ يطبع التقدير المناسب للدرجة المدخلة كما هو موضّح في الجدول التالي:

الدرجة	التقدير
٩٠ – ١٠٠	ممتاز
٨٠ – ٨٩	جيد جدا
٦٥ – ٧٩	جيد
٦٤ – ٥٠	مقبول
٤٩ أو أقل	ضعيف



قد لا تخلو البرمجة من بعض التحدِّيات؛ لذا احرص على الاستمتاع بكل سطر برمجي تكتبه، لتصبح محترفًا في هذا المجال.



الدرس الثالث

المصفوفات وحلقات التكرار

تُستخدم المصفوفة لجمع وتخزين مجموعة من القيم والعناصر، وتُعدُّ القوائم أكثر أنواع المصفوفات مرونة في التعامل مع عناصرها وإمكانية تحريرها، كما تتيح لك تكرار عناصرها أو تنفيذ عدة أوامر بناءً على شرط معين باستخدام حلقات التكرار.

المصفوفات Arrays

تُستخدم المصفوفة لجمع البيانات وتخزينها في الذاكرة تحت اسم متغير واحد بشكل منظم، بحيث تكون مرتبة بجانب بعضها، لتسهيل عملية جلب البيانات، وتطوير البرنامج. وتتعامل لغة البايثون مع أربعة أنواع من المصفوفات، هي:

القائمة List

تحتوي على مجموعة بيانات مرتبة، قابلة للتغيير، أي يمكن الحذف والإضافة عليها، وتسمح بتكرار بياناتها.

الصف Tuple

تحتوي على مجموعة بيانات مرتبة، غير قابلة للتغيير، وتسمح بتكرار بياناتها.

المجموعة Set

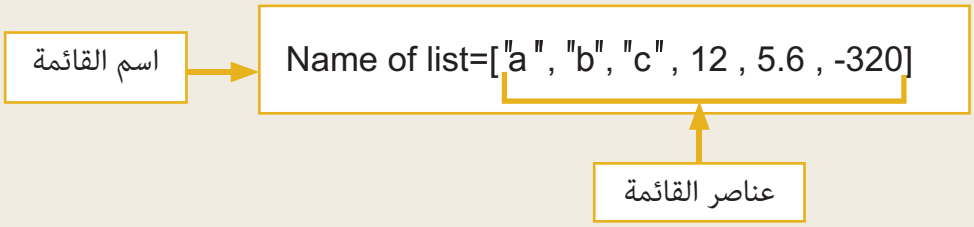
تحتوي على مجموعة بيانات غير مرتبة، أي تظهر بياناتها بشكل عشوائي عند تشغيل البرنامج، ولا تسمح بتكرار بياناتها.

القاموس Dictionary

تحتوي على مفاتيح ومجموعة بيانات غير مرتبة، وغير قابلة للتغيير، تسمح بتكرار بياناتها، ولا تسمح بتكرار المفاتيح.

التعامل مع القوائم Lists

تُعدُّ القوائم أكثر أنواع البيانات استخدامًا في لغة البايثون، ولا تشترط أن تكون عناصرها الموجودة فيها من نفس النوع، ويمكن كتابتها على شكل قيم متتالية، يفصل بينها بعلامة الفاصلة، وتكون القيم محصورة بين قوسين مربعين، كما في المثال الآتي:



ولمعرفة طول القائمة Length، والاستعلام عن عدد العناصر المخزنة فيها باستخدام دالة () len، اتبع ما يلي:

- افتح **ملف جديد New File**، لتفتح لك نافذة جديدة في **IDLE المحرِّر**، واحفظ الملف باسم.
- عرِّف القائمة ولتكن باسم (lst)، ثم اكتب العناصر الآتية:

تجنَّب استخدام كلمة (list) لتعريف اسم القائمة؛ لأنها تُعدُّ من الكلمات المحجوزة في لغة البايثون.

```
lst=["scince" , "IT" , "Maths" , 1998 , 2000]
print(2000 in lst)
lst[4]=2002      # تغيير قيمة العنصر الرابع
print(lst)       # طباعة عناصر القائمة بعد الإستبدال
```

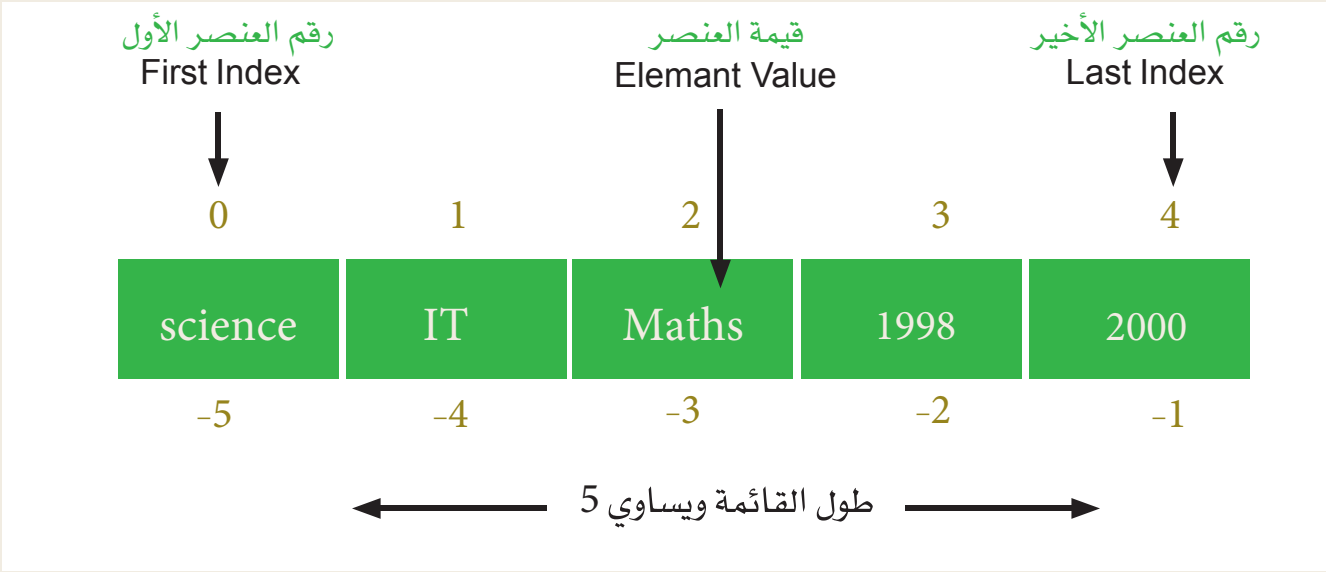
- شغِّل البرنامج باستخدام أمر Run Module.

لاحظ ظهور عدد عناصر القائمة

```
========== RESTART: C:/Users/hp/Desktop/lst.py ====
5
>>>
```

الوصول إلى عنصر محدّد

يمكنك الوصول إلى عنصر محّدد في القائمة من خلال تحديد رقم موقع العنصر في القائمة، يطلق عليه اسم **فهرس Index**، حيث يبدأ ترقيم فهرس العنصر الأول من جهة اليسار بالرقم **صفر**، ثم **واحد** وهكذا، ويكون رقم فهرس العنصر الأخير (طول القائمة – 1) أي:(**n-1**)، أما ترقيم فهرس العنصر الأول من جهة اليمين فيبدأ بـ(**-1**)، ثم (**-2**) وهكذا، كما يوضحه الشكل الآتي:



وللوصول إلى العنصر الثاني في القائمة على سبيل المثال باستخدام الفهرس، اتبع ما يلي:

- اكتب دالة الطباعة () print ، وبداخلها اسم القائمة ورقم فهرس العنصر المراد استدعائه، كمايلي:

```
lst=["science" , "IT" , "Maths" , 1998, 2000]
print(lst[1])    #للوصول إلى العنصر الثاني في القائمة
```

- شغّل البرنامج باستخدام أمر Run Module، ولاحظ العنصر الذي سيظهر لك.

لطباعة أكثر من عنصر من خلال تحديد نطاق بدء الفهرس وانتهاؤه، ولْيَكُن من الفهرس [1:4]، اتبع ما يأتي:

```
lst=["science" , "IT" , "Maths" , 1998, 2000]

print(lst[1:4])    # للوصول إلى عناصر القائمة من[1:4]
```

فكر

كيف يمكنك طباعة جميع عناصر القائمة بأكثر من طريقة؟

تحرير عناصر القائمة

يمكن إجراء العديد من العمليات على القائمة منها تحديث عناصر القائمة، وحذف عنصر وإضافة إلى القائمة.

تحديث عناصر القائمة Updating List Elements

يمكنك تغيير عنصر أو عدة عناصر في القائمة من خلال الوصول إلى العنصر المحدَّد عن طريق رقم الفهرس، وإعطاؤه قيمة جديدة باستخدام عامل الإسناد (=).

ولتغيير قيمة العنصر الخامس (2000)، إلى القيمة (2002) على سبيل المثال من القائمة lst، اتبع مايلي:

- تأكد من احتواء القائمة على العنصر **(2000)** باستخدام خاصية **in**، كما يلي:

```
lst=["scince" , "IT" , "Maths" , 1998 , 2000]
print(2000 in lst)    #  البحث عن العنصر الخامس باستخدام in،وطباعته
```

- شغّل البرنامج، ولاحظ أن النتيجة صحيحةٌ **True**؛ لأن العنصر موجود.
- حدّد رقم فهرس العنصر، وأسند له قيمة جديدة لتغييره، كما يلي:

```
lst=["scince" , "IT" , "Maths" , 1998 , 2000]
lst[4]= 2000   # تغيير قيمة العنصر الخامس
print(last)    # طباعة عناصر القائمة المحدثة
```

- شغّل البرنامج، ،ولاحظ قيمة العنصر الخامس بعد الاستبدال.

جرّب استبدال العنصر الثالث بـ **["Arabic"]**، والعنصر الرابع بـ **[2020]** بنفس الأمر البرمجي.

نشاط ٦: التحقق من وجود عنصر في القائمة

- اتبع خطوات إنشاء برنامج يتحقق من وجود عنصر معين في القائمة، مع مراعاة ما يلي:

- عرِّف القائمة ولتكن باسم (fruits).
- أدخل عناصر القائمة التالية: ("apple","orange","banana").
- تحقَّق من وجود العنصر ("cherry") في القائمة، لينفذ أحد الشرطين:

١- إذا كان ناتج الشرط صحيحًا True، سيطبع البرنامج العبارة : ("yes") مع جميع عناصر القائمة.

٢- إذا كان الشرط خطأ False، سيعمل البرنامج على استبدال قيمة العنصر الذي يحمل رقم الفهرس [2]، بالقيمة ("cherry") وطباعة عناصر القائمة بعد التحديث.

إضافة عنصر إلى القائمة Adding Elements

يمكنك إضافة عنصر أو أكثر، أو قائمة بداخل قائمة أخرى باستخدام الدالة ( ) append .

لإضافة عنصر (art) إلى القائمة (lst)، اتبع الآتي:

```
lst=["science" , "IT" , "Maths" , 1998, 2002]

lst.append("art")    # إضافة العنصر (art) إلى القائمة

print(lst)    # طباعة القائمة بعد الإضافة
```

- شغّل البرنامج، ولاحظ ظهور العنصر الجديد في القائمة.

جرّب حذف العنصر (1998) من القائمة باستخدام دالة( ) remove.

استكشف

كيفيَّة حذف عنصر من القائمة lst باستخدام أمر الحذف del، مستعينًا بقائمة المساعدة Help.

نشاط ٧: إضافة قائمة إلى أخرى

- **اتبع خطوات إنشاء برنامج يقارن بين عناصر قائمتين، مع مراعاة ما يلي:**

– عرّف القائمة الأولى **(a)**، وأدخل فيها قيم لأربعة أرقام صحيحة.

– عرّف القائمة الثانية **(b)**، وأدخل فيها قيمًا لأربعة أرقام صحيحة.

– قارن بين القائمتين باستخدام عوامل المقارنة، لتنفيذ أحد الشروط الآتية:

١– إذا كانت **a>b**، يضيف القائمة **(b)** إلى القائمة **(a)**، ثم يطبع القائمة **(a)**.

٢– إذا كانت **a<b**، يضيف القائمة **(a)** إلى القائمة **(b)**، ثم يطبع القائمة **(b)**.

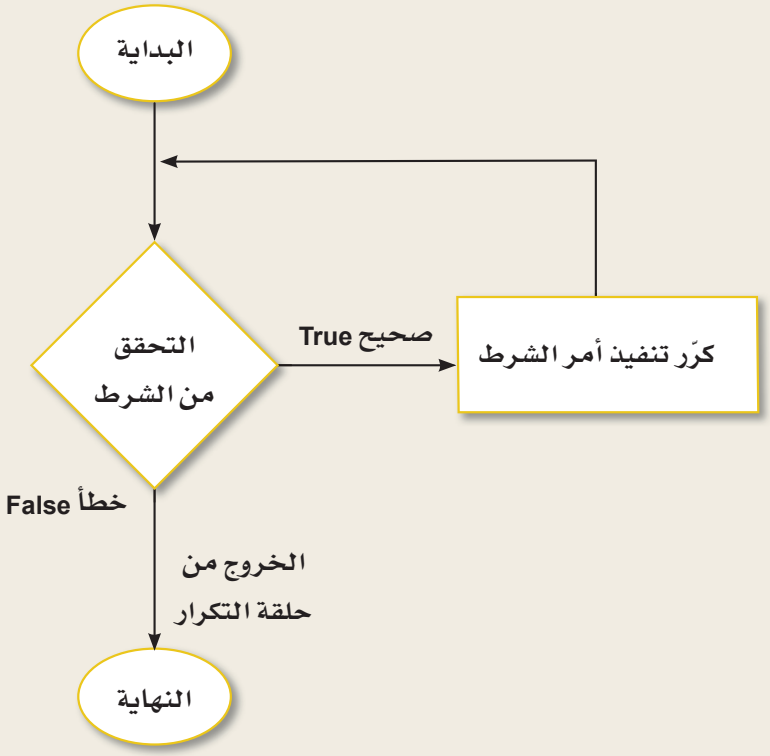
٣– إذا كانت كلتا القائمتين متساويتين، يضيف القائمة **(b)** إلى القائمة **(a)**، ثم يطبع عبارة **("a=b")**.

حلقات التكرار Loops

تُستخدم لتكرار جزء معين من الأوامر البرمجية إما لعدد معين من المرات، أو حتى يتم استيفاء تحقق الشرط. وتنفذ لغة البايثون حلقات التكرار باستخدام جملتين، هما:

١ جملة التكرار (While)

تُستخدم لتنفيذ مجموعة أوامر بشكل متكرِّر بناءً على شرط معين والتحقق من صحته، فإذا تحقق الشرط (True)، سيتم تنفيذ الأوامر الموجودة بداخل While، ثم يعود البرنامج للتحقق من صحة الشرط مرة أخرى، حتى تصبح نتيجة الشرط خطأ (False)، ليعمل البرنامج على تجاهل جملة التكرار والخروج منها، كما يوضحه المخطط الآتي:



ولإنشاء عدَّاد counter يبدأ العدّ من 1 إلى 10، ويطبع قيمة العدد، اتبع الآتي:

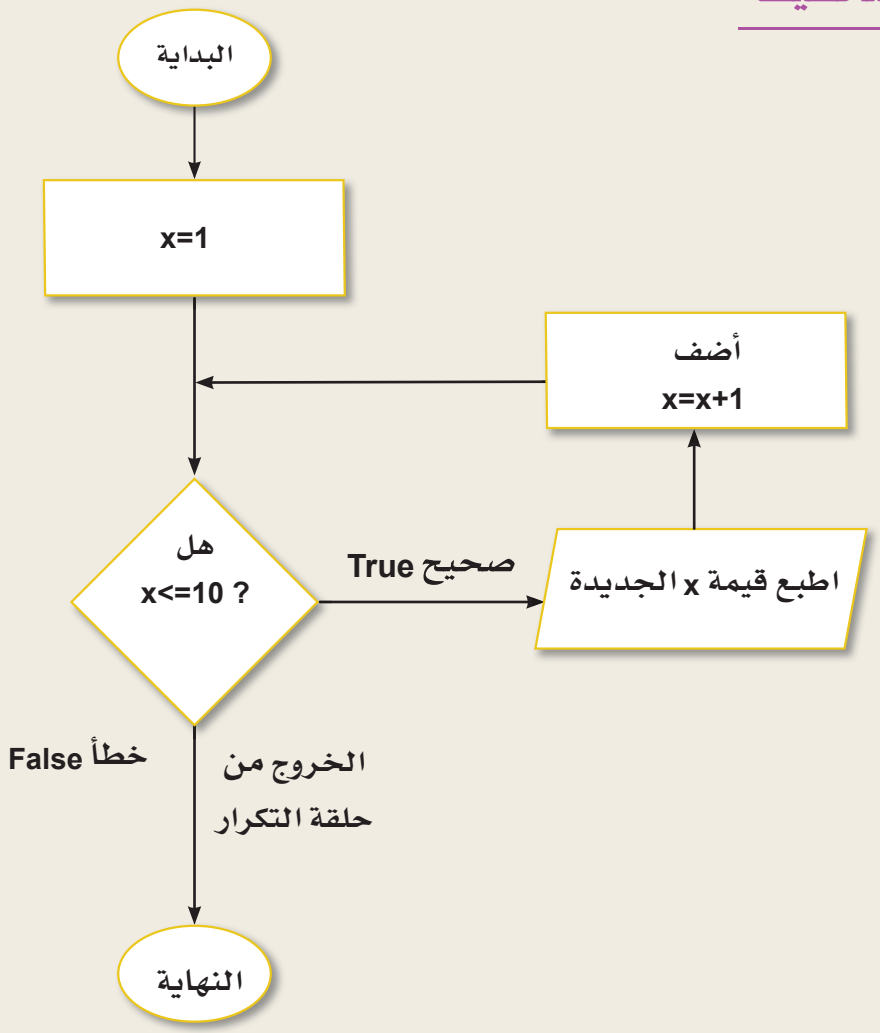
- اتبع خطوات إنشاء برنامج، لتحديد الآتي:

أولاً: التحليل

المعطيات: **الأعداد من 1 إلى 10 .**

المطلوب: **إنشاء عدّاد للعد من 1 إلى 10 وطباعته.**

ثانيا : التخطيط



ثالثا : كتابة الأوامر البرمجية

- افتح ملف جديد **New File**، في **IDLE** المحرّر، واحفظ الملف.
- عرّف متغيرًا وليكن **(x)**، وأسند له قيمة ابتدائية مباشرة **(x=1)**.
- كرّر عملية الزيادة في قيمة المتغير باستخدام حلقة التكرار **while**، كالآتي:

```
x=1          # القيمة الإبتدائية للمتغير
while x<=10:     # التحقق من الشرط
    print(x)     # طباعة القيمة الحالية للمتغير
```

- شغِّل البرنامج باستخدام أمر **تشغيل Run**، ولاحظ تكرار طباعة العدد **1**، **فسّر ذلك.**

لزيادة قيمة (x) بمقدار 1، وطباعة القيمة الجديدة:

```
x=1
while x<=10:
    print(x)
    x=x+1     # زيادة قيمة (x) واحد في كل دورة من حلقة التكرار
```

- شغّل البرنامج، ولاحظ قيم المتغير التي يتم طباعتها.

