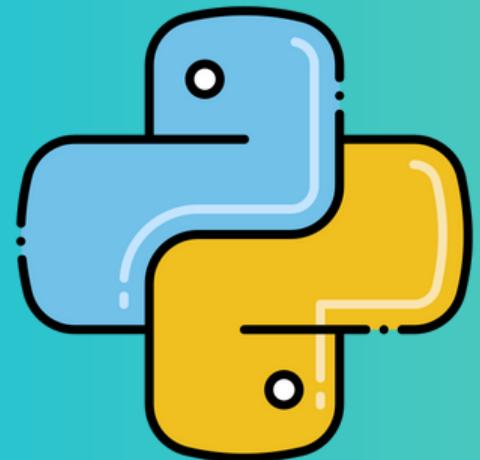


P Y T H O N C O U R S E

أساسيات لغة البايثون



BY. PYQONSOLE

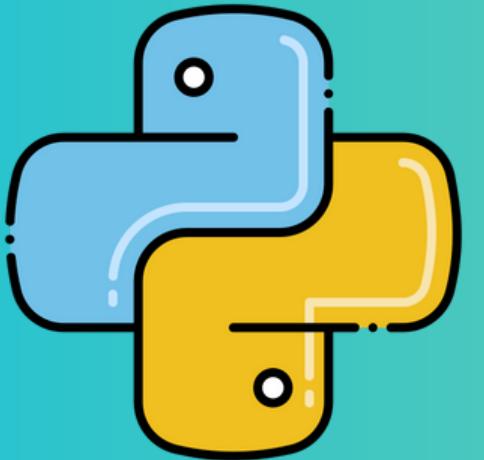


طريقك لاكتراك لغة البايثون

المستوى الأول

P Y T H O N C O U R S E

أساسيات لغة البايثون



BY. PYQONSOLE

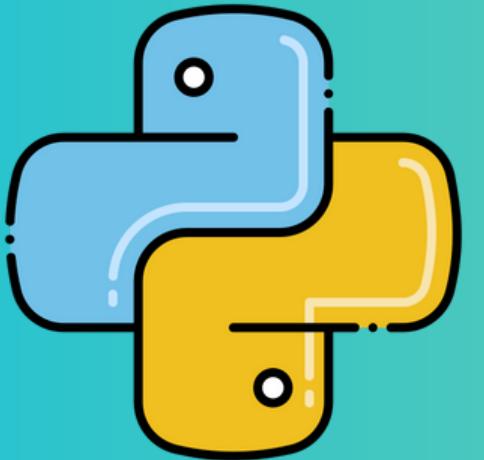


مقدمة

- تعریف لغات البرمجة
- الغرض من تعلم لغة برمجة
- نبذة قصيرة عن لغة البايثون
- خصائص لغة البايثون

P Y T H O N C O U R S E

أساسيات لغة البايثون



BY. PYQONSOLE

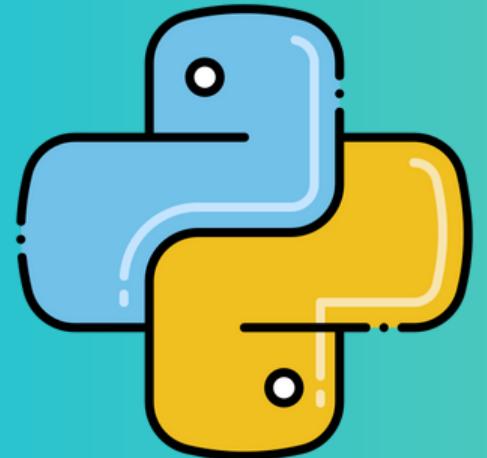


• تعریف لغات البرمجة

لغات البرمجة هي لغة تواصل بين المبرمج والآلة، والتي تكون على شكل كتابة لا ياهر مفهومها عند تحليلها من طرف محرر النصوص

P Y T H O N C O U R S E

أَسَاسِيَّات لغة الـ Python



BY. PYQONSOLE

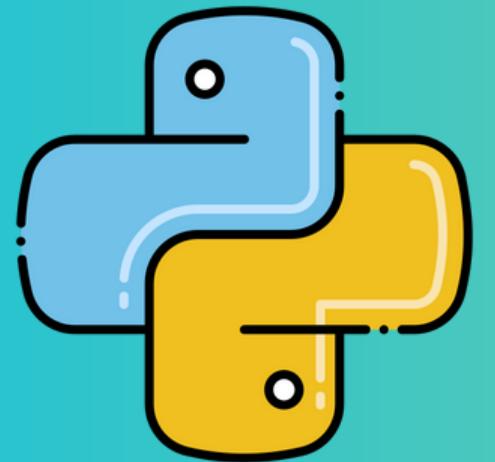


الغرض من تعلم لغة البرمجة

تلبيـة حاجة انسانـية لـتسهـيل المـهام
الـليـوـمـيـة و اـبـتكـار حلـول تقـنيـة
لـالمـشـاـكـل و الصـعـوبـات فـكـل مـادـة
برـمـجـة لـادـاء مـهـام مـا

P Y T H O N C O U R S E

أساسيات لغة البايثون



BY. PYQONSOLE

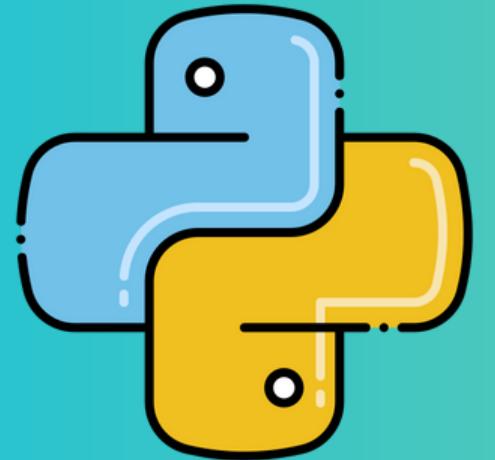
• نبذة قصيرة عن لغة البايثون

بايثون هي لغة برمجة عالية المستوى، تعتبر من أكثر اللغات شعبية وسهولة في التعلم. تم تطويرها في أواخر الثمانينيات على يد جودو فان روسم، وأصدرت لأول مرة في 1991.

تميز بايثون بوضوح تركيبها وبساطتها، مما يجعلها خياراً ممتازاً للمبتدئين والمطورين المحترفين على حد سواء.

P Y T H O N C O U R S E

أَسَاسِيَّات لغة البايُثُون



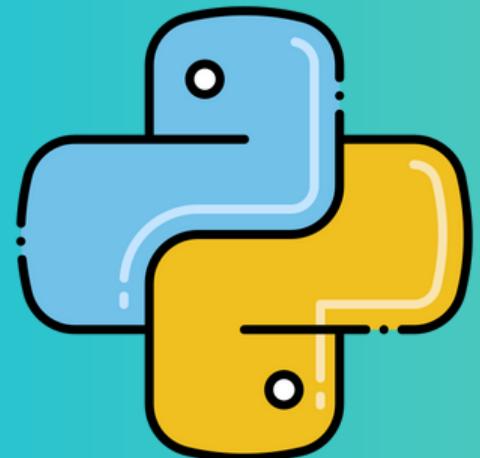
BY. PYQONSOLE

• خصائص لغة البايُثُون

- سهولة القراءة: تعتمد على تركيب بسيط وواضح
- متعددة الاستخدامات: يمكن استخدامها في تطوير البرمجيات، التطبيقات، تحليل البيانات، وغيرها
- مدعومة من قبل مجتمع كبير: يتوفر دعم واسع من خلال المنتديات والمكتبات المجانية

P Y T H O N C O U R S E

أساسيات لغة البايثون



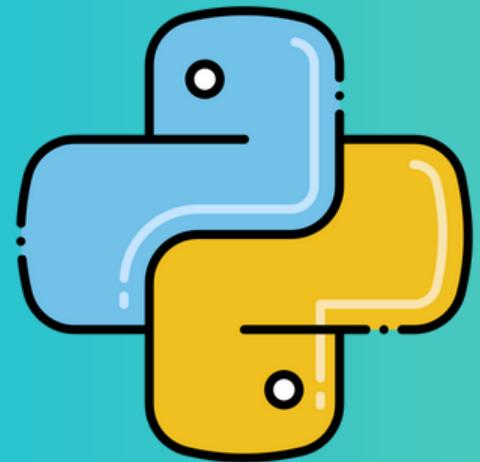
BY. PYQONSOLE

مدخل لغة البايثون

- التعليق Comment
- تسمية المتغيرات Variable Naming
- الطباعة Print
- الأعداد
- السلاسل
- الجمل الشرطية و العمليات المنطقية
- الحلقات التكرارية

P Y T H O N C O U R S E

رُسَاسِيَّات لغة البايُثُون



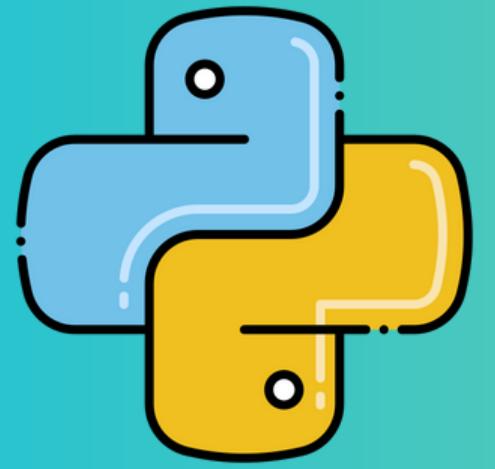
BY. PYQONSOLE



- التعليق - 1

P Y T H O N C O U R S E

أساسيات لغة البايثون



BY. PYQONSOLE



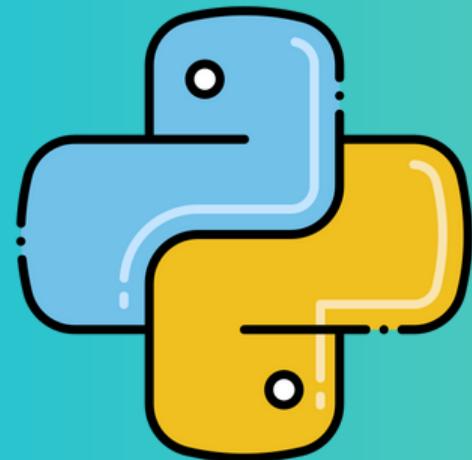
1 - التعليق - comment

الغرض من التعليق

- تساعد التعليقات في شرح الوظيفة أو المنطق وراء أجزاء معينة من الكود، مما يجعل من السهل فهمه لاحقاً.
- عند العودة إلى الكود بعد فترة، يمكن أن تساعد التعليقات المطوريين في تذكر ما كان يفعله الكود وكيفية تعديله.

