



# Python

**Ibrahim Eldesouky**  
**Technical Master Trainer**



ما هي علوم الحاسب



# علوم الحاسب

- ليس دراسة الحاسب الالى
- ليس كيفية كتابة البرامج او البرمجة
- ليس استخدام التطبيقات والبرامج



# علوم الحاسب

هو دراسة الخوارزميات

**الخوارزميات :**

هي مجموعة من الخطوات الرياضية والمنطقية والمتسلسلة اللازمة لحل مشكلة ما

A close-up photograph of a laptop's keyboard and screen. The keyboard is silver with black keys. The screen displays a color calibration tool with various color patches and numerical values. The background is a solid magenta color.

# مجالات الحاسب

- Graphics & Images Processing
- Programmer
- Database
- Network
- Operating System
- Hardware





## ما هي لغة python

لغة بايثون Python هي لغة برمجة عامة الهدف General-purposed مفتوحة المصدر، أي يمكن استخدامها في شتى مجالات البرمجة دون تقييد بمجال معين كما هو الحال في بعض لغات البرمجة الأخرى، كما يمكن لأي مطور التعديل عليها والتطوير فيها بما يتناسب مع احتياجاته وتطلعاته.



## نشأة python

طورت لغة بايثون في منتصف تسعينيات القرن الماضي لتكون لغة جذابة وقادرة على استقطاب المبرمجين من الجيل الشاب. إذ تتمتع اللغة بكونها سهلة التعلم والفهم، بالإضافة إلى تركيزها على تنفيذ العمليات التي يريدها المبرمج عوضاً عن التركيز على آلية الكتابة والأخطاء فيها. مما جعلها من أسهل لغات البرمجة تعلمًا وكتابةً.

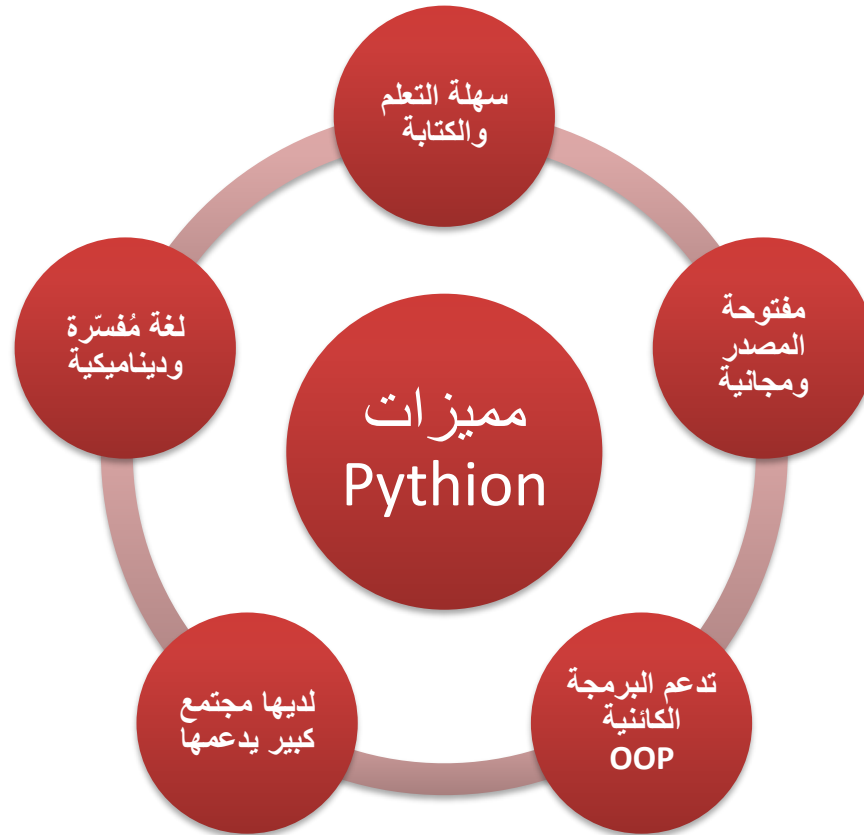
أطلق الإصدار الأول من بايثون Python 1 في عام ١٩٩٤، بعد ذلك أخذ مطورو لغة البرمجة يعملون على إضافة التحديثات والإضافات حتى إطلاق الإصدار الثاني Python 2 الذي يحمل الكثير من ميزات بايثون الأساسية عام ٢٠٠٠.