جرّب كتابة المقطع البرمجي السابق كما يلي، ثم شغّل البرنامج، **ماذا تلاحظ؟ فسّر ذلك.**

```
x=1
while x<=10:
    x=x+1
    print(x)
```

تنفيذ أمر آخر باستخدام تعليمة else

يمكنك تنفيذ أمر آخر بعد الانتهاء من حلقة التكرار، باستخدام تعليمة **else**، كالآتي:

- اتبع خطوات إنشاء برنامج، لتحديد الآتي:

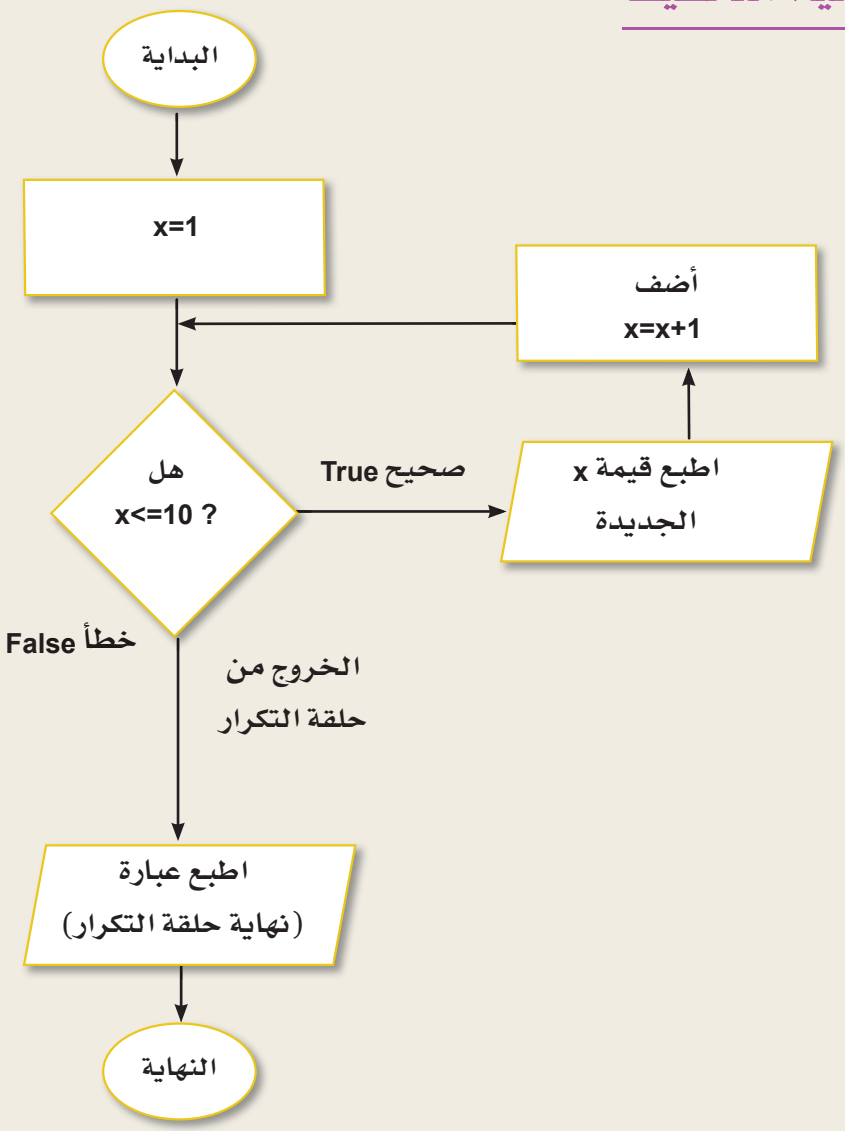
أولاً: التحليل

المعطيات: الأعداد من 1 إلى 10

المطلوب: ١- إنشاء عدّاد للعد من 1 إلى 10.

٢- تنفيذ أمر طباعة (« نهاية حلقة التكرار») بعد الانتهاء من علمية التكرار.

ثانيًا: التخطيط



ثالثا: كتابة الأوامر البرمجية

- أضف التعليمة else و أمر طباعة عبارة («نهاية حلقة التكرار»)، كالتالي:

```
x=1
while x<=10:
    print(x)
    x=x+1
else:  # إضافة شرط آخر نهاية حلقة التكرار
     print(" نهاية حلقة التكرار ")
```

إيقاف حلقة التكرار

يمكنك التحكم في حلقات التكرار وإيقافها باستخدام أمر **break**، ولإنشاء برنامج ينهي حلقة التكرار عندما تصبح قيمة المتغير (x==5)، باستخدام الجملة الشرطية **if** للتحقق من الشرط، اتبع الآتي:

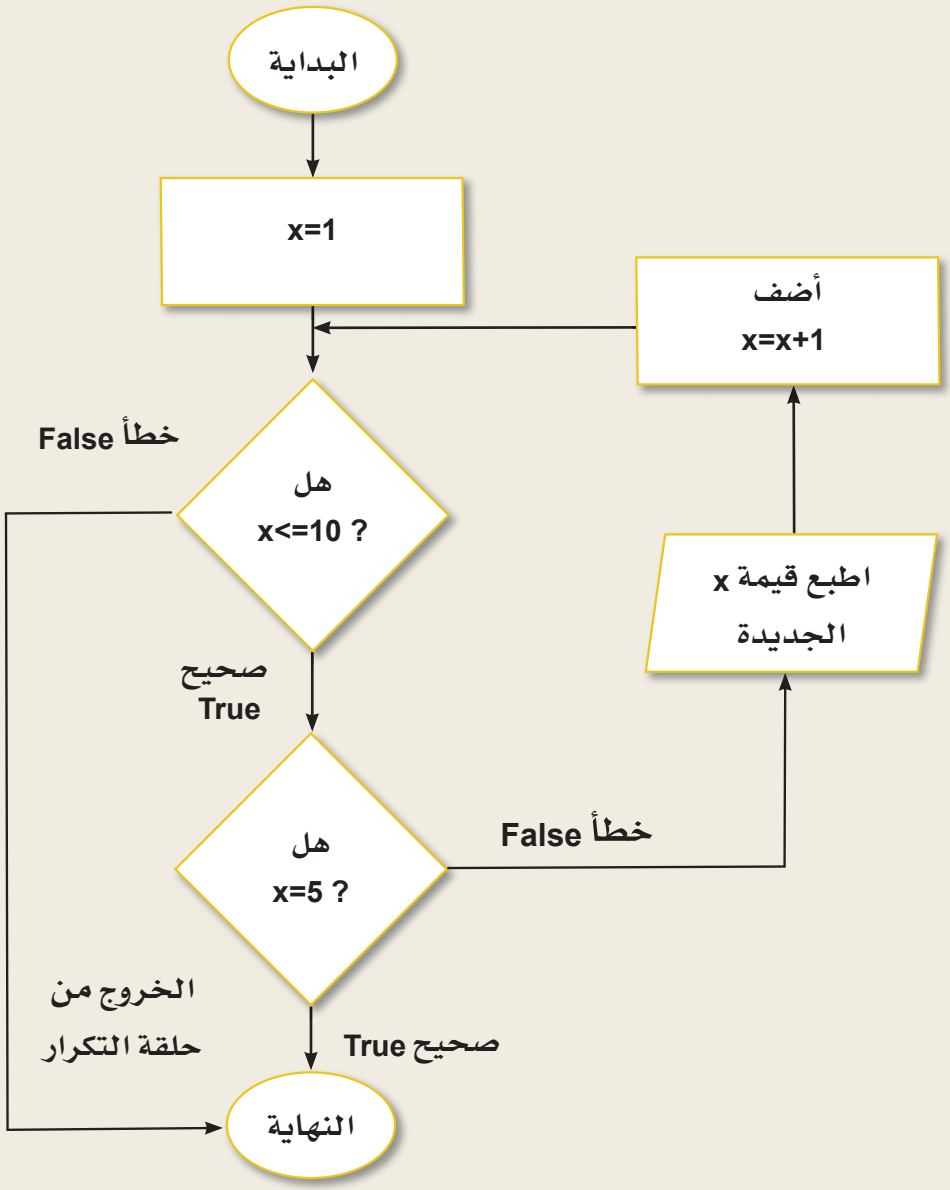
أولاً: التحليل

المعطيات: الأعداد من 1 إلى 10

المطلوب: ١- إنشاء عدّاد للعد من 1 إلى 10.

٢- إيقاف حلقة التكرار عندما تصبح قيمة x=5

ثانيا : التخطيط



ثالثًا : كتابة الأوامر البرمجية

• أضف في المقطع البرمجي السابق جملة الشرط (if) للتحقق من أن (n = 5)، وأمر إيقاف break لإنهاء حلقة التكرار بعد تحقق الشرط كالآتي:

```
x=1
while x<=10:
    if x==5:      # الجملة الشرطية لتحقق الشرط
        break     # إيقاف حلقة التكرار عندما تصبح قيمة المتغير المخزنة تساوي 5
    print(x)
    x=x+1
```

نشاط ٨: خمّن الرقم

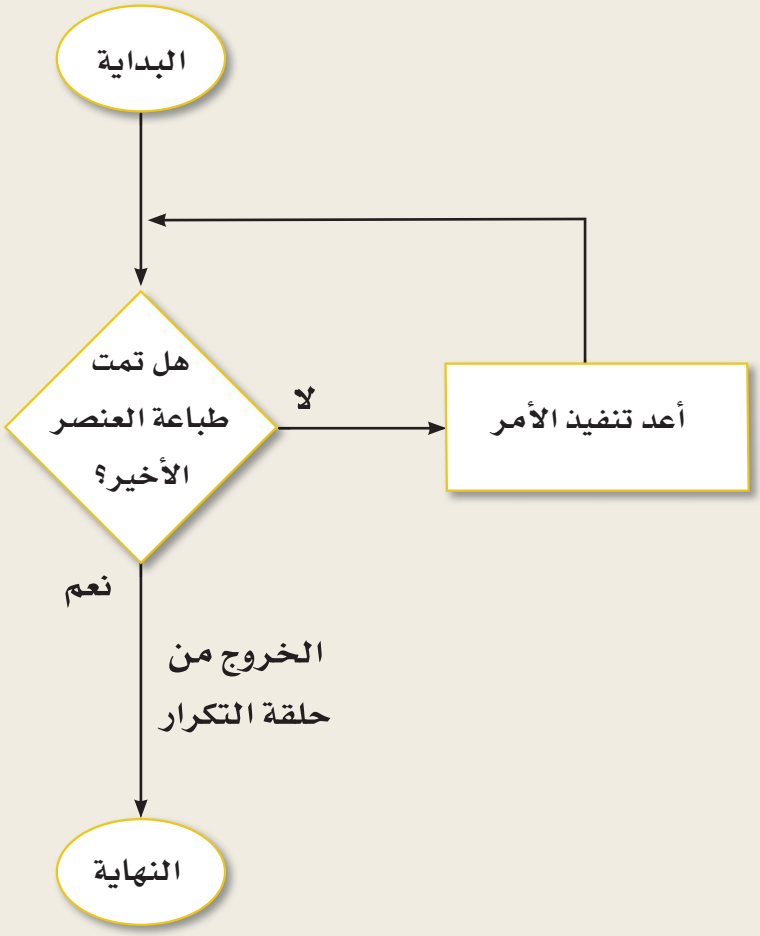
- **اتبع خطوات إنشاء برنامج يطلب من المستخدم تخمين رقم، مع مراعاة ما يلي:**
  - يكون الرقم محصورًا بين (١ إلى ٩).
  - يتم تكرار محاولة إدخال الرقم ثلاث مرات فقط، بحيث ينفِّذ أحد الشرطين:

١- إذا أدخل الرقم الصحيح، يطبع له عبارة (« أحسنت، لقد أدخلت الرقم الصحيح »)، ثم يتم إيقاف حلقة التكرار.

٢- إذا استنفد المحاولات الثلاث دون أن ينجح في معرفة الرقم المطلوب، يطبع له عبارة (« لقد استنفدت جميع المحاولات»).

٢ جملة التكرار (for)

تُستخدم لتنفيذ جزء معين من الأوامر البرمجية لعدد محدَّد من المرات حسب العناصر المسندة في القائمة، بحيث يتم فيها تعريف متغير العدّاد في بداية حلقة التكرار، وفي كل مرة يتم زيادة المتغير وإعادة تنفيذ الأمر بعد جملة for حتى تنتهي عدد مرات التكرار، كما يوضحه المخطط الآتي:



يمكنك تنفيذ عملية التكرار باستخدام جملة **for** بطريقتين، هما:

التكرار مرة واحدة

تتيح لك تنفيذ أوامر برمجية معينه مرة واحدة فقط لكل عنصر، ولطباعة كل عنصر في قائمة من خلال المرور على جميع عناصرها، اتبع الآتي:

- افتح ملف جديد **New File**، في **IDLE المحرّر**، وأحفظ الملف.
- أنشئ قائمة list، وعرّفها باسم وليكن **(fruits)**، وأسند لها القيم الآتية:
["orang", "banana", "apple", "grapes"]
- أسند عناصر القائمة إلى متغير في كل حلقة تكرار وليكن **(n)**، كما يلي:

```
fruits=["orange" , "banana" , "apple" , "grapes"]
for n in fruits:    # n المرور على عناصر القائمة وتخزينه في المتغير
    print(n)  # طباعة العنصر في حلقة التكرار الحالية
```

- شغّل البرنامج، ولاحظ طباعة عناصر القائمة بشكل متسلسل.
- أضف أمر **إيقاف break** لإيقاف حلقة التكرار بعد طباعة العنصر **["apple"]** .
- **جرّب** طباعة المتغير **(fruits)**، بكتابة أمر طباعة كما يلي: **print(fruits)** ، ثم شغّل البرنامج، **ماذا تلاحظ؟ فسّر ذلك.**

التكرار باستخدام دالة النطاق ( ) range

تتيح لك دالة **النطاق ( ) range** بتنفيذ عملية التكرار من خلال تحديد القيمة الابتدائية لبدء عملية التكرار ،بإحدى الطرق الآتية:

- **تحديد عدد المرات range (n)**: تُستخدم لتكرار أمر بعدد محدَّد من المرات يبدأ من صفر وينتهي بـ (n-1).

ولإنشاء برنامج يعمل على تكرار أوامر برمجية وطباعة عناصر القائمة من خلال تحديد رقم فهرس العنصر الذي ينتهي عنده التكرار، اتبع الآتي:

```
fruits=['orang' , 'banana' , 'apple' , 'grapes' ]
for n in range (3):     #n المرور على عناصر القائمة، وتخزين رقم فهرسها في المتغير
    print(fruits[n])    #n طباعة عناصر القائمة التي تم تخزينها في المتغير
```

- شغِّل البرنامج، ولاحظ العناصر التي سيتم طباعتها.

كما تتيح لك دالة النطاق range (n) توليد أرقام صحيحة محددة، ولإنشاء برنامج يعمل على توليد ستة أرقام صحيحة وطباعتها، اتبع الآتي:

```
for i in range (6):     #i المرور على ستة قيم، وتخزينها في المتغير
    print(i)    #طباعة القيم التي تم تخزينها في المتغير
```

- شغِّل البرنامج، ولاحظ قيم العناصر التي يتم طباعتها.

نشاط ٩: جدول ضرب ٢

- أضف بعض التعديلات في المقطع البرمجي السابق ليطبع لك حاصل ضرب العدد (2) بحيث يكرر الأمر حتى يصل إلى العدد (12).

- **تحديد البداية والنهاية range (start, end)** : تُستخدم لتكرار أمر بعدد محدَّد من المرات من خلال تحديد القيمة الابتدائية لبدء عملية التكرار والقيمة النهائية.

ولإنشاء برنامج يعمل على توليد مجموعة أرقام صحيحة تبدأ من(2)، وتنتهي بـ(9)، ومن ثم طباعتها، اتبع الآتي:

```
for i in range(2,10):        # يبدأ التكرار من العدد ٢ وينتهي عند ١٠-١
    print(i)
```

- شغّل البرنامج، ولاحظ القيم التي يتم طباعتها.

- **تحديد البداية، والنهاية، وعدد القفزات range (start, end, step size)**: تُستخدم لتكرار أمر بعدد محدد من المرات من خلال تحديد القيمة الابتدائية لبدء عملية التكرار والقيمة النهائية، وعدد القفزات، أي يتم تجاهل عناصر معينة وعدم طباعتها.

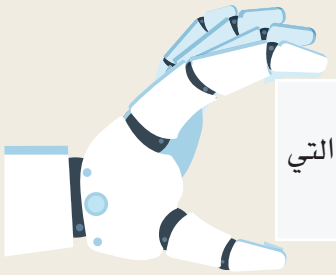
ولإنشاء برنامج يعمل على توليد مجموعة أرقام صحيحة تبدأ من (1)، وتنتهي بـ (9)، بحيث يزيد عددان في كل مرة، ومن ثم طباعة الأرقام الناتجة، اتبع الآتي:

```
for i in range(1,10,2):    # يبدأ من ١، وينتهي بـ٩، وفي كل مرة يقفز عددان
    print(i)
```

- شغّل البرنامج، ولاحظ القيم التي يتم طباعتها.

فكر

كيف يمكنك أن تجعل البرنامج يعمل على توليد الأرقام بطريقة عكسية، أي مِنَ الأكبر إلى الأصغر؟

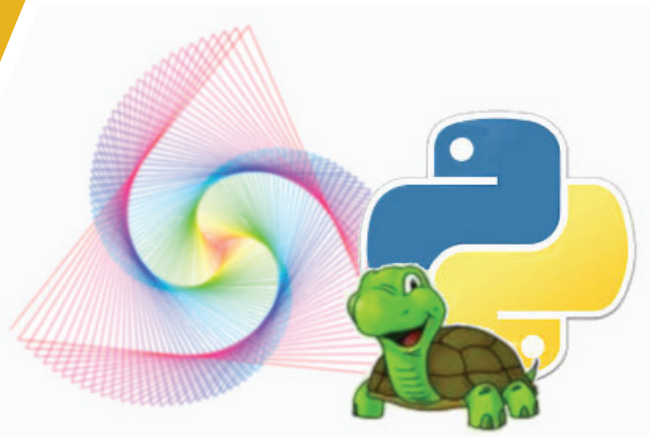


تذكَّر!