P Y T H O N C O U R S E

رُسَاسِيَّات لغة البايِثُون



BY. PYQONSOLE



- comment - 1

طريقة كتابة التعليق

1-1 التعليق في سطر واحد

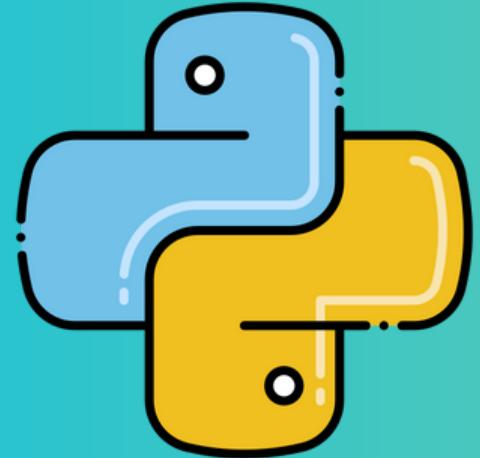
```
# this is a python course by PyQonsole
```

2-1 التعليق في عدة أسطر

```
''' this is a python course  
by - PyQonsole -  
2025 - 02 - 09  
Subscribe & Hit Like Button !!
```

P Y T H O N C O U R S E

أَسَاسِيَّات لغة البايُثُون



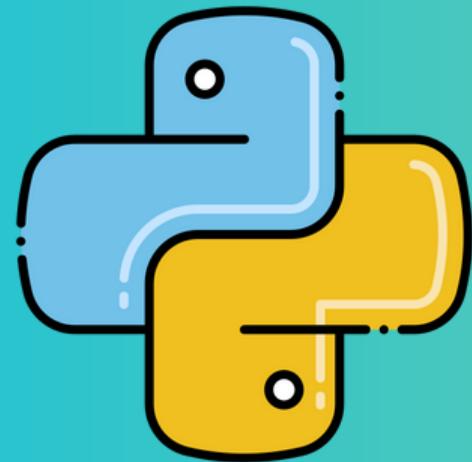
BY. PYQONSOLE



2 - تَسْمِيَّة المُتَغَيِّرَات

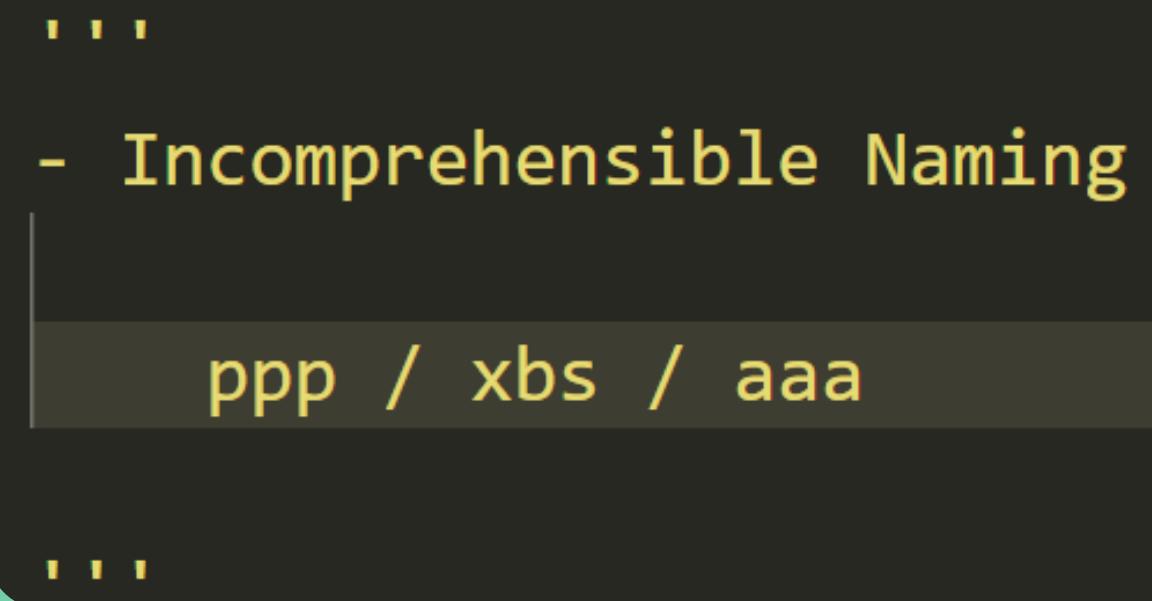
P Y T H O N C O U R S E

أساسيات لغة البايثون



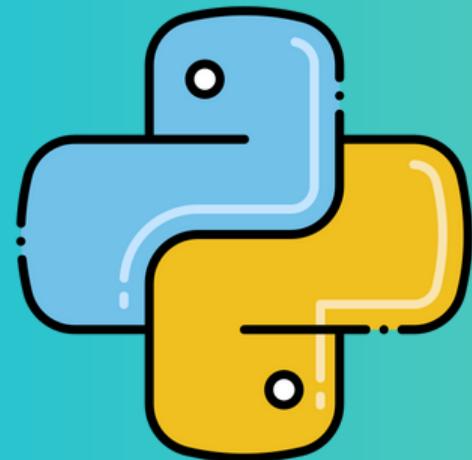
2 - تسمية المتغيرات

تتميز المتغيرات بأسماء مختلفة عن غيرها و من الأحسن اعطاءها أسماء معبرة و مميزة عوض تسمية سريعة و هامة. مثل :



P Y T H O N C O U R S E

أساسيات لغة البرمجة



BY. PYQONSOLE

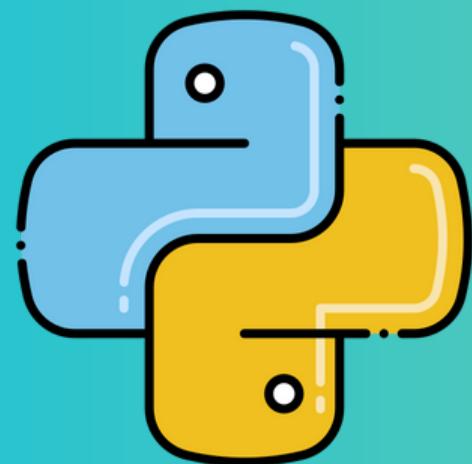
2 - تسمية المتغيرات

- التسمية الصحيحة للمتغير

```
...
- Understood Naming
product / name / address ...
...
```

P Y T H O N C O U R S E

أساسيات لغة البرمجة



BY. PYQONSOLE



2 - تسمية المتغيرات

- طرق تسمية المتغيرات
- طريقة حدبة الجمل

...

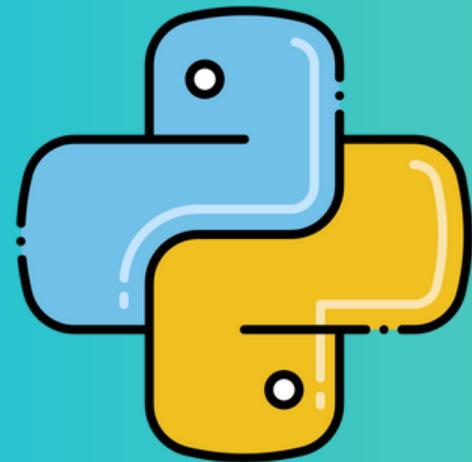
- camel case naming

...

myName / VariableName

P Y T H O N C O U R S E

أساسيات لغة البايثون



BY. PYQONSOLE

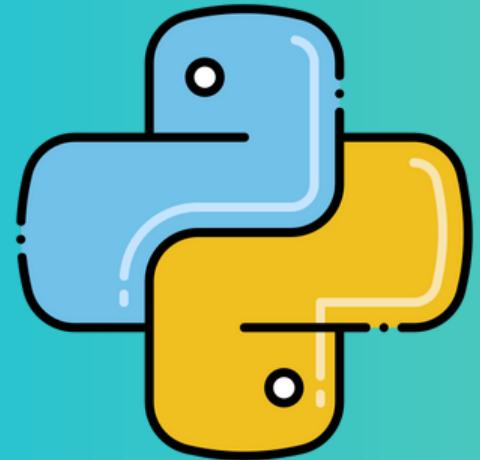
2 - تسمية المتغيرات

- طرق تسمية المتغيرات
- طريقة التسطير أسفل السطر Under Score

```
...
- Under Score Naming
product_name / object_id / _file ...
...
```