ساهم انتشار البايثون الكبير وزيادة شعبيتها في تمهيد الطريق أمام التحديث الثالث لإصدارات البايثون Python 3 في عام ٢٠٠٨. أضيفت للإصدار الثالث الكثير من التحديثات





# لماذا أستخدم Python؟



تعدّ لغة بايثون واحدة من أسهل اللغات البرمجية للتعلم، بسبب سهولة بناءها ومنطقية أوامرها البرمجية التي تشبه إلى حدٍ كبير الكتابة باللغة الانجليزية، مما يجعلها صديقة ومُحبّبة جدًا لمعظم المطورين، وبخاصةً المبتدئين منهم. إضافةً إلى أنها صُمّمت لتكون لغة برمجة عالية المستوى ((high-level programming، أي لا تحتاج أن تكون على دراية وخبرة ببنية الترميز وإدارة الذاكرة، وغيرها من الأمور التقنية المعقدة

A close-up photograph of a laptop keyboard, showing keys like 'D', 'F', 'T', '6', '7', '8', '9', '0', 'P', 'U', 'I', 'O', 'K', 'N', 'M', 'B', 'V', 'C', 'G', 'H', 'J', 'Z', 'X', 'C', 'V', 'B', 'N', 'M'. To the right of the keyboard is a color calibration chart with various color patches and numerical values. The background is a solid magenta color.

# مفتوحة المصدر ومجانية

بايثون لغة برمجة مفتوحة المصدر، بإمكان أي شخص إنشاء نسخة خاصة منها والمساهمة في تطويرها. لدى Python منتدى عبر الإنترنت، حيث يجتمع الآلاف من المبرمجين يوميًا لتحسين هذه اللغة بصورة أكبر. إضافةً إلى ذلك، تُعدّ لغة بايثون مجانية التنزيل والاستخدام في أي نظام تشغيل، سواء أكان نظام Windows أو Mac أو Linux .

A close-up photograph of a laptop keyboard, showing keys like 'D', 'F', 'T', '6', '7', '8', '9', '0', 'P', 'O', 'U', 'I', 'J', 'K', 'N', 'M'. To the right of the keyboard is a color calibration chart with various color patches and numerical values. The background of the slide is a vibrant pink with diagonal lines.

# تدعم البرمجة الكائنية OOP

أحد الميّزات الرئيسية للغة بايثون أنها تتبع وتدعم نهج البرمجة كائنية التوجه أو البرمجة الشيئية (Object-oriented programming)، مما يسهّل عملية التطوير ويجعلها فعّالة بصورة أكبر على المدى البعيد، إذ يتيح هذا النهج إعادة استخدام الشيفرات البرمجية ويمنع التكرار، مما يمنح المطورون مرونة أكبر وتوفير الوقت والجهد في أثناء العملية الإنتاجية.