لا يمكنك أن تحفظ جميع الأوامر البرمجية، تعلَّم الأساسيات التي تستخدمها باستمرار، واقرأ كثيرًا في مصادر البحث المختلفة.

الدرس الرابع

الوحدات البرمجية



الوحدة **Module** عبارة عن ملف يحتوي على أوامر ، ودوال برمجية جاهزة ،يتم استيرادها إلى البرنامج لتنفيذ وظائف مختلفة. ولتسهيل إدارة المشاريع توفر البايثون العديد من الوحدات البرمجية ، مثل: وحدة واجهة المستخدم الرسومية **tkinter module**، ووحدة العشوائية **random module** ، ووحدة الرسم **turtle module**.

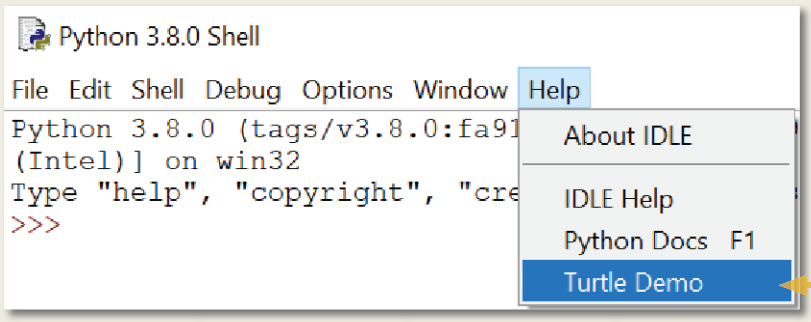
وحدة Turtle Module

تحتوي لغة البايثون على وحدة الرسم **Turtle Module** التي تتيح لك رسم رسومات متَّجِهة (**Vector graphics**) ثنائية الأبعاد **2D** بألوان مختلفة، وإنشاء أنظمة المحاكاة والألعاب التفاعلية وغيرها.

ولاستكشاف نماذج من الأمثلة الرسومية لوحدة الرسم المتاحة في لغة البايثون، اتبع الآتي:

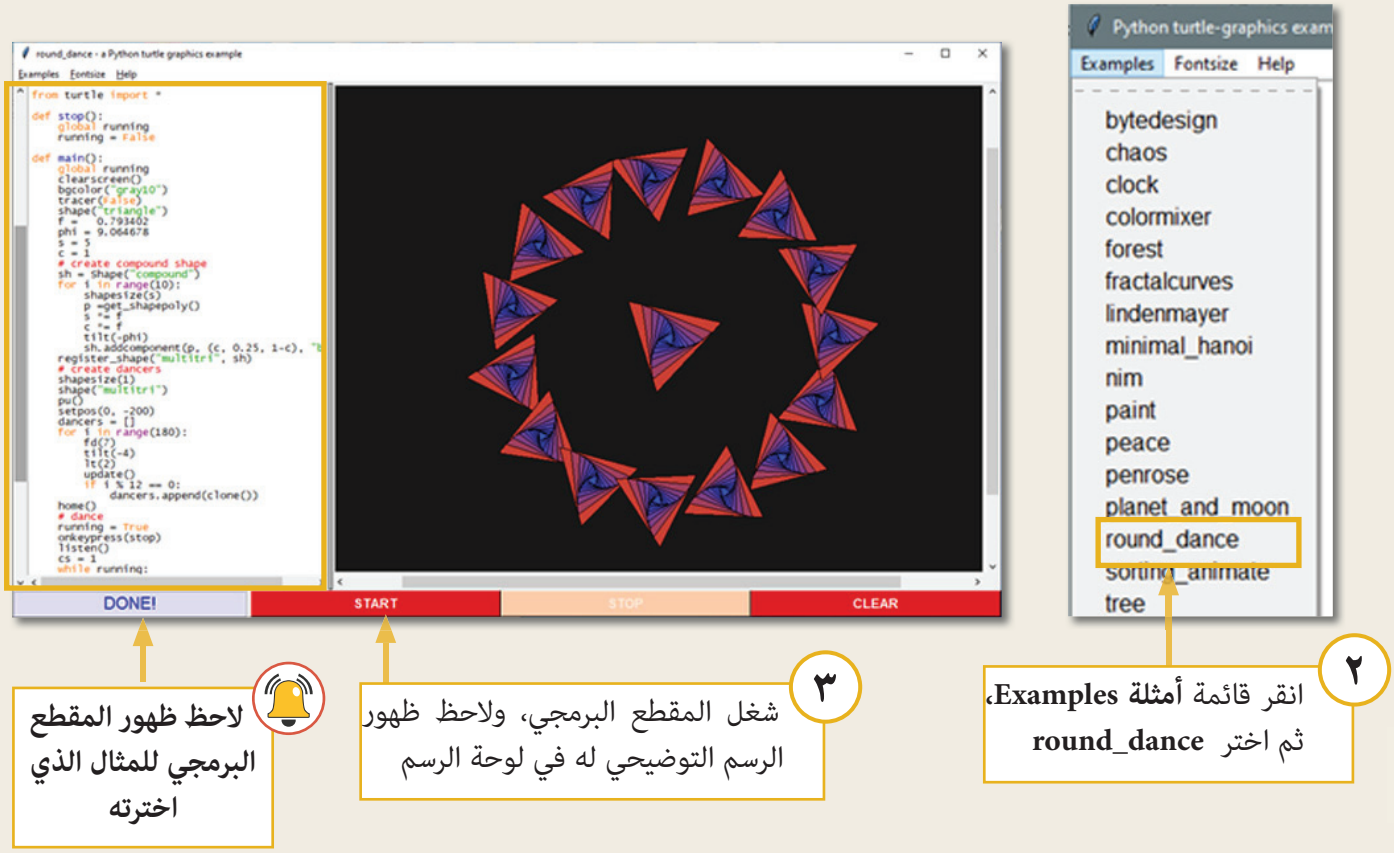
من نافذة الأوامر **Python Shell**:

- انقر قائمة المساعدة **Help**.



١ اختر أمر **Turtle Demo**

من النافذة التي تظهر لك:



نشاط ١٠: الأشكال المتحركة

- **تأمَّل المقطع البرمجي لمثال round_dance، وأجب عن الأسئلة الآتية:**
  - ما الأمر البرمجي المستخدم في رسم شكل المثلث؟
  - ما الأوامر البرمجية المستخدمة في رسم الأشكال المركَّبة؟
  - ما الأوامر البرمجية المستخدمة في تحريك الأشكال؟
- **استكشف نماذج أخرى لوحدة الرسم، ولاحظ نتيجة تنفيذها.**

استدعاء وحدة الرسم

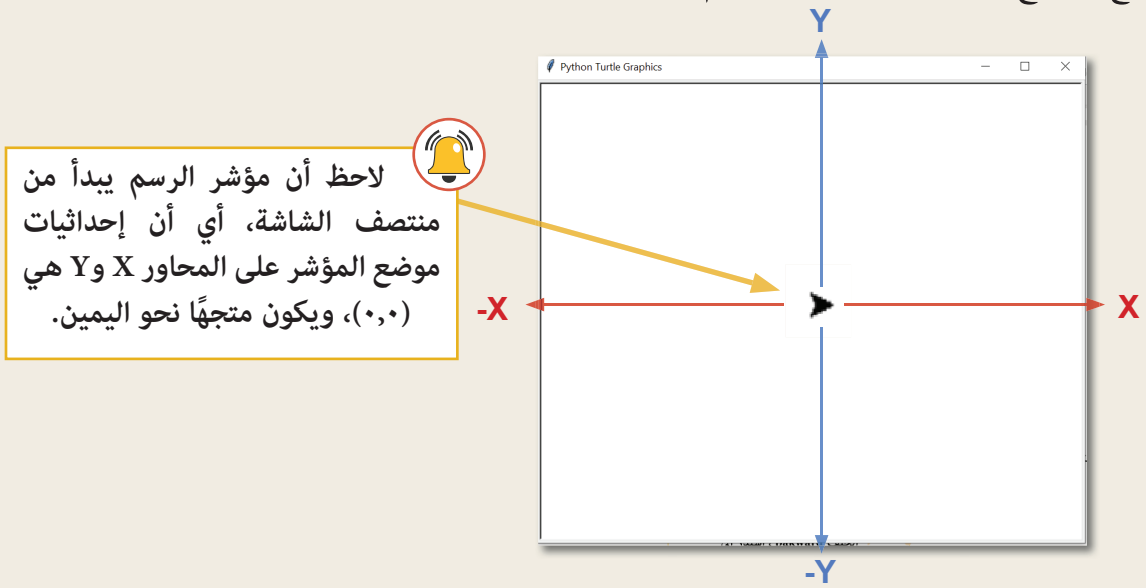
لتتمكن من استخدام وحدة السلحفاة في رسم الأشكال والاستفادة من وظائفها، عليك أولًا استدعاؤها، واستيراد لوحة الرسم الخاصة بها **Canvas**، باتباع الآتي:

- افتح نافذة جديدة في **IDLE المحرِّر**، واحفظ الملف.

تجنَّب حفظ الملف باسم **(turtle.py)**، حتى لا يتعارض مع الاسم الافتراضي لاستيراد وحدة الرسم **turtle**.

```
from turtle import*    # أمر استدعاء وحدة الرسم
Turtle()     # أمر استدعاء لوحة الرسم
```

- شغِّل البرنامج، لتفتح لك نافذة **لوحة الرسم Canvas**، الآتية:



لاحظ أن مؤشر الرسم يبدأ من منتصف الشاشة، أي أن إحداثيات موضع المؤشر على المحاور X وY هي (٠،٠)، ويكون متجهًا نحو اليمين.

رسم الأشكال

تتيح لك وحدة السلحفاة رسم كائنات وأشكال مختلفة، من خلال تحديد نقاط الرسم أو المسافة التي يقطعها المؤشر في أثناء الحركة.

يمكنك تحريك المؤشر ورسم الأشكال في وحدة السلحفاة، باستخدم إحدى الطريقتين:

١ تحديد إحداثيات موضع المؤشر:

يُستخدم أمر **(goto)** لتحريك المؤشر من خلال تحديد الإحداثيات التي سيتحرك بها المؤشر على الشاشة.

ولتحريك المؤشر إلى أعلى باتجاه المحور **(Y)**، لمسافة مقدارها **(100)** بكسل **Pixel**، اتبع الآتي:

```
from turtle import*
Turtle()
goto(0,100)
```

- شغِّل البرنامج، ولاحظ اتجاه حركة المؤشر.
- **جرّب** تغيير قيمة الإحداثيات، وشغل البرنامج، ولاحظ اتجاه حركة المؤشر.

معلومة تهمك

تُستخدم وحدة البكسل Pixel لقياس المسافة التي يتحركها المؤشر على لوحة الرسم.

٢ تحديد اتجاه المؤشر:

يمكنك تحريك المؤشر من خلال تحديد أحد الاتجاهات الأربعة (أمام ، وخلف، دوران مع عقارب الساعة، ودوران عكس عقارب الساعة).

ولتحريك المؤشر باتجاه الأمام باستخدام دالة ( ) forward، لمسافة مقدارها (100) بكسل، اتبع الآتي:

```
from turtle import*
Turtle()
forward(100)
```

- شغِّل البرنامج، ولاحظ حركة المؤشر.

ولرسم شكل هندسي، ولْيَكُن مربعًا من خلال تحديد اتجاه المؤشر، اتبع الآتي:

```
from turtle import*
Turtle()
forward(100)   # حركة المؤشر إلى الأمام مقدار ١٠٠بكسل
right(90)      # حركة المؤشر بزاوية ٩٠ باتجاه عقارب الساعة
forward(100)
right(90)
forward(100)
right(90)
forward(100)
```

- شغِّل البرنامج، ولاحظ رسم الشكل.

معلومة تهمك

- يمكنك تحريك مؤشرالرسم في اتجاه محدَّد، باستخدام الدوال الآتية:

اسم الدالة	الوظيفة	اختصارها
forward( )	تحريك مؤشر الرسم إلى الأمام مسافة محددة	fd( )
backward( )	تحريك مؤشر الرسم إلى الخلف مسافة محددة	bk( )
right( )	تحريك المؤشر باتجاه عقارب الساعة بزاوية محددة	rt( )
left( )	تحريك المؤشر باتجاه عكس عقارب الساعة بزاوية محددة	lt( )

نشاط ١١: رسم المضلعات

- استخدم حلقة التكرار for، وغيّر ما يلزم في المقطع البرمجي السابق، لرسم أشكال هندسية منتظمة أخرى (مثلث، خماسي، سداسي، ....).

لرسم شكل حلزوني، باستخدام دالة التكرار for، اكتب الأوامر البرمجية الآتية:

```
from turtle import*
Turtle()
color('red')        # لون خط الرسم

for i in range(200):      # عدد مرات تكرار عملية الرسم
    forward(i)
    left(91)
```

- شغِّل البرنامج، ولاحظ الشكل الناتج.

لتتمكن من رسم الشكل بعدة ألوان ، باستخدام وحدة العشوائية **random module** ، وتغيير سرعة رسم المؤشر باستخدام دالة سرعة **speed( )**، أضف إلى المقطع السابق مايلي:

```
from turtle import*
import random       # إستيراد وحدة العشوائية
Turtle()
c=['red' , 'yellow' , 'purple']    #أسند قيم قائمة الألوان إلى متغير
for i in range (200):
    color(random.choice(c))  # اختيار ألوان بشكل عشوائي من القائمة باستخدام وحدة random
    forward(i)
    left(91)
    speed(5)          # دالة تغيير سرعة رسم المؤشر
```

- شغِّل البرنامج، **ماذا تلاحظ؟**

معلومة تهمك

- تُستخدم وحدة العشوائية random module في اختيار عنصر من القائمة List ، أو لتوليد أرقام بشكل عشوائي.
- تتراوح سرعة دالة السرعة () speed من (1) أبطأ سرعة إلى (10) أعلى سرعة.
- عند إعطاء دالة السرعة () speed قيمة أكبر من (10) أو أصغر من (0.5) فإن البرنامج يعيدها إلى (0) تلقائيا، وسيتحرك الشكل بأقصى سرعة ممكنة.

ابحث عن الدالة التي تتيح لك التحكم بخاصية سُمْك الخط.

التحكم بخصائص الرسم

يمكنك إجراء تغييرات في خصائص لوحة الرسم، مثل: تغيير لون شاشة الخلفية، وإضافة نص وغيرها، كما يمكن إجراء تغييرات في خصائص المؤشر، مثل: تغيير شكله، ولون الخط الذي يرسمه، وسمك الخط وغيرها.

تغيير خصائص خلفية الرسم

عند استدعاء وحدة الرسم، فإنها تفتح نافذة بخلفية بيضاء بشكل افتراضي، ولتغيير لون الخلفية، اتبع الآتي:

- افتح نافذة جديدة في **IDLE المحرِّر**، واحفظ الملف.
- اكتب أمر استدعاء وحدة الرسم **Turtle Module**، وأمر استدعاء لوحة الرسم **Canvas**.
- غيّر خلفية الرسم **back ground** بكتابة الأمر **bgcolor ( )**، كما الآتي:

```
from turtle import*
Turtle()
bgcolor("blue")        # أمر تغيير لون الخلفية
```

- شغِّل البرنامج، ولاحظ لون لوحة الرسم.

لكتابة نص على لوحة الرسم، وتنسيقه باستخدام أمر كتابة **write ( )**، اتبع الآتية:

```
from turtle import*
Turtle()
bgcolor("blue")
write("Oman" , align="left" , font=("Arial" ,12))      #أمر كتابة النص على الشاشة
```

محاذاة النص

حجم ونوع النص

- شغِّل البرنامج، ولاحظ ظهور النص.

تغيير خصائص المؤشر

يمكنك تغيير شكل المؤشر ولونه إلى عدة أشكال وألوان مختلفة، ولتغيير شكل المؤشر إلى شكل سلحفاة باستخدام **دالة الشكل ( ) shape**، اتبع الآتي:

```
from turtle import*
Turtle()
bgcolor("blue")
write("Oman" ,align="left" , font=("Arial",12))
shape("turtle")     #  أكتب القيمة المرجعية turtle، لتغيير شكل المؤشر إلى لحفاة
```

- شغِّل البرنامج، ولاحظ تغير شكل المؤشر.

معلومة تهمك

- **يمكنك تغيير شكل مؤشر الرسم إلى أشكال مختلفة باستخدام دالة ( ) shape، وتحديد القيمة المرجعية للشكل، كالآتي:**

القيمة المرجعية	الوظيفة
"circle"	دائرة
"triangle"	مثلث
"square"	مربع

نشاط ١٢: الأشكال الهندسية

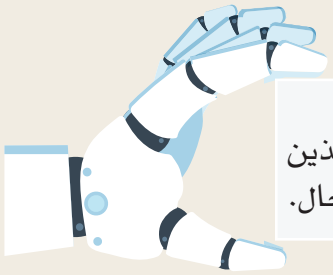
- **اتَّبع خطوات إنشاء برنامج، يعمل على رسم أشكال هندسية مختلفة ، ويكتب اسم الشكل في منتصف الشاشة، مع مراعاة مايلي:**

- عـرِّف متغيـرًا وأسند إليه قيمة باستخـدام دالة الإدخال **( ) input**، بحيـث تظهر للمستخدم رسالة تطلب منه اختيار رقم صحيح **integer** من **1** إلى **3**.
- وظِّف الجمل الشرطية في التحقق من الشرط وصحة الرقم المدخل، لتنفيذ الآتي:

١- إذا كان الرقـم المُدخَـل (**1**)، تفتـح لوحـة الرسـم بشاشة زرقـاء، ويرسـم المؤشر قطعة مستقيمة.

٢- إذا كان الرقم المُدخَل (**2**)، تفتح لوحة الرسم بشاشة حمراء، ويرسم المؤشر زاوية قائمة.

٣- إذا كان الرقم المُدخَل (**3**)، تفتح لوحة الرسم بشاشة صفراء، ويرسم المؤشر مثلثًا.



تذكَّر!
استمر في تطبيق ما تعلمته، وشارك أعمالك مع الذين لديهم اهتمامات في البرمجة، لتتميز أكثر في هذا المجال.