P Y T H O N C O U R S E

أساسيات لغة البايثون



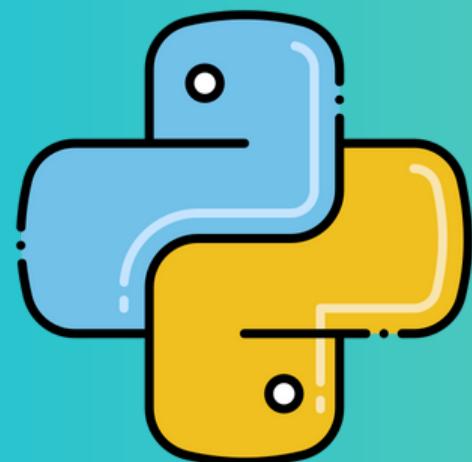
BY. PYQONSOLE



-Print - الطباعة 3

P Y T H O N C O U R S E

أساسيات لغة البايثون



BY. PYQONSOLE

-Print - 3 - الطباعة-

الطباعة توضح نتيجة الكود و التحقق من
خلو البرنامج من أي خطأ

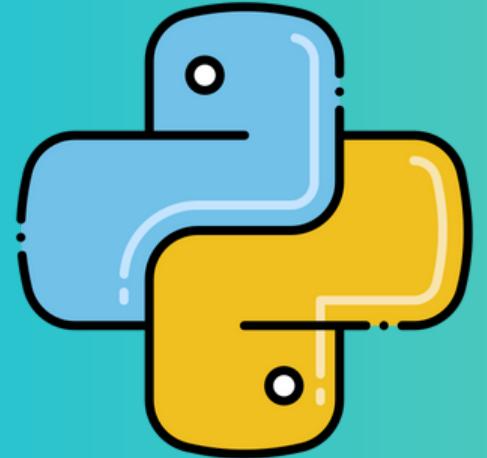
```
print(1)  
# output : 1
```

و كذلك

```
print(1.0)  
# output : 1.0
```

P Y T H O N C O U R S E

أساسيات لغة البرمجة



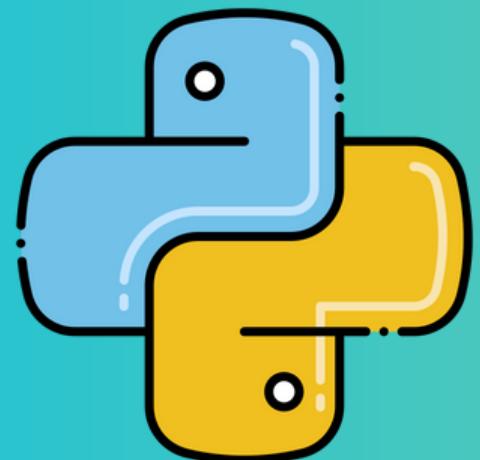
BY. PYQONSOLE



الاعداد - 4

P Y T H O N C O U R S E

أساسيات لغة البرمجة



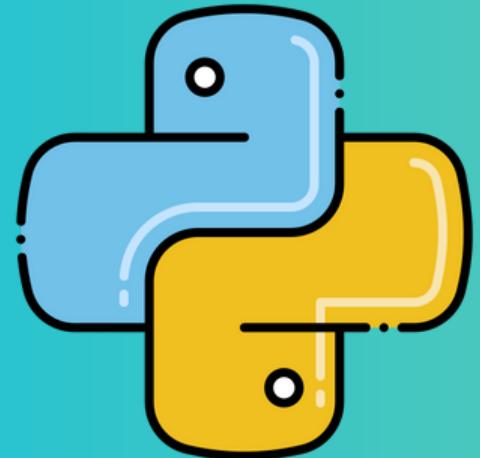
BY. PYQONSOLE

4 - الأعداد

تُستعمل الأعداد لإجراء عمليات حسابية على أنواع بيانات رقمية للحصول على ناتج كلي.

P Y T H O N C O U R S E

أساسيات لغة البرمجة



BY. PYQONSOLE



4 - الأعداد

• أنواع الأعداد

• الأعداد الصحيحة (Integer)

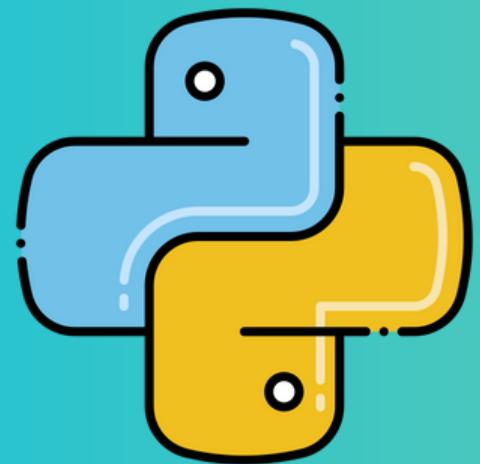
```
integer_number = 1
```

• الأعداد العشرية (float)

```
float_number = 1.5
```

P Y T H O N C O U R S E

رُسَاسِيَّات لغة البايُثُون



BY. PYQONSOLE



4 - الأعداد

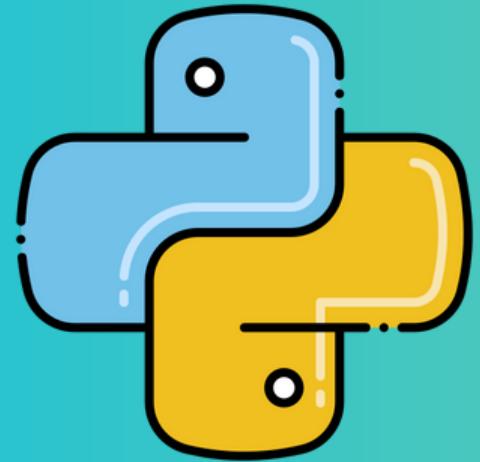
- التعامل بالأعداد
- أمثلة في العمليات الحسابية

- عملية الجمع

```
number_1 = 1
number_2 = 1
result = number_1 + number_2
# output : result = 2
```

P Y T H O N C O U R S E

أساسيات لغة البرمجة



BY. PYQONSOLE

4 - الأعداد

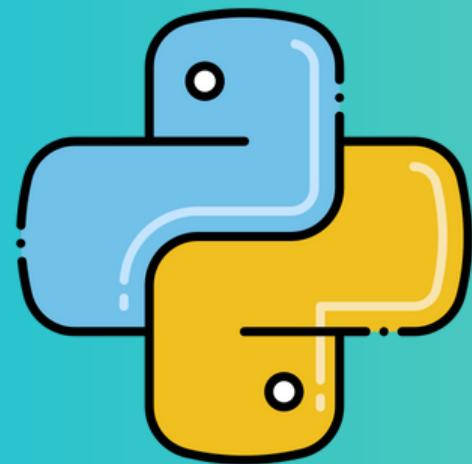
• التعامل بالأعداد
أمثلة في العمليات الحسابية

• عملية الطرح

```
number_1 = 1
number_2 = 1
result = number_1 - number_2
# output : result = 0
```

P Y T H O N C O U R S E

رُسَاسِيَّات لغة البايُثُون



BY. PYQONSOLE

4 - الأعداد

• التعامل بالأعداد

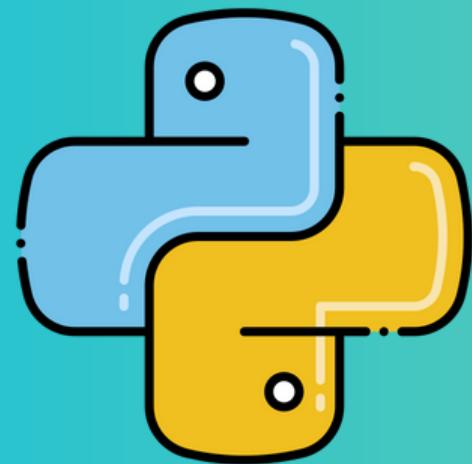
أمثلة في العمليات الحسابية

• عملية الضرب

```
number_1 = 1  
number_2 = 1  
result = number_1 * number_2  
# output : result = 1
```

P Y T H O N C O U R S E

أساسيات لغة البرمجة



BY. PYQONSOLE



4 - الأعداد

- التعامل بالأعداد

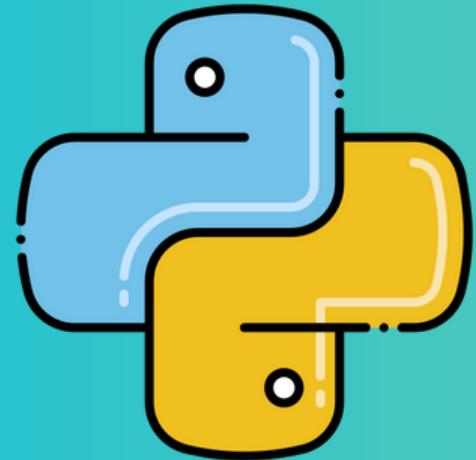
أمثلة في العمليات الحسابية

- عملية القسمة

```
number_1 = 1  
number_2 = 1  
result = number_1 / number_2  
# output : result = 1.0
```

P Y T H O N C O U R S E

أساسيات لغة البايثون



4 - الأعداد

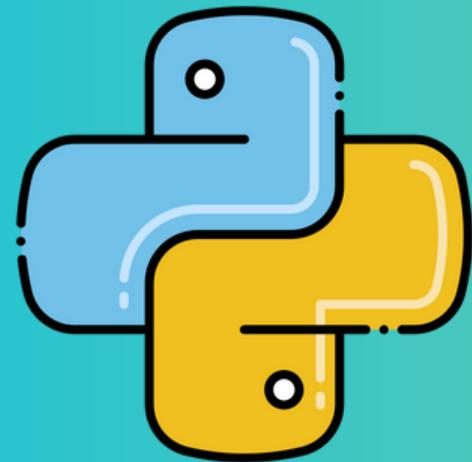
أمثلة في العمليات الحسابية

عملية باقي القسمة

```
number_1 = 1
number_2 = 1
result = number_1 % number_2
# output : result = 0
```

P Y T H O N C O U R S E

أساسيات لغة البرمجة



BY. PYQONSOLE



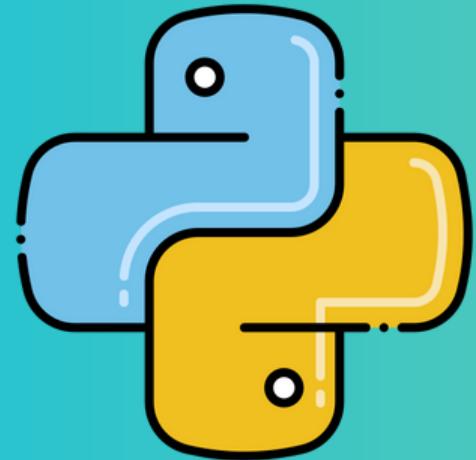
4 - الأعداد

- التعامل بالأعداد
- أمثلة في العمليات الحسابية
- عملية القسمة التحتية //

```
number_1 = 1
number_2 = 1
result = number_1 // number_2
# output : result = 1
```

