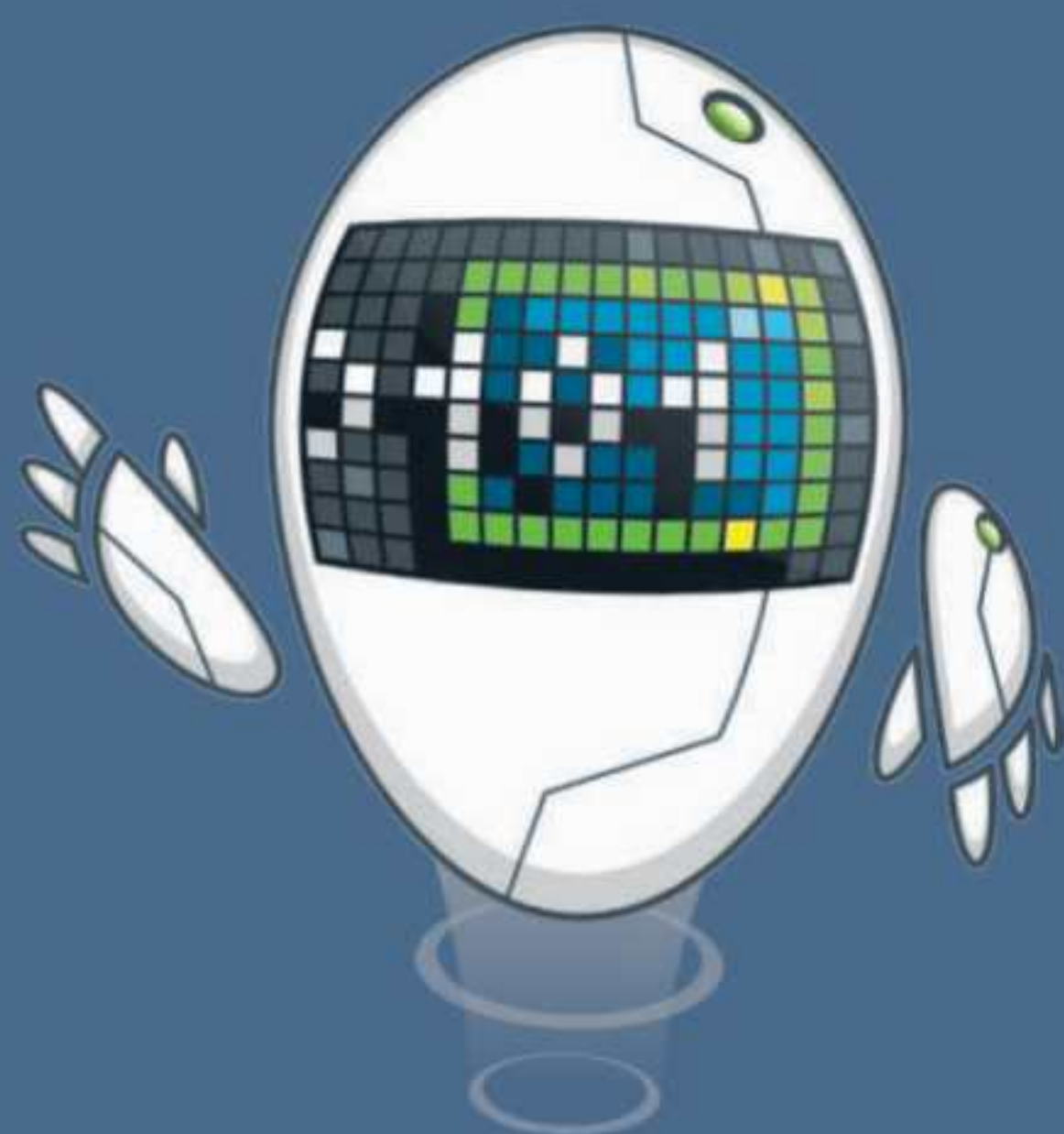


الوحدة الرابعة: البرمجة مع بايثون

في هذه الوحدة، ستتعلم المزيد عن عالم البرمجة، وستتعرف على أشكال التعليمات البرمجية بلغات البرمجة المختلفة، كما ستتعلم كيف يمكنك تمثيل الخوارزمية باستخدام المخطط الإنسيابي. بعد ذلك، ستكتب برنامجك الأول بلغة بايثون، وستتعلم ماهية المتغيرات في البرمجة وطرق استخدامها، وطريقة التفاعل مع المستخدمين للحصول على البيانات أو تقديم نتيجة، وكذلك طريقة استخدام بايثون لإجراء العمليات الحسابية وحل المشكلات.



الأدوات

< بيئة التطوير المتكاملة للغة بايثون
(Python IDLE)

أهداف التعلم

- ستتعلم في هذه الوحدة:
- < ماهية لغة برمجة بايثون.
- < مفهوم الخوارزمية.
- < كتابة الخطوات الخوارزمية.
- < تقسيم المشكلة إلى مهام أصغر.
- < مفهوم المخطط الإنسيابي.
- < تمثيل الخطوات الخوارزمية باستخدام مخططات الانسياب.
- < مراحل تكوين البرنامج.
- < التمييز بين أنواع المتغيرات.
- < تعيين قيم للمتغيرات بطرق مختلفة.
- < إدخال قيمة للمتغير.
- < تمييز الفروق بين أنواع البيانات المختلفة.
- < تعيين قيم للمتغيرات بطرق مختلفة.
- < استخدام معاملات الإسناد لإسناد قيم المتغيرات.
- < استخدام المعاملات الرياضية لإجراء العمليات الحسابية على الأرقام والمتغيرات.
- < استخدام نموذج السلحفاة في بايثون.
- < إنشاء مقطع برمجي لرسم أشكال بسيطة باستخدام مقطع برمجي في بايثون.



الدرس الأول: ما البرنامج

يتعلم الإنسان اتباع القواعد منذ نعومة أظفاره، وهكذا يستمر باتباع القواعد طوال حياته. إن ما يقوم الإنسان بعمله عند الاستيقاظ كل صباح هو أيسر مثال على تلك القواعد. قد لا تكون القواعد واضحة دائماً، وفي بعض الأحيان قد يحتاج الأشخاص إلى وضع قواعد جديدة وفقاً لمواقف محددة. لا يمكن لأجهزة الحاسب اتخاذ القرارات بنفسها، بل تتبع تعليمات محددة للغاية.

المقطع البرمجي هو مجموعة من الأوامر المكتوبة بلغة برمجة معينة لتنفيذ مهمة محددة.

كيف أكتب مقطعاً برمجياً

يُكتب المقطع البرمجي من قبل المبرمج، ويقوم جهاز الحاسب بقراءة التعليمات المقدمة له بلغة الآلة وهي اللغة الوحيدة التي يفهمها جهاز الحاسب وتتكون من 0 و 1. في الواقع يستحيل على المبرمج كتابة برنامج بلغة الآلة المكونة من 0 و 1، ولذلك يستخدم المبرمجون لغات برمجة "ذات مستوى أعلى" لكتابة المقطع البرمجي بكلمات مفهومة تصف التعليمات لجهاز الحاسب. وبمجرد كتابة البرنامج بلغة برمجة معينة، يستخدم المبرمج أدوات لتحويل هذه التعليمات إلى لغة الآلة التي يمكن تنفيذها بواسطة جهاز الحاسب.



python لغة برمجة بايثون

هناك المئات من لغات البرمجة، وستستخدم لغة برمجة بايثون (Python) لكتابة برنامجك الأول. لغة برمجة بايثون هي لغة عالمية عالية المستوى مفتوحة المصدر وسهلة التعلم، تعتمد على كتابة الأكواد (التعليمات البرمجية). يمكن استخدامها مع مجموعة متنوعة من التطبيقات وتُعد سهلة وممتعة للمبتدئين بالبرمجة.

لمحة تاريخية

تم إنشاء بايثون على يد جويدو فان روسوم عام 1991.

التعليمات البرمجية بلغات برمجة مختلفة



Scratch

سكراش

```
#Hello world! in Python  
print("Hello world!")
```

python

لغة برمجة بايثون

```
' Hello world! in Small Basic  
TextWindow.WriteLine ("Hello world!")
```



سمول بيزيك

```
/* Hello world! in C Ansi */  
#include <stdio.h>  
#include <stdlib.h>  
int main(void)  
{  
    puts("Hello world!");  
    return EXIT_SUCCESS;  
}
```



أنسي سي

```
# Hello world! in Ruby  
puts "Hello world!"
```



روبي

```
//Hello world! in C#  
class Hello world  
{  
    static void Main()  
    {  
        System.Console.WriteLine("Hello  
world!");  
    }  
}
```



سي شارب

يعتمد اختيار لغة
البرمجة على التطبيق
الذي تريد إنشائه.

معلومة

تنفذ أجهزة الحاسب التعليمات، لذلك إذا تم إعطاؤها تعليمات خطأ، فستكون النتيجة خطأ ولن يتم إنجاز العمل بالشكل المطلوب.

قبل بدء البرمجة، يجب أن تتعرف على بعض المفاهيم الأساسية في البرمجة.

الخوارزمية

تعلمت سابقاً أن الخوارزمية هي قائمة من التعليمات يتم اتباعها خطوة بخطوة لحل مشكلة معينة، ومن المهم أن تكون هذه التعليمات واضحة وسهلة ليتم تنفيذها دون أخطاء.



الخوارزميات في حياتنا اليومية

تجد الخوارزميات في كثير من شؤون حياتك اليومية. فاتجاهات الوصول إلى المتنزه هي خوارزمية، وكذلك وصفة إعداد الطعام هي خوارزمية، حيث تخبرك بالخطوات التي يجب اتباعها لإعداد طبق لذيذ.



المخرجات
(طبق شهية)



الخوارزمية
(إجراءات الطهي)



المدخلات
(المكونات)



اكتب خوارزمية لإعداد فطيرة محلاة

1. ضع جميع المكونات الصلبة (دقيق، بيكنج بودر وملح، سكر) في وعاء كبير.
2. ضع جميع المكونات السائلة (الحليب والبيض والزبدة المذابة) في نفس الوعاء.
3. حرك المكونات حتى يصبح لديك خليط ناعم.
4. سخن المقلاة.
5. صب المزيج في المقلاة.
6. اطبخها حتى تصبح الفطائر بنية على كلا الجانبين.
7. قدمها مع الشراب المحبب لديك.

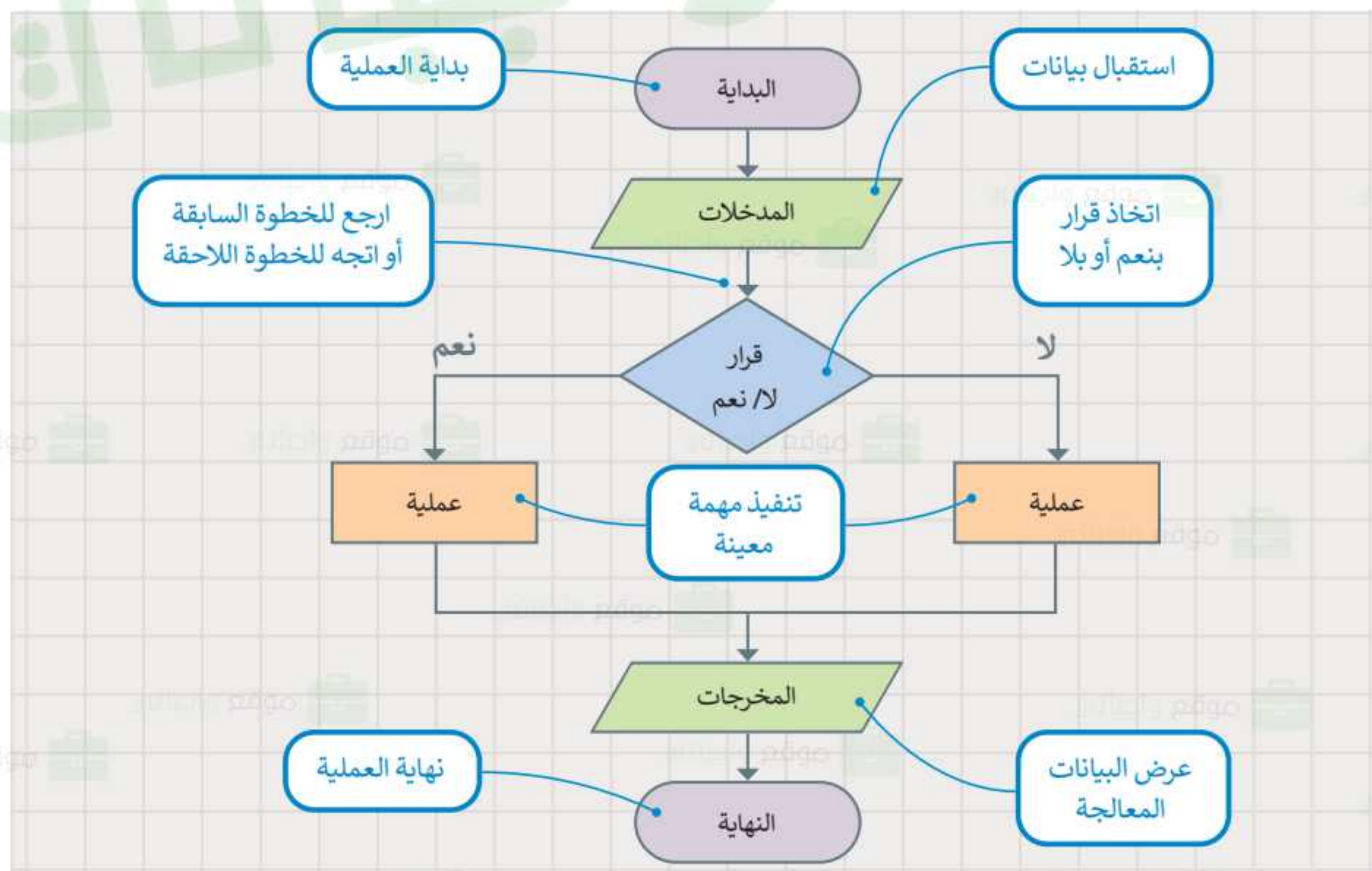


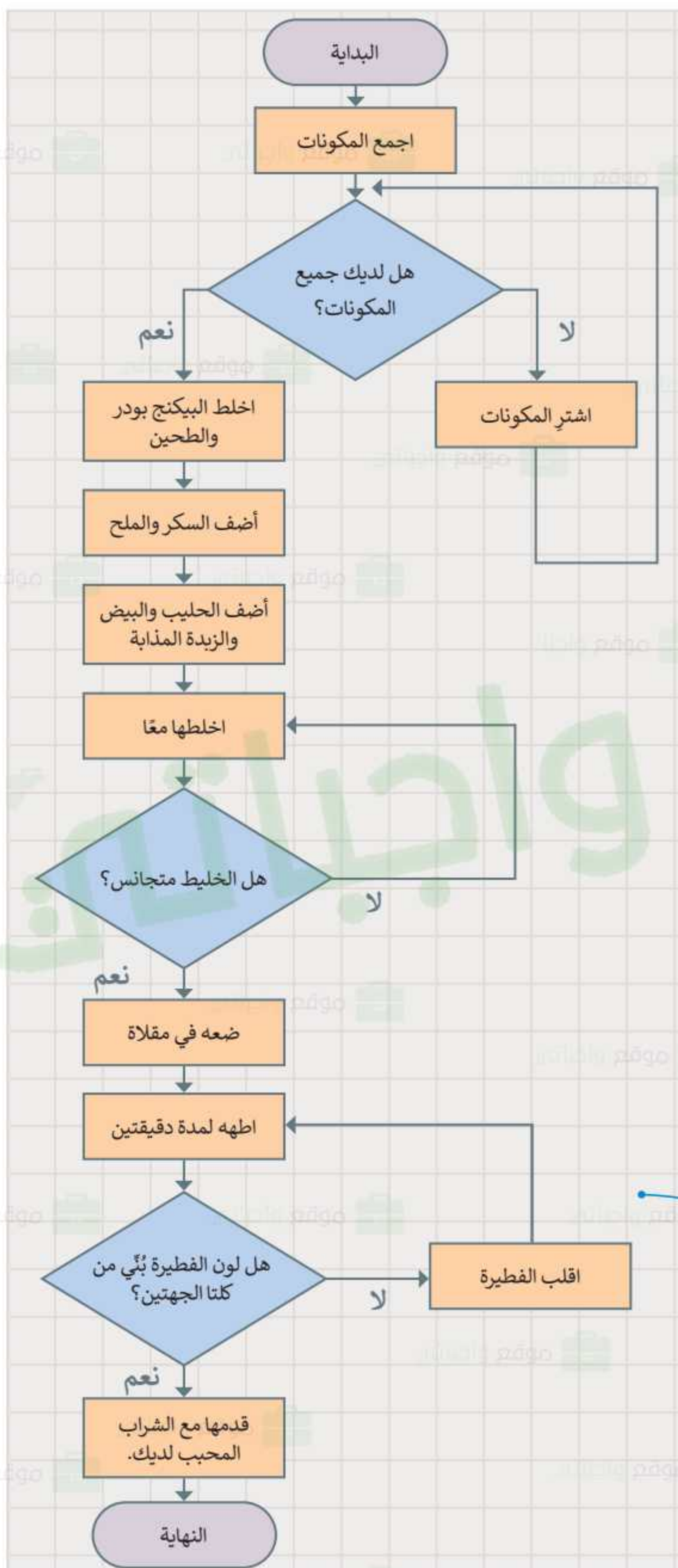
المخطط الانسيابي

المخطط الانسيابي (Flowchart) هو نوع من أنواع المخططات البيانية يستخدم لتمثيل الخوارزمية ويعرض الخطوات التي تحتاج إلى اتباعها بالترتيب الصحيح. يقدم هذا المخطط حل المشكلة خطوة بخطوة وبصورة واضحة وذلك بتقسيمها إلى مهام أصغر أو تعليمات محددة. يمكنك إنشاء مخططات انسيابية لوصف أفكارك حول كيفية حل مشكلة باستخدام جهاز الحاسب قبل كتابة المقطع البرمجي فعليًا.