اختبر معلوماتك لتقيس ما تعلمته في وحدة البرمجة النصية، ثم ناقش زملاءك مع معلمك لتتحقق من صحة إجاباتك:

أجب عن الأسئلة الآتية:

- وضح كيف تُسهم البرمجة في تسهيل بعض مهام الحياة اليومية.
- تتضمن بيئة التطوير IDLE نوعين من النوافذ. اذكرهما ووضّح الفرق بينهما.
- ما النتيجة التي تتوقع أن تحصل عليها عند تنفيذ المقطع البرمجي التالي:
- `print ("Oman"*5)` ؟ هل يمكنك تغيير علامة الضرب إلى القسمة؟ فسّر إجابتك.
- ما فائدة إضافة علامة (#) في المقطع البرمجي؟
- كيف تنشئ متغيرًا يتضمن القيمة العددية (6)؟
- ما الخطأ في كتابة المتغيِّر للمقطع البرمجي: `2my-name="Ahmed"` ؟
- ما العدد الذي ستحصل عليه عند تنفيذ المقطع البرمجي: `int(float(100*2)+5)` ؟
- صحّح المقطع البرمجي الآتي ليتم تنفيذه:

```
if 5 > 2:
print("5 أكبر من 2!")
```

اكتب نتيجة المقارنة بين قيمتين (صحيح True، خطأ False) للتعبيرات الآتية:

`x=3, y=5` `x< y and x==y`	`5!=6 or 5==6`	`int (0.1)==(1/10 )`

- كيف يمكنك الوصول إلى العنصر الأول في كلمة ("Fatma")؟
- ما الدالة المستخدمة لإضافة عنصر إلى القائمة؟

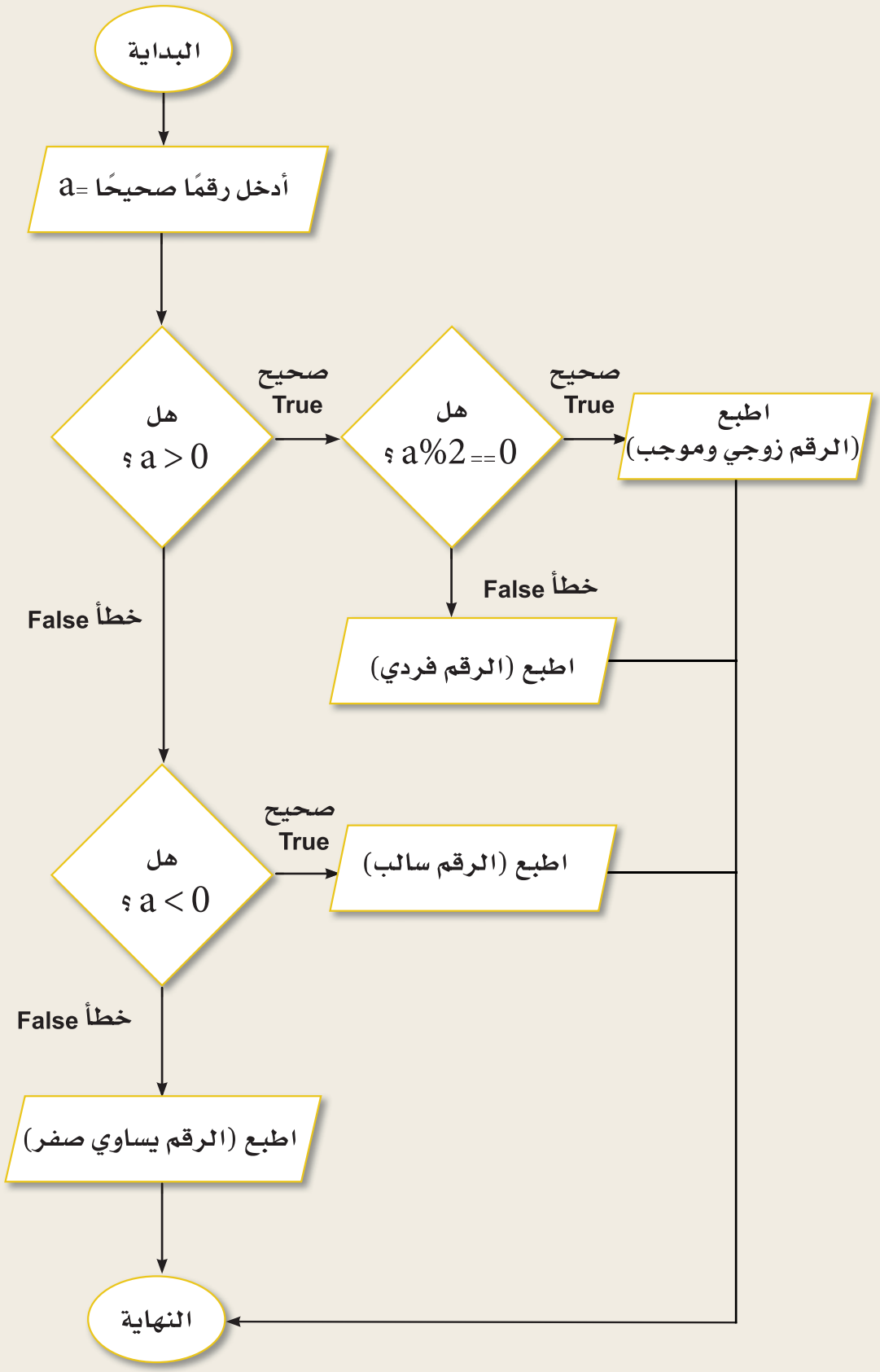
- كم عدد المرات التي سيتم فيها طباعة كلمة ("Oman") في المقطع البرمجي الآتية؟

```
a= 0
b= "Oman"
while (a<30):
    print(b)
    a=a+5
```

- اكتب الخوارزمية المستخدمة لكتابة المقطع البرمجي الآتي:

```
user= input(" enter your user name \n")
password= int(input (" enter your password \n"))
if (user == "admin"  and password == 222) or (user == "salah" and password==123):
    print(" welcome " , user)
else:
    print("user name or password is wrong" )
```

- تأمل المخطط الانسيابي التالي، ثم اكتب البرمجة اللازمة لتنفيذ هذا البرنامج:



المراجع والمصادر

- مصباح، خالد محمد. (٢٠١٧م) **Python** المهام الأساسية – المركز المصري لتبسيط العلوم، مصر.
- مصباح، خالد محمد. (٢٠١٧م) **Python** المهام المتقدمة – المركز المصري لتبسيط العلوم، مصر.

- Cannon, Jason. Python Programming for Beginners. Retrieved from:
https://kupdf.net/queue/python-programming-for-beginner-jason-cannon_58da6f8ddc0d602665c3461e_pdf?queue_id=-1&x=1602062266&z=OTYuOS4xNTAuMTE2

- Halterman, Richard L. (2011) LEARNING TO PROGRAM WITH PYTHON. Retrieved from: http://ce.sharif.edu/courses/962/97-/ce1536-/resources/root/Learn-to-Program-with-Python.pdf

- Kuhlman, Dave. (December 15, 2013) A Python Book: Beginning Python, Advanced Python, and Python Exercises. Retrieved from:
https://www.davekuhlman.org/python_book_01.pdf

- Learn Python simply easy learning. Retrieved from:
https://www.tutorialspoint.com/python/index.htm

الوحدة الثانية

الحوسبة السحابية

مقدمة

مع تطور التقنيات الحديثة المتاحة على شبكة الويب والزيادة المطردة في سرعات الإنترنت ظهرت تقنية الحوسبة السحابية Cloud Computing، التي قدَّمت لمستخدميها مزايا عديدة متطورة تتمثل في توفير البرمجيات والشبكات وأنظمة التشغيل عبر الإنترنت، بالإضافة إلى إمكانية معالجة البيانات وتخزين المعلومات ومشاركتها في أي وقت، ومن أي مكان من خلال أجهزة و خوادم خارجية متاحة على شبكة الإنترنت.

وتهدف هذه الوحدة إلى توجيهك نحو التعرف إلى تقنية الحوسبة السحابية وكيفية الاستفادة من خدماتها، واستخدام بعض تطبيقاتها بطريقة آمنة مع الالتزام بإجراءات الحماية المطلوبة بما يضمن تحقيق الفائدة المرجوة.

يتوقَّع منك عزيزي الطالب / عزيزتي الطالبة في هذه الوحدة:

- تتعرف إلى مفهوم الحوسبة السحابية وأنواعها ومكوناتها.
- تدرك أهمية الحوسبة السحابية في حياتنا اليومية.
- تستخدم أحد تطبيقات الحوسبة السحابية في تخزين البيانات.
- تستخدم أحد تطبيقات الحوسبة السحابية في تصميم نماذج إلكترونية.
- تتبع إجراءات الاستخدام الآمن لتطبيقات الحوسبة السحابية.

الدرس الأول

مقدمة في الحوسبة السحابية

الحوسبة السَّحابية عبارة عن تقنية تهدف إلى توفير مجموعة من الخدمات عبر شبكة الإنترنت، وتشمل هذه الخدمات تقديم الموارد الحاسوبية (البرمجيات والعتاد)، وتوفير مساحات لتخزين الملفات، وعمل النسخ الاحتياطية، كما تهدف إلى تقليل تكلفة شراء الموارد الحاسوبية وتوفير مستوى من الأمان والخصوصية.

نشاط ١: نشأة الحوسبة السحابية

ابحث عن إجابات الأسئلة التالية، وناقشها مع زملائك:

١. كيف ظهرت فكرة الحوسبة السحابية؟
٢. لماذا نستخدم الحوسبة السحابية؟
٣. ما متطلبات الوصول للحوسبة السحابية؟
٤. ناقش مع زملائك مخاطر الحوسبة السحابية.
٥. هل ترى أن الحوسبة السحابية قد تغنينا عن استخدام وسائط التخزين الأخرى؟ ولماذا؟

مزايا الحوسبة السحابية

توجد العديد من المزايا التي تجعل استخدام الحوسبة السحابية ضروريا ومفيدا للأفراد والمؤسسات، منها:

مزايا الحوسبة السحابية

سرعة معالجة البيانات

تمكّن أي كمبيوتر من القيام بمعالجات معقدة، وتنفيذ العديد من المهام بسرعة هائلة ويعتمد ذلك على سرعة الإنترنت.

خفض التكلفة المادية

تسهم في الحدّ من شراء الأجهزة والبرمجيات والمصادر المختلفة مثل: الخوادم، ووسائط التخزين؛ مما يقلل من نفقات الشراء والصيانة.

تأمين وحماية المعلومات

تنشئ نُسخًا احتياطية للملفات لحفظ البيانات، وتوفر الصيانة الدورية والدعم الفني في حال حدوث أي خلل مفاجئ.

تبادل المعلومات ونشرها

تُمكِّنك من نشر البيانات والمعلومات ومشاركتها مع الآخرين بشكل فوري ومرن.

التطوير والتحديث المستمر

توفِّر لك تطوير الخدمات والبرامج المقدَّمة عبر السحابة الإلكترونية وتحديثها بشكل مستمر، وغالبًا ما تكون هذه الخدمات مجانية.

سهولة الوصول إلى البيانات

يُمكنك الوصول إلى بياناتك الخاصة باستخدام أي جهاز حاسوب شخصي في أي وقت ومن أي مكان.

مكونات الحوسبة السحابية

للوصول إلى خدمات الحوسبة السحابية تحتاج إلى أربعة عناصر أساسية، هي:

المستخدم User

المستفيد من خدمات هذه التقنية من خلال أجهزة الحواسيب الشخصية المختلفة.

البنية التحتية Infrastructure

الأنظمة والعتاد التي يُعتمد عليها في تقديم الخدمة مثل: أجهزة وخـوادم، ومساحات تخزينية للمعلومات، وأنظمة تشغيل.

التطبيقات Applications

البرمجيات التي يستخدمها المستفيد في السحابة مثل: برمجيات معالجة النصوص والعروض التقديمية، وخدمات نقل المعلومات ومشاركتها.

مزود الخدمة Service Provider

الجهة التي توفِّر الخدمة السحابية عبر خوادم عملاقة في سعاتها التخزينية وسرعة معالجتها للبيانات ليستفيد منها المستخدمون والمطوِّرون.

معلومة تهمك

- **من الأمثلة على الشركات المزوِّدة للخدمات السحابية:**
  شركة أمازون Amazon التي توفِّر خدمة (Amazon Web Services)، وشركة مايكروسوفت Microsoft التي توفِّر خدمة (Windows Azure)، وشركة جوجل Google التي توفِّر خدمة (Google Apps).
- **يعتمد اختيارك لمزود الخدمة السحابية على احتياجاتك ونوع الخدمة التي تحتاج إليها.**

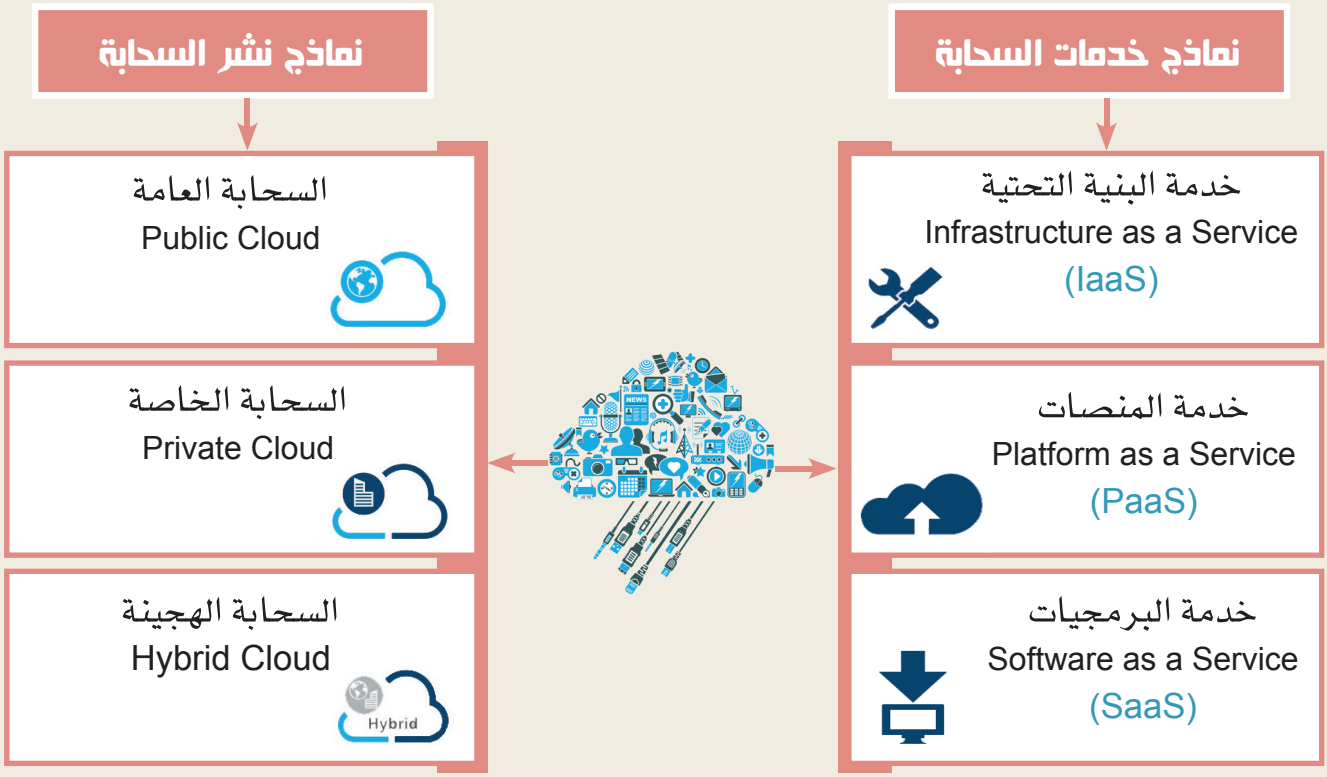
نماذج الحوسبة السحابية

ظهرت الحوسبة السحابية بأنواع مختلفة لتلبية الاحتياجات المتنوعة للمستفيدين، حيث تم تصنيفها كالآتي:

١. **نماذج خدمات السحابة:** تعتمد على نوع الخدمة التي تقدمها السحابة، حيث تختلف فيما بينها في الصلاحيات المتاحة للمستفيد، والخدمات التي يوفرها مزود الخدمة.

٢. **نماذج نشر السحابة:** تعتمد على مدى انتشار المعلومات ونطاق الوصول إليها وفئة المستفيدين منها، وتختلف فيما بينها في مستويات الأمن والإدارة المطلوبة.

ويوضح المخطط الآتي نماذج الحوسبة السحابية:



نشاط ٢: أنواع الحوسبة السحابية

اختر أحد نماذج الحوسبة السحابية مستعينًا بالمخطط السابق، ثم ابحث عن النقاط الآتية، وناقش زملاءك فيما ما توصَّلت إليه من معلومات:

١. مواصفاتها واستخداماتها.
٢. الصلاحيات المتاحة للمستفيد.
٣. أمثلة عليها.
٤. أَيُّ من نماذج الحوسبة السحابية أعلاه يتم استخدامه على النطاق الشخصي أو على نطاق مدرستك؟

الدرس الثاني

خدمات الحوسبة السحابية

توفِّر الحوسبة السحابية مجموعة من الخدمات عبر شبكة الإنترنت التي تخدم فئات مختلفة من المستفيدين عند الطلب، حيث تسهِّل عليهم استخدام التطبيقات وتخزين المعلومات ومشاركتها وحمايتها. وتتضمن خدمات الحوسبة السحابية التطبيقات ومنصات التطوير وأجهزة وخوادم افتراضية.

الفئات المستفيدة من خدمات الحوسبة

يوفِّر مزوِّدو الخدمة مجموعة من خدمات الحوسبة السحابية لثلاث فئات مختلفة من المستفيدين، يتم تصنيفهم كما هو موضح في المخطط الآتي:

الأفراد	المبرمجون	المؤسسات
خدمات البريد الإلكتروني	منصات افتراضية لتطوير البرمجيات وتحديثها	بيئات افتراضية لتقديم خدمات التجارة الالكترونية
خدمات وسائل التواصل الاجتماعي	بيئات اختبار للتأكد من خلو البرمجيات من الأخطاء	مراقبة عمليات الشركة وتحسين الإنتاجية
خدمات التخزين السحابي	خوادم بسرعة عالية مع توفير مساحة تخزين كبيرة ودعم فني	العمل عن بعد والاجتماعات المباشرة

نشاط ٣: تطبيقات الحوسبة السحابية

اختر إحدى الخدمات السحابية التالية، وتصفح رابط الموقع الإلكتروني للتعرف على نوع الخدمة التي يقدمها. استعن بالمرفق (Cloud Applications.pdf) للإجابة عن الأسئلة الآتية وناقش الإجابات مع بقية زملائك:

OneDrive
https://onedrive.live.com/about/signin/

Outlook
https://outlook.live.com/owa/

Chromium OS
https://www.chromium.org/chromium-os

Amazon Cloud Player
https://music.amazon.com/home

Renderores
https://www.renderforest.com/ar/video-maker

Google Calendar
https://calendar.google.com/calendar/r?pli=1

- صِف الخدمة التي يقدمها الموقع.
- كيفية وصولك للخدمة: ☐ يتطلب إنشاء حساب ☐ لا يتطلب إنشاء حساب
- ما مدى استفادتك من الموقع؟ وهل يغنيك عن استخدام تطبيق مشابه له مثبت على جهاز الحاسوب؟
- هل الخدمة التي يقدِّمها الموقع: ☐ مجانية ؟ ☐ مدفوعة؟
- صنِّف نوع الخدمة التي يقدمها الموقع إلى أحد الأنواع التالية:

☐ بريد إلكتروني. ☐ تخزين سحابي. ☐ صوتيات سحابية.