P Y T H O N C O U R S E

أساسيات لغة البايثون



BY. PYQONSOLE

PYTHON
Py-Qonsole

PROGRAMMING DEVELOPMENT

4 - الأعداد

- التعامل بالأعداد

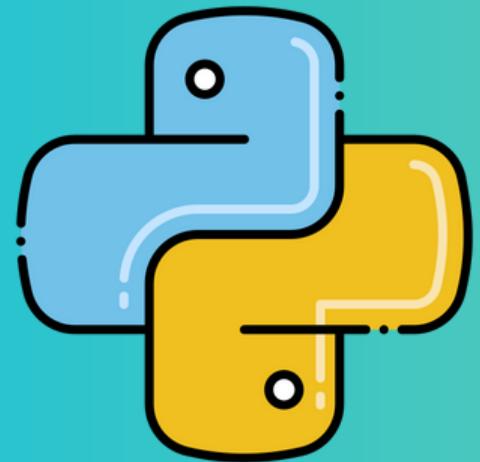
أمثلة في العمليات الحسابية

- تحويل قيمة العدد من قيمة موجبة إلى سالبة

```
number_1 = 1
number_2 = 1
result = - number_1 / number_2
print(result)
# output : result = -1.0
```

P Y T H O N C O U R S E

أساسيات لغة البايثون

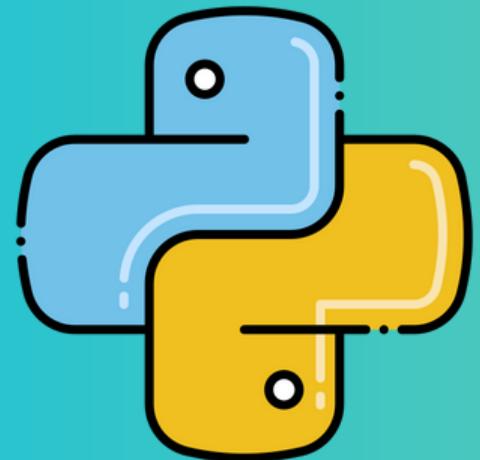


BY. PYQONSOLE

4 - الأعداد

- التعامل بالأعداد
 - أمثلة في العمليات الحسابية
 - تحويل قيمة العدد من قيمة موجبة إلى سالبة
- ```
number_1 = 1
number_2 = 1
result = number_1 /- number_2
output : result = -1.0
```

## أُسَاسِيَّات لغة البايُثُون

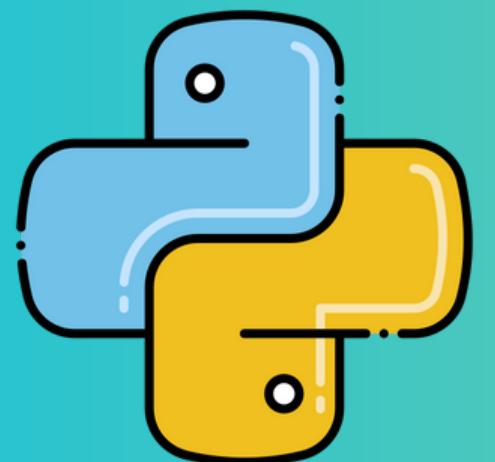


BY. PYQONSOLE

السلسلة هي أي شيء يمكن تجزئه إلى عدّة أقسام و يمكن دمج سلسلة بأخرى

# P Y T H O N   C O U R S E

أساسيات لغة البايثون



BY. PYQONSOLE

## 5 - السلسل

1 - سلسلة النصية String

2 - سلسلة القائمة List

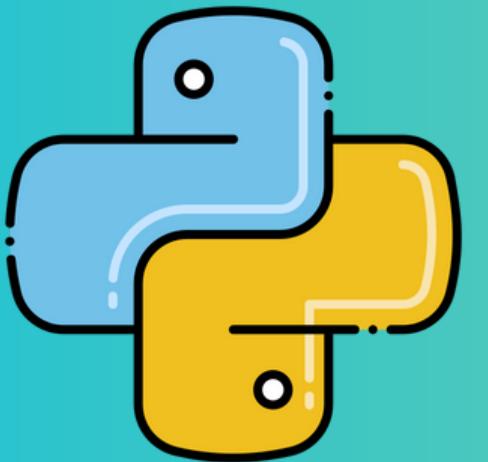
3 - سلسلة المترابطة البيانية Tuple

4 - سلسلة التعين Set

5 - سلسلة القاموس Dictionary

# P Y T H O N   C O U R S E

أساسيات لغة البرمجة



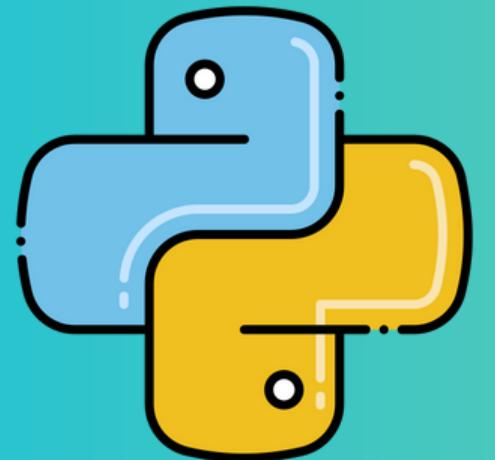
BY. PYQONSOLE

5 - المقادير

( String ) 1 - 5

# P Y T H O N   C O U R S E

أساسيات لغة البايثون



## ( String ) - 1 - 5 السلاسل النصية

طريقة كتابة السلسلة النصية

سلسلة نصية بين علامتي اقتباس واحد

```
string Variable
```

```
string_var = 'Single Quote String'
```

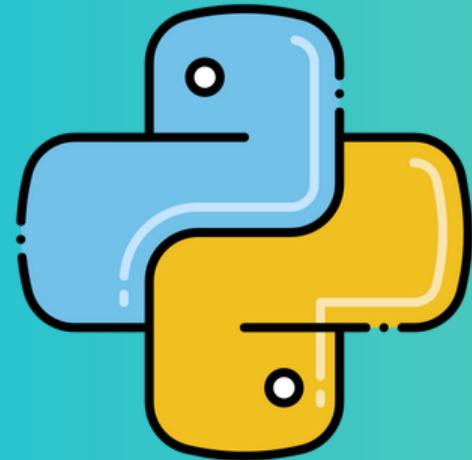
سلسلة نصية بين علامتي اقتباس هزدوج

```
string Variable
```

```
string_var = " Double Quote String "
```

# P Y T H O N   C O U R S E

أُسَاسِيَّات لغة البايُثُون



B Y . P Y Q O N S O L E

## ١ - ٥ اللسلة النصية ( String )

### طباعة السلسلة النصية

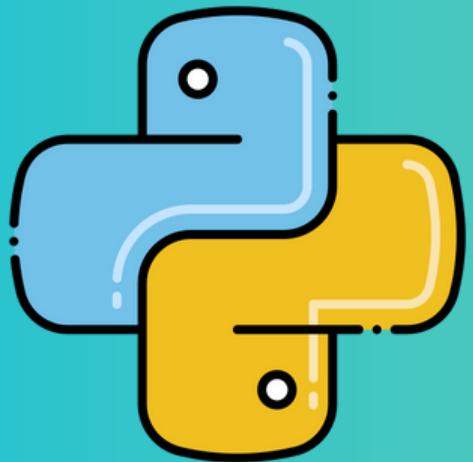
- طباعة سلسلة نصية ذات سطر واحد

```
print in python
print('subscribe !!')
```

- طباعة سلسلة نصية تتكون من عدة أسطر

```
print(
 """
 subscribe bro !!
 """
```

## أساسيات لغة البايثون



## ٥ - ١ . السلسلة النصية ( String )

### التعامل مع السلسل النصية

- طباعة العناصر الموجودة بـ **تقنية الفهرسة**

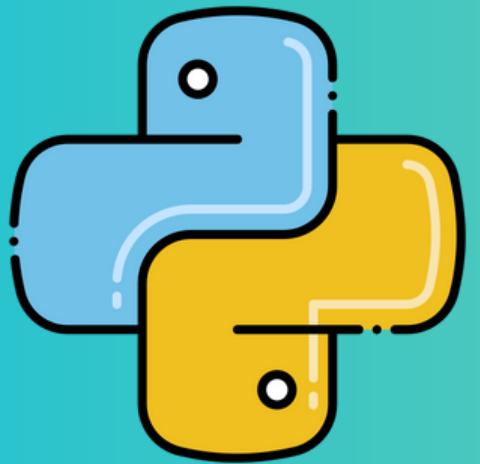
**الفهرسة في جميع لغات البرمجة تبدأ من الصفر**

- اختيار العنصر الأول من السلسلة

```
string_var = 'PyQonsole'
print(string_var[0])
```

# P Y T H O N   C O U R S E

## أُسَاسِيَّات لغة البِيْثُون



BY. PYQONSOLE

PYTHON  
Py-Qonsole

PROGRAMMING DEVELOPMENT

## ٥ - ١ السلاسل النصية ( String )

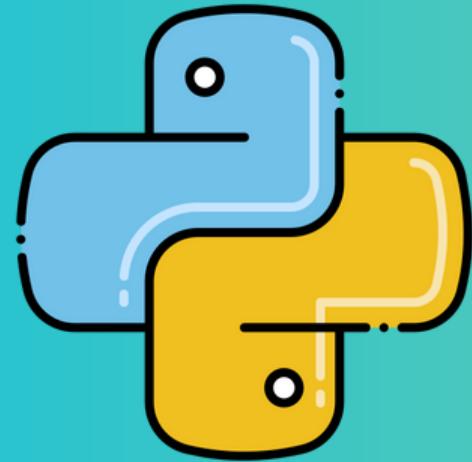
### التعامل مع السلاسل النصية

- طباعة العناصر الموجودة بـ تقنية الفهرسة
- اختيار العنصر الآخر لأن كانت السلسلة معلومة النهاية

```
string_var = 'PyQonsole'
print(string_var[8])
```

# P Y T H O N   C O U R S E

أساسيات لغة البرمجة



BY. PYQONSOLE



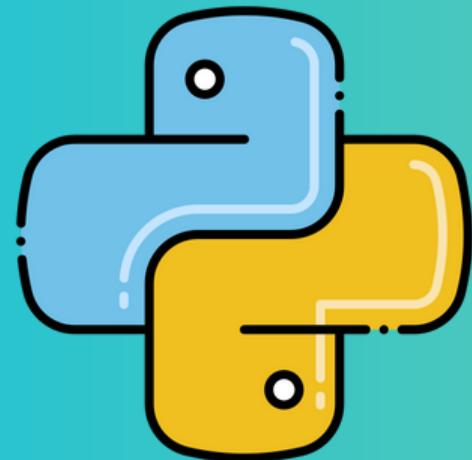
## ( String ) - 1 - 5

التعامل مع السلسلة النصية

- استخراج العناصر الموجودة بـ **تقنية الفهرسة**
- اختيار العنصر الآخر لأن كانت السلسلة مجهولة **النهاية**

```
string_var = 'PyQonsole'
print(string_var[-1])
```