يمكنك تمثيل خطوات الخوارزمية برسم أربعة أنواع مختلفة من الأشكال تعكس إجراءاتها المختلفة ثم ربط الأشكال بالأسهم لإظهار ترتيبها.

نوع الشكل	الوصف
البداية / النهاية	الإشارة إلى بداية ونهاية المقطع البرمجي.
الإدخال / الإخراج	استقبال وعرض البيانات التي سيتم معالجتها (إدخال وإخراج).
العمليات	تنفيذ عملية رياضية.
اتخاذ قرار	اتخاذ قرارات (نعم أو لا) أو اختبارات تحقق (صواب أو خطأ).
	عرض التسلسل الذي يجب تنفيذ الخطوات به.





عند رسم مخطط انسيابي، تأكد من:

< أن يحتوي المخطط الانسيابي على نقطة بداية ونقطة نهاية.

< عدم تقاطع خطوط الأسهم التي تربط بعض الإجراءات ببعض.

< تضمين جميع الإجراءات في المخطط الانسيابي.

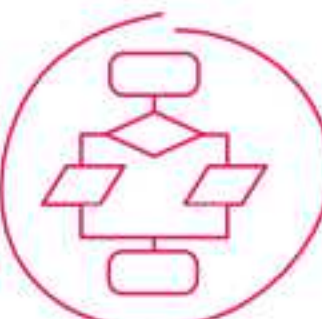
مراحل إنشاء البرنامج



أولاً: تحديد المشكلة.



ثانياً: كتابة الخطوات الخوارزمية اللازمة لحلها بترتيب متسلسل.



ثالثاً: رسم المخطط الانسيابي بناءً على الخطوات الخوارزمية.



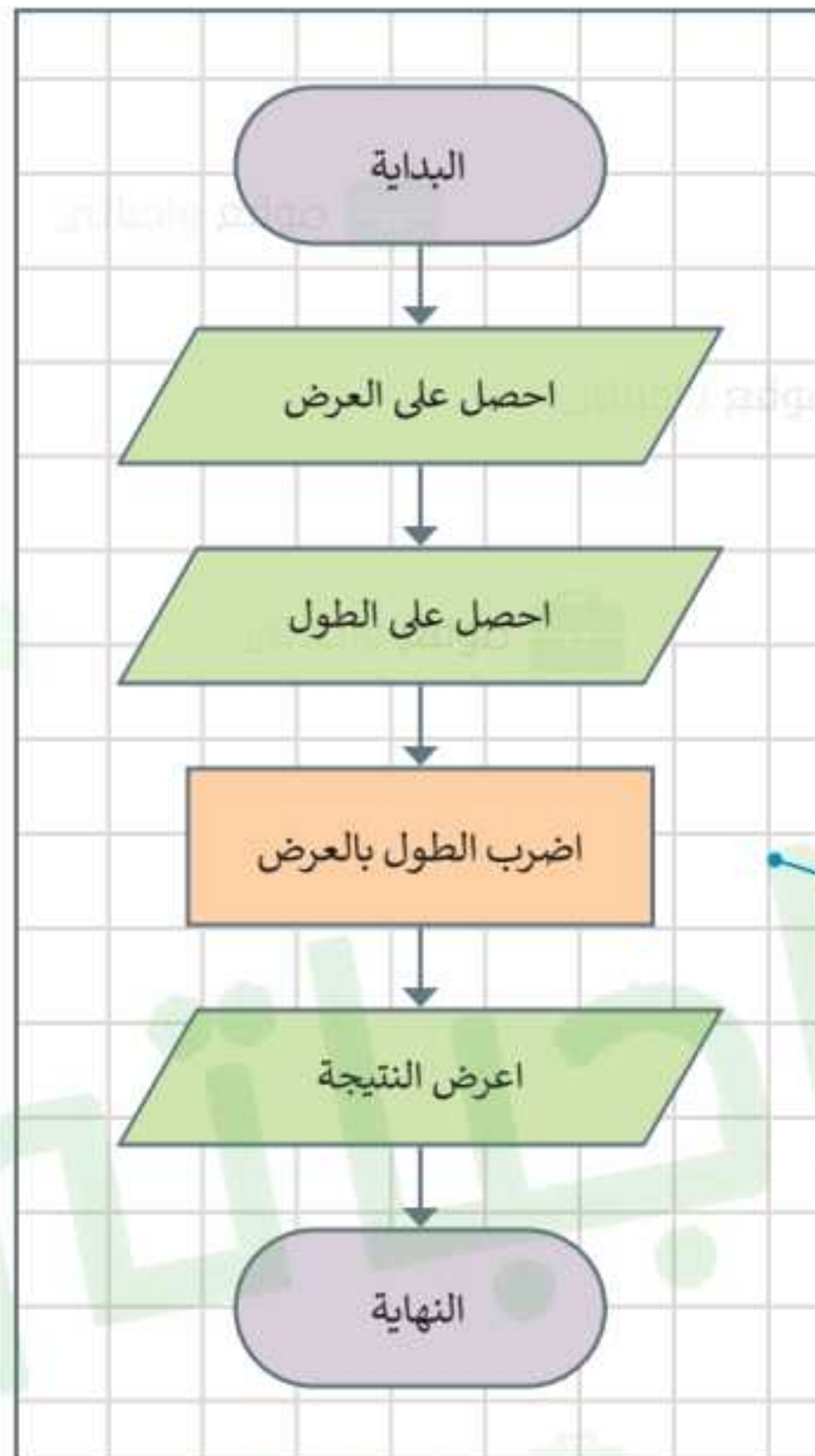
رابعاً: كتابة البرنامج بلغة البرمجة.

المخطط الانسيابي لعمل فطيرة محلاة

تعريف المشكلة

قبل البدء بتصميم البرنامج، يتعين عليك فهم المشكلة التي يجب عليك حلها لتحديد المدخلات والمخرجات وما يجب فعله لتحقيق هدفك. على سبيل المثال، افترض أنك تريد حساب مساحة المستطيل. أولاً، عليك التفكير في الخطوات اللازمة للحصول على إجابتك. ستحتاج في هذا المثال إلى معرفة بُعدَي الشكل (العرض والطول). تُستخدم المعادلة التالية لحساب المساحة:

$$\text{المساحة} = \text{الطول} \times \text{العرض} \quad (\text{Area} = \text{Width} \times \text{Length})$$



1 احصل على العرض

2 احصل على الطول

3 اضرب الطول بالعرض

4 اعرض النتيجة

خطوات الخوارزمية

المخطط الانسيابي

يُطلق على الأداة المستخدمة لتحويل تعليمات لغة البرمجة عالية المستوى إلى لغة الآلة لكي يفهمها جهاز الحاسب اسم محول برمجي (Compiler).

هيا لنبرمج

لكتابة مقطع البرمجي بلغة البايثون، يجب عليك تحويل المخطط الانسيابي إلى لغة برمجة. سيحسب المقطع البرمجي التالي مساحة المستطيل في بايثون. ستتعلم قريباً كيفية كتابة تعليماتك البرمجية بنفسك.

```
print("لتحسب مساحة المستطيل")
print("اكتب طول المستطيل:")
length=int(input())
print("اكتب عرض المستطيل:")
width=int(input())
area=length * width
print("مساحة المستطيل هي:",area)
```

التعليمات البرمجية

لنطبق معًا

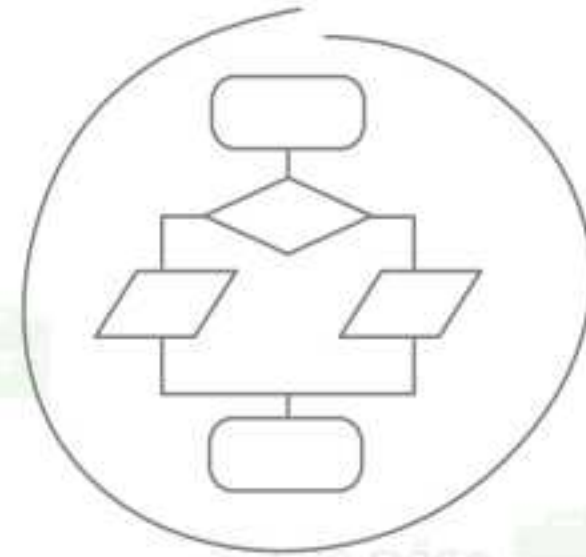
تدريب 1

طلب منك معلمك إنشاء مقطع برمجي لحساب متوسط رقمين، وزودك بالخوارزمية اللازمة ولكنها غير مرتبة، رتب الخطوات بشكل صحيح ثم ارسم المخطط الانسيابي للخوارزمية.



تدريب 2

◀ ارسم مخططًا انسيابيًا للخوارزمية.

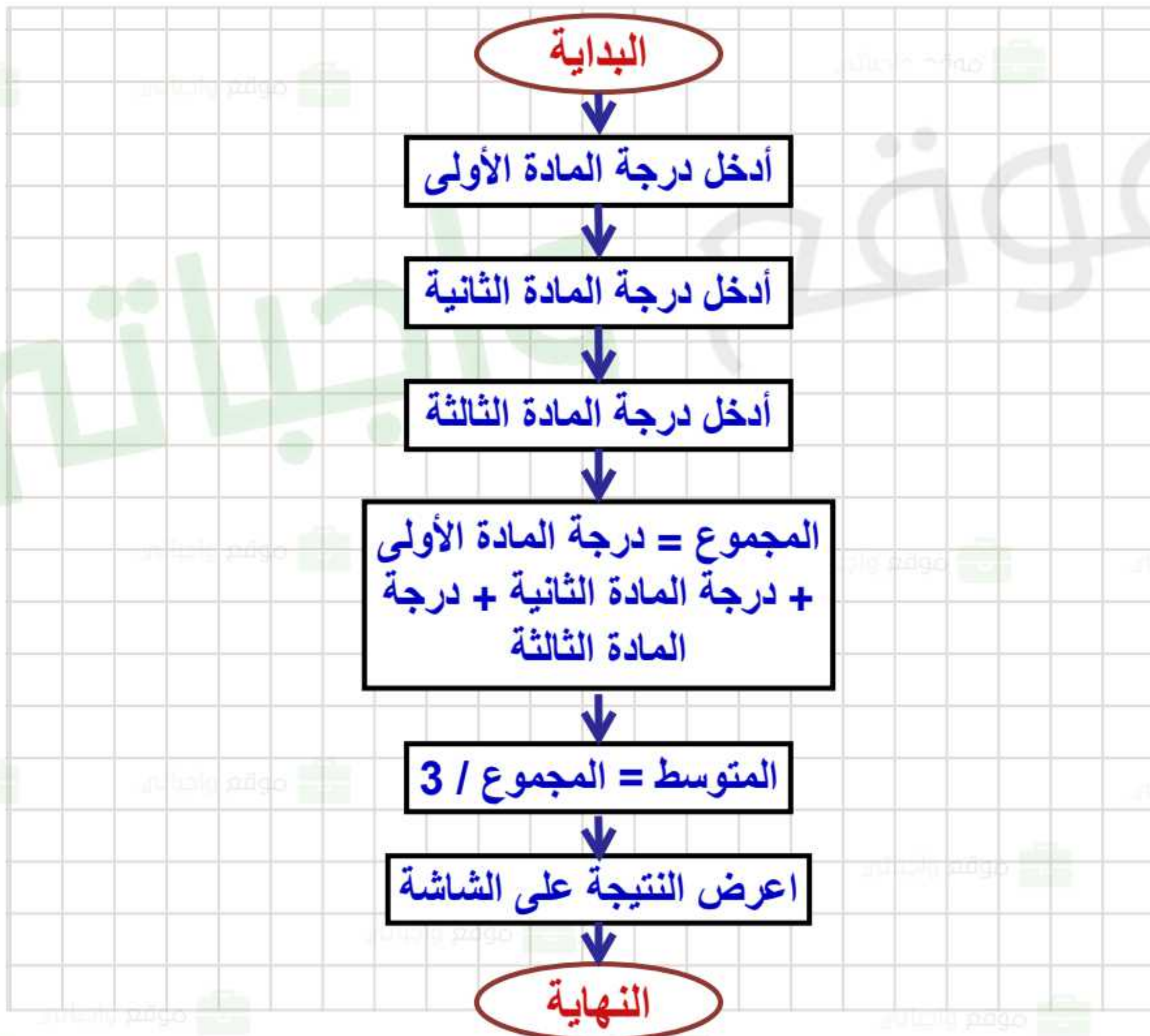


أدخل درجاتك في ثلاث مواد.

احسب المجموع.

احسب المتوسط بقسمة المجموع على 3.

اطبع النتيجة على الشاشة.



تدريب 3

أعدّ رسم المخطط الانسيابي التالي الذي يحوّل الكيلو مترات إلى أميال. بعد ترتيب الأشكال بشكل صحيح، أكتب خطوات الخوارزمية للمخطط الانسيابي.

علمًا أن الميل = 1.61 كيلو متر.

خطوات الخوارزمية

1

البداية

2

احصل على عدد الكيلومترات

3

الأميال = الكيلومترات / 1.61

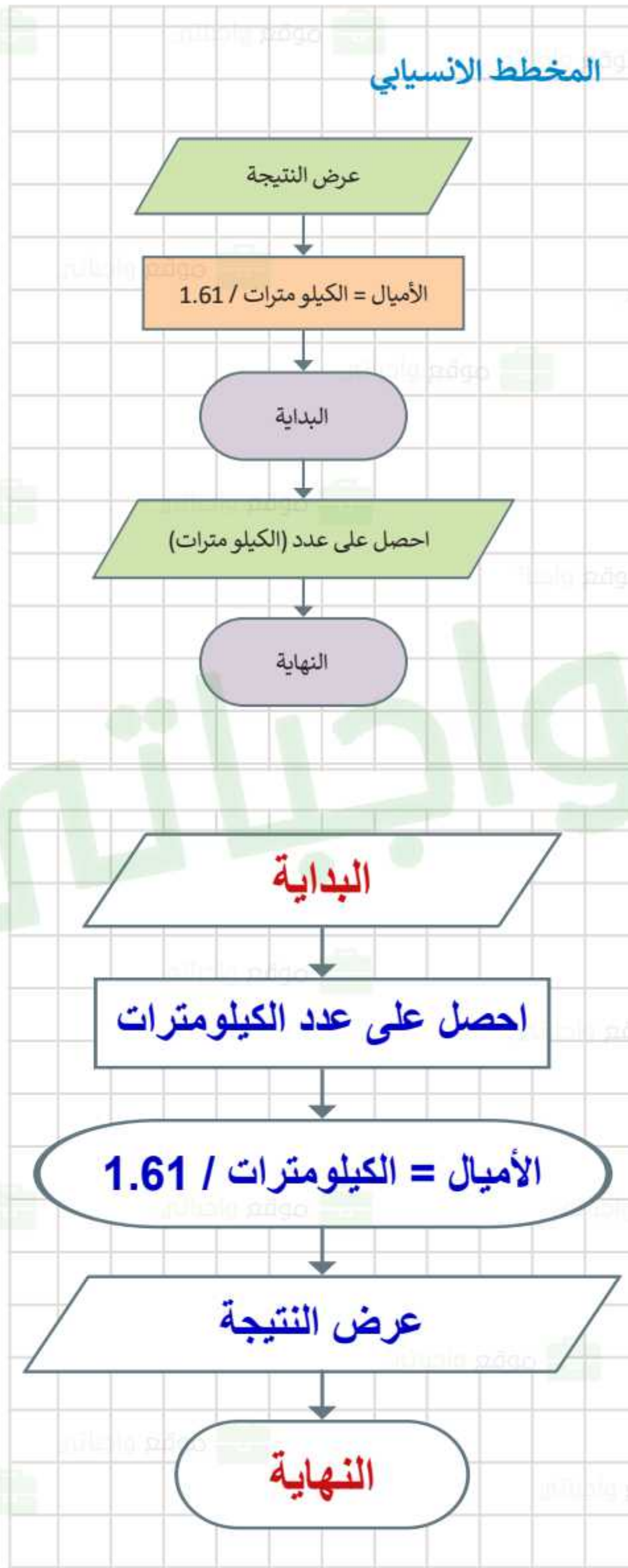
4

عرض النتيجة

5

النهاية

المخطط الانسيابي





المتغيرات والثوابت

في هذا الدرس ستستخدم بيئة التطوير المتكاملة للغة بايثون (Python IDLE) الموجودة في جهازك، لكتابة وتشغيل مقاطع البرمجة بلغة بايثون، كما ستتعلم كيفية إنشاء واستخدام المتغيرات والثوابت في مقطعك البرمجي.