☐ تطبيقات وبرامج سحابية. ☐ أنظمة تشغيل سحابية.

ابحث عن مواقع أخرى تقدِّم خدمات سحابية مماثلة.

الدرس الثالث

التخزين السحابي

يُعـد التخزيـن السـحابي **Cloud storage** أحد الخدمات السـحابية التـي تتيح لك حفظ ملفاتـك علـى خادم مـزود الخدمة عبر الإنترنت، وتسـمح لـك بمزامنة الملفات لتسـتطيع الوصول إليها عند الطلب. وتُعدُّ خدمة التخزين السحابي أحد الأمثلة على نموذج خدمة البنية التحتية (**IaaS**).

نشاط ٤: خدمة التخزين السحابي

شاهد مقطع الفيديو (التخزين السحابي)، ثم ناقش الأسئلة التالية مع بقية زملائك:

- مفهوم التخزين السحابي.
- مزايا خدمة التخزين السحابي.
- المقصود بالمزامنة التلقائية للملفات.
- أمثلة لبعض التطبيقات المتخصصة في التخزين السحابي.

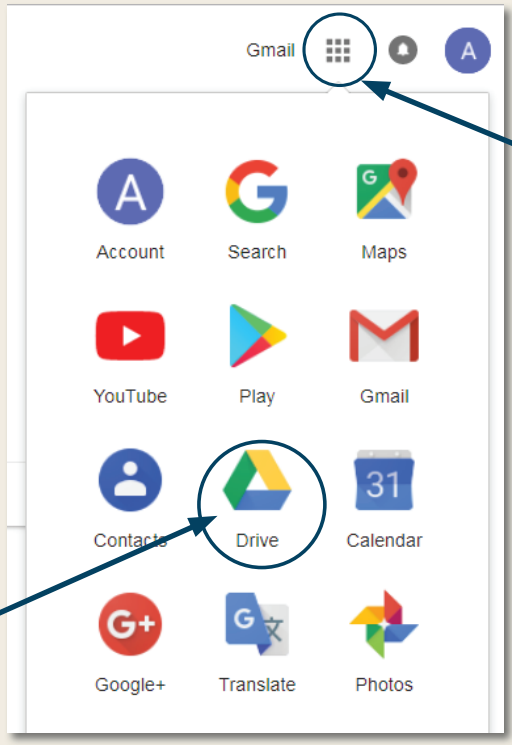
خدمة Google Drive

يعد تطبيق **Drive** الذي تقدمه شركة **Google** أحد الأمثلة على تطبيقات خدمة التخزين السحابي التي يمكن الاستفادة منها لإنشاء وتعديل و تحميل الملفات وتخزينها في السحابة، بالإضافة إلى إمكانية مشاركتها، وهو تطبيق متاح لجميع أنظمة التشغيل.

للوصول إلى هذه الخدمة، اتبع الآتي:

- افتح الصفحة الرئيسية لموقع Google عبر الرابط: www.google.com

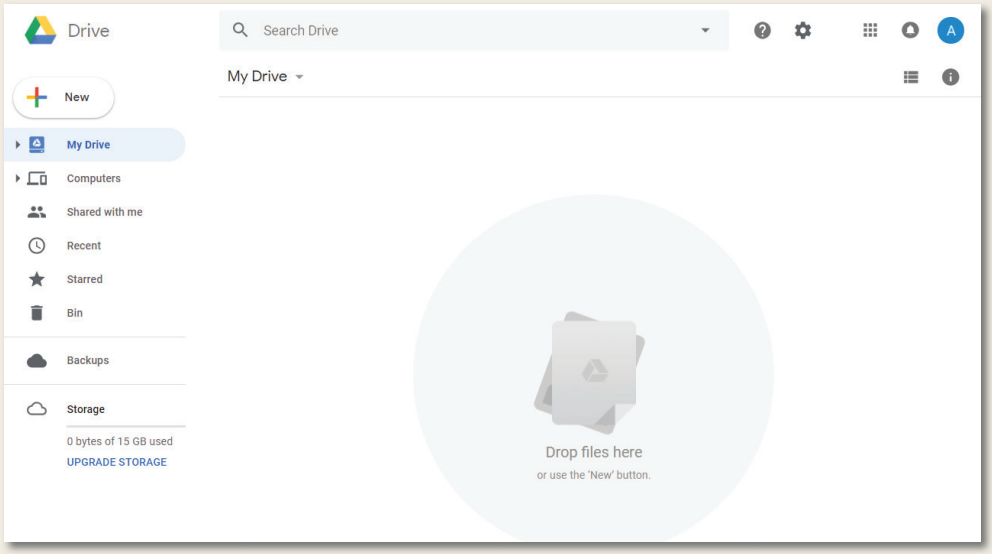
١ قم **بتسجيل دخول Sign In** لحساب البريد الإلكتروني الخاص بك



٢ انقر زر **تطبيقات Google apps**

٣ اختر تطبيق Drive

ستظهر لك صفحة واجهة تطبيق Drive الآتية :



استعرض مكونات واجهة خدمة **Google Drive**، ونفِّذ مايلي:

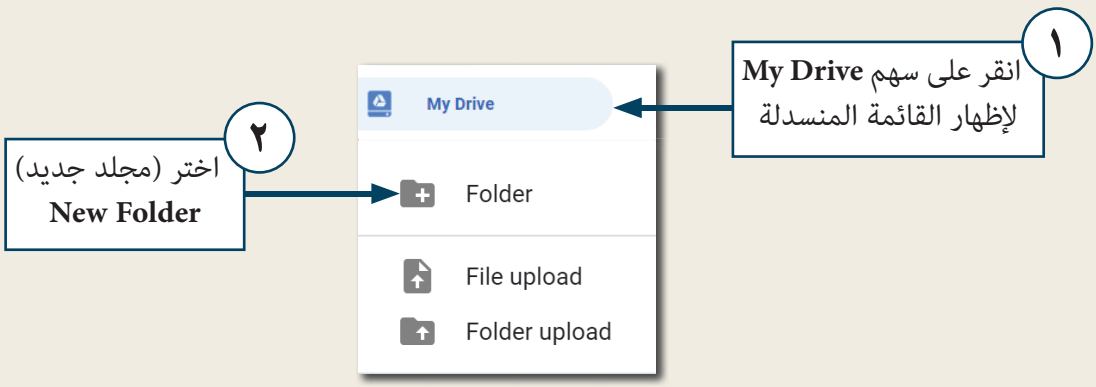
١. تغيير لغة واجهة الخدمة.
٢. تغيير طريقة عرض الملفات إما على شكل شبكة أو قائمة.
٣. البحث Search عن ملف داخل التطبيق.
٤. ترقية المساحة التخزينية المتاحة Upgrade Storage.

استكشف

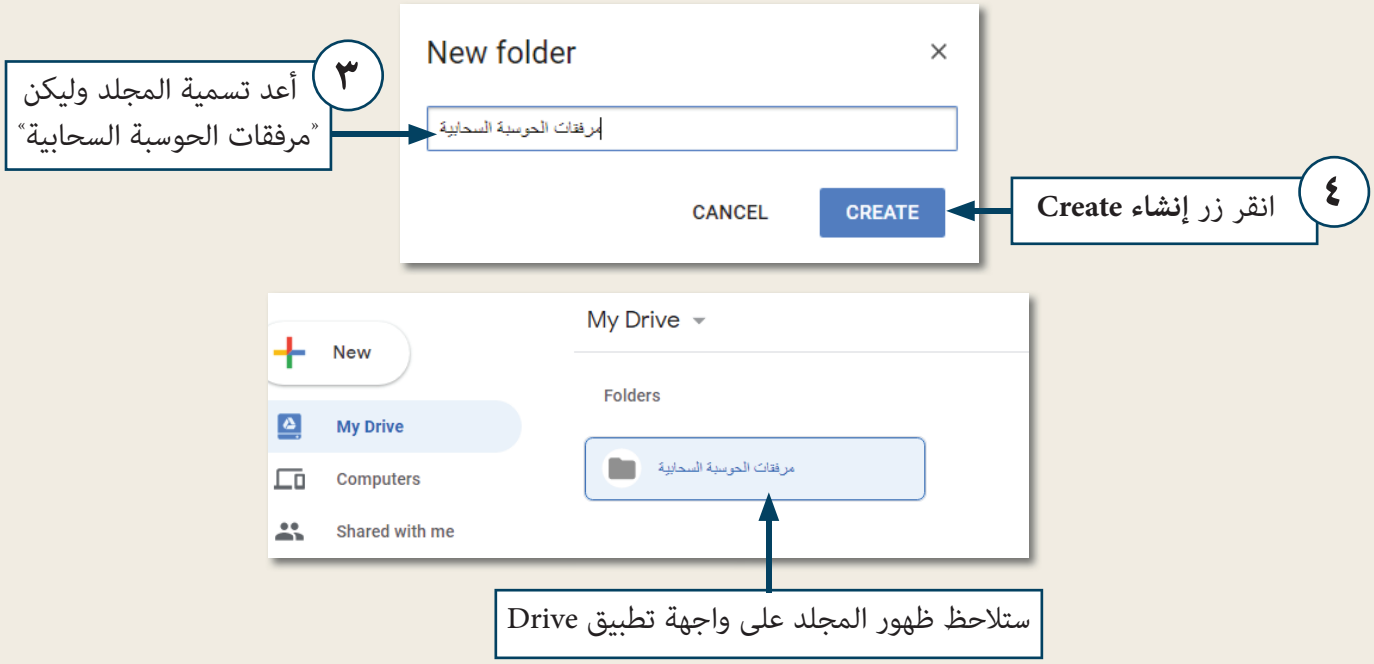
التعامل مع المجلدات والملفات

يتيح لك تطبيق Drive إنشاء وتحميل الملفات بكافة أنواعها (مثل : مستندات وصور وصوت وفيديو)، بالإضافة إلى عرضها وتعديلها ومشاركتها.

لإنشاء مجلد جديد:

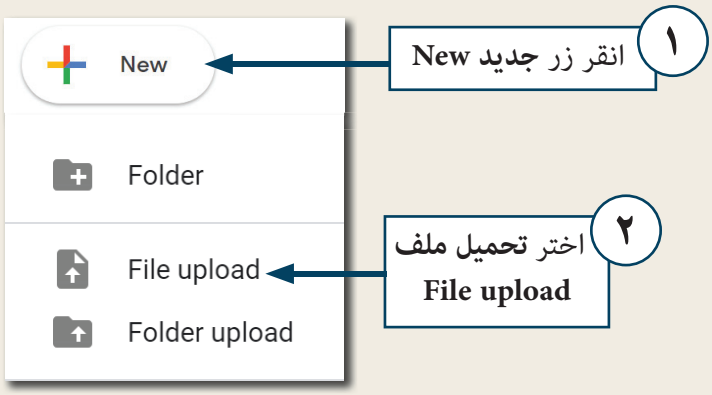


من النافذة التي ستظهر:

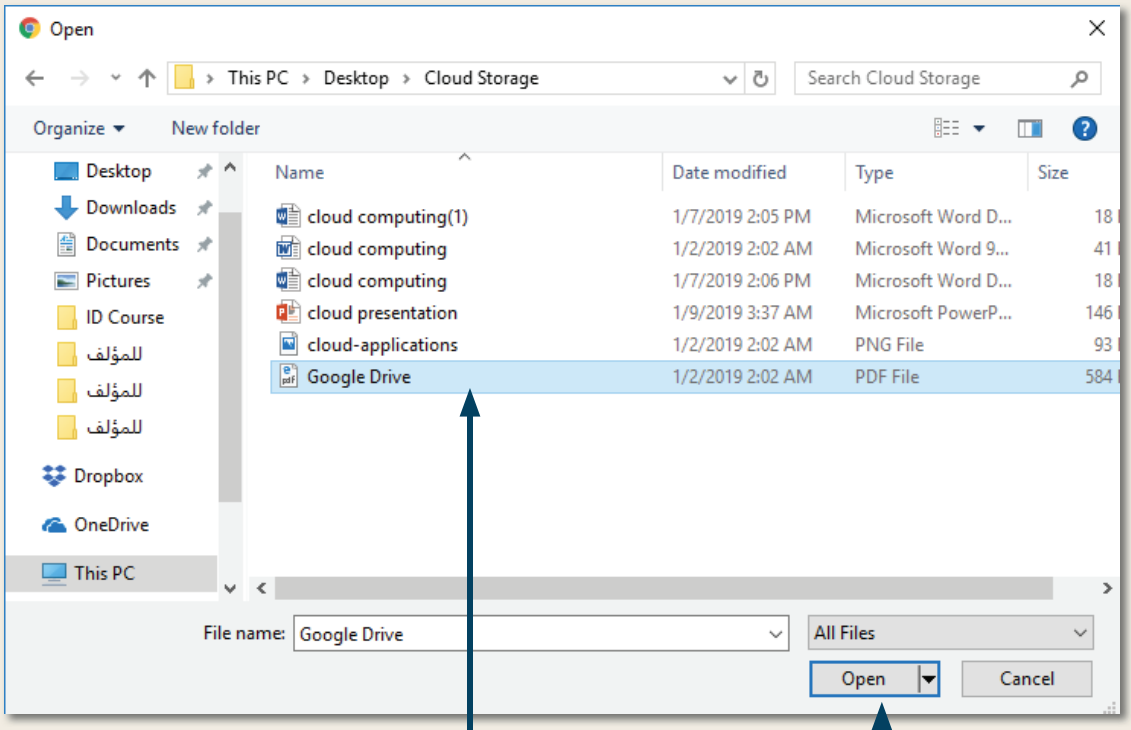


لتحميل ملف بداخل المجلد:

- افتح مجلد (مرفقات الحوسبة السحابية)، الذي أنشأته في الخطوة السابقة :



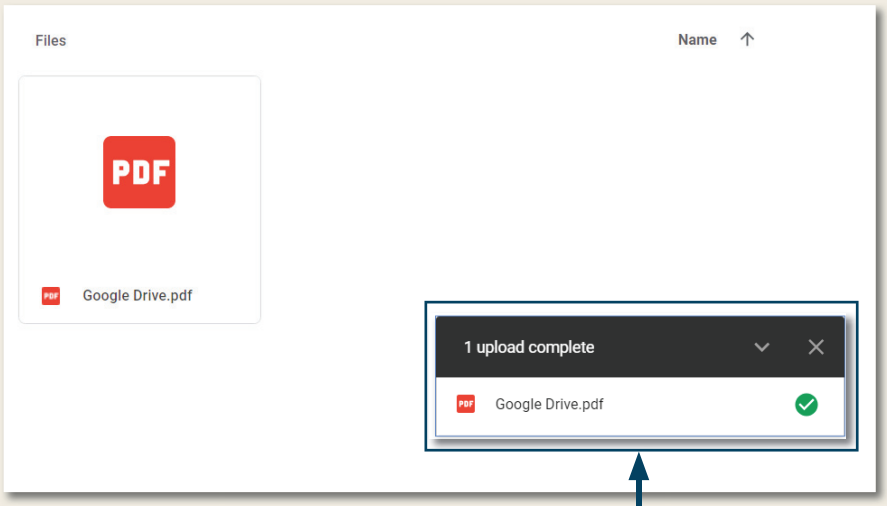
من مربع الحوار فتح Open :



١ حدّد الملف المرفق (Google Drive. pdf) من المجلد Cloud Storage

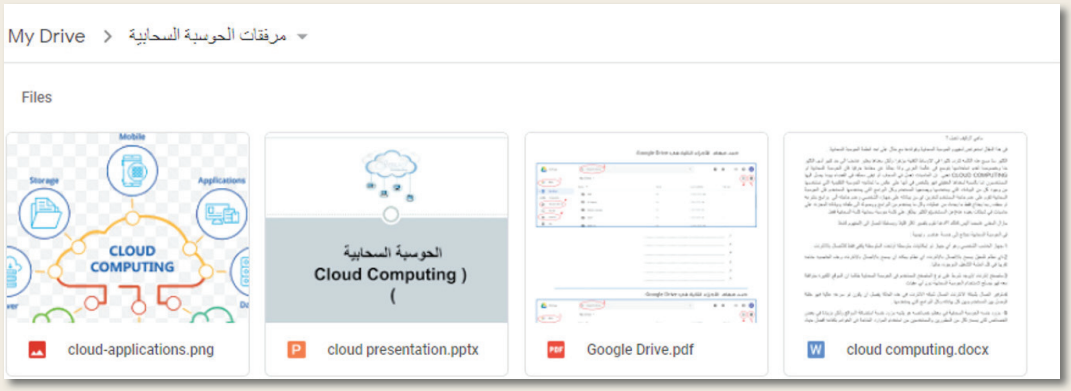
٢ انقر فتح Open

عنـد تحميـل الملفـات فـي تطبيق Drive، ستلاحظ أنها تشغـل حيِّـزًا مـن المساحة التخزينية المتاحة لك.