## أساسيات لغة الباتشون



BY. PYQONSOLE

## ١ - ٥ للسلسلة النصية ( String )

### التعامل مع السلسل النصية

- استخراج العناصر الموجود بـ **تقنية الفهرسة**

اختيار عدة عناصر حسب اختيار المبرمج

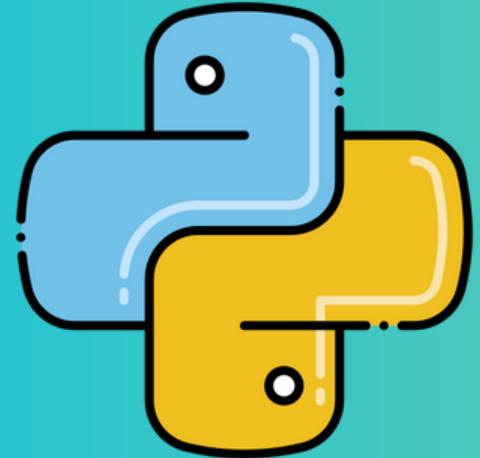
```
print(string_var[0:5])
```

أو

```
print(string_var[5:])
```

# P Y T H O N   C O U R S E

أساسيات لغة البرمجة



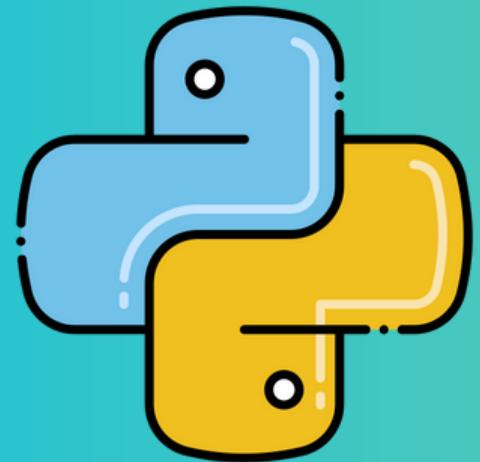
BY. PYQONSOLE

السلسل

2 - القائمة List

# P Y T H O N   C O U R S E

أساسيات لغة البرمجة



BY. PYQONSOLE



## 2 - سلسلة القائمة List

طريقة الكتابة

• قائمة فارغة

```
list_var = []
```

• قائمة أعداد

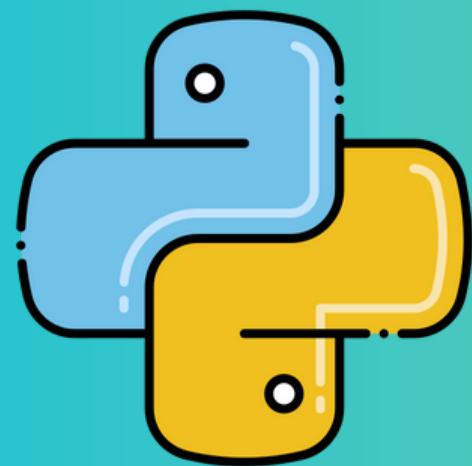
```
list_var = [10, 2, 2025]
```

• قائمة سلسلة نصية

```
list_var = ['10', 'February', '2025']
```

# P Y T H O N   C O U R S E

أساسيات لغة البرمجة



## 2 - سلسلة القائمة List

### طريقة الكتابة

- قائمة مختلفة وعشوائية

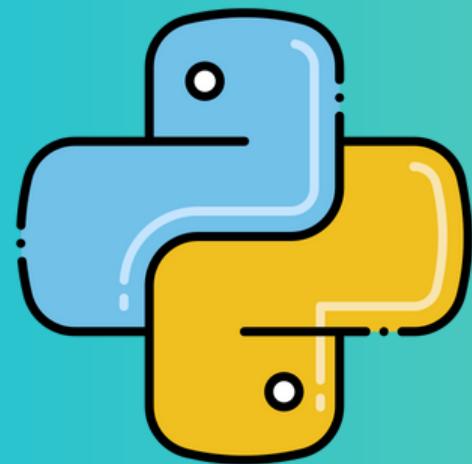
```
list_var = [10, 'February', 20.25]
```

- قائمة بداخل قائمة

```
list_var = [
 ['Subscribe', 'Bro!!']
]
```

# P Y T H O N   C O U R S E

أساسيات لغة البايثون



BY. PYQONSOLE



## 2 - سلسلة القائمة List

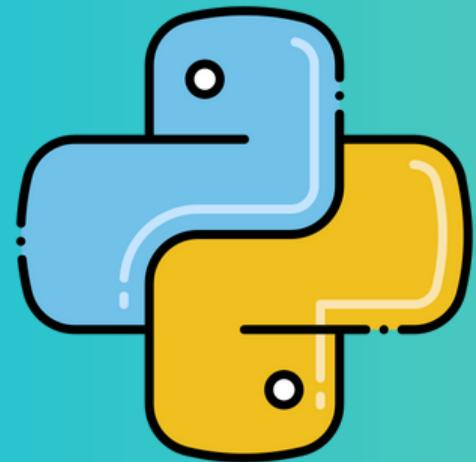
التعامل مع سلسلة القائمة

- استبدال عنصر بعنصر آخر من القائمة حسب اختيار المبرمج

```
list_var = [1, 2, 3]
list_var[0] = 1.5
print(list_var)
output : list_var = [1.5, 2, 3]
```

# P Y T H O N   C O U R S E

أساسيات لغة البايثون



BY. PYQONSOLE

## 2 - سلسلة القائمة List

التعامل مع سلسلة القائمة

- طباعة عناصر القائمة بـتقنية الفهرسة
- طباعة أول عنصر من القائمة

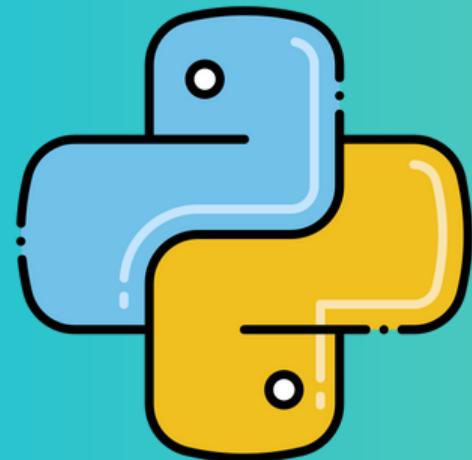
```
print(list_var[0])
```

- طباعة آخر عنصر من القائمة

```
print(list_var[-1])
```

# P Y T H O N   C O U R S E

## رُسَاسِيَّات لغة البايُثُون



BY. PYQONSOLE

PYTHON  
Py-Qonsole

PROGRAMMING DEVELOPMENT

## 2 - سلسلة القائمة List

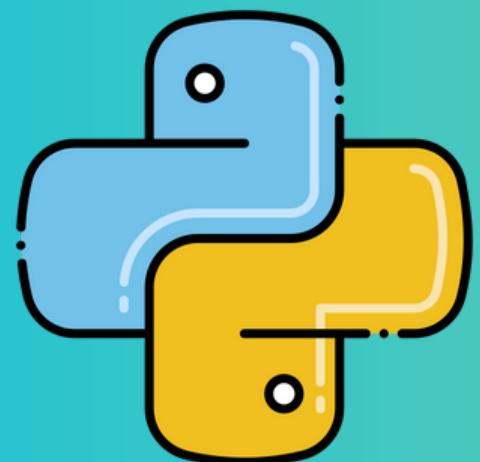
### التعامل مع سلسلة القائمة

- دمج قائمتين معاً باستعمال علامة الجمع

```
list_1 = [1, 2, 3]
list_2 = [4, 5, 6]
print(list_1 + list_2)
```

# P Y T H O N   C O U R S E

أساسيات لغة البايثون



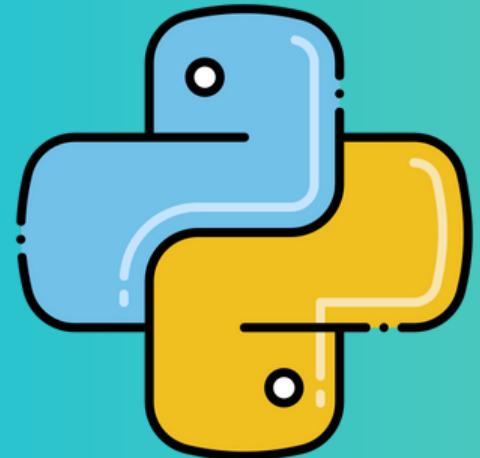
B Y . P Y Q O N S O L E

السلسل

3 - المترابطة البيانات Tuple

# P Y T H O N   C O U R S E

أساسيات لغة البرمجة



## 3 - المترابطة البيانية Tuple

طريقة الكتابة

• مترابطة فارغة

```
tuple_var = ()
```

• مترابطة بيانية تحتوي على أرقام

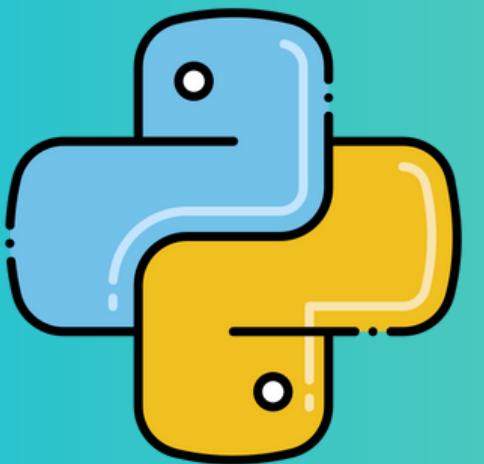
```
tuple_var = (1, 2, 3)
```

• مترابطة بيانية تحتوي على سلاسل نصية

```
tuple_var = ('1', '2', '3')
```

# P Y T H O N   C O U R S E

## أساسيات لغة البايثون



BY. PYQONSOLE

## 3 - المترابطة البيانية Tuple

### طريقة الكتابة

- مترابطة بيانية عشوائية

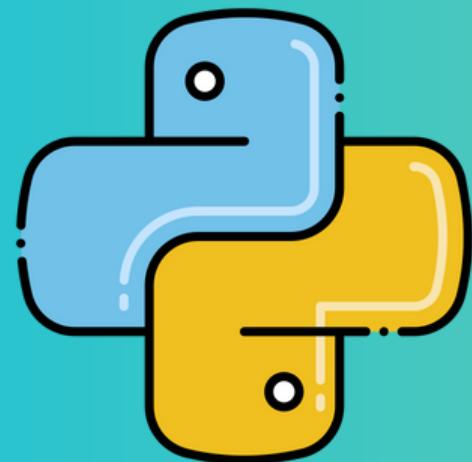
```
tuple_var = ('1', "2", 1.0, 2025)
```