فتح بيئة التطوير

لفتح بيئة التطوير المتكاملة للغة بايثون:

1 < اضغط على Start (بدء).

2 < مرر للأسفل ثم اضغط على Python3.9 (مجلد بايثون 3.9).

3 < اضغط على IDLE (Python 3.9 64-bit).



تتيح لك قائمة ملف (File) إجراء العمليات الأساسية مثل فتح البرامج أو الحفظ أو الطباعة.

تستخدم قائمة شيل (Shell) لإعادة تعيين بيئة التعليمات البرمجية.

تستخدم قائمة تصحيح (Debug) لتتبع وتشخيص الأخطاء داخل مقطعك البرمجي.



لغة البرمجة عالية المستوى تستخدم كلمات من اللغة الاعتيادية، وتحتوي أيضًا على كلمات ومصطلحات وتراكيب بناء الجملة تحتاج إلى تعلمها. على سبيل المثال: يستخدم بايثون كلمات "print" و "input" من اللغة الإنجليزية لتشكيل التعليمات. في المقابل هناك بعض العبارات والكلمات الرئيسية مثل: "elif" و "def" تستخدم فقط في لغة البرمجة.

تُعد لغات فيجوال بيسك (Visual Basic) وجافا سكريبت (JavaScript) لغات برمجة عالية المستوى.



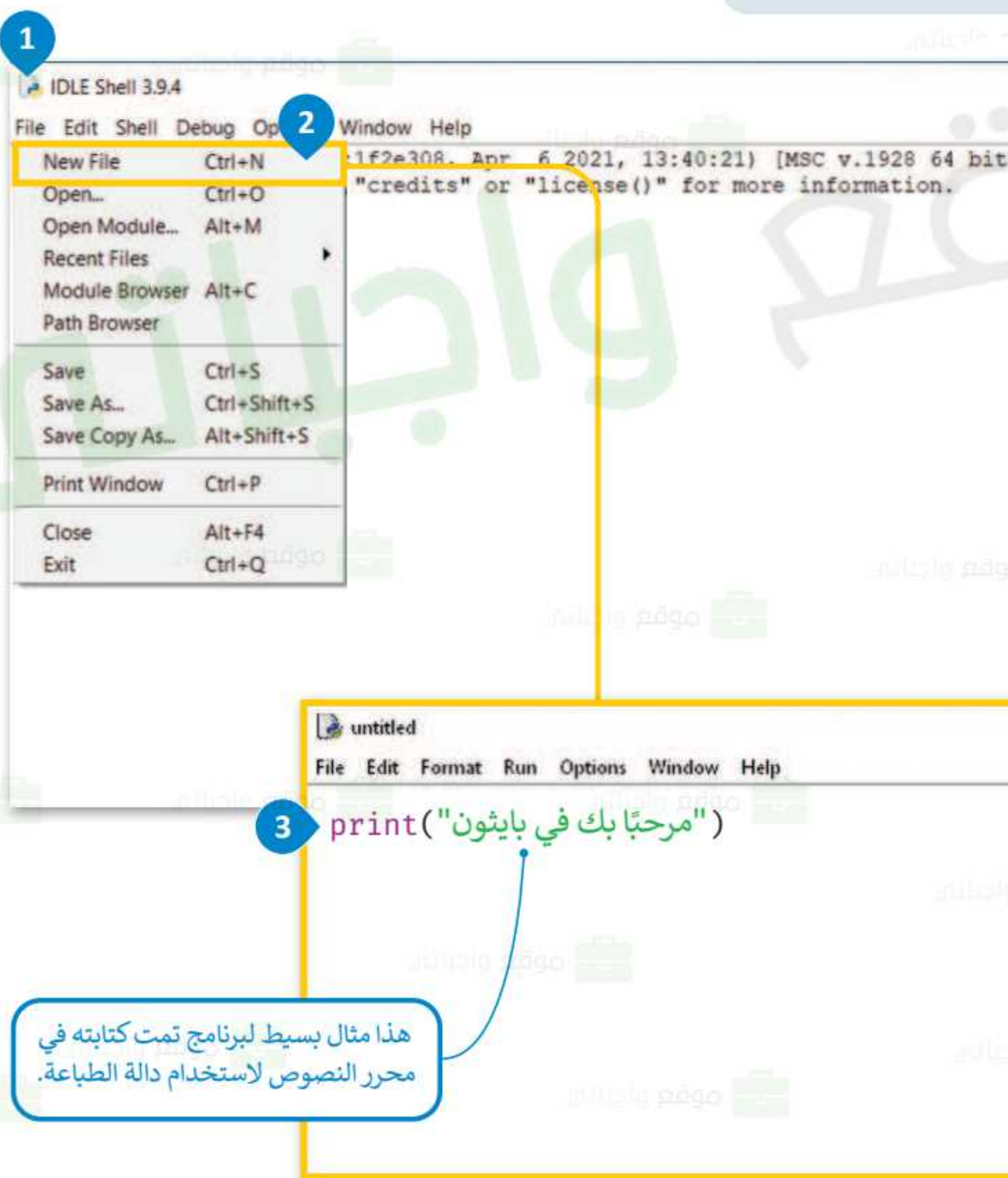
إنشاء ملف البرمجة

يمكنك كتابة التعليمات البرمجية في بايثون مباشرة من خلال بيئة التطوير المتكاملة للغة بايثون (Python IDLE)، أو استخدامها لتشغيل البرنامج فقط، ولكن في هذه الحالة لا يمكنك إجراء أي تغييرات على الأوامر التي تم تنفيذها بالفعل.

حل هذه المشكلة عن طريق إنشاء المقطع البرمجي في ملف باستخدام محرر النصوص المتوفر في IDLE، وحفظه، ثم تشغيله في بيئة IDLE. بهذه الطريقة يمكنك إجراء العديد من التغييرات التي تحتاجها في الملف والإبقاء فقط على بيئة IDLE Shell لتشغيل المقطع البرمجي فقط.

لإنشاء ملف البرمجة:

- < اضغط على قائمة **File** (ملف)، **1** ثم اضغط على
- New File** (ملف جديد) لإنشاء ملف جديد. **2**
- < سيفتح محرر النصوص.
- < اكتب مقطعك البرمجي. **3**



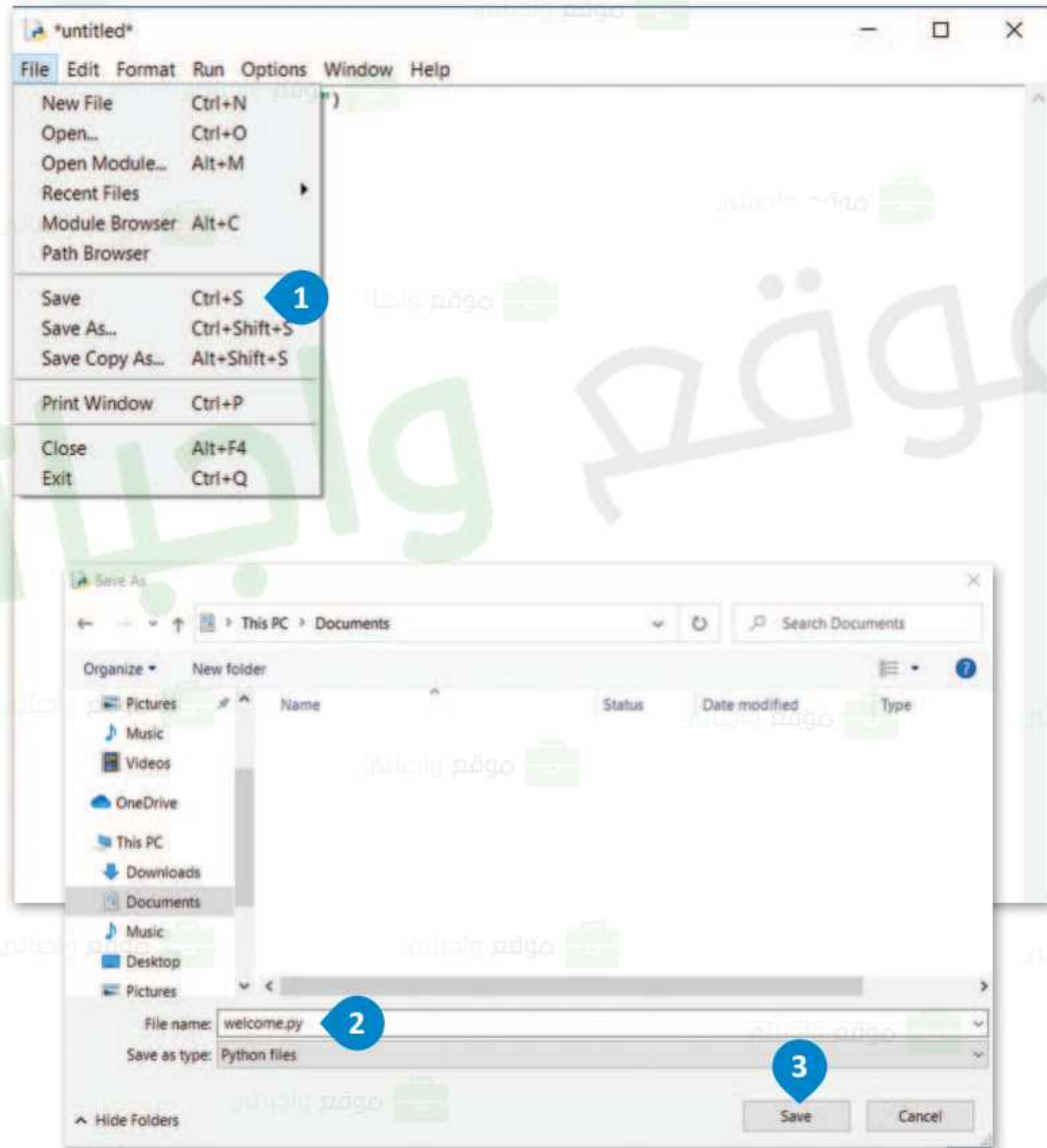
محرر النصوص

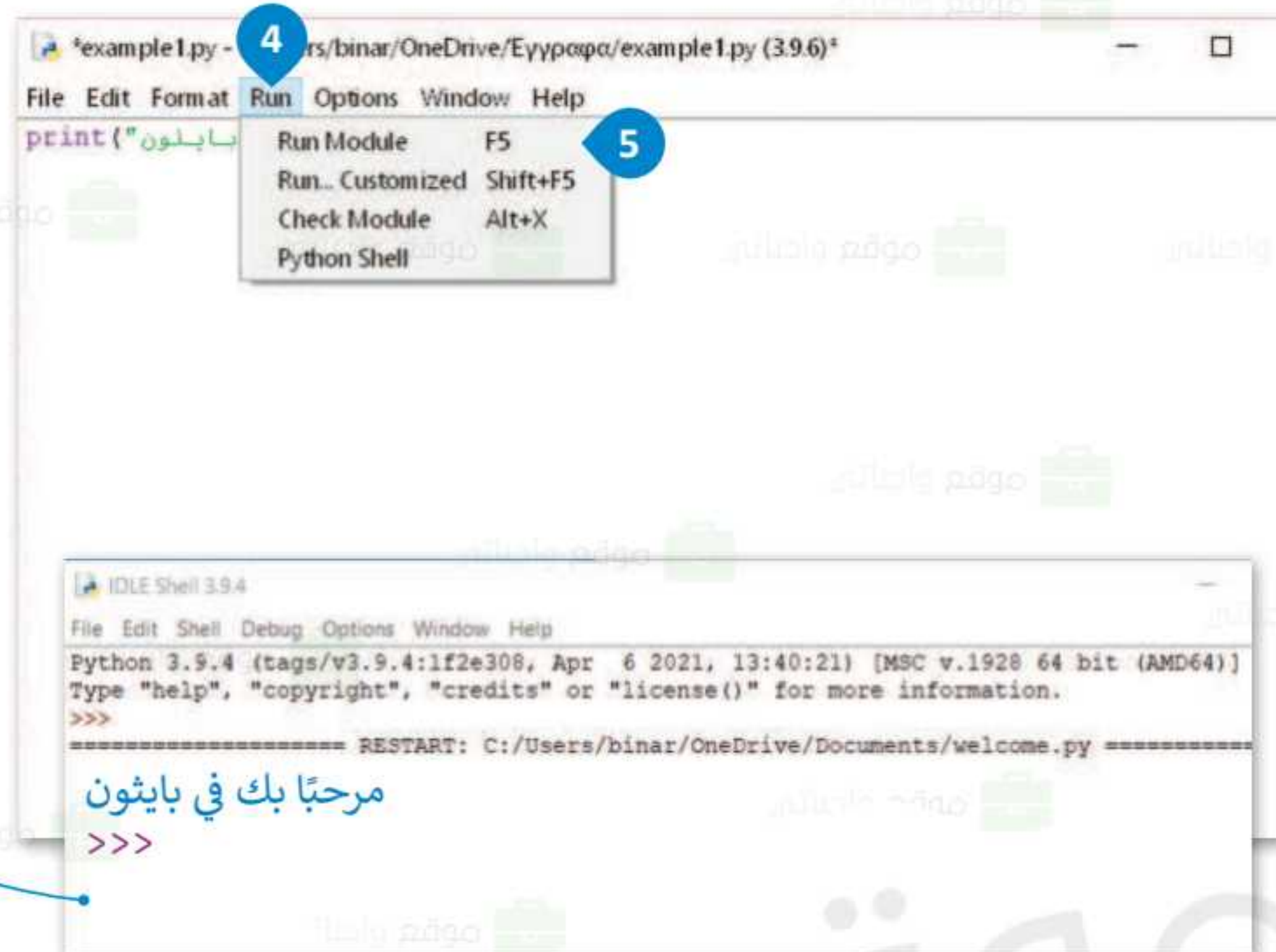
هذا مثال بسيط لبرنامج تمت كتابته في محرر النصوص لاستخدام دالة الطباعة.

الآن وبعد أن كتبت السطر الأول من التعليمات البرمجية، احفظ مقطعك البرمجي وبذلك يكون جاهزًا لتشغيله.

لحفظ المقطع البرمجي وتشغيله:

- 1 < من قائمة **File** (ملف)، اضغط على **Save** (حفظ).
- 2 < اكتب اسم الملف، ثم اضغط على **Save** (حفظ) لحفظ المقطع البرمجي.
- 3 < اضغط على **Run** (تشغيل)، ثم اضغط على **Run Module** (تشغيل الوحدة النمطية) لتشغيل مقطعك البرمجي.





بيئة التشغيل

يستخدم برنامج بايثون دوال مدمجة ودوال يُعرفها المستخدم. الدالة المدمجة يتم تحديدها بالفعل في المقطع البرمجي ويمكن للمستخدمين استخدامها مباشرة في برامجهم. تعد دوال `print()` و `input()` بعضًا من هذه الدوال التي ستستخدمها في هذا الدرس.