# لديها مجتمع كبير يدعمها

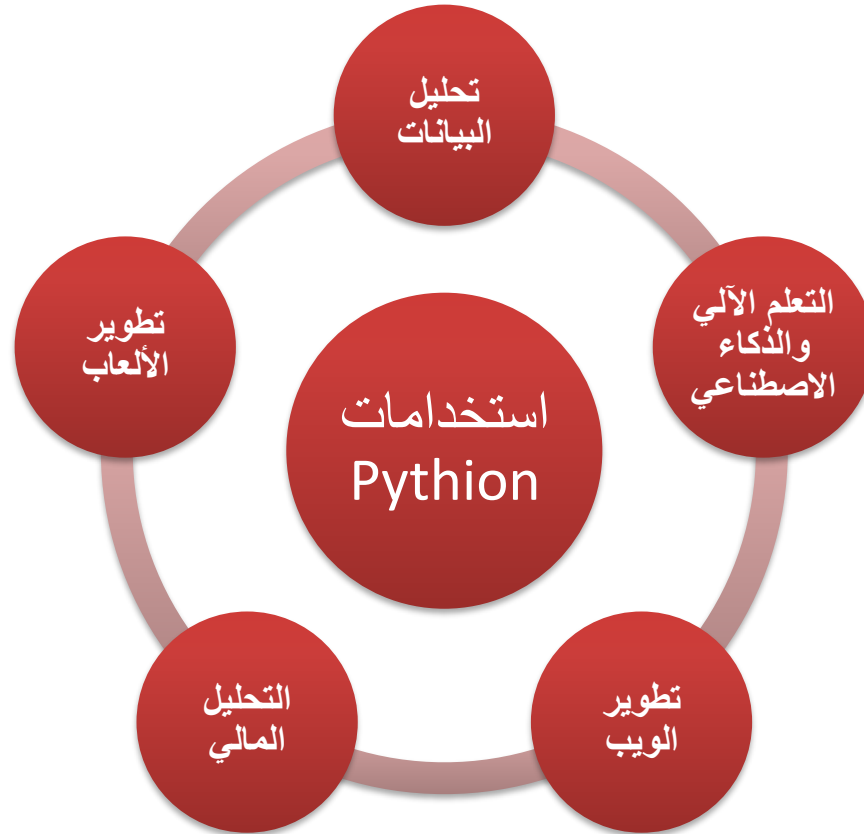
تمتلك لغة بايثون أحد أكبر المجتمعات في منصتي StackOverflow وMeetup، التي أكسبتها شعبية كبيرة على مر السنين. إذا احتجت إلى أي نوع من المساعدة المتعلقة بلغة Python، فإن المجتمع الضخم موجود دائمًا للإجابة على استفساراتك.

A close-up photograph of a laptop keyboard, showing keys like 'D', 'F', 'T', '6', '7', '8', '9', '0', 'P', 'U', 'J', 'K', 'N', 'M'. To the right of the keyboard is a color calibration chart with various color patches and numerical values. The background is a solid magenta color.

# لغة مُفسّرة وديناميكية

تُعدُّ لغة Python لغة مُفسّرة، أي أنها لا تحتاج ترجمة الشيفرات إلى لغة الآلة في ملف منفصل، ثم بعد ذلك يتم التشغيل، إنما تُقرأ الشيفرات مباشرةً وقت التشغيل Run Time عبر برنامج يدعى «المفسّر» interpreter، مما يمنحها ميزة تعريف المتغيرات دون الاضطرار إلى تخزينها في مكانٍ في الذاكرة، وتسمح كذلك باستدعاء الدوال المُعرّفة عدّة مرات بتمرير قيم مختلفة في كل مرة.

# استخدامات Python؟








تحميل Python

# معرفة اصدار Python؟



```
Command Prompt
Microsoft Windows [Version 10.0.19045.3996]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ibrahim>py --version
Python 3.12.2

C:\Users\Ibrahim>
```



# انشاء اول كود Python



# المتغيرات variables

- عبارة عن أماكن يتم حجزها في الذاكرة بهدف تخزين بيانات فيها أثناء تشغيل البرنامج.
- في بايثون، المبرمج غير مسؤول عن تحديد أنواع المتغيرات التي يعرفها في برنامجهِ فعلياً، عندما تقوم بتعريف متغير وتضع فيه أي قيمة، سيقوم مفسر لغة بايثون بتحديد نوع هذا المتغير بناءً على القيمة التي أسندتها إليه بشكل تلقائي وقت التشغيل.