- مترابطة بيانية بداخل المترابطة الأصلية

```
tuple_var = ('1', "2", (1.0, 2025))
```

# P Y T H O N   C O U R S E

## أساسيات لغة البايثون



BY. PYQONSOLE



## 3 - المترابطة البيانية Tuple

### التعامل مع المترابطة بيانية

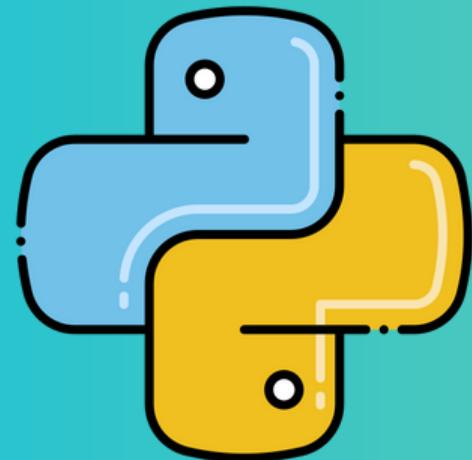
- طباعة عناصر المترابطة البيانية بـ **تقنية الفهرسة**
- طباعة أول عنصر

```
print(tuple_var[0])
```

- طباعة آخر عنصر

```
tuple_var = ('1', "2", (1.0, 2025))
print(tuple_var[-1])
output : (1.0, 2025)
```

## أُسَاسِيَّات لغة البايُثُون



BY. PYQONSOLE

## 3 - المترابطة البيانية Tuple

### التعامل مع المترابطة بيانية

- دمج قائمتين معا بـ استعمال علامة الجمع +

```
tuple_1 = (1, 2, 3)
tuple_2 = (4, 5, 6)
print(tuple_1 + tuple_2)
```

- ملاحظة :

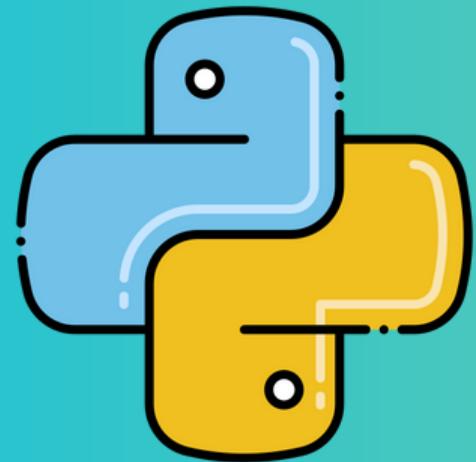
المترابطة البيانية tuple لا تقبل استبدال العناصر

```
tuple_1 = (1, 2, 3)
tuple_1[0] = 2
print(tuple_1)
output : TypeError: 'tuple' object does not support item assignment
```

# P Y T H O N   C O U R S E

السلسل

أساسيات لغة البايثون

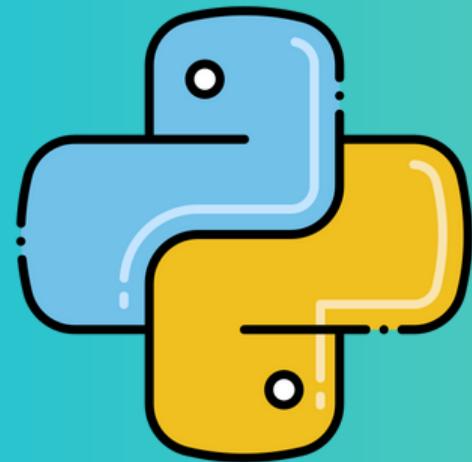


BY. PYQONSOLE

Set - التعريف 4

# P Y T H O N   C O U R S E

أساسيات لغة البرمجة



BY. PYQONSOLE

## 4 - التعريف

طريقة الكتابة

- سلسلة تعريف فارغة

```
set_var = {}
```

- سلسلة تعريف تحتوي على أرقام

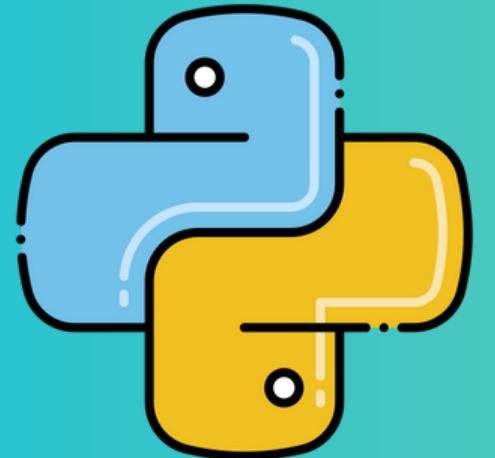
```
set_var = {1, 2, 3}
```

- سلسلة تعريف تحتوي على سلاسل نصية

```
set_var = {'1', "2", "2025"}
```

# P Y T H O N   C O U R S E

## أساسيات لغة البايثون



BY. PYQONSOLE

## Set - التعبيـن 4

### طريقة الكتابة

- سلسلة تعبيـن عشوائـية

```
set_var = {1, "2", 20.25}
```

- ملاحظـة : سلسلة التعبيـن لا تقبل

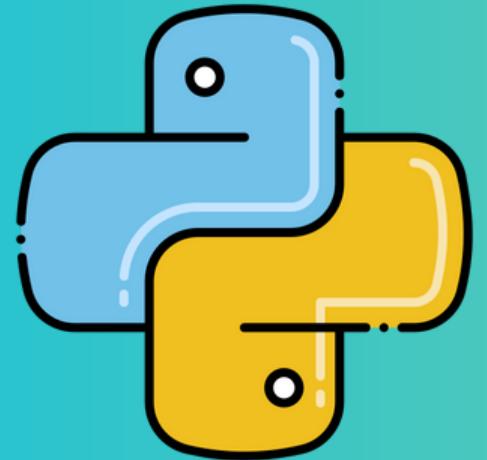
الدـج

- الطـبـاعـة بالفـهـرـسـة

- تواجد سلسلة تعبيـن بـداخـلـها

# P Y T H O N   C O U R S E

رُسَاسِيَّات لغة البايُثُون



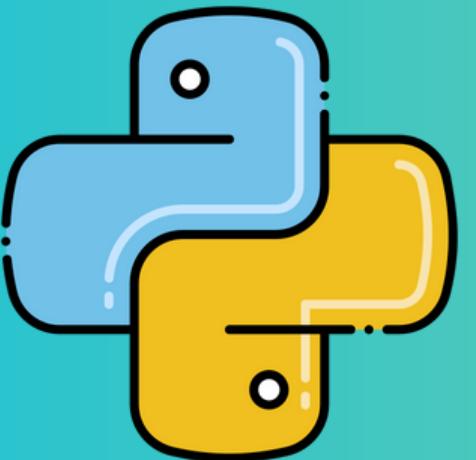
BY. PYQONSOLE



السلسلة

5 - سلسلة القاموس Dictionary

## أساسيات لغة البايثون



## 5 - سلسلة القاموس Dictionary

### طريقة الكتابة

القاموس يحتوي على جزئين

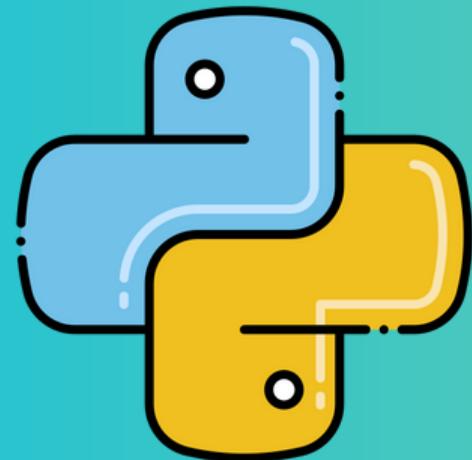
الجزء الأول يسمى المفتاح والجزء الثاني يسمى

القيمة

```
dict_var = {'key': 'value'}
```

# P Y T H O N   C O U R S E

أساسيات لغة البرمجة



BY. PYQONSOLE

## 5 - سلسلة القيم الموسن Dictionary

طريقة الكتابة

• أمثلة

```
dict_var = {0: 10}
```

```
dict_var = {0: '10'}
```

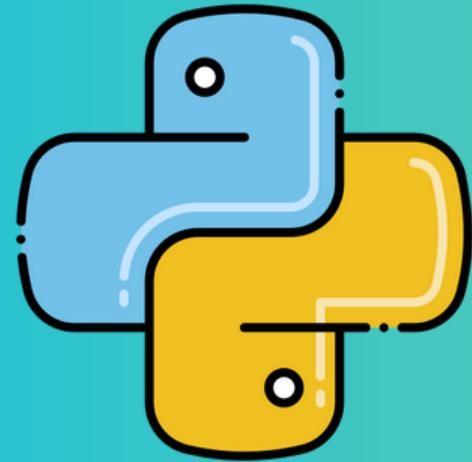
```
dict_var = {'0': '10'}
```

```
dict_var = {'0': [0, "2", '2025']}
```

```
dict_var = {(2, 5): {20, 20}}
```