كما رأيت في المثال السابق، فإن أبسط أمر يمكن كتابته في بايثون أمر طباعة نص على الشاشة. الدالة التي يتم استخدامها لعرض النص على الشاشة في بايثون هي دالة الطباعة `print()`. إذا أردت كتابة نص معين، يتعين عليك وضعه داخل علامتي تنصيص.



حاول أن تطبع الرسائل التالية:

`print("أتعلم لغة بايثون")`

أتعلم لغة بايثون

`print("أحب البرمجة")`

أحب البرمجة



مثال تطبيقي: لعبة السباق الثلاثي.

ستنشئ في هذا الدرس مقطعًا برمجيًا خاصًا بلعبة السباق الثلاثي. سيسأل المقطع البرمجي المستخدم عن اسم الرياضي ونتائجه في الرياضات الثلاث، بعد ذلك ستُحسب النتيجة النهائية للرياضي وأفضل أداء له.

السباق الثلاثي هو سباق تحمل يقوم فيه الرياضيين بالمنافسة على ثلاث رياضات متواصلة (السباحة ثم ركوب الدراجة ثم الجري) ويتم حساب الوقت لكل رياضة، ثم الإجمالي لكل الرياضات.



```
print("لعبة السباق الثلاثي")
```

المتغيرات

المتغير هو مكان محجوز في ذاكرة جهاز الحاسب يُستخدم لتخزين قيمة يتم إدخالها. تمثل المتغيرات أنواعًا مختلفة من البيانات، ويمكن أن تتغير قيمة المتغير أثناء تنفيذ المقطع البرمجي. تنقسم المتغيرات إلى فئتين رئيسيتين هما الأرقام والنصوص. تدعم بايثون نوعين من الأرقام: الأعداد الصحيحة والأعداد بفاصلة عشرية (العشرية). وكما في سكراتش (Scratch) فإن المتغيرات النصية تُسمى سلسلة نصية (String).

الأعداد (المتغيرات العددية)

```
level=3  
score=1200  
TotalAmount=120.50
```

النص (المتغيرات النصية)

```
Message="هل تريد اللعب مرة أخرى؟ نعم/لا"  
MyName="محمد"  
EmailAddress="mohammad@binary-academy.com"
```

أسماء المتغيرات

قد يكون للمتغير اسم قصير مثل X أو Y، أو اسمًا وصفيًا مثل total_volume، carname، age.

الشروط الواجب توفرها في اسم المتغير:

< أن يبدأ بحرف أو بشرطة سفلية.

< ألا يبدأ برقم.

< يمكن استخدام الأحرف الإنجليزية والأرقام والشرطة السفلية (أرقام من 0-9، أحرف كبيرة من A-Z، أحرف صغيرة من a-z، _).

< حالة الحروف الإنجليزية مهمة، فمثلًا تعتبر الأسماء التالية AGE، Age، age بمثابة ثلاثة متغيرات مختلفة.



تحقق من الأسماء أدناه وحدد أسماء المتغيرات الصحيحة بوضع إشارة ○.

2AB

☐

CourseName

☒

GrAdE

☒

True

☐

تخصيص قيمة لمتغير

يمكنك استخدام علامة يساوي (=) لتعيين قيمة لأحد المتغيرات. لا تستخدم علامة يساوي (=) في البرمجة كما في الرياضيات. فعلى سبيل المثال، $x = 15$ تعني أنك تأخذ القيمة 15 كرقم وتخصصها للمتغير المسمى x.

يمكنك أيضًا حساب أي شيء على الجانب الأيمن من علامة المساواة ثم إسناد النتيجة إلى المتغير الموجود على الجانب الأيسر. شاهد المثال الآتي:

تحديد قيمة المتغير x.

تحديد قيمة المتغير y.

x=15

y=20

Total=x+y

print(Total)

إضافة متغير إلى متغير آخر. في هذا المثال يعمل (+) كمعامل رياضي وتسند النتيجة في المتغير total.

يعرض قيمة المتغير Total على الشاشة.

35

معلومة

يفضل إعطاء أسماء للمتغيرات تمثل المحتوى وذلك لفهم ما يمثله كل متغير داخل التعليمات البرمجية ولكن، لا يمكن استخدام بعض الأسماء لكونها كلمات خاصة تستعمل بالفعل في لغة البرمجة، ويُطلق عليها الكلمات المحجوزة مثل:

print return while else global break and not import True False None

المتغيرات النصية

كما ذكر سابقًا؛ لا يقتصر استخدام المتغيرات على تخزين الأرقام فقط، بل يمكن استخدامها لتخزين النصوص أيضًا. المتغيرات التي تخزن النص تسمى متغيرات من نوع String. ولتحويل النص إلى متغير معين كل ما عليك فعله هو وضع النص داخل علامتي تنصيص "".

```
name="محمد"  
print (name)
```

محمد

يجب استخدام الفواصل إذا أردت إضافة المزيد من العبارات وإذا كنت ترغب في جعل المعلومات التي تعرضها واضحة وقابلة للاستخدام بشكل أكبر، فأنت بحاجة إلى تغيير التعليمات البرمجية كالتالي:

```
name="محمد"  
print ("اسمي:", name)
```

هل يمكنك ملاحظة الاختلاف؟
هكذا أفضل بلا شك.

اسمي: محمد

التعليقات

تُستخدم التعليقات لإضافة تلميحات حول التعليمات البرمجية ولأُعد من خطوات المقطع البرمجي. قد تحتاج إلى مراجعة التعليقات في المستقبل إذا رغبت بإجراء أي تغييرات على برنامجك. يمكنك إضافة ما تريده من تعليقات باستخدام علامة (#) في بداية العبارة، وسيجاهله جهاز الحاسب.

#....

```
# عيّن قيمة لاسم المتغير  
name="محمد"  
# أطبع قيمة المتغير  
print("اسمي:", name)
```

هذه عبارة عن تعليقات
لا ينفذها جهاز الحاسب

اسمي: محمد

معلومة

يجب عليك كتابة النص دائمًا بين علامتي تنصيص "" عند استخدام المتغيرات النصية.



مثال تطبيقي: لعبة السباق الثلاثي.

لتطبيق ما تعلمته في لعبة السباق الثلاثي.

```
print("لعبة السباق الثلاثي")
SwimmingScore=70
CyclingScore=40
RunningScore=60
#احسب نتيجة الرياضي في لعبة السباق الثلاثي
TriathlonScore=SwimmingScore+CyclingScore+RunningScore
print("لعبة السباق الثلاثي")
print(TriathlonScore,"مجموع نقاط لعبة السباق الثلاثي: ")
```

لعبة السباق الثلاثي

مجموع نقاط لعبة السباق الثلاثي: 170

الثوابت

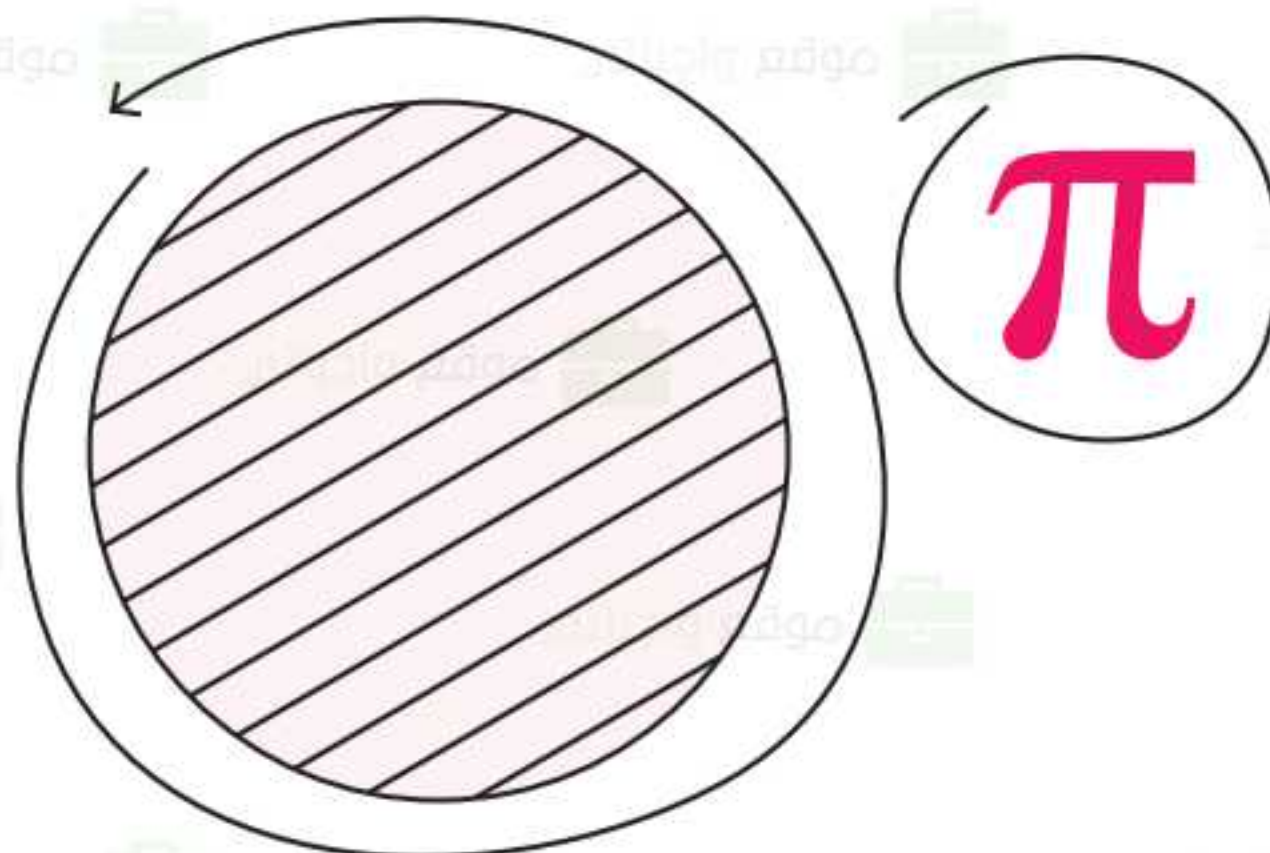
قد تحتاج أحياناً إلى متغير ذي قيمة ثابتة أثناء البرمجة، ولتحقيق ذلك يمكنك استخدام المتغيرات الثابتة. على سبيل المثال: إذا كنت تريد حساب مساحة الدائرة فإن برنامجك يحتاج إلى استخدام قيمة باي والتي تساوي 3.14، وبالنظر إلى أن القيمة لا تتغير أبداً فيمكن تخزينها بصورة ثابتة باسم باي (pi).

عادة ما يتم تخزين الثوابت في ملف مختلف عن المقطع البرمجي الرئيس ويجب عليك استيرادها لاستخدامها.

```
from math import pi
print(pi,"قيمة باي: ")
```

استخدم
السطر للوصول
إلى الدوال
الرياضية.

قيمة باي: 3.141592653589793



لنطبق معًا

تدريب 1

اختر الإجابة الصحيحة:

`input ("اكتب رقمين")`



`print ("اكتب رقمين")`



اكتب رقمين



`print (Num1, Num2)`



`print ("الرقم 2 , الرقم 1")`



`print (N1, N2)`



1. الأمر الذي تريد كتابته لعرض رسالة على الشاشة تطلب من المستخدم كتابة رقمين:

2. الأمر الذي يجب عليك كتابته لعرض قيم المتغيرات Num1 و Num2:

تدريب 2

اكتب مقطعًا برمجيًا لإضافة درجتك في الرياضيات في المتغير myGrade، ثم اطبع درجتك على الشاشة.

المقطع البرمجي

```
# عين قيمة لاسم المتغير
myGrade=30
# اطبع قيمة المتغير
print("درجتي في الرياضيات:", myGrade)
```

تدريب 3

اكتب ناتج المقطع البرمجي التالي:

```
x=20
y=30
Total=x+y
print(Total, "مجموع x و y يساوي:")
```

الناتج

مجموع x و y يساوي: 50



إدخال البيانات

تعلمت سابقاً تعيين قيم للمتغيرات في المقطع البرمجي، ولكن أحياناً كمبرمج قد ترغب بالحصول على قيم المتغيرات من مستخدم المقطع البرمجي. في لغة بايثون، توجد دالة الإدخال (`input()`) لإدخال البيانات. عند استخدام الدالة يتوقف المقطع البرمجي وينتظر أن يدخل المستخدم البيانات.

على سبيل المثال: يُطلب من المستخدم في المقطع البرمجي التالي إدخال قيمة للمتغير `x`، وعندما يُدخل المستخدم القيمة 10 ويضغط على **Enter** فيتم تعيين القيمة 10 للمتغير `x`.

أنت تطلب من المستخدم إدخال قيمة للمتغير `x`.

```
print("الرجاء إدخال قيمة للمتغير x:")
x=input()
print("قيمة x:",x)
```

عند البدء بكتابة السطر البرمجي في بيئة التطوير المتكاملة للغة بايثون (Python IDLE)، لا تترك مسافة. ابدأ مباشرة بكتابة السطر البرمجي بدون فراغ حتى لا يظهر خطأ عند تنفيذ البرنامج.

يُدخل المستخدم القيمة 10 ويضغط على **Enter**.

الرجاء إدخال قيمة للمتغير x:
10
قيمة x: 10

تم إسناد القيمة 10 للمتغير `x`.



اطلب من المستخدم إدخال اسمه ثم احفظه في المتغير `myName`.

```
print("الرجاء إدخال اسمك:")
myName=input( )
```

أنواع البيانات

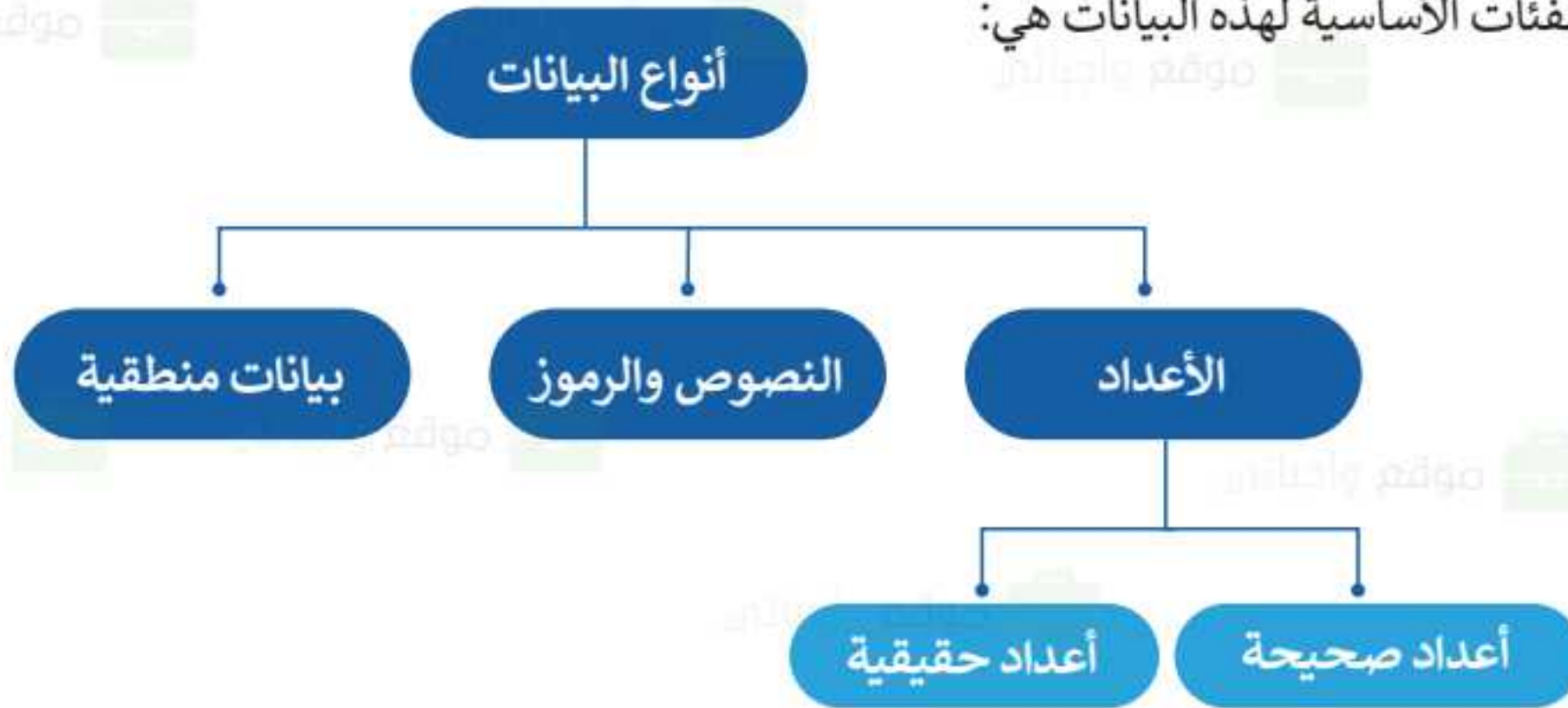
نوع البيانات هو تصنيف لأنواع مختلفة من البيانات. يحدد نوع البيانات مجموعة من القيم مع العمليات التي يمكن إجراؤها على هذه القيم.

