

لغة برمجة بايثون (Python) من اللغات عالية المستوى، المجانية والمفتوحة المصدر، والتي تُعد سهلة وممتعة للمبتدئين بالبرمجة. سنستخدم بيئة التطوير المتكاملة للغة بايثون (Python IDLE) الموجودة في جهازك، لكتابة وتشغيل برامج بايثون الخاصة بك.

افتح بيئة التطوير المتكاملة للغة بايثون:

1 < اضغط على Start (ابدأ).

< مرر للأسفل ثم اضغط على

2 Python3.9 (مجلد بايثون 3.9).

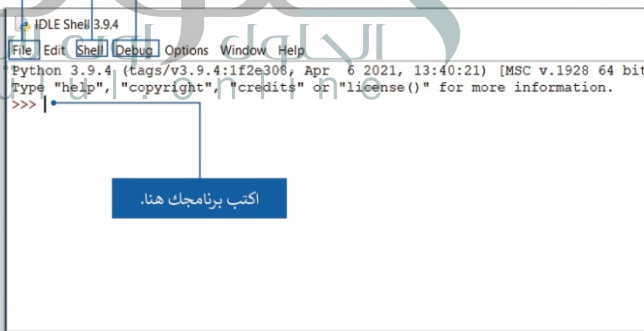
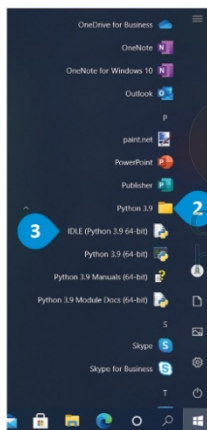
3 < اضغط على IDLE (Python 3.9 64-bit).



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

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

استخدم قائمة تصحيح (Debug) لتتبع وتشخيص الأخطاء داخل برنامجك.



اكتب برنامجك هنا.

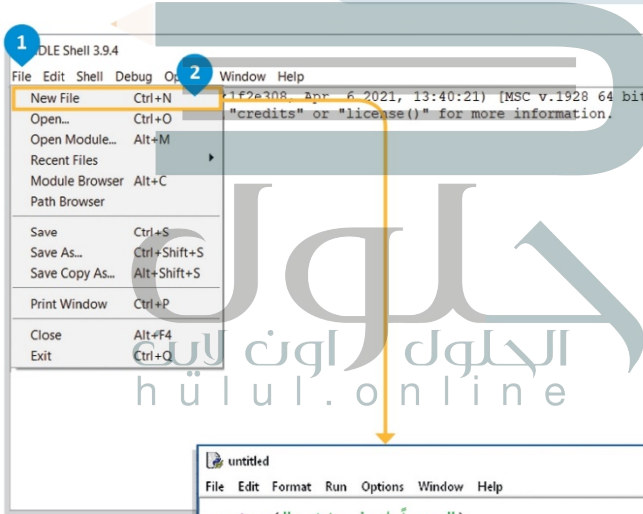
لغة البرمجة عالية المستوى تستخدم كلمات من اللغة الاعتيادية، وقد تحتوي أيضًا على كلمات ومصطلحات وتراكيب بناء الجملة نحتاج إلى تعلمها. على سبيل المثال: يستخدم بايثون كلمات "print" و "input" من اللغة الإنجليزية لتشكيل التعليمات. في المقابل هناك بعض العبارات والكلمات الرئيسة مثل: "elif" و "def" تستخدم فقط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

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

يمكنك كتابة التعليمات البرمجية في بايثون مباشرة من خلال بيئة التطوير المتكاملة للغة بايثون (Python IDLE)، أو استخدام بايثون (Python Shell)، أو استخدام بايثون (Python Shell).  
حل هذه المشكلة عن طريق إنشاء البرنامج 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 البرنامج فقط.

### إنشاء ملف البرمجة:

- < اضغط على قائمة **File** (ملف)، ثم اضغط على **New File** (ملف جديد) لإنشاء ملف جديد. ❶
- < سيفتح محرر النصوص.
- < اكتب برنامجك. ❷



الآن وبعد أن كتبت السطر الأول من التعليمات البرمجية، أصبح برنامجك جاهزاً لتشغيله ولكن انتبه يجب عليك حفظه أولاً قبل إعادة تشغيله.

### لحفظ البرنامج وتشغيله:

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



The screenshot illustrates the steps to save and run a Python program in IDLE. It shows the 'File' menu with 'Save' highlighted (1), the 'Save As' dialog box where the file is named 'welcome.py' (2), and the 'Run' menu with 'Run Module' highlighted (4). The 'Run Module' dialog box is also shown (5). The final output in the IDLE Shell shows the program running successfully and displaying 'مرحباً بك في بايثون' (Hello in Python).

1 Save

2 Save As

3 Save

4 Run Module

5 Run Module

مرحباً بك في بايثون

بيئة التشغيل

## الطباعة

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.

يستخدم برنامج بايثون دوال مدمجة ودوال مدمجة. الدالة المدمجة يتم تحديدها بالفعل في البرنامج. ويمكن للمستخدمين استخدامها مباشرة في برامجهم. تعد دوال () print و () input بعضاً من هذه الدوال التي ستستخدمها في هذه الوحدة.

هذا مثال بسيط لبرنامج تمت كتابته في محرر النصوص.

```
print.py - C:/python/print.py
File Edit Format Run Options Window Help

print("أستخدم دالة الطباعة")
```

```
IDLE Shell 3.9.4
File Edit Shell Debug Options Window Help
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 6 2021, 13:40:21) [MSC v.1928 64 bit (AMD64)] on win
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/binar/OneDrive/Documents/welcome.py =====
>>>
```

أستخدم دالة الطباعة !  
>>>

هنا يتم عرض النتيجة على الشاشة.