# P Y T H O N   C O U R S E

أساسيات لغة البرمجة



BY. PYQONSOLE

## 5 - سلسلة القاموس Dictionary

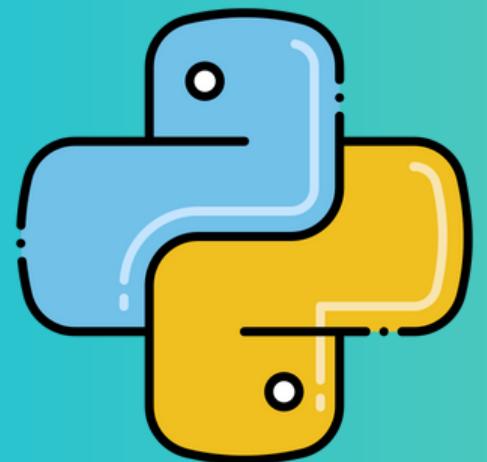
طباعة قيم القاموس بفهرسة المفتاح

• أمثلة

```
dict_var = {0: 10, 2: 20}
print(dict_var[0])
output : 10
print(dict_var[2])
output : 20
```

```
dict_var = {'My-Youtube-Channel': 'PyQonsole'}
print(dict_var['My-Youtube-Channel'])
output : PyQonsole
```

## أساسيات لغة البايثون



## 5 - سلسلة القاموس Dictionary

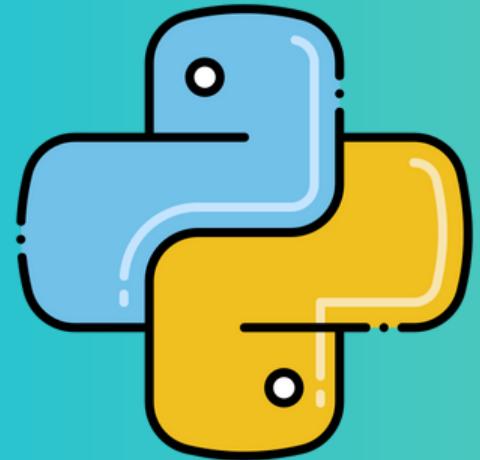
التعامل مع القاموس

استبدال قيمة مفتاح معين بقيمة أخرى

```
dict_var = {0: 10, 2: 20}
dict_var[0] = 'PyQonsole'
print(dict_var[0])
output : 'PyQonsole'
```

# P Y T H O N   C O U R S E

أَسْاسِيَّات لغة البايُثُون



BY. PYQONSOLE

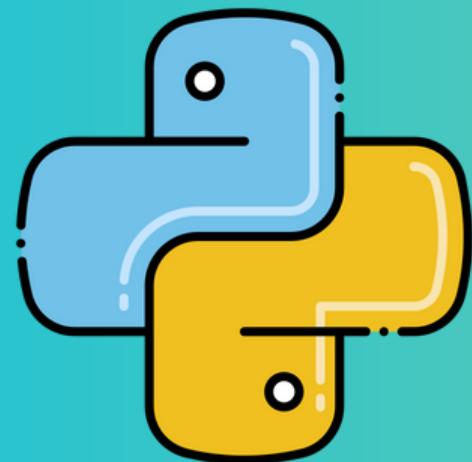


السلسل

6 - العمليات المنطقية و الجمل الشرطية

# PYTHON COURSE

أساسيات لغة البايثون



BY. PYQONSOLE

## 6 - العمليات المنطقية و الجمل الشرطية

### • العمليات المنطقية

المتغير المنطقي يحتمل قيمة واحدة فاما صحيح او خطأ

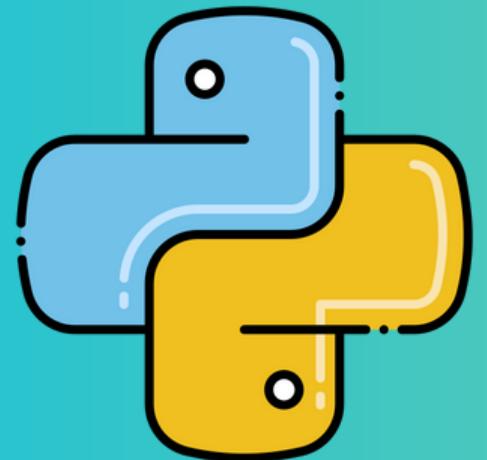
طريقة الكتابة

```
bool_var = True
```

```
bool_var = False
```

# PYTHON COURSE

أساسيات لغة البرمجة



BY. PYQONSOLE

## 6 - العمليات المنطقية و الجمل الشرطية

### • العمليات المنطقية

العمليات المنطقية تكون باستخدام علامات خاصة  
سنوضحها في أمثلة

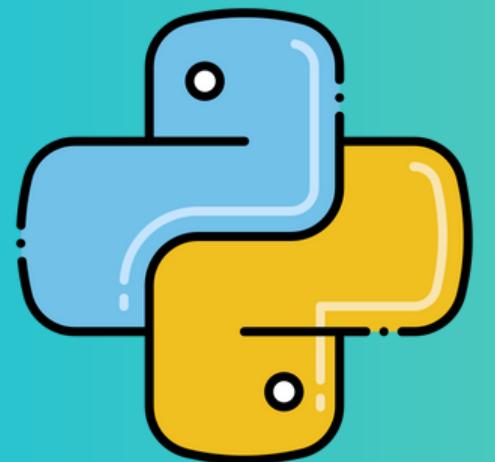
### • علامة (<) أقل من

```
print(0 < 1)
output : True
```

نتيجة هذه العملية صحيحة لأن الصفر أقل من الواحد

# PYTHON COURSE

أساسيات لغة البايثون



BY. PYQONSOLE

## 6 - العمليات المنطقية و الجمل الشرطية

### • العمليات المنطقية

#### • علامة (>) أكير من

نتيجة هذه العملية خاطئة لأن الصفر ليس أكبر من الواحد و الواحد أكبر من الصفر

```
print(0 > 1)
output : False
```

#### • علامة (==) تساوي

نتيجة هذه العملية خاطئة لأن الصفر لا يساوي الواحد

```
print(0 == 1)
output : False
```

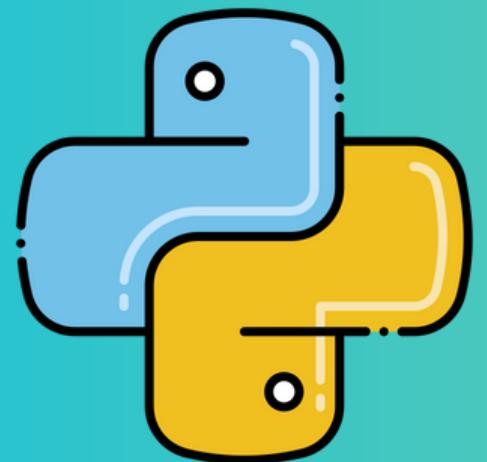
#### • علامة (!=) يخالف

نتيجة هذه العملية صحيحة لأن الصفر يخالف الواحد

```
print(0 != 1)
output : True
```

# PYTHON COURSE

أساسيات لغة الباتشون



## 6 - العمليات المترافقية و الجمل الشرطية

### • العمليات المترافقية

#### • علامة ( $\geq$ ) أكبر أو تساوي

```
print(0 >= 1)
output : False
```

نتيجة هذه العملية خاطئة لأن الصفر أقل من الواحد وكذلك لا يساوي الواحد

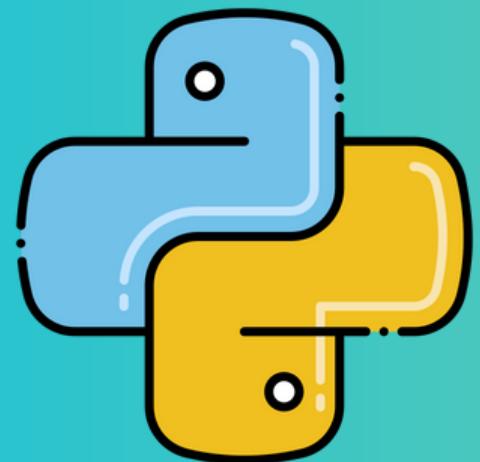
#### • علامة ( $\leq$ ) أقل أو تساوي

```
print(0 <= 1)
output : True
```

نتيجة هذه العملية صحيحة لأن الصفر أقل من الواحد وكذلك الصفر لا يساوي الواحد

# PYTHON COURSE

## أساسيات لغة البايثون



BY. PYQONSOLE



## 6 - العمليات المنطقية و الجمل الشرطية

### • العمليات المنطقية

### • مثال مع الشرح

```
print(1 < (2.5 >= 3))
output : False
```

عند تواجد الأقواس تتم العملية ابتداءاً من الأقواس ثم تنتقل إلى خارجها في هذا المثال تبدأ العملية من ( $3 \geq 2.5$ ) و النتيجة True أي أنها تصبح

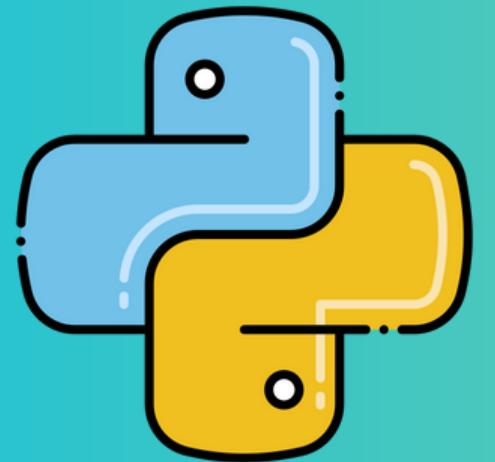
$1 < \text{True}$

والواحد ليس أقل من صحيح بل الواحد صحيح أي عند كتابة

```
print(bool(1))
output : True
```

# PYTHON COURSE

أساسيات لغة البايثون



## 6 - العمليات المنطقية و الجمل الشرطية

### • الجمل الشرطية

الجمل الشرطية تستخدم عند توفر عدة احتمالات في الكود تشرط تنفيذ أمر واحد أو عدة أوامر

طريقة الكتابة

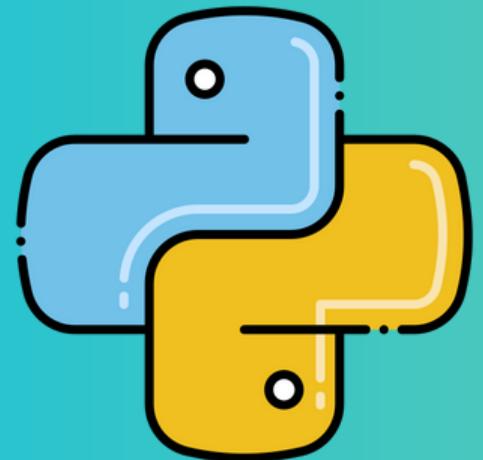
تكتب الدالة الشرطية ابتداء ب الكلمة **If**

```
if 1:
```