يوفر بايثون بعض أنواع البيانات المدمجة. الفئات الأساسية لهذه البيانات هي:

< الأعداد

< النصوص والرموز

< بيانات منطقية



أمثلة على أنواع البيانات

نوع البيانات	التعريف في بايثون	مثال
الأعداد الصحيحة	integer	900000, 0-, 999, 12
الأعداد الحقيقية	float number	3.0, -90.5, 0.0003, 4.5
النصوص والرموز	string	"\$\$\$" , "hello" , "Saad"
البيانات المنطقية	boolean	True, False

إذا كنت تريد أن يكتب المستخدم أرقامًا لإجراء عمليات حسابية، فعليك استخدام الأوامر:

> `int(input())` للأعداد الصحيحة
> `float(input())` للأعداد العشرية

```

a=False
print(a)
b=True
print(b)
  
```

a و b متغيرات منطقية.

False
True

يمكنك هنا مشاهدة مثال لجمع عددين (x و y):

```
print("الرجاء إدخال قيمة x:")  
x=int(input())  
print("الرجاء إدخال قيمة y:")  
y=int(input())  
Total=x+y  
print(Total,"مجموع x و y يساوي")
```

الرجاء إدخال قيمة x:
10
الرجاء إدخال قيمة y:
5
مجموع x و y يساوي: 15

من المفيد استخدام دالة الطباعة (`print()`) مع دالة الإدخال (`input()`)؛ لمساعدة المستخدم على فهم نوع البيانات التي يجب إدخالها. بدلاً من استخدام الوظيفتين، يمكن كتابة الرسالة النصية في دالة الإدخال (`input()`).
جرب المقطع البرمجي التالي وتحقق من النتيجة.

```
x=int(input("الرجاء إدخال قيمة x:"))  
y=int(input("الرجاء إدخال قيمة y:"))  
Total=x+y  
print(Total,"مجموع x و y يساوي")
```

الرجاء إدخال قيمة x:
10
الرجاء إدخال قيمة y:
5
مجموع x و y يساوي: 15



اطلب من المستخدم إدخال اسمه ثم احفظه في المتغير `myName`، وإدخال عمره ثم احفظه في المتغير `myAge`، بعد ذلك اطبع اسم وعمر المستخدم.

```
myName=input("الرجاء إدخال اسمك:")  
myAge=int(input("الرجاء إدخال عمرك"))  
print("اسمك:",myName,"عمرك:",myAge)
```

لعبة السباق الثلاثي

يمكنك أن تطلب من المستخدم إدخال اسم الرياضي.

```
print("اكتب اسم الرياضي:")  
AthleteName=input()
```

أو يمكنك كتابتها بهذه الطريقة:

```
AthleteName=input("اكتب اسم الرياضي:")
```

أكمل المقطع البرمجي، يجب أن يسأل المقطع البرمجي عن اسم الرياضي وعن نتائج اللاعب في الرياضات الثلاث.

التعليقات تساعدك على فهم المقطع البرمجي.

```
print("لعبة السباق الثلاثي")  
# نقرأ اسم الرياضي  
print("اكتب اسم الرياضي:")  
AthleteName=input()  
# نقرأ رقم الرياضي  
AthleteNumber=int(input("اكتب رقم الرياضي:"))  
# نقرأ نتيجة الرياضي في الرياضات الثلاث  
SwimmingScore=float(input("نتيجة رياضة السباحة:"))  
CyclingScore=float(input("نتيجة رياضة ركوب الدراجات:"))  
RunningScore=float(input("نتيجة رياضة الجري:"))  
print("نتيجة رياضة السباحة:", SwimmingScore)  
print("نتيجة رياضة ركوب الدراجات:", CyclingScore)  
print("نتيجة رياضة الجري:", RunningScore)
```

السباق الثلاثي

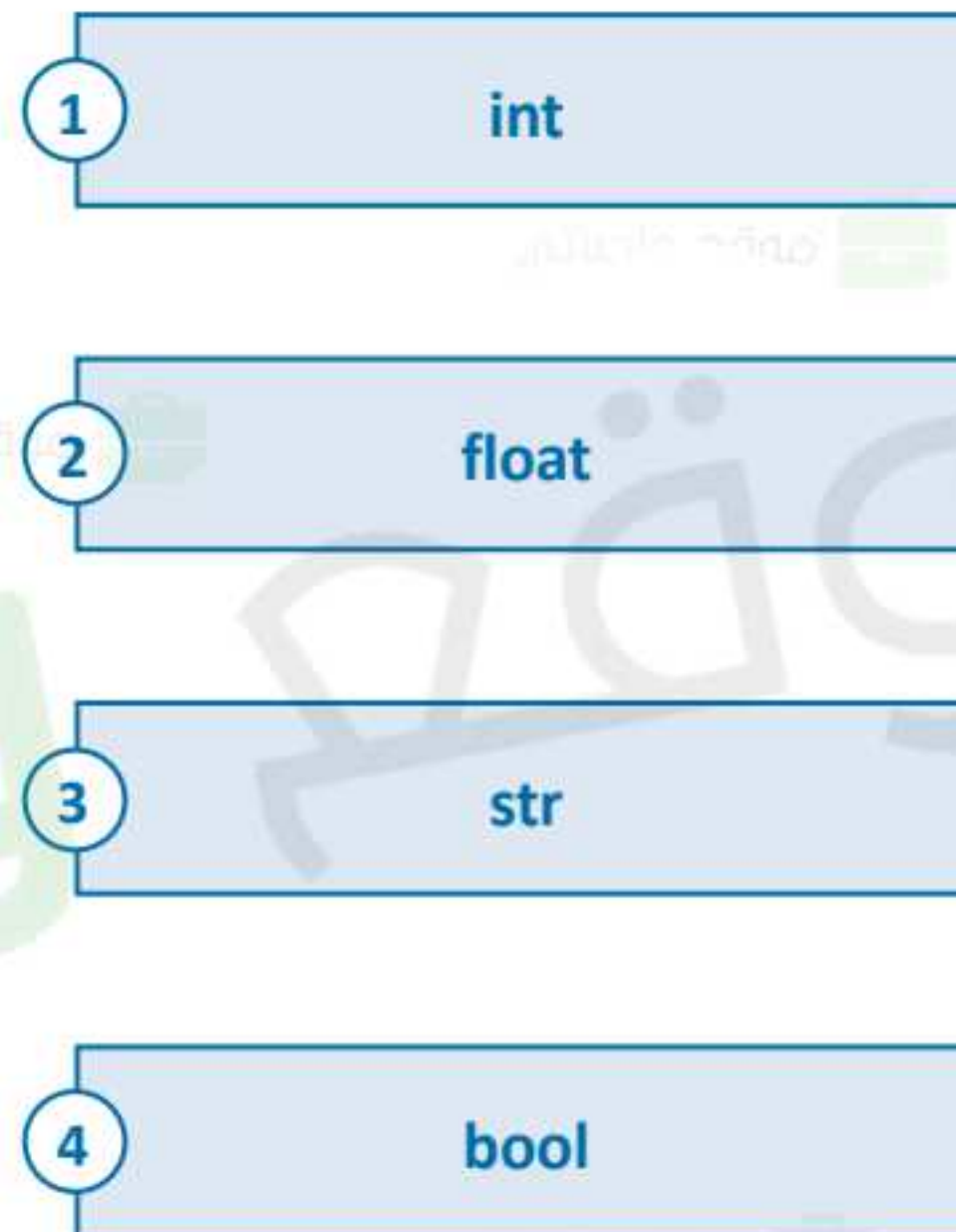


لنطبق معًا

تدريب 1

↩ صل قيمة كل متغير بنوعه.

7.23	2
True	4
36	1
-98.27	2
"Python"	3
3756	1
False	4
"Khaled"	3



تدريب 2

◀ اكتب مقطعًا برمجياً في بايثون يطلب من المستخدم إدخال اسم اليوم ودرجة الحرارة وتخزينها في المتغيرات المناسبة. سيقوم المقطع البرمجي بعد ذلك بطباعة الرسالة: "اليوم هو (يوم) ودرجة الحرارة (درجة الحرارة)".

```
day=input("الرجاء إدخال اسم اليوم:")
degree=float(input("الرجاء إدخال درجة الحرارة:"))
print("درجة الحرارة",degree,"اليوم هو",day)
```

تدريب 3

◀ املأ الفراغات في المقطع البرمجي التالي للحصول على نتائج الإخراج التالية:

```
print("الرجاء إدخال قيمة a:")
a=int(input())
print("الرجاء إدخال قيمة b:")
b=int(input())
s=a+b
print("مجموع a و b يساوي:",s)
```

النتائج

الرجاء إدخال قيمة a:

12

الرجاء إدخال قيمة b:

34

مجموع a و b يساوي:

46



المعاملات في بايثون

المعاملات في بايثون

المعامل (operator) في لغة البرمجة هو رمز يستخدم لإجراء عملية محددة على المتغيرات والقيم. يقدم بايثون فئات مختلفة من المعاملات. الفئات الأربع للمعاملات الأكثر استخدامًا في بايثون هي:

< المعاملات الرياضية.

< معاملات الإسناد.

< المعاملات الشرطية.

< المعاملات المنطقية.

المعاملات (operators) في بايثون

المعاملات الرياضية	+ - * / **	تُستخدم لإجراء العمليات الحسابية: الجمع، الطرح، الضرب، و القسمة وما إلى ذلك.
معاملات الإسناد	= += -= *= /= 	تُستخدم لإسناد قيم للمتغيرات.
المعاملات الشرطية	> < == <= >= !=	تُستخدم في مقارنة القيم أثناء كتابة الجمل الشرطية.
المعاملات المنطقية	and or not	تُستخدم لفحص أكثر من شرط في جملة شرطية واحدة، وأيضًا لفحص نقيض الشرط. وتُمكن هذه المعاملات من اتخاذ قرارات لجمل شرطية مركبة.

الآن ستتعرف على المعاملات الرياضية ومعاملات الإسناد.



المعاملات الرياضية

في بايثون، يمكنك القيام بأي نوع من الحسابات: الجمع والطرح والضرب والقسمة إلخ. تكتب الحسابات برمجياً بطريقة مختلفة عن كتابتها رياضياً. في البرمجة تُستخدم المعاملات الرياضية الرموز لتمثيل العمليات الرياضية الأساسية.

رياضياً	في بايثون
$2 + 4$	$2 + 4$
$2 - 4$	$2 - 4$
2×4	$2 * 4$
$2 \div 4$	$2 / 4$
x^2	$x ** 2$

المعامل

المعنى

+

الجمع

-

الطرح

*

الضرب

/

القسمة

**

الأس

مثال 1: الحسابات الرياضية.

يمكنك كتابة هذه المعادلة في بايثون كالتالي:

$$x = a^2 + 2ab + b^2$$

$$x = a ** 2 + 2 * a * b + b ** 2$$



جرب بنفسك

كيف تكتب التعبيرات الرياضية التالية في بايثون؟

رياضياً

في بايثون

$$ax^2 + bx + c$$

$$a * x ** 2 + b * x + c$$

$$2x - 3(x - ac^{35}/ac^5)$$

$$2 * x - 3(x - a * c^{35}/a * c^5)$$

ترتيب العمليات	
()	الأقواس
**	الأس
/*	الضرب القسمة
- +	الجمع الطرح

في بايثون، يتم تنفيذ العمليات بأولويات محددة تمامًا كما تعلمتها سابقًا في مايكروسوفت إكسل (Microsoft Excel).

فتكون الأولوية لتنفيذ ما بداخل الأقواس، ويتم حساب الضرب والقسمة قبل الجمع والطرح، وهذا يعني أن ناتج $5 * 2 + 4$ هو 14 وليس 30.

يجب عليك استخدام الأقواس إذا أردت تحديد تسلسل مختلف للعمليات الحسابية. في الجدول التالي يتم تحديد العمليات الحسابية بتنفيذ المعاملات الموجودة بنفس المستوى وذلك بالترتيب من اليسار إلى اليمين.



جرب بعض الحسابات. احسب النتيجة أولاً على الورق ثم جربها بلغة بايثون.

	على الورق	في بايثون
$2**3 + 3 * 2$	14	14
$20 * 3 - 60 * 3$	-120	-120
$(20 + 4) / 4$	6	6
$12 * 3 / 6 + 5$	11	11

لعبة السباق الثلاثي

لتجري بعض العمليات الرياضية على مقطعك البرمجي.

تريد عرض النقاط التي حصل عليها اللاعب في لعبة السباق الثلاثي على الشاشة، وبالتالي فإن متوسط النقاط هو مجموع الأرقام مقسومًا على عدد الألعاب الرياضية.

```
print("لعبة السباق الثلاثي")
# نقرأ اسم الرياضي
print("اكتب اسم الرياضي:")
AthleteName=input()
# نقرأ رقم الرياضي
AthleteNumber=int(input("اكتب رقم الرياضي:"))
# نقرأ نتيجة الرياضي في الرياضات الثلاث
SwimmingScore=float(input("نتيجة رياضة السباحة:"))
CyclingScore=float(input("نتيجة رياضة ركوب الدراجات:"))
RunningScore=float(input("نتيجة رياضة الجري:"))
# حساب متوسط النقاط في لعبة السباق الثلاثي
TriathlonPoints=(SwimmingScore+CyclingScore+RunningScore)/3
print(TriathlonPoints,"نقاطك في لعبة السباق الثلاثي",AthleteName)
```



معاملات الإسناد

تستخدم معاملات الإسناد لإسناد قيم للمتغيرات.

المعامل	مثال
	$s = x + y$
$x += y$	$x = x + y$
$x -= y$	$x = x - y$
$x *= y$	$x = x * y$
$x /= y$	$x = x / y$

المعامل

المعنى

=	إسناد القيمة
+=	جمع وإسناد القيمة
-=	طرح وإسناد القيمة
*=	ضرب وإسناد القيمة
/=	قسمة وإسناد القيمة

مثال 2: معاملات الإسناد

```
x=6
x+=3
print(x)
```

9

```
x=6
x-=3
print(x)
```

3

```
x=6
x*=3
print(x)
```

18

```
x=6
x/=3
print(x)
```

2.0

لنطبق معًا

تدريب 1

أوجد قيمة المتغير e في نهاية المقطع البرمجي باستخدام بايثون.

قيمة المتغير e هي:

```
a=15  
b=5  
c = (a + b) / 4  
d = 2 * b - a / 3  
e = (c + d) * 2
```

20

تدريب 2

اكتب التعبيرات الرياضية التالية بطريقة برمجية:

$$E = \pi \cdot R^2$$

$$E = \pi * R^{**2}$$

$$A = \frac{1}{7-x} \cdot 20$$

$$A = 1/(7-x) * 20$$

$$A = \frac{5x^3 + 7x^2 + 8}{8x - 6}$$

$$A = (5 * x^{**3} + 7 * x^{**2} + 8) / (8 * x - 6)$$

تدريب 3

اكتب مقطعًا برمجيًا في بايثون يطلب من المستخدم إدخال عدد المنتجات التي باعها المتجر في أربعة أيام. يجب تخزين عدد المنتجات المباعة كل يوم في متغير منفصل. المقطع البرمجي سيحسب ويطبّع متوسط عدد منتجات المتجر المباعة.

المقطع البرمجي

ادخال عدد المنتجات

day1=int(input("الرجاء إدخال عدد المنتجات في اليوم الأول:"))

day2=int(input("الرجاء إدخال عدد المنتجات في اليوم الثاني:"))

day3=int(input("الرجاء إدخال عدد المنتجات في اليوم الثالث:"))

day4=int(input("الرجاء إدخال عدد المنتجات في اليوم الرابع:"))

حساب متوسط عدد المنتجات المباعة

avg = (day1 + day2 + day3 + day4) / 4

طباعة نتيجة المتوسط

print ("متوسط عدد المنتجات المباعة:", avg)

تدريب 4

أنشئ مقطعًا برمجيًا في بايثون يقرأ درجاتك في ثلاث مواد، ويحسب متوسط الدرجات.