ستلاحظ بدء عمليّة تحميل الملف و من ثم ظهور علامة « ✓ » بعد الانتهاء من التحميل

كرِّر الخطوات السابقة لتحميل بقية الملفات من المجلد المرفق **(Cloud storage)**، لتظهر لك الواجهة كالتالي:

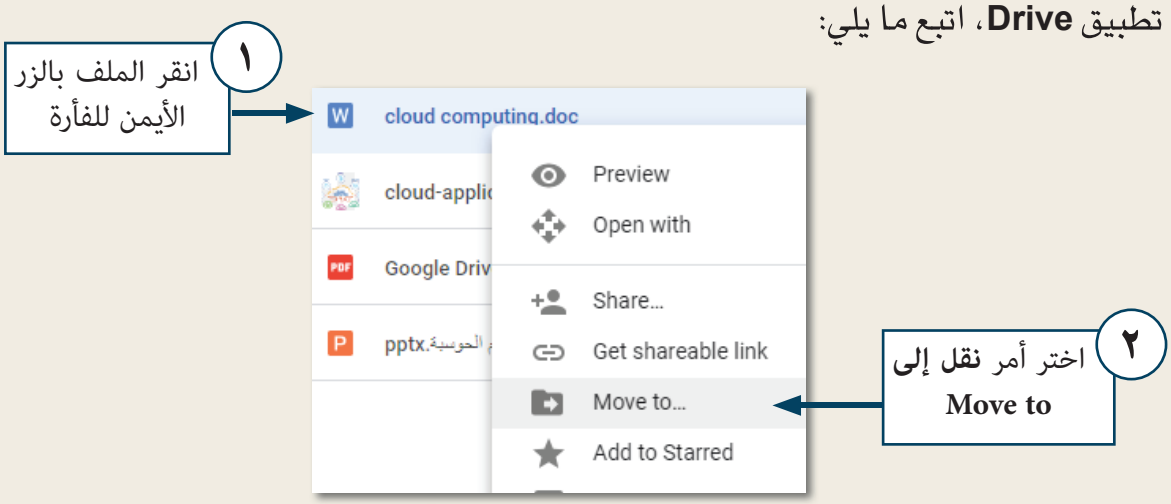


جرّب: إعادة ترتيب الملفات تصاعديًّا أو تنازليًّا.

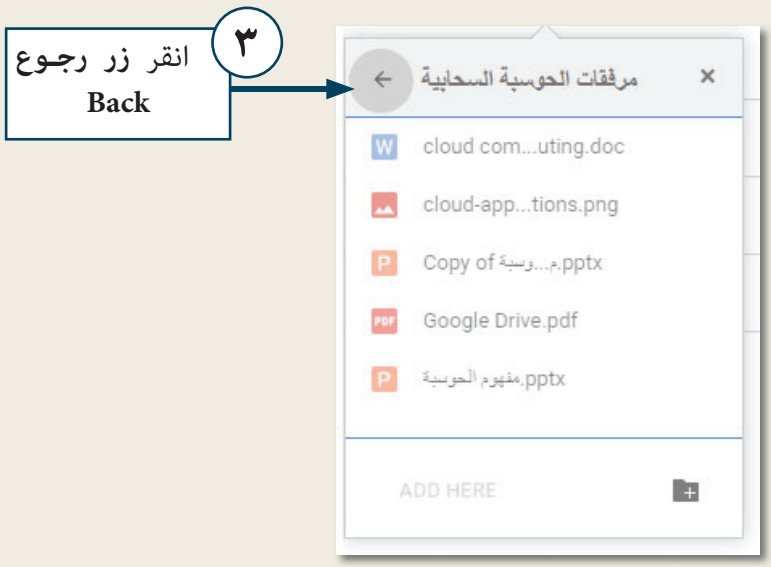
استكشف

- طرقًا أخرى لتحميل الملفات في تطبيق **Drive** .
- حجم المساحة التخزينية **Storage** الذي شغلته الملفات بعد رفعها من المساحة المتاحة لديك.
- كيفية تنزيل ملف **Download** .

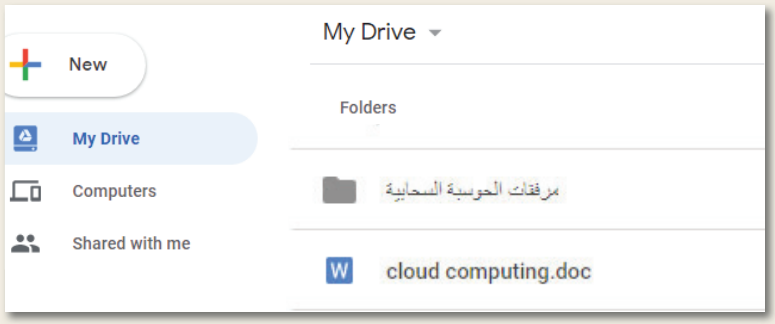
لنقل ملف (**cloud computing.doc**) من مجلد (مرفقات الحوسبة السحابية) إلى النافذة الرئيسية في تطبيق **Drive**، اتبع ما يلي:



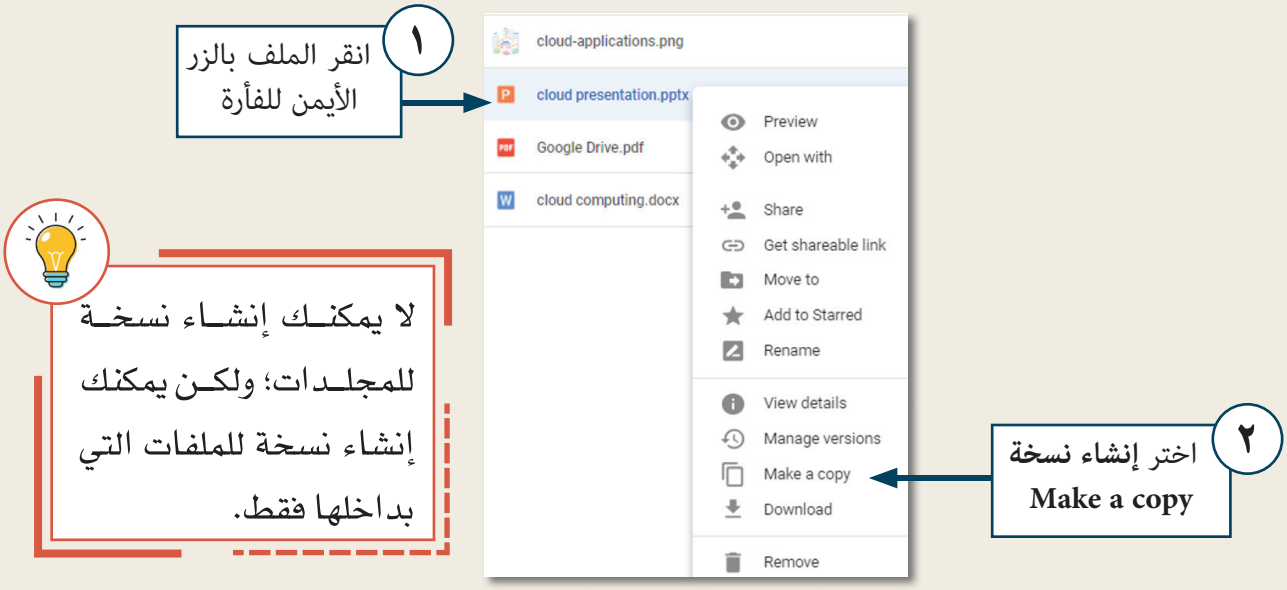
من النافذة التي ستظهر لك:



ستلاحظ انتقال الملف إلى واجهة تطبيق Drive:



لإنشاء نسخة لملف (**cloud presentation.pptx**):



ستلاحظ ظهور نسخة من الملف بنفس الاسم مع إضافة (**copy of**).

- أعد تسمية النسخة الجديدة.

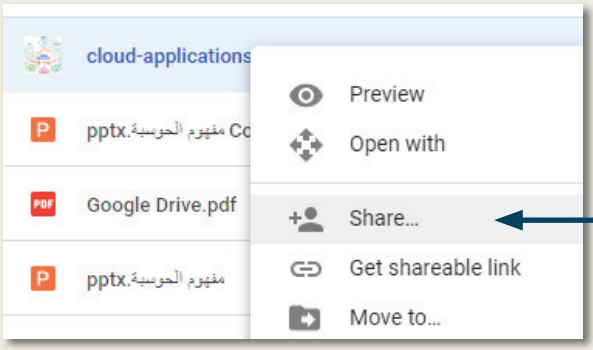
استكشف

كيفية حذف الملفات من تطبيق **Drive** واستعادة العناصر المحذوفة.

مشاركة المجلدات والملفات

يمكنك مشاركة الملفات والمجلدات التي خزَّنْتهَا في **Google Drive**، كما يمكنك تحديد صلاحيات المستخدمين للتعامل مع هذه الملفات سواء تعديل الملف أو التعليق عليه أو عرضه فقط.

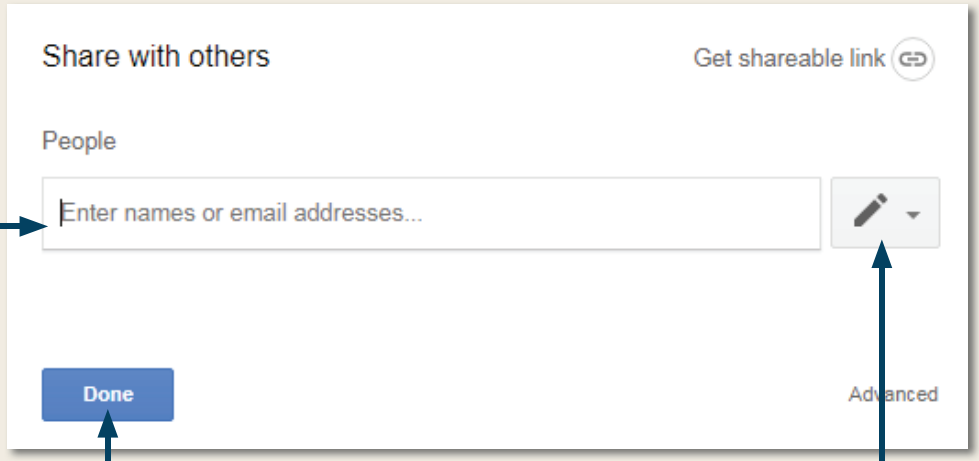
لمشاركة الآخرين صورة (cloud-applications.png) المخزنة في مجلد(مرفقات الحوسبة السحابية)، اتبع الآتي:



١ انقر الصورة بالزر الأيمن للفأرة، واختر أمر **مشاركة Share**

إذا قمت بمشاركة الملفات مع مستخدمين لديهم حسابات بريد إلكتروني غير Google، سيتمكّن المستخدم من عرض الملف فقط، ولا يمكن التعليق عليه أو تعديله.

ستظهر لك نافذة مشاركة مع الآخرين **Share with other** :



٢ أدخل أسماء أو عناوين البريد الإلكتروني للأشخاص الذين تود مشاركتهم

٣ حدِّد الصلاحيات التي يمكن للأشخاص تنفيذها في ملفك، وليكن **للتعديل Can Edit**

٤ انقر على **تم Done**

ستلاحظ ظهور رمز المشاركة بجانب الملف .

جرّب مشاركة عدة ملفات في نفس الوقت.

معلومة تهمك

عند مشاركة الملفات يمكنك تحديد الصلاحيات التي تمنحها للمستخدمين، كالآتي:

- **العرض View:** يتيح للمستخدمين الاطّلاع على الملف، دون تغييره أو مشاركته مع الآخرين.
- **التعليق Comment:** يتيح للمستخدمين إدخال تعليقات واقتراحات دون إمكانية تغيير الملف أو مشاركته مع الآخرين.
- **التحرير Edit:** يتيح للمستخدمين إجراء تغييرات وقبول الاقتراحات أو رفضها ومشاركة الملف مع الآخرين.

- طريقة مشاركة الملفات مع الآخرين من خلال رابط أو ارسال دعوة.
- خيارات التحكم في مدى مشاركة الملف مع الآخرين ومن لديه حق الدخول للملف.

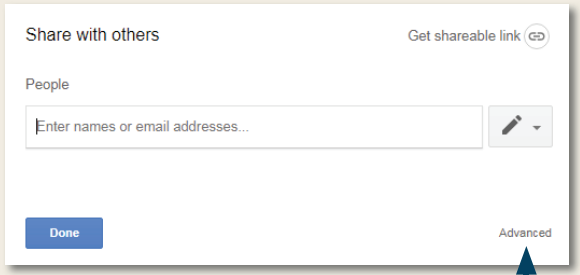
نقل ملكية الملفات

يمكنك نقل ملكية ملف أنشأته في **Google Drive** إلى مستخدم آخر لديه حساب في **Google** بعد مشاركته الملف.

ولنقل ملكية ملفك إلى شخص آخر، اتبع ما يلي:

- حدِّد الصورة (**cloud-applications.png**) التي شاركتها مسبقًا، ثم اضغط رمز المشاركة أعلى النافذة .

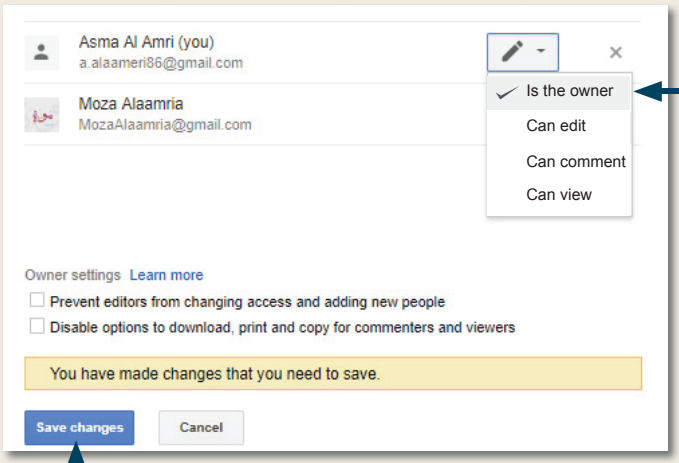
من نافذة شارك مع الآخرين Share with other:



٢ انقر خيارات متقدمة Advanced

إذا نقلت ملكية عدد كبير من الملفات والمجلدات دفعة واحدة، فقد يستغرق بعض الوقت حتى تظهر لك التغييرات.

ستظهر لك نافذة إعدادات المشاركة Sharing Settings:



٣ انقر السهم، ثم اختر المالك Is the owner

٣ انقر حفظ التغييرات Save changes

ابحث

طريقة تمكنك من تحويل شخصًا آخر مالكًا لأحد مجلداتك، تظل أنت مالك الملفات التي يحويها المجلد، حتى يقرر المالك الجديد تغيير مستوى وصولك.

الدرس الرابع

النماذج الإلكترونية

تعد النماذج الإلكترونية أداة سهلة لجمع البيانات و حفظها بشكل تلقائي وتنظيمها في جدول عبر الإنترنت، حيث يمكنك إنشاء نماذج متنوعة تساعدك في إنجاز العديد من المهام مثل: نموذج اتصال،أو استبيان، أو اختبار إلكتروني. وتوجد عدة أمثلة لتطبيقات ومواقع تقدِّم خدمة تصميم النماذج الإلكترونية كخدمة سحابية، مثل: **Google Forms**، **Survey Monkey**، **Microsoft Forms**، وغيرها.

مزايا استخدام النماذج الإلكترونية

- سهولة التصميم
- تجميع البيانات وتحليلها
- مشاركتها مع الآخرين
- التقييم التلقائي للأسئلة

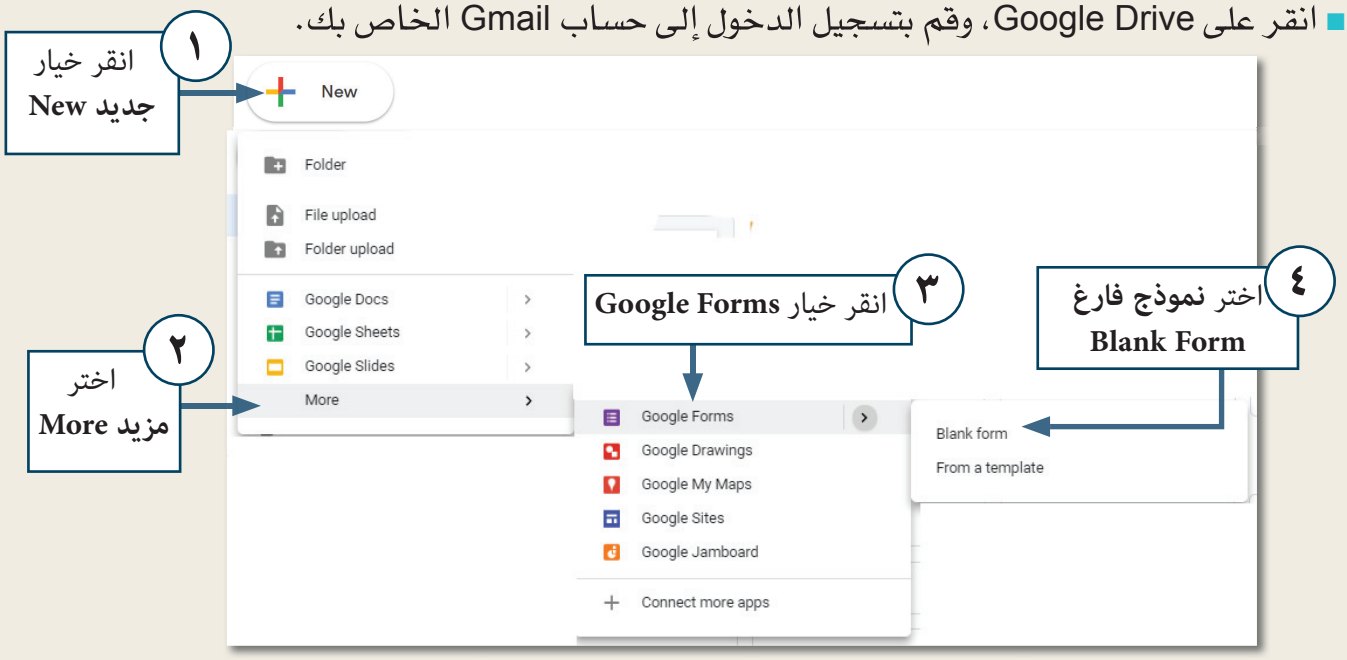
خدمة نماذج Google

تُعدُّ نماذج جوجل (Google Forms) إحدى الخدمات المدمجة ضمن خدمة التخزين السحابي (Google Drive)، التي تستخدم لتصميم الإستبانات والاختبارات الإلكترونية بسهولة، مع إمكانية متابعة ردود المشاركين وتحليلها التي يتم تخزينها بشكل تلقائي في جدول لمحرِّر مستندات Google.