## مثال لإنشاء المتغيرات

```
x = 5
```

```
print("x = " , 5)
```

```
y = z = a = 10
```

```
print("y = " , y)
```

```
print("z = " , z)
```

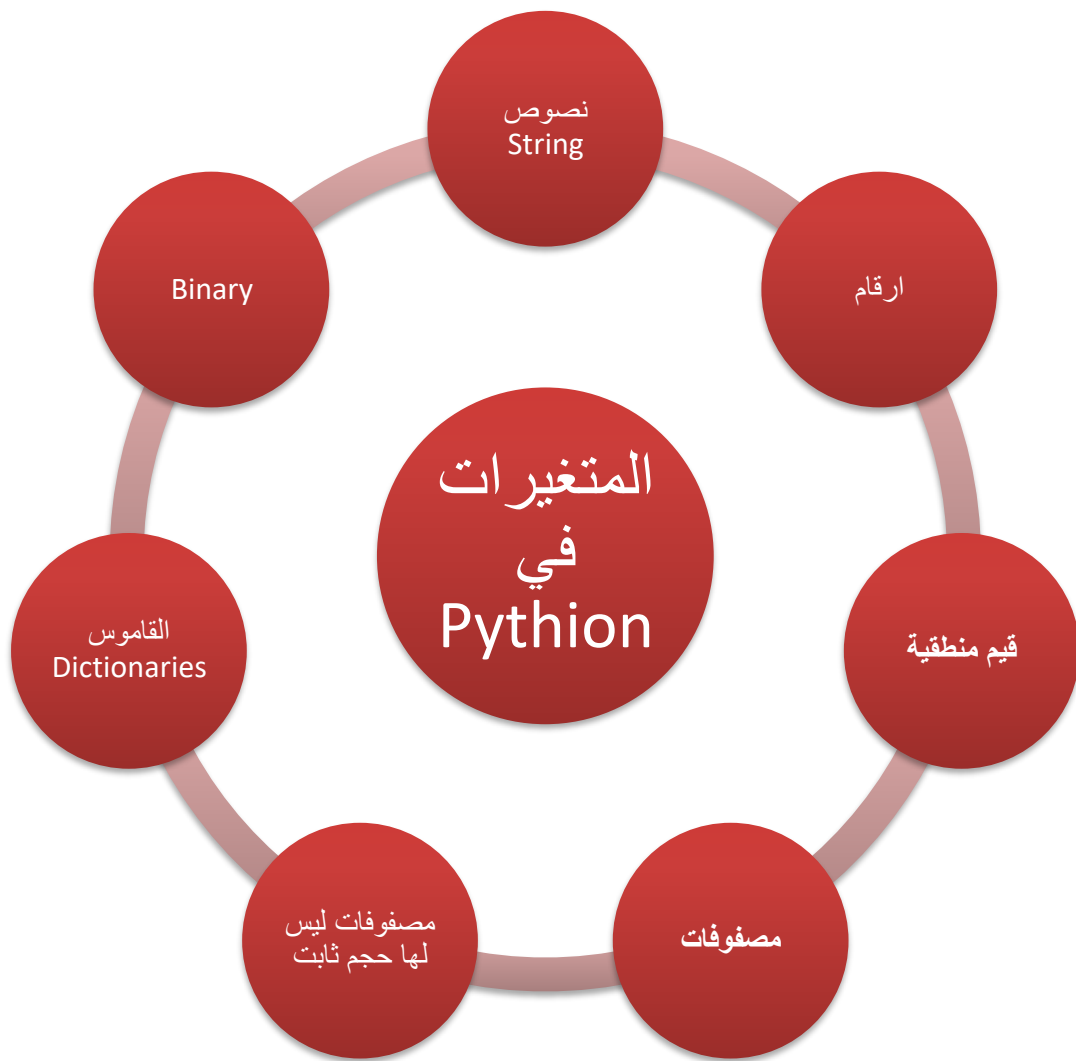
```
print("a = " , a)
```

```
# لمعرفة نوع المتغير
```

```
name = "ibrahim Eldesouky"
```

```
print(type(x))
```

```
print(type(name))
```