المقطع البرمجي

ادخال الدرجات

mark1=float(input("الرجاء إدخال درجتك في المادة الأولى"))

mark2=float(input("الرجاء إدخال درجتك في المادة الثانية"))

mark3=float(input("الرجاء إدخال درجتك في المادة الثالثة"))

حساب متوسط الدرجات في المواد الثلاث

avg = (mark1 + mark2 + mark3) / 3

طباعة نتيجة المتوسط

print ("متوسط الدرجات : ", avg)



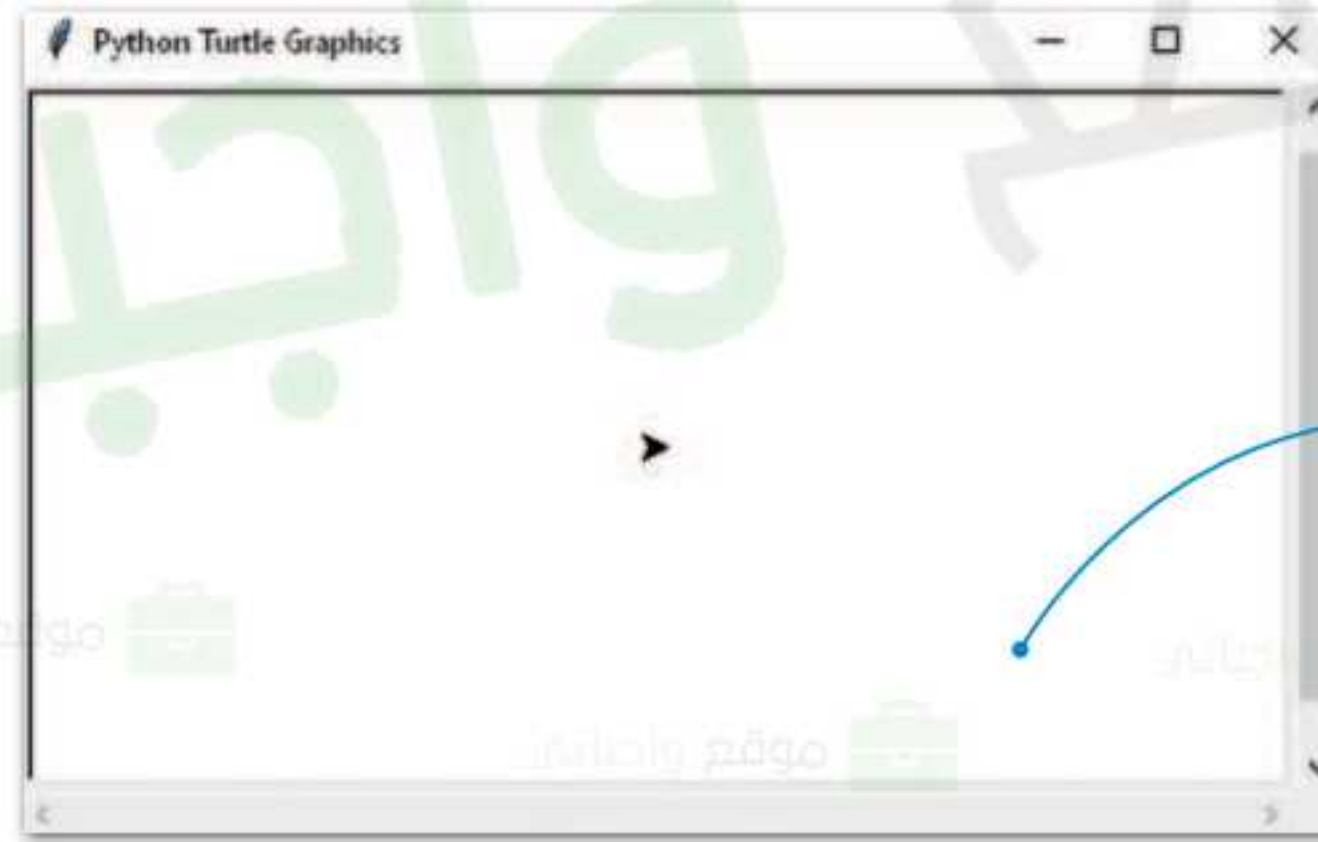
الرسم باستخدام البرمجة

إنشاء الرسومات

توجد العديد من الأشياء المثيرة للاهتمام التي يمكنك القيام بها باستخدام البرمجة. على سبيل المثال: يمكن رسم أشكال ورسومات رائعة. ويمكن برمجة سلاحف افتراضية (Virtual Turtles) بلغة بايثون بحيث تتحرك حول الشاشة. وترسم السلحفاة خطوطًا أثناء حركتها، لتصمم أشكالًا جميلة. تقدم لغة بايثون عددًا من الأوامر البرمجية الجاهزة لعمل الرسومات وذلك باستخدام ما يسمى بالنماذج البرمجية (Modules)، مثل نموذج السلحفاة (turtle)، والذي يقوم برسم الأشكال.

```
from turtle import *
miniTurtle=Turtle()
```

استيراد نموذج السلحفاة
(turtle module).



عند تشغيل المقطع الرمجي
السابق، ستعرض هذه
النافذة.

يوجد سهم في منتصف النافذة يعدّ الموضع الافتراضي. تُحدد وضعية السلحفاة باستخدام نظام إحداثيات ثنائي الأبعاد (x,y)، حيث إن الموضع الافتراضي للسلحفاة هو (0,0) وهو منتصف النافذة.



شكل السلحفاة

يمنحك البرنامج القدرة على تغيير شكل السلحفاة باستخدام دالة الشكل (`shape()`)، وتغيير لونها باستخدام دالة اللون (`color()`)، وتغيير الحجم باستخدام دالة تغيير الحجم (`shapesize()`).

أمثلة عن الأشكال التي يمكنك استخدامها:

<code>miniTurtle.shape("arrow")</code>	Arrow	سهم
<code>miniTurtle.shape("circle")</code>	Circle	دائرة
<code>miniTurtle.shape("square")</code>	Square	مربع
<code>miniTurtle.shape("triangle")</code>	Triangle	مثلث
<code>miniTurtle.shape("turtle")</code>	Turtle	سلحفاة
<code>miniTurtle.shape("classic")</code>	Classic	تقليدي

```
from turtle import*
miniTurtle=Turtle()
miniTurtle.shape("turtle")
```



يمكنك تغيير لون وحجم السلحفاة.

```
from turtle import*
miniTurtle=Turtle()
miniTurtle.shape("turtle")
miniTurtle.color("green")
miniTurtle.shapesize(2)
```

```
from turtle import*
miniTurtle=Turtle()
miniTurtle.shape("turtle")
miniTurtle.color("green")
```



الوظائف المستخدمة في الرسم

تقدم نموذج السلحفاة (Turtle Module) العديد من الدوال جاهزة الاستخدام للرسم.

الوصف	الدالة	دوال مفيدة للرسم
تحريك السلحفاة للأمام بالمقدار المحدد.	<code>forward()</code>	
تحريك السلحفاة إلى الخلف بالمقدار المحدد.	<code>backward()</code>	
تحريك السلحفاة إلى جهة اليمين.	<code>right()</code>	
تحريك السلحفاة إلى جهة اليسار.	<code>left()</code>	
تحريك السلحفاة إلى إحداثيات x و y المحددة.	<code>goto()</code>	
خفض القلم بحيث ترسم السلحفاة خطًا أثناء حركتها.	<code>pendown()</code>	
رفع القلم بحيث تتحرك السلحفاة دون رسم.	<code>penup()</code>	
التعبئة عند استدعاء الدالة <code>begin_fill()</code> والانتهاء عند استدعاء الدالة <code>end_fill()</code> .	<code>fillcolor()</code>	
إخفاء السلحفاة.	<code>hideturtle()</code>	

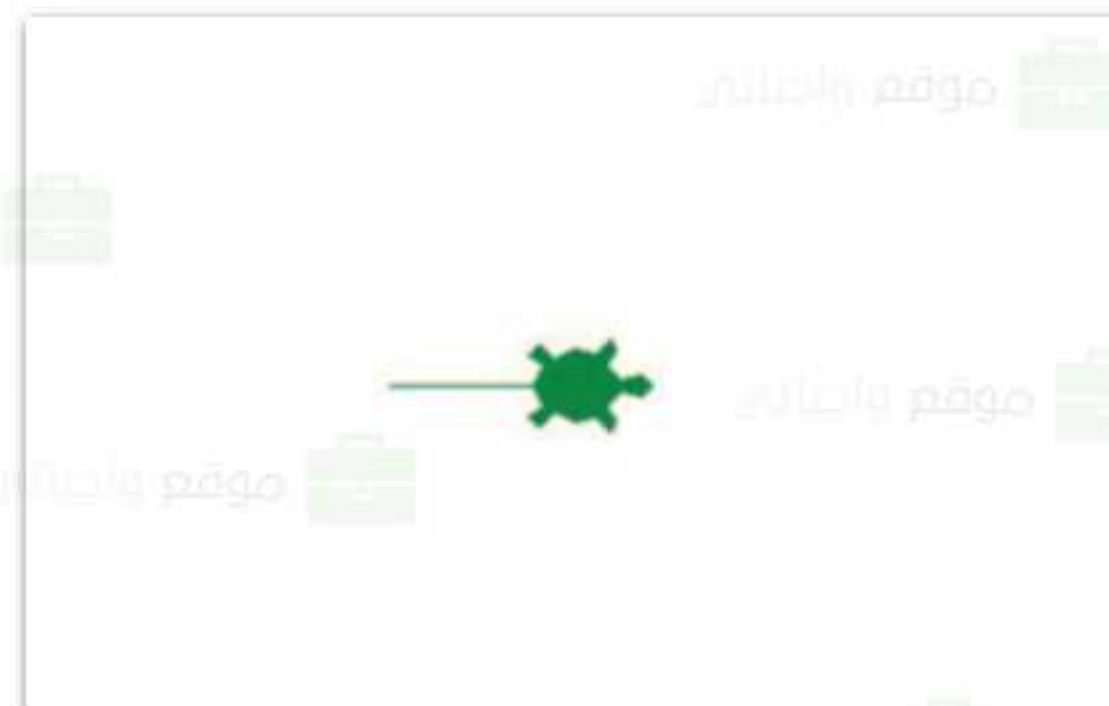
رسم خط

إن أحد أبسط المهام التي يمكنك القيام بها باستخدام نموذج السلحفاة (Turtle Module) هي رسم الخطوط والأشكال.

```
from turtle import*
miniTurtle=Turtle()
miniTurtle.shape("turtle")
miniTurtle.color("green")
miniTurtle.shapesize(2)
miniTurtle.forward(70)
```

التحرك في الاتجاه الذي تقابله السلحفاة لمسافة 70 بكسل.

البكسل هو أصغر عنصر في الصورة على شاشة جهاز الحاسب.





رسم مربع

لرسم شكل ما كمربع بواسطة السلحفاة، عليك تدوير السلحفاة. يمكنك ضبط الدرجات اللازمة لدوران السلحفاة في الاتجاه الذي تريده من خلال القيم التي تضعها في الأقواس.

```
from turtle import*
miniTurtle=Turtle()
miniTurtle.shape("turtle")
miniTurtle.color("green")
miniTurtle.shapesize(2)
miniTurtle.forward(70)
miniTurtle.left(90)
miniTurtle.forward(70)
miniTurtle.left(90)
miniTurtle.forward(70)
miniTurtle.left(90)
miniTurtle.forward(70)
```



يمكنك استخدام وظيفة `turtle.reset()` لحذف ما رسمته السلحفاة حتى الآن.



جرب بنفسك

استخدم الأوامر البرمجية الصحيحة لجعل السلحفاة ترسم مستطيلاً.

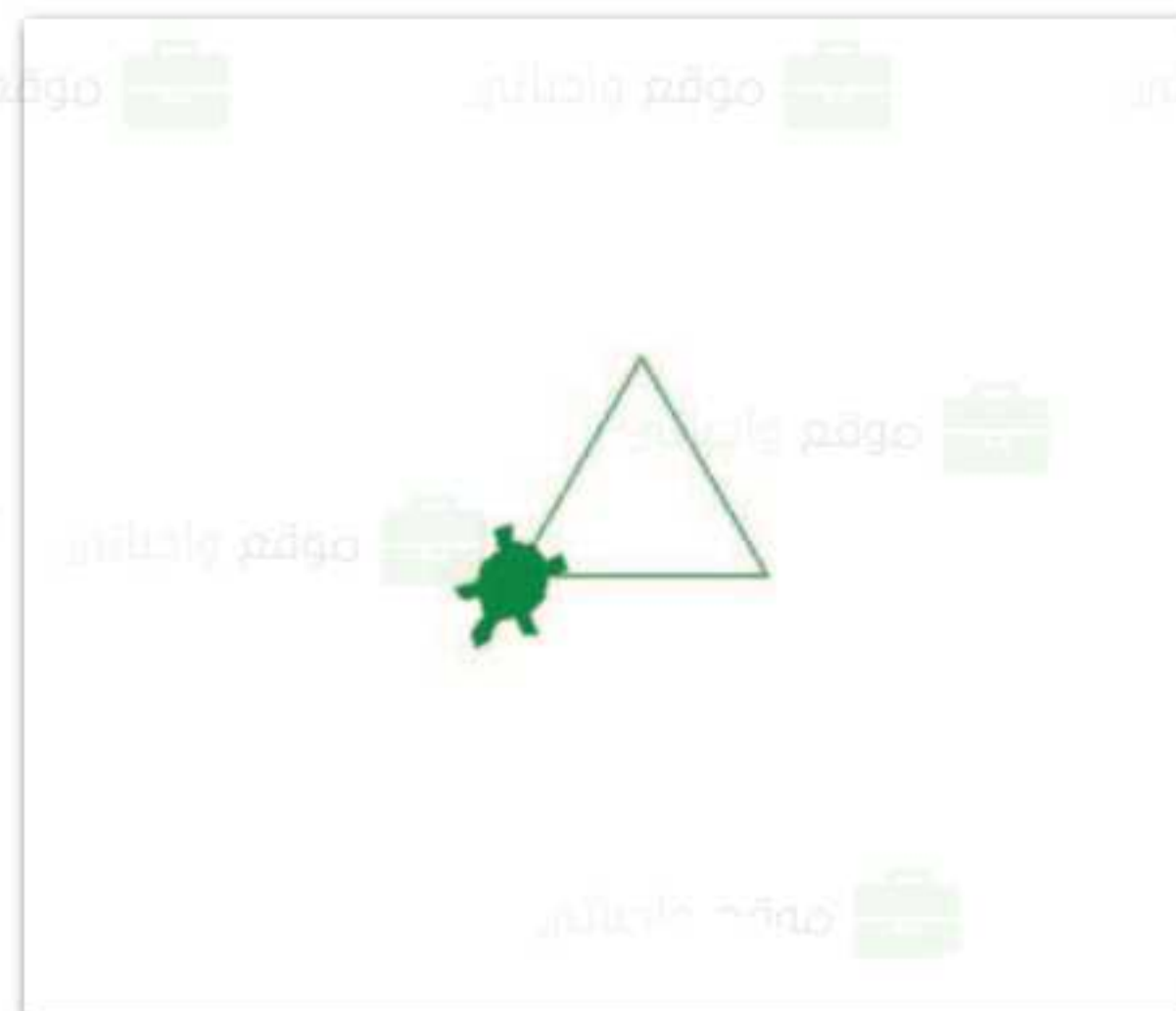
```
from turtle import *
t = Turtle( )
t.shapesize(2)
t.color("green")
t.goto(0, 0)
t.forward(60)
t.right(90)
t.forward(100)
t.left(90)
t.backward(60)
t.right(90)
t.backward(100)
```



رسم مثلث

لرسم مثلث متطابق الأضلاع تحتاج السلحفاة إلى الدوران 120 درجة.

```
from turtle import*  
miniTurtle=Turtle()  
miniTurtle.shape("turtle")  
miniTurtle.color("green")  
miniTurtle.shapesize(2)  
miniTurtle.forward(100)  
miniTurtle.left(120)  
miniTurtle.forward(100)  
miniTurtle.left(120)  
miniTurtle.forward(100)
```



رسم دائرة

لرسم دائرة تحتاج إلى تحديد نصف قطر الدائرة.

```
from turtle import*  
miniTurtle=Turtle()  
miniTurtle.shape("turtle")  
miniTurtle.color("green")  
miniTurtle.shapesize(2)  
miniTurtle.circle(50)
```



كتابة نص

تمنحك السلحفاة القدرة على الكتابة على الشاشة باستخدام دالة الكتابة `.write()`.

```
from turtle import *
miniTurtle=Turtle()
miniTurtle.shape("turtle")
miniTurtle.color("green")
miniTurtle.shapesize(2)
miniTurtle.forward(120)
miniTurtle.hideturtle()
miniTurtle.pencolor("blue")
miniTurtle.write("هذا خط", align="center", font=(12, "calibri"))
```

يمكنك تغيير لون النص باستخدام
دالة لون القلم `.pencolor()`.