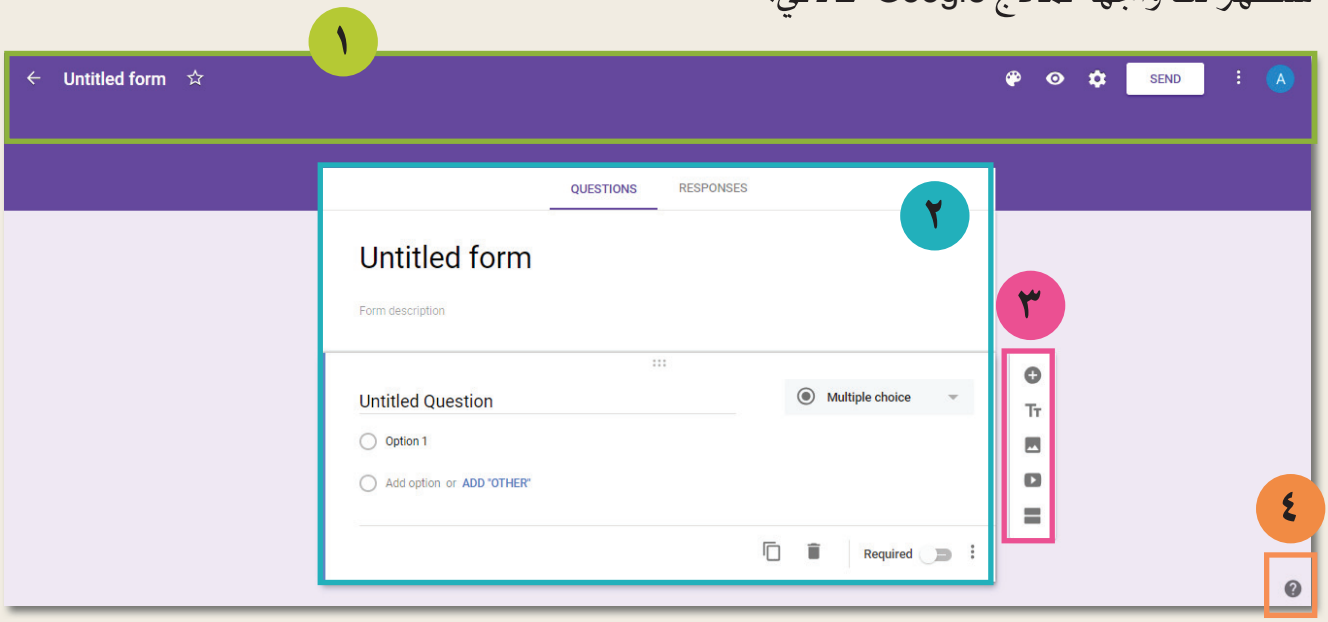
استكشاف واجهة نماذج Google

للوصول إلى خدمة نماذج Google، اتبع الآتي:

- انقر على Google Drive، وقم بتسجيل الدخول إلى حساب Gmail الخاص بك.



ستظهر لك واجهة نماذج Google كالآتي:



١ منطقة إدارة النموذج Form Management Area : تتضمن عنوان النموذج ومجموعة خيارات تمكّنك من التحكم بسمات وخصائص النموذج وغيرها من الإعدادات، بالإضافة إلى زر الإرسال لمشاركة النموذج.

٢ منطقة العمل Work Area: وتحتوي على تبويبين:

- **تبويب الأسئلة Questions:** يتضمن مساحة لعنوان النموذج ووصفه، وحقول الأسئلة، كما يسمح للمستخدم التحكم في خصائص الأسئلة كنسخها وحذفها وترتيبها.
- **تبويب الردود Responses:** يتيح لك مشاهدة الردود على النموذج وربطها بجدول بيانات بالإضافة إلى خيارات أخرى لإدارة هذه الردود.

٣ شريط أدوات السؤال Question Toolbar: يتيح لك إضافة عناصر مختلفة إلى جزء السؤال مثل: سؤال جديد، أو صورة، أو مقطع فيديو، أو قسم جديد.

٤ زر المساعدة والتعليقات Help & Feedback: يتيح لك معرفة نبذة مختصرة حول بعض أزرار النموذج، والإبلاغ عن أي مشكلة قد تواجهك، والإجابة عن استفساراتك بالإضافة إلى إمكانية الرجوع إلى النماذج السابقة.

استكشف

- طرقًا أخرى للوصول إلى خدمة نماذج Google .

معلومة تهمك

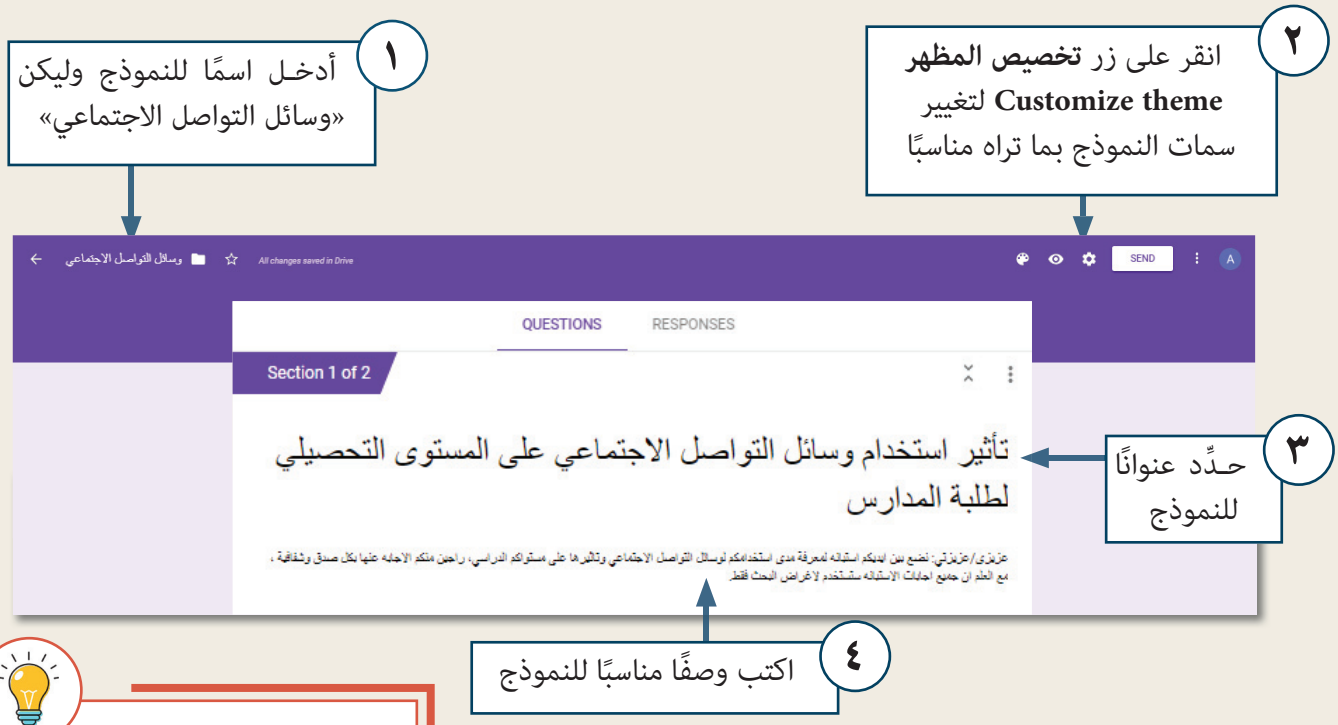
عند إنشاء إستبانة، احرص على إضافة البيانات الآتية:

- معلومات عن الإستبانة لتساعدهم على فهم الغرض منه.
- عدد الأسئلة التي تتضمنها الاستبانة.
- المدة اللازمة للإجابة عن الأسئلة.
- شكر الأفراد الذين ستَجمع منهم الردود على الوقت الذي سيأخذونه للرد.

تصميم نموذج استبانة

قبل البدء في تصميم نموذج استبانة حول (تأثير استخدام وسائل التواصل الاجتماعي على المستوى التحصيلي لطلبة المدارس)، خطِّط لعمل الاستبانة مستعينًا بالمرفق (Social media questionnaire.doc)**، ثم اتبع الآتي:**

- أضف عنوانًا واسمًا للنموذج.



ستلاحظ حفظ العمل في نماذج Google بشكل تلقائي.

لتغيير مكان حفظ النموذج:

- انقل النموذج إلى مجلد (مرفقات الحوسبة السحابية) الذي أنشأته سابقًا في **Google Drive** بالنقر على زر **نقل إلى مجلد Move to folder** الموجود في منطقة إدارة النموذج.

عند إدخال اسم النموذج، يقوم التطبيق بإضافته كعنوان افتراضي للنموذج .

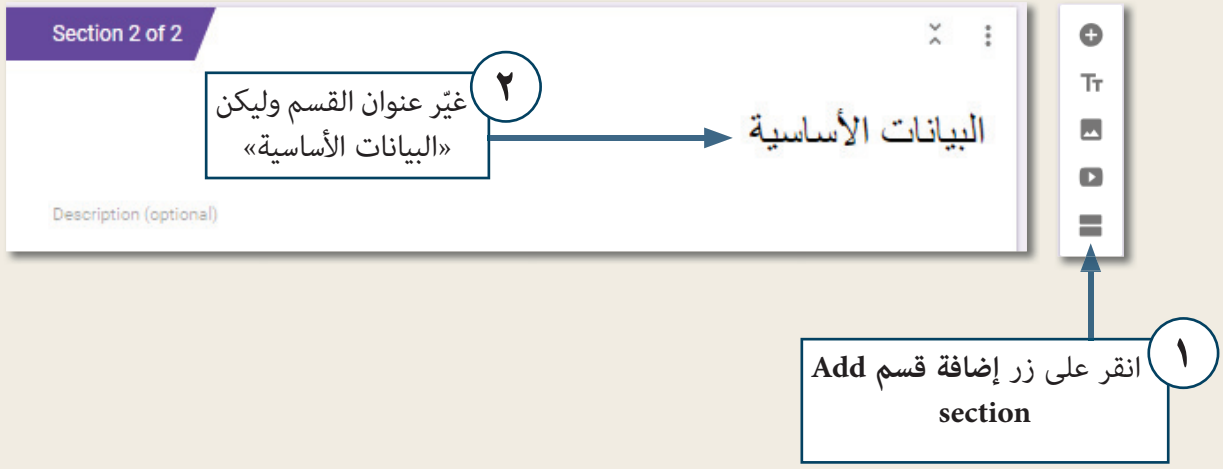
تحرير الأسئلة

تتيح لك نماذج Google إضافة أنواع مختلفة من الأسئلة مثل: الاختيار من متعدد، وإجابات قصيرة وغيرها، بالإضافة إلى إمكانية إدراج الصور ومقاطع الفيديو.

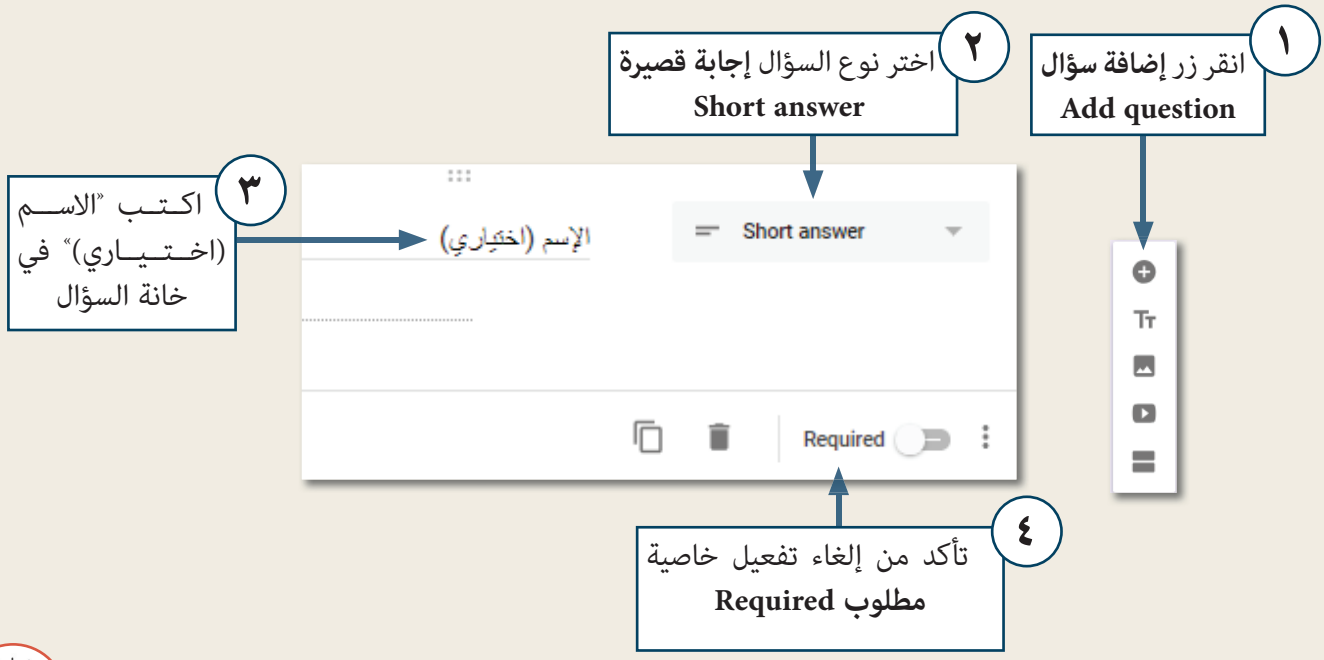
لإضافة قسم خاص ببيانات المشاركين، اتبع الآتي:

- تأكد من تنشيط تبويب **الأسئلة Questions**.

عند تصميمك للاستبيان يجب مراعاة تقسيمه إلى أجزاء مختلفة طبقًا لطبيعة البيانات المطلوبة من المشاركين، ليتم طرح الأسئلة في محاور واضحة و مستقلة؛ مما يُسهِّل عليك تجميع الردود وتحليل النتائج.



- ابدأ بكتابة البيانات الأساسية المطلوب تعبئتها من المشاركين، كما يلي:



لإضافة بقية البيانات الشخصية المطلوبة، اتبع الآتي:

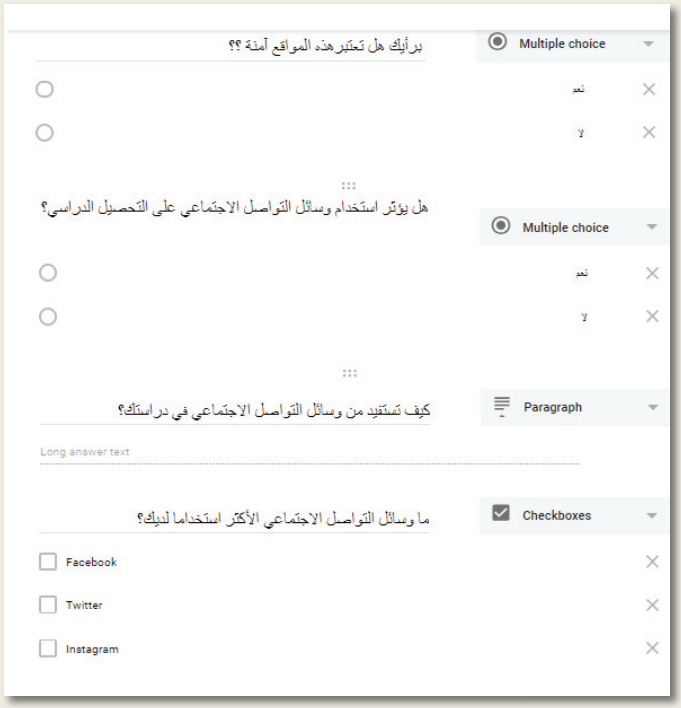
- أضف سؤالًا جديدًا من شريط أدوات السؤال.

عند تفعيلك لخاصية مطلوب Required فإنك تجبر الشخص على الإجابة عن السؤال قبل النقر على التالي.

لإضافة أسئلة الاستبانة:

- أضف قسمًا جديدًا، وغيّر اسمه إلى (أسئلة الاستبانة).
- افتح المجلد المرفق (**Google Forms**) واستعن بالمرفق (**social media questionnaire.doc**) لإضافة بقية الأسئلة.
- تأكد من تفعيلك لخاصية مطلوب **Required** في جميع الأسئلة.

سيظهر لك نموذج الاستبانة كالآتي:



استكشف

- بقية أنواع الأسئلة التي توفِّرها نماذج **Google**.
- كيفية إعادة ترتيب الأسئلة بعد إنشائها.
- كيفية حذف سؤال.

- افتح المجلد المرفق (**Google Forms**) وأدرج الصورة (**social-media.jpg**) بجانب السؤال الرابع في النموذج.
- أعد تحجيم الصورة وضبط **محاذاتها في الوسط center align**، لتبدو كالآتي:

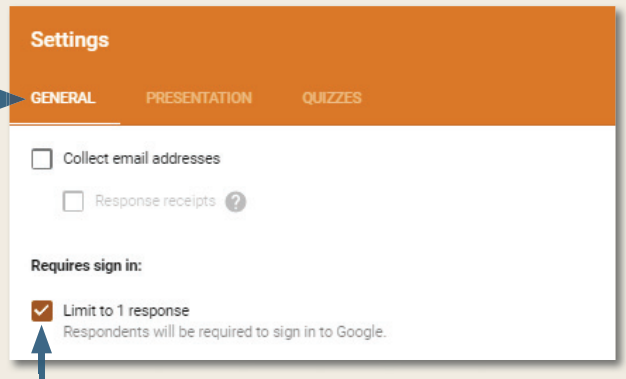
معاينة النموذج ونشره

بعد الانتهاء من تصميم النموذج، اضبط الإعدادات المخصَّصة، باتباع ما يلي:

- انقر خيار إعدادات **Setting** .