# الروابط الرياضية

الرابط	دوره
+	عملية الجمع
-	عملية الطرح
/	عملية القسمة
*	عملية الضرب
%	باقي القسمة





task



قم بإنشاء برنامج يجمع قيم  
يقوم بإدخالها المستخدم

# الروابط المقارنة

المؤثر الاسم	الوصف
$==$ هل يساوي	يقارن اذا كان متغير يساوي اخر
$!=$ لا تساوي	يقارن اذا كان متغير لا يساوي اخر
$<$ اكبر من	يقارن اذا كان متغير اكبر من اخر
$>$ اصغر من	يقارن اذا كان متغير اصغر من اخر
$=<$ اكبر من او يساوي	يقارن اذا كان متغير اكبر من او يساوي اخر
$>$ اصغر من او يساوي	يقارن اذا كان متغير اصغر من او يساوي اخر


# الروابط المنطقية



العامِل	الشرح
and	إن كان كلا التعبيرين صحيحين True
or	إن كان أحد التعبيرين على الأقل صحيحًا True



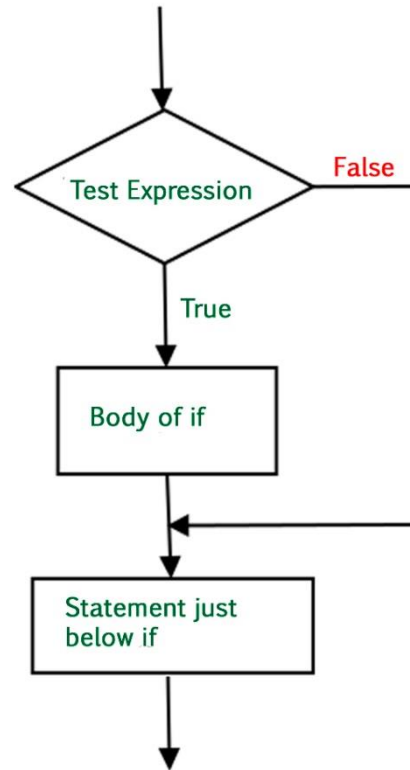
task



قم بإنشاء برنامج يقوم بحساب  
العمر بناء على سنة الميلاد الذي  
يدخلها المستخدم

# البنية الشرطية

## IF باستخدام




مثال

```
Users > Ibrahim > Desktop > py.py > ...  
1   x = int(input("Type Your Age"))  
2  
3   if x < 18 :  
4       | print("You are A Young")  
5   elif x < 60:  
6       | print("You are Adult")  
7   else:  
8       | print("Error Age")
```



task

- 
- قم بإنشاء برنامج ينفذ العمليات الحسابية بناء على قيم وروابط رياضية يقوم بادخالها المستخدم باستخدام البنية الشرطية If



# Lists

القوائم في بايثون هي أحد أنواع البيانات المركبة و التي تستخدم لتجميع القيم معًا. يمكن كتابة القوائم كمجموعة من القيم المفصولة بفواصل ومحاطة بأقواس مربعة وتدعم أنواعًا مختلفة من العناصر

```
mylist = ["Ibrahim","eldesuky","Mohamed","Ahmed"]  
  
print(mylist)
```

# Access List Items

```
mylist = ["Ibrahim", "eldesuky", "Mohamed", "Ahmed"]
```

```
mylist[2] = "Ali"
```

```
print(mylist[1:3])
```



# Add & Remove Item

Append

Insert

Extend

Remove

Pop

Del

Clear



## List Methods

len

type

Sort

count

## البنية التكرارية

- وهي تعني مجموعة من الأساليب والأدوات تساعد علي تكرار امر معين عوضا عن كتابته اكثر من مرة