يمكنك تغيير المحاذاة إلى اليسار،
أو اليمين، أو التوسيط أو تغيير نوع
خط النص.



```
from turtle import *
t = Turtle()
t.shape("turtle")
t.color("green")
t.shapesize(2)
t.goto(0, 0)
t.forward(100)
t.right(90)
t.forward(100)
t.right(90)
t.forward(100)
t.right(90)
t.forward(100)
t.penup()
t.hideturtle()
t.goto(0, -100)
t.pencolor("blue")
t.write("هذا مربع", align="left", font=(12, "calibri"))
```

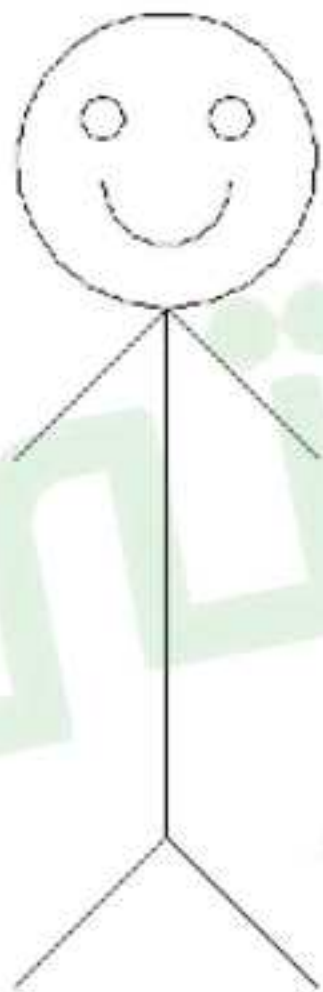
استخدم الأوامر البرمجية الصحيحة لإنشاء الشكل التالي والنص الذي بداخله.





رسم أشكال معقدة

ابدأ الآن بإنشاء رسومات أكثر تعقيدًا. ادمج بين الأشكال المختلفة لرسم شخص بسيط.



```
from turtle import *
#create the the turtle
t=Turtle()
t.shapesize(2)

# draw the head
t.color("black")
t.circle(70)
t.penup()

# draw the eyes
t.goto(-30, 80)
t.pendown()
t.color("black")
t.circle(10)
t.penup()

t.goto(30, 80)
t.pendown()
t.color("black")
t.circle(10)
t.penup()

#draw mouth
t.goto(-30, 60)
t.pendown()
t.right(90)
t.circle(30,180)
t.penup()

#draw body
t.goto(0, 0)
t.pendown()
t.right(180)
```

```
t.forward(250)
t.up
```

#draw arms

#right arm

```
t.goto(0, 0)
t.pendown()
t.right(45)
t.forward(100)
t.up
```

#left arm

```
t.goto(0, 0)
t.pendown()
t.right(270)
t.forward(100)
t.up
```

#draw feet

```
t.goto(0, 0)
t.right(45)
t.forward(250)
```

#left foot

```
t.pendown()
t.right(45)
t.forward(100)
t.up
```

#right foot

```
t.backward(100)
t.pendown()
t.right(270)
t.forward(100)
t.up
```

```
t.hideturtle()
```



لنطبق معًا

تدريب 1

صل الأوامر مع الشرح:

`miniTurtle.pensize(4)`

4

1

يحدد لون السلحفاة.

`miniTurtle.color("green")`

1

2

يخفي السلحفاة.

`miniTurtle.shapesize(2)`

3

3

يحدد حجم السلحفاة.

`miniTurtle.hideturtle()`

2

4

يحدد حجم القلم للرسم.

تدريب 2

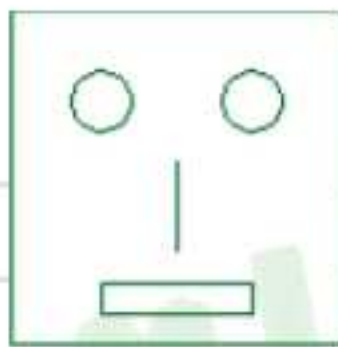
➤ ارسم أشكالاً خاصة بك عن طريق الجمع بين هذه الدوال وإضافة قيم داخل الأقواس.

```
forward()  
backward()  
left()  
right()
```

الحل في الصفحة التالية

تدريب 3

➤ ادمج بين الأشكال التي تعلمتها في الدرس لرسم الشكل التالي.



النص البرمجي

الحل في الصفحة التالية

حل تدريب 2

رسم شكل مستطيل

```
from turtle import *
t = Turtle( )
t.shapesize(2)
t.color("green")
t.goto(0, 0)
t.forward(60)
t.right(90)
t.forward(100)
t.left(90)
t.backward(60)
t.right(90)
t.backward(100)
```

رسم شكل دائرة

```
from turtle import *
t=Turtle( )
t.shape("turtle")
t.color("green")
t.shapesize(2)
t.circle(50)
```

رسم شكل مثلث

```
from turtle import *
t=Turtle( )
t.shape("turtle")
t.color("green")
t.shapesize(2)
t.forward(100)
t.left(120)
t.forward(100)
t.left(120)
t.forward(100)
```

حل تدريب 3

from turtle import *

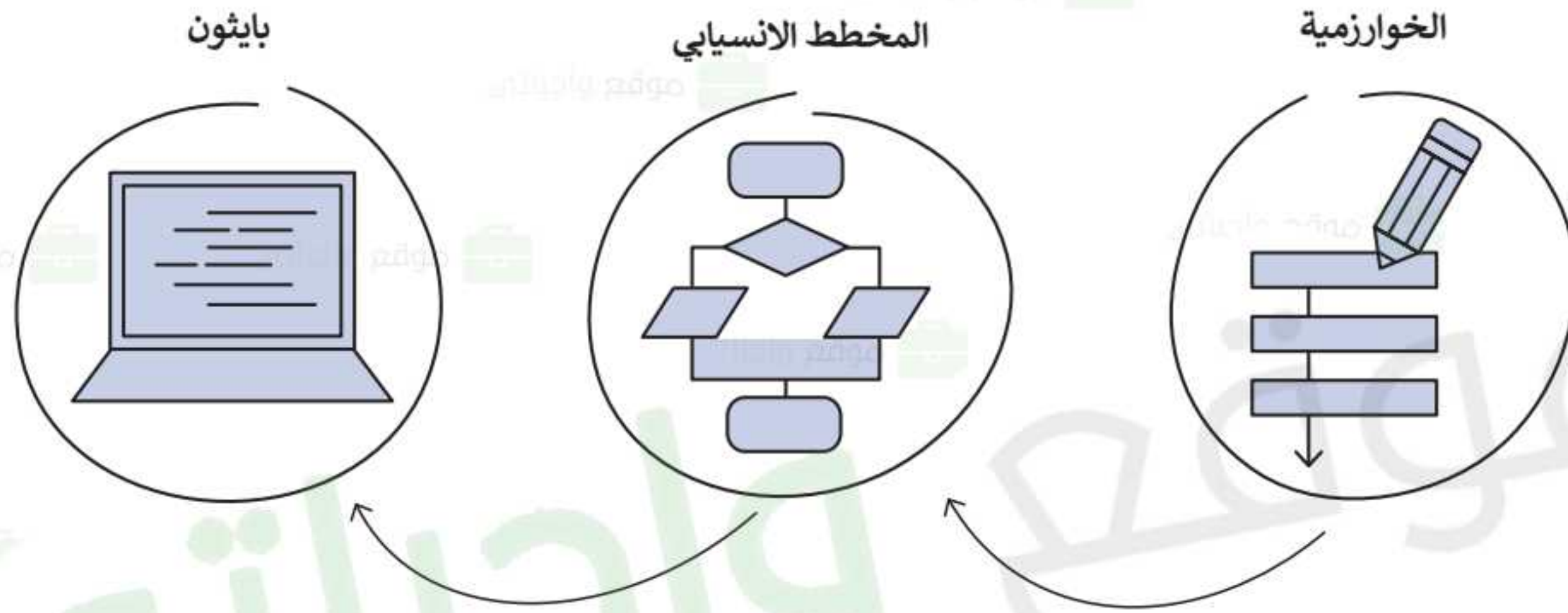
```
t = Turtle( )
t.shapesize(2)
t.color("green")
#draw the head
t.goto(0,0)
t.forward(100)
t.right(90)
t.forward(100)
t.right(90)
t.forward(100)
t.right(90)
t.forward(100)
t.penup( )
#draw the eyes
t.goto(35, -25)
t.pendown( )
t.circle(10)
t.penup( )
t.goto(80, -25)
t.pendown( )
t.circle(10)
t.penup( )
#draw the nose
t.goto(50, -45)
t.pendown( )
t.right(180)
t.forward(25)
t.penup( )
#draw the mouth
t.goto(25, -80)
t.pendown( )
t.left(90)
t.forward(50)
t.right(90)
t.forward(10)
t.right(90)
t.forward(50)
t.right(90)
t.forward(10)
t.penup( )
```



مشروع الوحدة

فكرت مع زميلك بإنشاء مشروع صغير لجمع وبيع الأجهزة الإلكترونية المستعملة وملحقاتها.
طريقة السداد: دفع 30% من المبلغ مقدماً والباقي على 6 أقساط متساوية.

صمم خوارزمية ومخطط انسيابي واكتب مقطعاً برمجياً في بايثون ينفذ ما يلي:



1 قراءة سعر الجهاز الإلكتروني وملحقاته من المستخدم.

2 حساب وعرض:
- التكلفة الإجمالية.
- المبلغ الواجب دفعه مقدماً.
- قيمة كل قسط.

3 التأكد من أن المقطع البرمجي يتبع الخطوات وفقاً للمخطط الانسيابي.

4 اختبار مقطعك البرمجي، وللقيام بذلك أجر العمليات الحسابية على الورق واستخدم نفس القيم المدخلة في المقطع البرمجي.

قارن بين النتيجةين.

في الختام

جدول المهارات

درجة الإتقان		المهارة
لم يتقن	أتقن	
		1. فهم المشكلة وتحديد مدخلاتها ومخرجاتها.
		2. كتابة خوارزمية لمشكلة محددة.
		3. رسم مخطط انسيابي لخوارزمية.
		4. التمييز بين المتغيرات في لغة البايثون.
		5. إنشاء برنامج في بايثون.
		6. التمييز بين أنواع البيانات المختلفة.
		7. إنشاء مقطع برمجي بلغة البايثون باستخدام دالتي الإدخال (input) والطباعة (print).
		8. إجراء العمليات الحسابية باستخدام المعاملات الرياضية.
		9. إسناد قيم للمتغيرات.
		10. رسم الأشكال باستخدام نموذج السلحفاة في بايثون.

المصطلحات

Function	دالة	Algorithm	خوارزمية
Modules	نماذج برمجية	Arithmetic Operator	معامل رياضي
Output	مُخْرَج	Assignment Operator	معاملات الإسناد
Problem	مشكلة	Code	مقطع برمجي
Program	برنامج	Comment	تعليق
Programming language	لغة برمجة	Constant	ثابت
Save	حفظ	Data type	نوع البيانات
Variable	متغير	Flowchart	مخطط انسيابي



اختبر نفسك

السؤال الأول

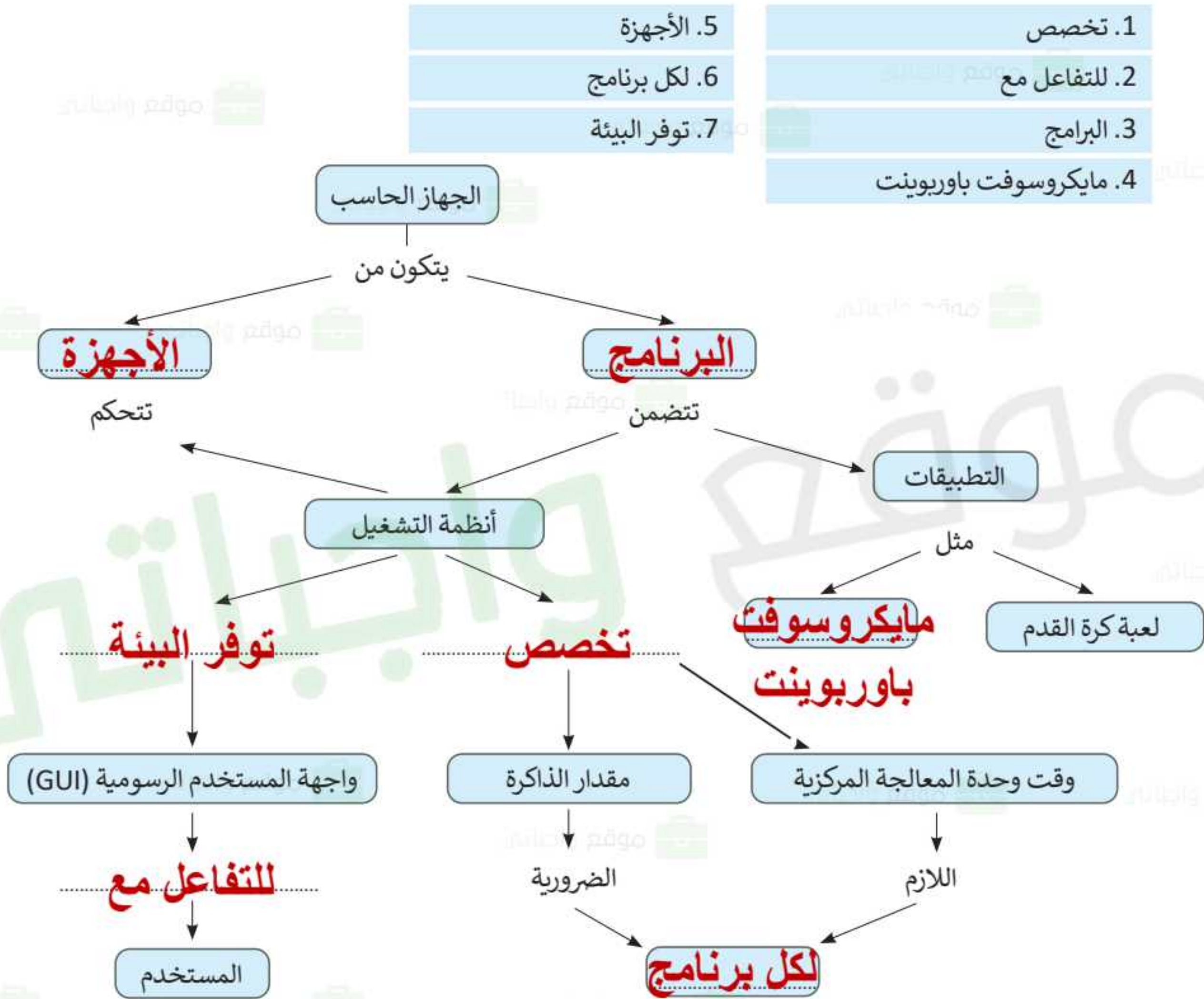
خطأ	صحيحة	حدد الجمل التالية هل هي صحيحة أم خطأ.
	✓	1. تحتوي الكثير من أجهزة غسيل الصحون والملابس الحديثة على جهاز الحاسب بداخلها.
✗		2. تقوم وحدة المعالجة المركزية بجميع العمليات الحسابية والعمليات المنطقية ولكنها لا تتحكم في عمليات البيانات المتعلقة بالذاكرة.
	✓	3. تتصل جميع الأجهزة الطرفية باللوحة الأم.
✗		4. يتم تخزين جميع الملفات مثل الصور في شاشة جهاز الحاسب.
	✓	5. تتمتع بعض بطاقات الذاكرة ووحدة الذاكرة الفلاشية بسعة أعلى من أقراص الفيديو الرقمي.
✗		6. يُعد نظام التشغيل نوعًا من أنواع المكونات المادية لجهاز الحاسب.
	✓	7. يقوم نظام التشغيل بإدارة ذاكرة جهاز الحاسب.
✗		8. لا تسمح لك واجهة المستخدم الرسومية برؤية جميع ملفاتك ومجلداتك على شكل أيقونات وصور.
	✓	9. يمكنك إغلاق الجهاز الحاسب الخاص بك حتى قبل الانتهاء من عملك.
	✓	10. يمكنك عرض محتوى بعض الملفات دون الحاجة إلى فتحها.
✗		11. لا يمكنك سحب وإفلات ملف أو مجلد من موقع إلى آخر.