ولتقييد عدد مرات تهيئة المشاركين للنموذج مرة واحدة فقط:

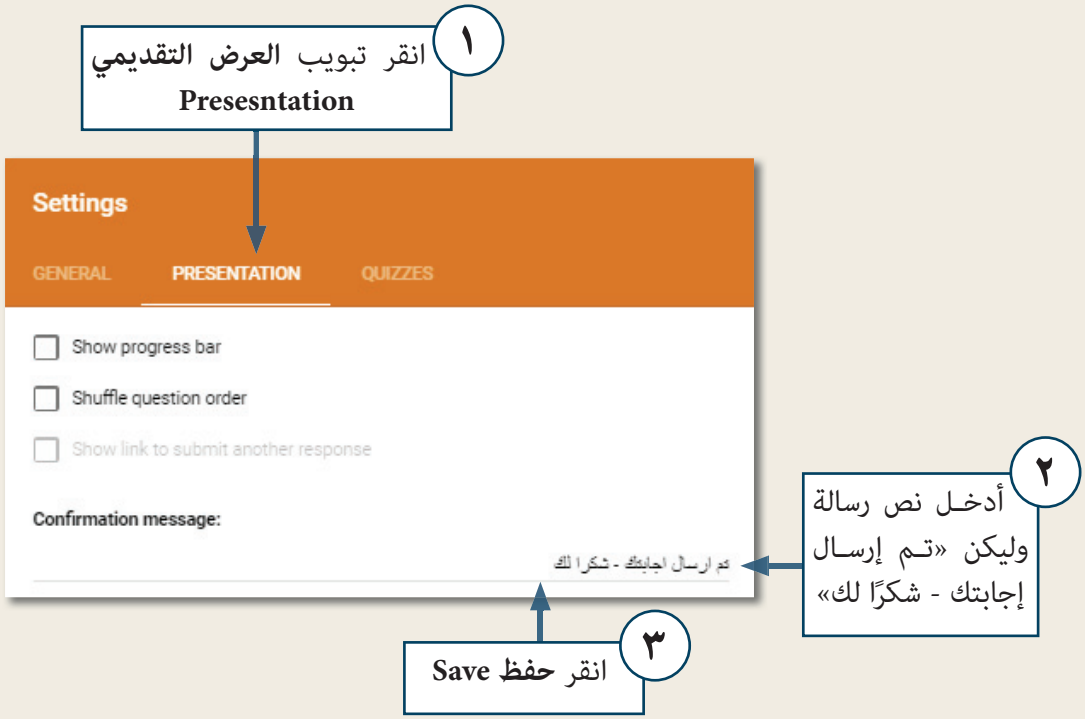
١ انقر تبويب عام **General**



٢ فعِّل خيار **تقييد عدد مرات تعبئة النموذج لمرة واحدة فقط Limit to 1 response**

تفعيلك لخيار (تقييد عدد مرات الرد لمرة واحدة)، يُلزم المشاركين على تسجيل الدخول لحساب Google قبل تعبئة النموذج وتمنع مشاركة الآخرين الذين لا يملكون حسابًا في Google.

لإضافة رسالة التأكيد التي ستظهر للمشارك بعد إرسال النموذج:



استكشف

الخيارات الموجودة في تبويب الاختبارات **Quizzes** .

- قُم بمعاينة النموذج قبل نشره، والتأكد من ضبط الإعدادات ومعرفة كيف سيكون بشكله النهائي، باتباع الآتي:

- انقر على أداة **المعاينة Preview** ليظهر لك كالآتي:

يمكنك النقر على زر تعديل هذا النموذج
Edit this form لتحرير النموذج وتعديل الأخطاء إن وجدت.

استكشف

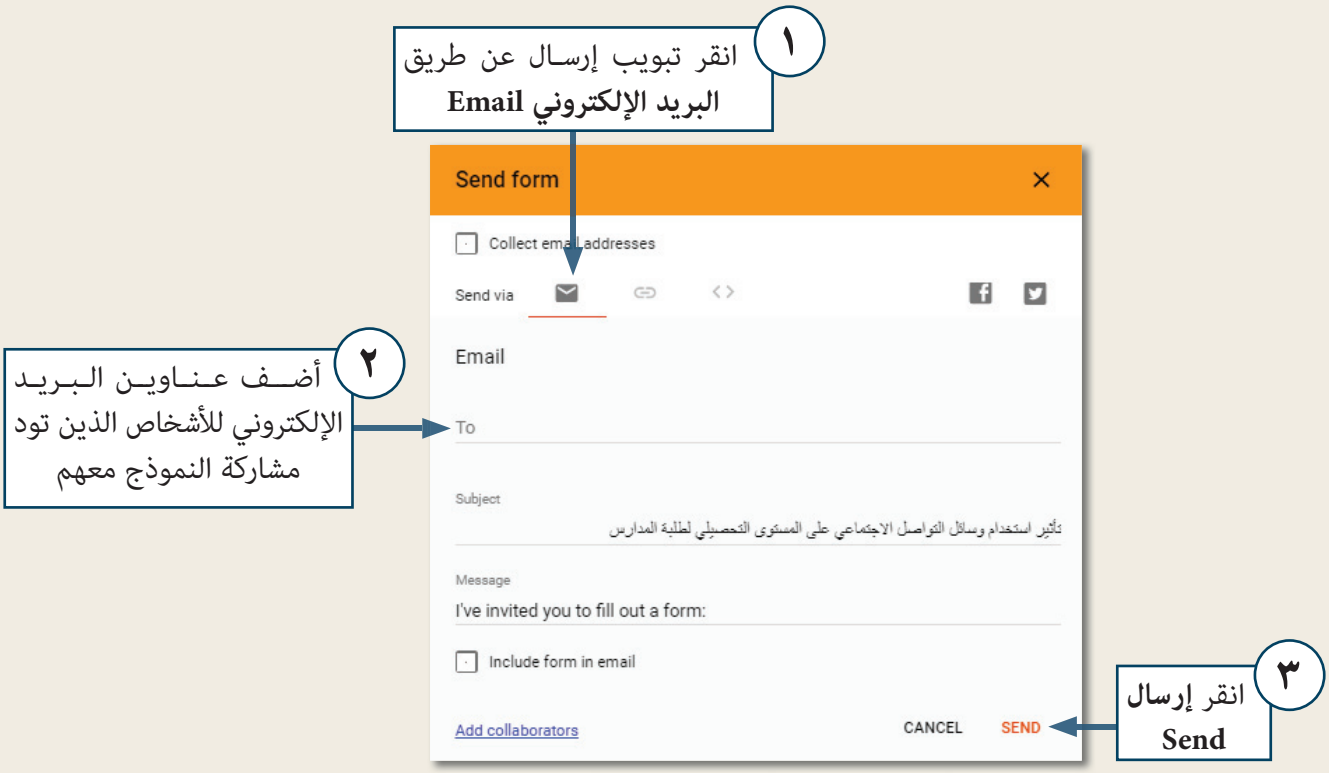
كيفية إضافة متعاونيين وتحديد صلاحياتهم.

يمكنك مشاركة النموذج مع الآخرين والاستعانة بوسائل التواصل الاجتماعي twitter ، Facebook باستخدام الطرق الآتية:

- **المشاركة عن طريق الرابط URL Link** : الذي من الممكن الحصول عليه من أسفل الشاشة.
- **المشاركة بتضمين HTML** : ويكون ذلك بأخذ الرمز الخاص بالاستبيان وتضمينه في مواقع الإنترنت، و لإتمام ذلك يجب النقر على خيار المزيد من الإجراءات (**More actions**).
- **المشاركة عن طريق البريد الإلكتروني Email**.

لمشاركة النموذج عن طريق البريد الإلكتروني، اتبع الآتي:

- انقر زر إرسال **Send** لتظهر لك نافذة إرسال النموذج **Send form**.



نشاط ٥: نشر النموذج

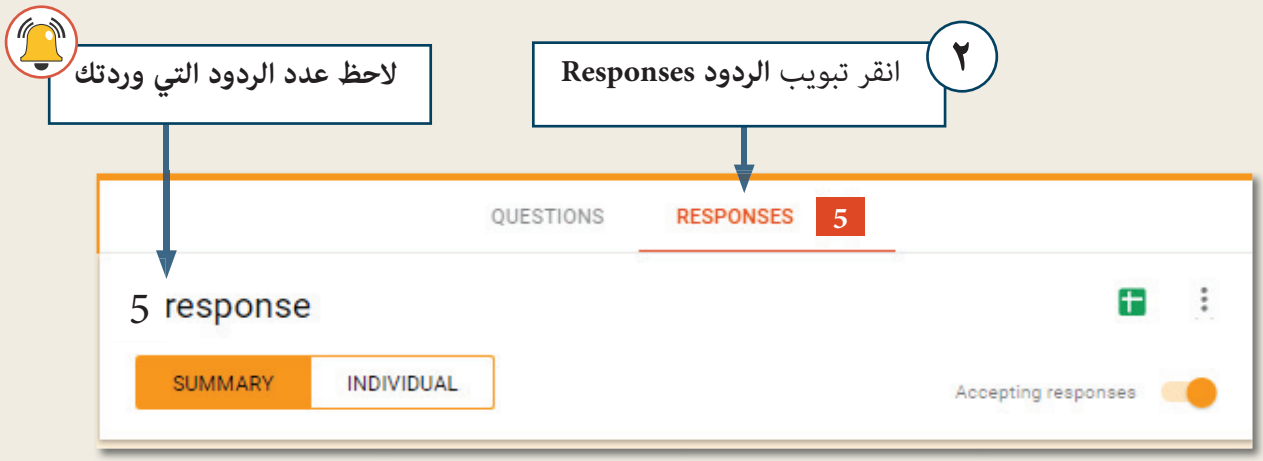
شـارك نموذج الاستبانة الذي أنشأته مع زملائك مستعينًا بإحدى وسائل التواصل الاجتماعي في نشر النموذج، واطلب منهم تعبئة النموذج وإعادة إرساله إليك.

تجميع الردود

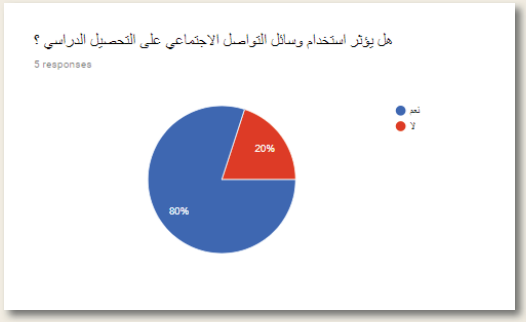
توفِّر لك **نماذج Google** إمكانية تجميع الردود وتخزينها بشكل تلقائي في تبويب **الردود Responses**، مع إظهار الرسومات البيانية وقوائم بالإجابات.

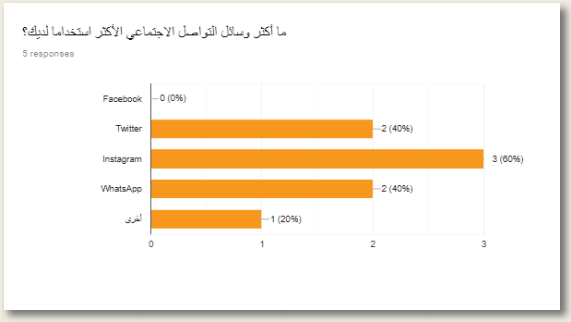
لمعاينة ردود المشاركين، اتبع الآتي:

من منطقة العمل:



- تصفّح الرسومات البيانية وقوائم الإجابات التي ظهرت معك في صفحة الردود.





الدرس الخامس

أمن الحوسبة السحابية

مع انتشار استخدام الحوسبة السحابية، وجَبَ على مزودي الخدمة اتخاذ التدابير الأمنية اللازمة لحماية الأنظمة والتطبيقات والبيانات المخزنة في السحابة، بهدف ضمان تقديم الخدمة بشكل آمن والتصدي للتهديدات التي قد تواجه الحوسبة السحابية؛ مثل اختراق الحسابات، وسرقة البيانات، وحجب الخدمة وغيرها.

عناصر أمن الحوسبة السحابية

تعتمد كفاءة تقديم خدمة الحوسبة السحابية على مدى توفر عدة عناصر في الأمن، منها:

إدارة الهوية

عملية إدارة تسجيل دخول المستخدم إلى السحابة، والتحقق من صحة هويته.

حماية الخصوصية

توفير مزود الخدمة السياسات والإجراءات المناسبة لحفظ حقوق المستخدم عند استخدام بياناته.

حماية البيانات

تقييد الوصول إلى البيانات والمعلومات الشخصية لغير المصرَّح لهم، سواء من قبل مزود الخدمة أو المستخدم.

أمن التطبيقات

رفع مستوى نظم الأمن والحماية عند تطوير التطبيقات من قبل المبرمجين لمنع اختراق السحابة.

تهديدات أمن الحوسبة السحابية

بالرغم من الخدمات التي تقدمها الحوسبة السحابية، إلا أن هناك العديد من التهديدات الأمنية التي يواجهها كُلٌّ من المستخدم ومزود الخدمة، ومن الأمثلة عليها:

فقدان البيانات Data Loss

يحدث فقدان المستخدم لبياناته إما بسبب الاختراق أو أخطاء من مزود الخدمة؛ مما يؤدي إلى تبعات مالية وقانونية للأفراد أو الشركات المزوِّدة للخدمة.

هجمات حجب الخدمة
Denial of Service Attacks

إرسال هجمات عن طريق أجهزة مصابة ببرامج ضارة بهدف التحكم بالسحابة؛ مما يؤدي إلى بطء الخدمة أو منع وصول المستخدم إليها.

أخطار موظفي الخدمة
Malicious Insiders

استغلال الموظفين لصلاحيات الوصول إلى الأنظمة الداخلية لمزود الخدمة ومعرفة نقاط ضعفها؛ مما يؤدي إلى انتهاك خصوصية المستخدمين وتسريب بياناتهم.

التطبيقات غير الآمنة
Insecure APIs

بناء تطبيقات غير آمنة من قبل المخترقين بهدف إلحاق الضرر ببيانات المستخدمين من خلال استغلال واجهة برمجة التطبيقات.

نشاط ٦: مخاطر الحوسبة السحابية

تأمَّل المخطط التالي الذي يوضح محاولة أحد المخترقين لاختراق حساب أحد المستفيدين من خدمات الحوسبة السحابية، ثم أجب عن الأسئلة الآتية:

- هل يمكن اختراق السحابة ؟ ولماذا؟
- ما المخاطر التي سيتعرض لها المستخدم إذا تم اختراق السحابة؟
- ما التدابير الأمنية التي تتخذها الشركة المزوِّدة للخدمة لحماية المستخدمين في حالة حدوث ثغرة أمنية أو اختراق لحساباتهم الخاصة؟
- من خلال المهارات التي اكتسبتها في مجال أمن المعلومات وحماية البيانات، ما التدابير الأمنية التي يجب عليك أن تتبعها كمستخدم لحماية بياناتك؟

اختبر معلوماتك لتقيس ما تعلمته عن قواعد البيانات، ثم ناقش معلمك لتتحقق من صحة إجابتك:

أولا: أجب عن الأسئلة الآتية:

- ما مزايا استخدام تقنية الحوسبة السحابية؟
- ما متطلبات الوصول إلى خدمات الحوسبة السحابية؟
- عدّد الفئات المستفيدة من خدمات الحوسبة، والخدمات التي يوفِّرها مزودو الخدمة لكل فئة.
- قارن بين خاصية مشاركة الملفات وخاصية نقل ملكية الملفات في التخزين السحابي.
- اذكر بعض التهديدات الأمنية التي يواجهها كُلٌّ من المستخدم ومزود الخدمة في الخدمات السحابية.

ثانيا: أكمل الفراغات بالكلمة المناسبة:

- أحد الأمثلة على الشركات المزوِّدة للخدمات السحابية شركة..... .
- نماذج تعتمد على نوع الخدمة التي تقدمها السحابة، وتختلف فيما بينها في الخدمات والصلاحيات المتاحة للمستخدم، هي نماذج .....
- يتم نقل الملفات المخزنة على تطبيق Drive الخاص بك من مجلد إلى آخر، باستخدام أمر .....
- يمكنك إنشاء نماذج إلكترونية متنوعة تساعدك في إنجاز العديد من المهام مثل: ..... ، .....
- عناصر الأمن اللازم توافرها عند تقديم خدمة الحوسبة السحابية هي: ..... ، ..... ، ..... .

ثالثا: استخدم المهارات التي اكتسبتها في كيفية إعداد نماذج إلكترونية لإنشاء اختبار إلكتروني يشمل خمسة أسئلة متنوعة في وحدة «الحوسبة السحابية» ، من خلال تطبيق المهارات الآتية:

- أعد تسمية الاختبار بالاسم الذي تراه مناسبًا.
- اكتب عنوانًا ووصفًا يوضِّحان الهدف من الاختبار.
- استخدم خاصية (الترتيب العشوائي للأسئلة) لتغيير ترتيب ظهور الأسئلة لزملائك.
- خصِّص درجتين لكل سؤال، باستخدام خيار تفضيلات Preferences من زر المزيد More.
- قيّد عدد مرات الإجابة بحيث لا تزيد على مرة واحدة فقط.
- غيّر مظهر الاختبار، وخصّص صورة مناسبة كخلفية للاختبار.
- حدّد مكان حفظ الردود أو الإجابات عن الاختبار.
- شارك نموذج الاختبار الذي أعددته مع زملائك، ثم شارك نتائج الإجابات مع معلمك.

المراجع والمصادر

- أبو العطا، مجدي. (2010م) المرجع الأساسي لمستخدمي شبكات الكمبيوتر - القرية لعلوم الحاسب، مصر.

- Waller, David, Computer Science - Cambridge, Oxford.

- Using Packet Tracer to Build A Network. Retrieved from:

https://fas-web.sunderland.ac.uk/~cs0ssw/PacketTracerTutorial.pdf

- Packet Tracer Help Menu.

- حسنين، رجب عبد الحميد. أمن شبكات المعلومات الإلكترونية: المخاطر والحلول Cybrarians Journal. - ديسمبر 2012)، متاح في:

http://www.journal.cybrarians.org/index.php?option=com_content&view=article&id=629:networks&catid=257:studies&Itemid=90

- Hayward, D. (2016, April 17). How to secure your home network from hackers? Retrieved from Trusted Reviews:

http://www.trustedreviews.com/how-tos/how-to-secure-your-homenetwork-from-hackers

- Rubens, P. (October 10, 2018). How to secure your Wi-Fi at home and in your business, Retrieved from:

https://www.techradar.com/news/networking/wi-fi/five-tips-for-a-secure-wireless-network-1161225

- كوكس، جويس، لامبرت، جوان. (2010) خطوة خطوة Access2010 – الدار العربية للعلوم ناشرون – لبنان.

- أبو العطا، مجدي محمد. (2012م) Microsot Access 2010 – شركة علوم الحاسب – القاهرة.

- MARY LEMONS. (2016) Microsoft Official Academic Course MICROSOFT ACCESS 2016. Retrieved from:

https://www.dit.ie/media/ittraining/msoffice/MOAC_Access_2016.pdf

- University Information Technology Services. (2016) Microsoft Office Access 2016 for Windows Relational Databases & Subforums. Retrieved from:

https://apps.kennesaw.edu/files/pr_app_uni_cdoc/doc/Access_2016_PC_Relational_Databases_and_Subforms.pdf

- Microsoft. (2018) MS Access. Retrieved from:

https://www.tutorialspoint.com/ms_access/ms_access_tutorial.pdf

- University Information Technology Services. Microsoft Office Access 2016 for Windows Reports and Queries. Retrieved from:

http://www.gti.bh/Library/assets/access-2016-pc-reports-and-queries-1uzygs.pdf

رقم الإيداع : ٢٠٢١/٣٦٨٤م

طبع بالمطابع الفنية، سلطنة عُمان - هاتف: ٢٤٨١٤١٥٣

ISBN 978-99969-3-713-2

www.moe.gov.om