# اساليب وادوات البنية التكرارية

For

While

While

```
x = 1
```

```
while x <= 12 :
```


```
    print(x)
```

```
    x+=1
```






task



قم بإنشاء برنامج يقوم بعرض  
جدول الضرب بدا من جدول ١  
حتى جدول ١٢ باستخدام البنية  
التكرارية While




task



قم بإنشاء برنامج يقوم بعرض جدول  
الضرب بدا من جدول القيمة التي  
يدخلها المستخدم حتي جدول القيمة  
التي يدخلها المستخدم باستخدام البنية  
التكرارية while



task



قم بإنشاء برنامج يطبع الأعداد  
الزوجية بين ١ حتى ٢٠ باستخدام  
while حلقة

**For**


```
name = ["Ibrahim", "Eldesouky", "Mohamed", "Ahmed"]  
for x in name:  
    print(x)
```

## Range

```
for x in range(0 , 100 , 3):  
    print(x)  
else :  
    print("finished")
```




task



قم بإنشاء برنامج يقوم بعرض  
جدول الضرب بدا من جدول ١  
حتى جدول ١٢ باستخدام البنية  
التكرارية For



task



قم بإنشاء برنامج يقوم بعرض جدول  
الضرب بدا من جدول القيمة التي  
يدخلها المستخدم حتي جدول القيمة  
التي يدخلها المستخدم باستخدام البنية  
التكرارية For

A close-up photograph of a laptop keyboard and a color calibration chart. The keyboard is silver with black keys. The chart shows various color patches and numerical values for color management.

# Functoin

```
def greet(name):  
    print("Hello", {name})  
  
greet("Ibrahim")
```

- الدوال هي مجموعة من البيانات التي تقوم بتنفيذ مهمة محددة. الفكرة هي تجميع المهام المتكررة معًا وإنشاء دالة بحيث يمكننا استدعاءها لإعادة استخدام الشيفرة الموجودة فيها مرارًا وتكرارًا بدلاً من كتابة نفس الشيفرة مرارًا وتكرارًا لإدخالات مختلفة.
- عند تعريف دالة في بايثون، يمكنك تحديد مجموعة من البيانات التي تشمل الأوامر التي تريد تنفيذها. يمكنك استدعاء هذه الدالة في أي مكان آخر في برنامجك





# Date & Calendar

today

Now

prcal


Month

isleap

leapdays



task



قم بإنشاء برنامج يقوم بعرض اذا  
كانت السنة كبيسة ام لا من خلال  
السنة التي يدخلها المستخدم ويعرض  
جدول شهر فبراير اذا كانت السنة  
كبيسة



# واجهات المستخدم الرسومية

يستخدم المبرمج في بداية تعلم البرمجة واجهة سطر الأوامر `command-line interface`، وهي واجهة تعتمد اعتمادًا كليًا على المدخلات من خلال لوحة المفاتيح، يكون فيها ترتيب الأحداث معتمدًا على طريقة كتابة المبرمج للأوامر ولا رأي للمستخدم في ترتيب سير الأحداث، وعلى العكس من ذلك تكون الواجهات الرسومية `Graphical Interface` التي يتفاعل فيها المستخدم مع البرنامج من خلال مؤشر الفأرة ويكون المتحكم في سير الأوامر من خلال ما يختار أن ينقر عليه أو أن يقوم بتعبئته من المدخلات بالترتيب الذي يرغبه، فيتفاعل مع العديد من العناصر كالأزرار والقوائم وصناديق المدخلات



تعد المكتبة Tkinter في بايثون اختيارًا جيدًا لإنشاء الواجهات الرسومية لعدة أسباب أهمها أنها سهلة التعلم، ويستخدم فيها القليل جدًا من التعليمات البرمجية لإنشاء تطبيق سطح مكتب يعمل بشكل ممتاز، ونستطيع تشغيلها على مختلف أنظمة التشغيل