السؤال الثاني

اختر الإجابة الصحيحة.		
☐	ذاكرة الوصول العشوائي وأجهزة التخزين.	1. من أنواع أجهزة الجهاز الحاسب:
☒	أجهزة الألعاب.	
☐	الأجهزة والبرامج.	
☐	مايكروسوفت ويندوز .	2. أي مما يلي لا يُعد نظام تشغيل؟
☐	ماك أو إس.	
☒	مايكروسوفت وورد.	
☒	نظام التشغيل.	3. تُعد إدارة الملفات سمة أساسية من سمات:
☐	وحدة المعالجة المركزية.	
☐	ذاكرة الوصول العشوائي.	
☐	سحب وإفلات شريط المهام في أي مكان تريده على سطح المكتب.	4. لكي تنقل شريط المهام إلى جانب آخر من سطح المكتب عليك:
☒	فتح شريط المهام ثم سحبه وإفلاته إلى أحد الجوانب الأربعة فقط لسطح المكتب.	
☐	قفل شريط المهام ثم سحبه وإفلاته إلى أحد الجوانب الأربعة فقط لسطح المكتب.	

السؤال الثالث

أكمل الخرائط المفاهيمية التالية:



السؤال الرابع

خطأ	صحيحة	حدد الجمل التالية هل هي صحيحة أم خطأ.
X		1. لا يمكنك إنشاء حد مخصص في مايكروسوفت وورد.
	✓	2. تمنحك مجموعة الخطوط جميع الأدوات اللازمة لتنسيق خط النص الخاص.
	✓	3. يمكنك حفظ مستند مايكروسوفت وورد كملف PDF.
	✓	4. يمكنك تطبيق مسافة بادئة معلقة عن طريق الضغط على خيار إعدادات الفقرة.
	✓	5. دمج المراسلات الإلكترونية هو إحدى عمليات معالجة النصوص التي يمكنك من دمج مستند مع ملف قاعدة البيانات.
	✓	6. المسافة البادئة المعلقة تنشئ مسافة بادئة لنص الفقرة بالكامل باستثناء السطر الأول.
	✓	7. لا يمكنك إنشاء حدود وتظليل مُخصصين في المستند.
X		8. في الخطوة رقم 3 من دمج المراسلات، يمكنك إضافة حقول لبيانات المستلمين.
	✓	9. يمكنك إرسال رسائل متعددة باستخدام دمج المراسلات.
	✓	10. بعد إنشاء قائمة عناوين في دمج المراسلات، عليك حفظها يدويًا لاستخدامها مرة أخرى.

السؤال الخامس

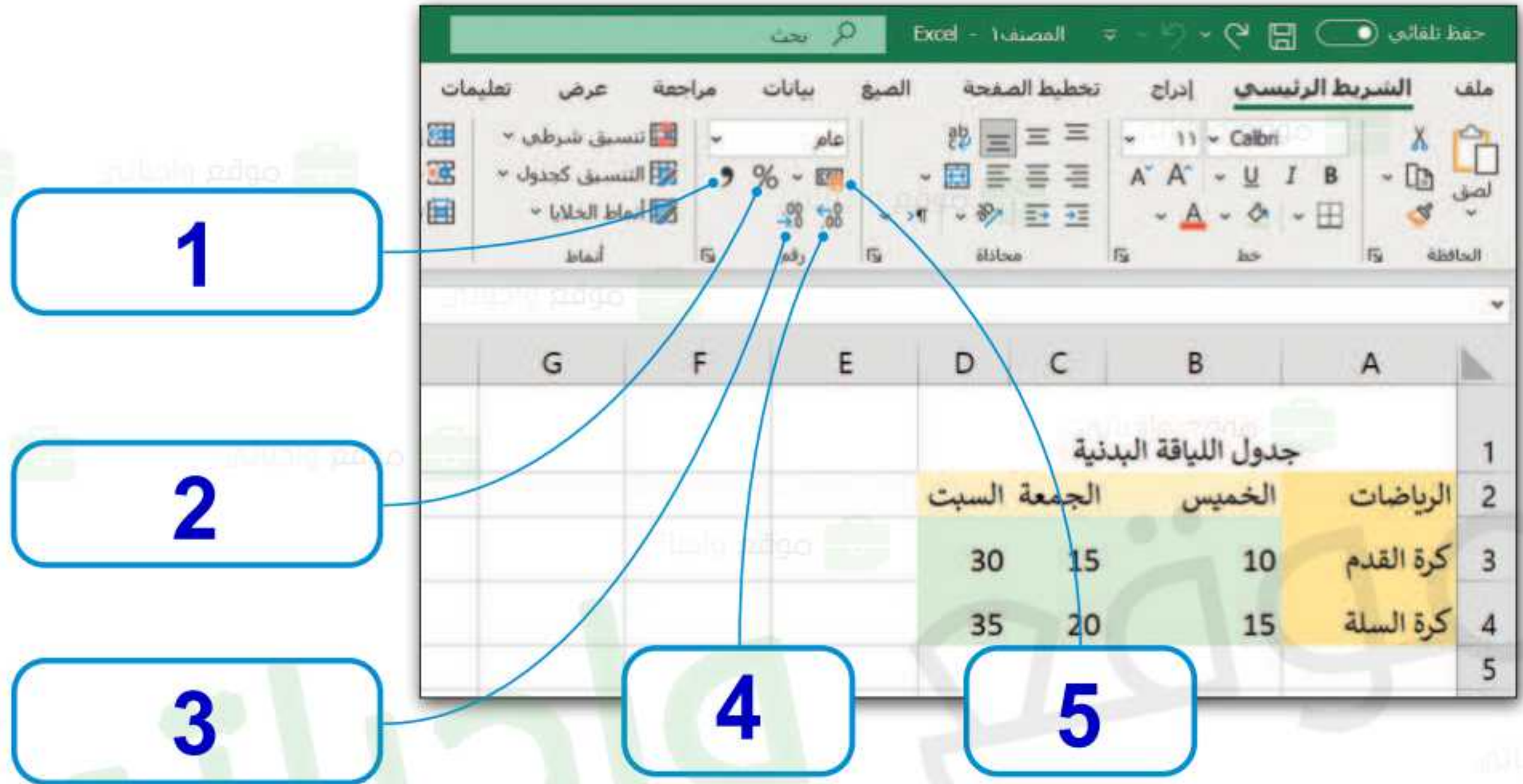
اختر الإجابة الصحيحة.		
☐	إرسال مجموعة من الرسائل لمستلم واحد.	1. أهم ميزة في خاصية دمج المراسلات هي:
☐	إنشاء مجموعة من الرسائل المختلفة.	
☒	إرسال رسالة لمجموعة من المستلمين.	
☐	كتابة رسالتك.	2. الخطوة الثالثة عند تشغيل معالج دمج المراسلات هي:
☐	تحديد مستند البداية.	
☒	تحديد المستلمين.	
☒	الأشخاص الذين تريد إرسال رسالة لهم.	3. تعد قائمة المستلمين قائمة خاصة بـ:
☐	الرسائل التي سيستقبلها الأشخاص.	
☐	الأشخاص الذين يكتبون المستند المدمج.	

السؤال السادس

خطأ	صحيحة	حدد الجملة الصحيحة والجملة الخطأ فيما يلي:
	✓	1. يمكنك في إكسل تطبيق تنسيق العملة على أي قيمة تمثل قيمة المال.
	✓	2. يقوم تنسيق المحاسبة (Accounting) بمحاذاة الأرقام بعلاماتها العشرية.
✗		3. عند تطبيق تنسيق التاريخ في إكسل، لا يمكنك تطبيق التقويم الهجري.
✗		4. لا يمكنك تنسيق أو تعديل الرموز التي أدخلتها.
✗		5. إذا كانت ورقة العمل المراد حذفها لا تحتوي على بيانات، عندها ستظهر دائماً نافذة تحذير.
	✓	6. تُرجع الدالة COUNT عدد الخلايا التي تحتوي على بيانات رقمية في جدول البيانات.
	✓	7. تعدّ LEN و CONCAT دالتان نصيتان.
	✓	8. لا تأخذ الدالتان TODAY و NOW أي وسيطات.
✗		9. ترجع الدالة TODAY الوقت الحالي جنباً إلى جنب مع التاريخ الحالي.
✗		10. بعد تطبيق الدالة NOW لا يمكنك تغيير تنسيقها.
✗		11. عند استخدام الدالة CONCAT لا يمكنك إضافة مسافة بين النص المدمج.
	✓	12. تُعدّ المسافات الفارغة ضمن عدد الأحرف التي تحسبها الدالة LEN.

السؤال السابع

املا الفراغات بأرقام الجمل الصحيحة أدناه، فيما يتعلق بالتنسيقات التي يمكنك تطبيقها باستخدام هذه الأزرار.



1. إضافة فاصلة للأرقام بالآلاف، ونقطة للأرقام بالعشرات في محتوى الخلية.

2. تحويل محتوى الخلية إلى نسبة مئوية.

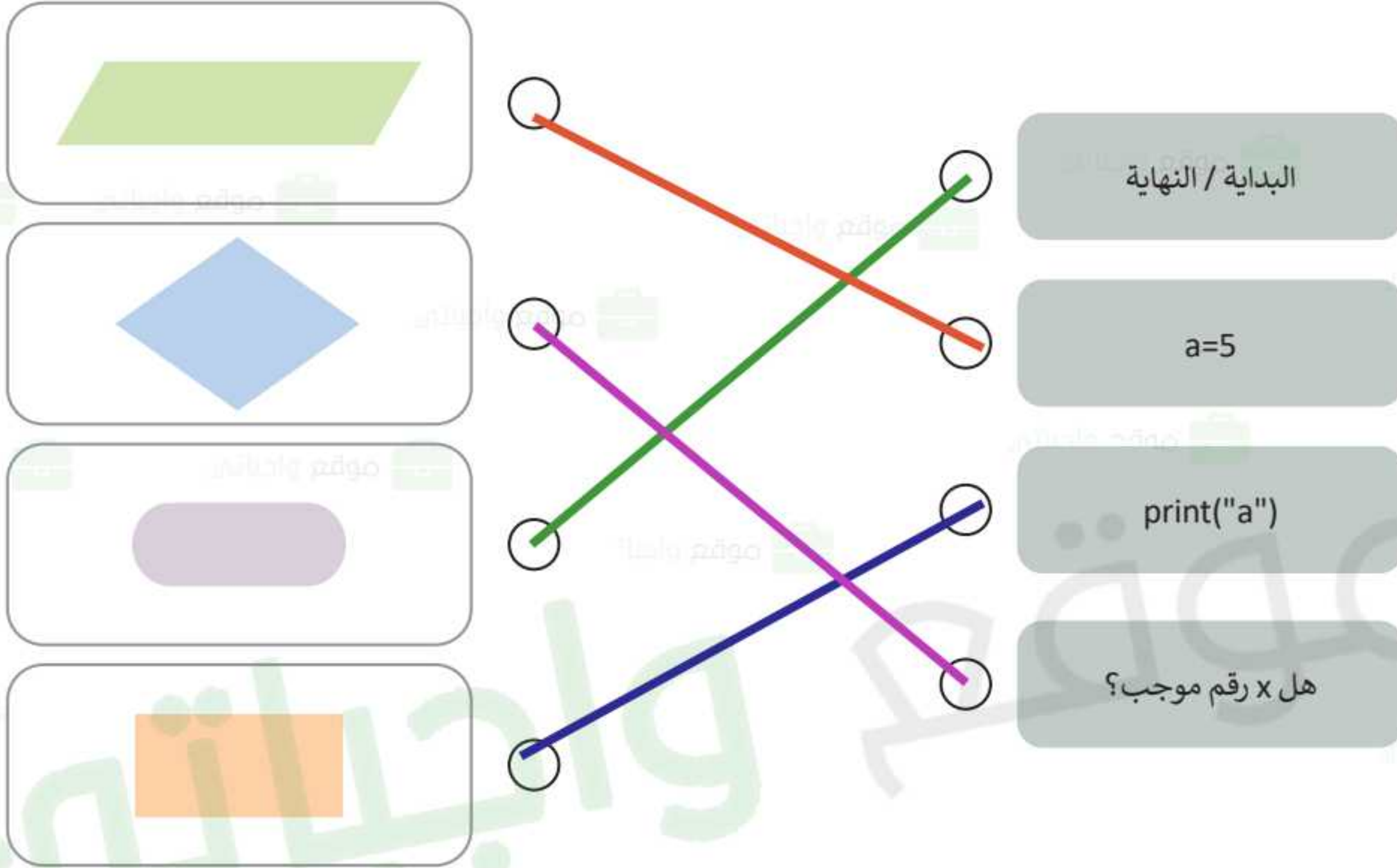
3. إنقاص عدد المنازل العشرية.

4. زيادة عدد المنازل العشرية.

5. تطبيق تنسيق العملة على خلية.

السؤال الثامن

صل بين أشكال المخطط الانسيابي وأسمائها.



السؤال التاسع

رتب مراحل إنشاء برنامج.

- 3
 - 1
 - 4
 - 2
- رسم المخطط الانسيابي الذي يوضح التسلسل المنطقي للخوارزمية.
 - تحديد المشكلة وكتابة الخطوات اللازمة لحلها.
 - كتابة المقطع البرمجي بلغة بايثون.
 - وضع الخطوات بترتيب منطقي تسلسلي لتشكيل الخوارزمية.

السؤال العاشر

اختر الإجابة الصحيحة.		
☐	A/2	1. الاسم الذي يُعدُّ اسمًا لمتغير:
☐	234	
☒	Name	
☒	محمد	2. name="محمد" print (name,"اسمي:") النص الذي يعرضه الأمر:
☐	name	
☐	print	
☒	متغير نصي.	3. MyVar = "محمد" نوع المتغير MyVar:
☐	متغير عددي.	
☐	ليس بمتغير.	

السؤال الحادي عشر

اختر الإجابة الصحيحة:		ناتج المقطع البرمجي التالي هو:
☐	50	<pre> x=20 y=30 x=y y=x Total=x+y print(Total,"مجموع x و y يساوي:") </pre>
☒	60	
☐	70	

السؤال الثاني عشر

اختر الإجابة الصحيحة عند التعامل مع لغة البرمجة "بايثون":

☐	<code>input (" اكتب المجموع ")</code>	1. الأمر الذي تحتاجه لعرض المجموع.
☐	<code>print (x,y,n)</code>	
☒	<code>print("مجموع x و y و n يساوي:", Total)</code>	
☒	<code>print("أدخل قيمة i")</code> <code>i=int(input())</code>	2. الأمر الذي تحتاجه لتطلب من المستخدم إدخال قيمة متغير.
☐	<code>print (" اكتب الرقم ")</code>	
☐	<code>int(input(i))</code>	
☐	<code>int (" أدخل قيمة y : ")</code>	3. الأمر الذي تحتاجه لتطلب من المستخدم إدخال قيمة المتغير y.
☒	<code>y=int(input (" أدخل قيمة y : "))</code>	
☐	<code>print("أدخل قيمة y:", y)</code>	

السؤال الثالث عشر

عند التعامل مع لغة البرمجة "بايثون" ضع رقم نوع البيانات المناسب في الفراغ لكل قيمة من القيم المقابلة.

"email"

3

1.23

2

"my name"

3

5

1

"5"

3

integer

1

float

2

string

3



السؤال الرابع عشر

املاً الفراغات الموجودة في المقطع البرمجي التالي لرسم الصورة.



```
from turtle import *
t=Turtle()
t.shapesize(2)

t.color(" black ")
t.begin_fill()
t.circle(100)
t.fillcolor(" green ")
t.end_fill()
t.penup()

t.goto(-40, 100)
t.color(" black ")
t.begin_ fill ()
t.circle(15)
t.color(" black ")
t.end_fill()
t. penup ()

t.goto(40, 100)
t. ..... ()
t.color(" black ")
t.begin_fill()
t.circle(15)
t.color(" black ")
t. end fill ()
t.penup()

t.goto(-40, 60)
t. pendown ()
t.pensize( 12 )
t.right(90)
t.circle(40,180)
t.penup()
t. hideturtle ( )
```