حاول أن تطبع الرسائل التالية:

أتعلم لغة بايثون



أحب البرمجة



جرب بنفسك

سننشئ في هذه الوحدة برنامجًا خاصًا بلعبة السباق الثلاثي. سييسأل البرنامج المستخدم عن اسم الرياضي ونتائجه في الرياضات الثلاث، بعد ذلك ستُحسب النتيجة النهائية للرياضي وأفضل أداء له. هيا لنبدأ.

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 الحاسب نستخدمه لتخزين قيمة يتم ادخالها. تمثل المتغيرات أنواعًا مختلفة من البيانات، ويمكن أن تتغير قيمة المتغير أثناء تنفيذ البرنامج. تنقسم المتغيرات إلى فئتين رئيسيتين هما الأرقام والنصوص. ندعم 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
```



## استكشاف الأخطاء

هل لديك مشكلة؟  
يرجى إعلامنا بـ "ملاحظة" عند  
استخدام المتغيرات النصية.

وزارة التعليم  
Ministry of Education  
2021 - 1443

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 اسماً وصفيًا مثل age, carname, total\_volume.  
الشروط الواجب توفرها في اسم المتغير:  
< أن يبدأ بحرف أو بشرطة سفلية.  
< ألا يبدأ برقم.

< يمكن استخدام الأحرف الإنجليزية والأرقام والشرطات السفلية (أرقام من 0-9، أحرف كبيرة من A-Z، أحرف صغيرة a-z، \_).  
< حالة الحروف الإنجليزية مهمة، فمثلاً تعتبر الأسماء التالية age, Age, AGE بمثابة ثلاثة متغيرات مختلفة.

< يفضل إعطاء أسماء للمتغيرات تمثل المحتوى وذلك لفهم ما يمثله كل متغير داخل التعليمات البرمجية.

### استكشاف الأخطاء

لا يمكن استخدام بعض الأسماء لكونها كلمات خاصة تستعمل بالفعل في لغة البرمجة، ويُطلق عليها الكلمات المحجوزة مثل:

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

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

2AB

CourseName

GrAdE

True

جرب  
بنفسك

تحديد قيمة المتغير x.

تحديد قيمة المتغير y.

إضافة متغير إلى متغير آخر.  
في هذا المثال يعمل (+)  
كمعامل رياضي.

x=15

y=20

Total=x+y

print(Total)

يعرض قيمة المتغير Total على الشاشة.

## تخصيص قيمة لمتغير

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

يمكنك أيضاً حساب أي شيء على الجانب الأيمن من علامة المساواة ثم إسناد النتيجة إلى المتغير الموجود على الجانب الأيسر. دعونا نرى مثالاً:

## المتغيرات النصية

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 علامات 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)
```

اسمي: محمد

هل يمكنك ملاحظة الاختلاف؟ هكذا أفضل بلا شك.

## التعليقات

تُستخدم التعليقات لتذكرك بوظيفة التعليمات البرمجية. قد تحتاج إلى مراجعة التعليقات في المستقبل إذا رغبت بإجراء أي تغييرات على برنامجك. يمكنك إضافة التعليقات باستخدام علامة (#) في بداية العبارة. يمكنك كتابة ما تريد في سطر تعليق حيث سيتجاهله الحاسب.

#....

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

اسمي: محمد



```

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

لعبة السباق الثلاثي  
 مجموع نقاط لعبة السباق الثلاثي: 170

### الثوابت

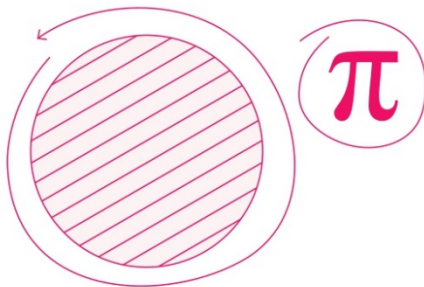
قد تحتاج أحياناً إلى متغير ذي قيمة ثابتة أثناء البرمجة. لتحقيق ذلك يمكنك استخدام المتغيرات الثابتة. على سبيل المثال: إذا كنت تريد حساب مساحة الدائرة فإن برنامجك يحتاج إلى استخدام قيمة باي والتي تساوي 3.14 وبالنظر إلى أن القيمة لا تتغير أبداً فيمكن تخزينها بصورة ثابتة باسم باي (pi). عادة ما يتم تخزين الثوابت في ملف مختلف عن البرنامج الرئيسي ويجب عليك استيرادها لاستخدامها.

```

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

قيمة باي: 3.141592653589793

استخدم السطر للوصول إلى  
 الدوال الرياضية.





## تدريب 1

اختر الإجابة الصحيحة: ◀

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

`input ("اكتب رقمين")`

`print ("اكتب رقمين")`

اكتب رقمين

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

`print (Num1, Num2)`

`print ("الرقم 2 , الرقم 1")`

`print (N1, N2)`



## تدريب 2

➤ أنشئ برنامج لإضافة درجتك في الرياضيات في المتغير myGrade، ثم أطلع درجتك على الشاشة.

النص البرمجي

```
# تعيين قيمة لاسم المتغير
myGrade=10
# طباعة قيمة المتغير
print ("درجتي في الرياضيات:", myGrade)
```

## تدريب 3

➤ اكتب ناتج البرنامج التالي:

```
x=20
y=30
Total=x+y
print(Total, "مجموع x و y يساوي:", 50)
```

الناتج

مجموع x و y يساوي: 50