و في آخر السطر يجب وضع نقطتين فوق بعضه والانتقال إلى سطر جديد تاركًا أربع مسافات بيضاء

# PYTHON COURSE

أساسيات لغة البايثون



BY. PYQONSOLE

## 6 - العمليات المنطقية و الجمل الشرطية

### • الجمل الشرطية

و نستطيع استخدام جميع علامات العمليات  
المنطقية عند كتابة جملة شرطية

أقل من

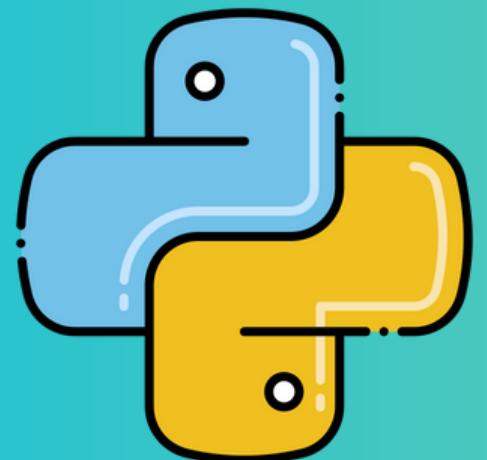
```
if 1 < 2:
```

أكبر من

```
if 1 > 2:
```

# PYTHON COURSE

أساسيات لغة البايثون



BY. PYQONSOLE



## 6 - العمليات المتنطقية و الجمل الشرطية

### • الجمل الشرطية

if 1 == 2:

يساوي

if 1 != 2:

يختلف

if 1 >= 2:

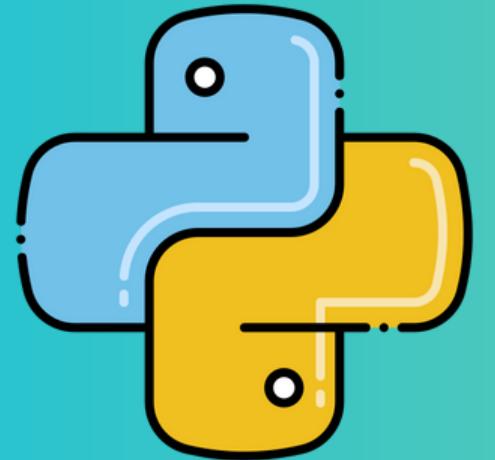
أكبر أو يساوي

if 1 <= 2:

أقل أو يساوي

# P Y T H O N   C O U R S E

أساسيات لغة البرمجة



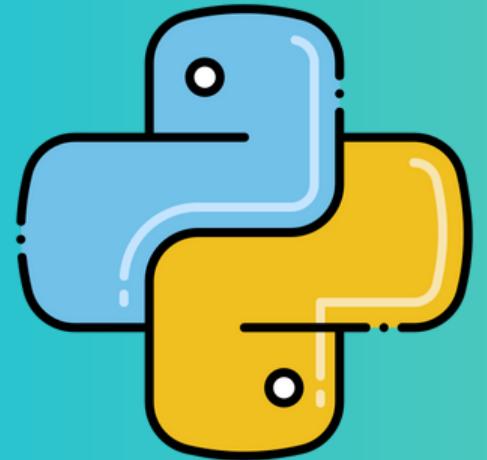
BY. PYQONSOLE

السلسل

7 - الحلقات التكرارية

# P Y T H O N   C O U R S E

## أساسيات لغة البايثون



## 7 - الحلقات التكرارية

أحياناً يحتاج المبرمج لتقنية التكرار التلقائي تفادياً لكتابة الكود عدة مرات  
تنقسم الحلقات التكرارية إلى نوعين

- حلقة تكرارية معلومة البداية و معلومة النهاية
- حلقة تكرارية معلومة البداية و غير معلومة النهاية

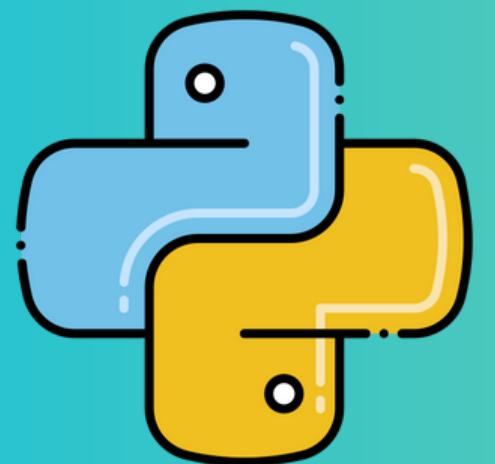
### مثال



نسبة بطارية الهاتف الذكي تمثل حلقة تكرارية معلومة البداية و معلومة النهاية نفرض أن نسبة البطارية 100% و الهاتف مفصول عن الشاحن فالحلقة ستكون تنازلية إلى بلوغ حد يستدعي شحن البطارية فأن لم يتتوفر الشحن ستبقى الحلقة شغاله إلى غاية نسبة الصفر التي تستدعي نهاية الحلقة

# P Y T H O N   C O U R S E

أُسَاسِيَّات لغة البايُثُون



BY. PYQONSOLE

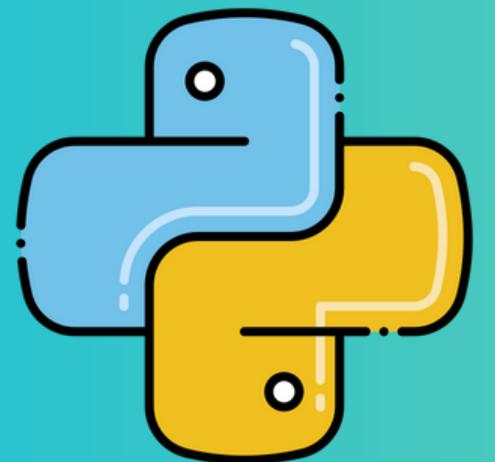
## 7 - الحلقات التكرارية



مثال  
ساعة الحائط المترددة تشتعل  
بموج الحلقة التكرارية معلومة  
البداية و غير معلومة النهاية  
لأنها لا تحتوي على أي مؤشر  
يوضح نسبة لبطارية الساعة

# P Y T H O N   C O U R S E

أُسَاسِيَّات لغة البايثون



## 7 - الحلقات التكرارية

- الحلقة التكرارية معلومة البداية و معلومة النهاية

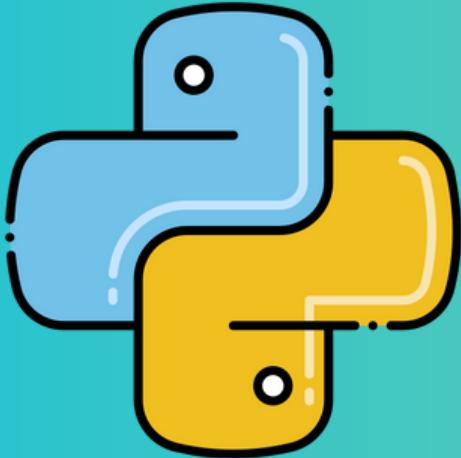
for .1  
while .2

- الحلقة التكرارية **for** طريقة الكتابة

لاستدعاء الحلقة التكرارية نكتب كلمة **for** أول السطر ثم نكتب اسم متغير جديد ثم نكتب **in** ثم اسم المتغير المراد تكراره

# PYTHON COURSE

أساسيات لغة البايثون



## 7 - الحلقات التكرارية

### طريقة الكتابة

• مثال

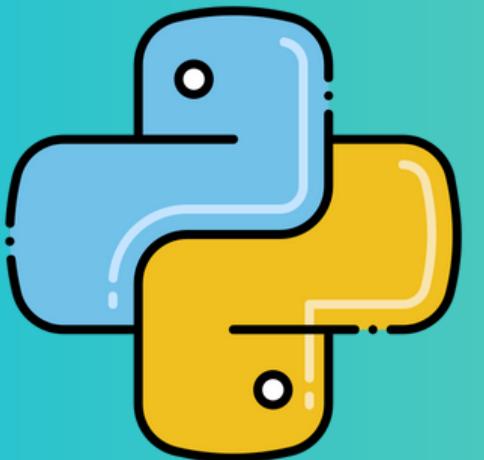
```
loops = 'loops'
for loop in loops:
 print(loop)
```

l  
o  
o  
p  
s

في هذا المثال ستتم  
طباعة loops مجزأة  
وكل جزء على حدي

# PYTHON COURSE

## أسسات لغة البايثون



## 7 - الحلقات التكرارية

- ملاحظة : الحلقة التكرارية تنفذ على المتغيرات التي تقبل التجزئة كالسلسل و لا تقبل التكرار . رقم صحيح الا عند استدعاء دالة تسمى range .

- مثال

```
for loop in range(0, 5):
 print(loop)
```

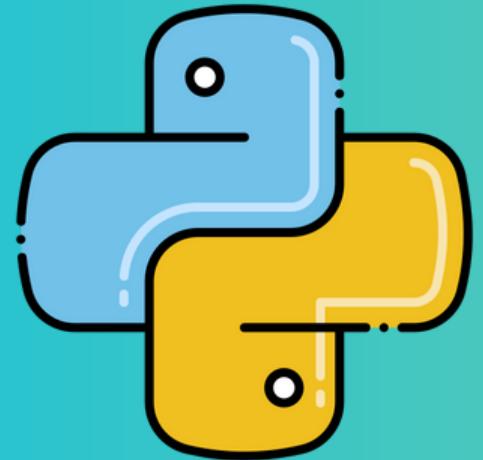
0  
1  
2  
3  
4

في هذا المثال ستم طباعة 5  
أجزاء ابتداء من الصفر الى  
الأربعة

- ملاحظة : كلمة range محجوزة في لغة البايثون و لا تحتاج لاي أكواد لاستدعائها و ستطرق لها في وقت لاحق

# P Y T H O N   C O U R S E

أساسيات لغة البايثون



BY. PYQONSOLE

## 7 - الحلقات التكرارية

### • الحلقة التكرارية while

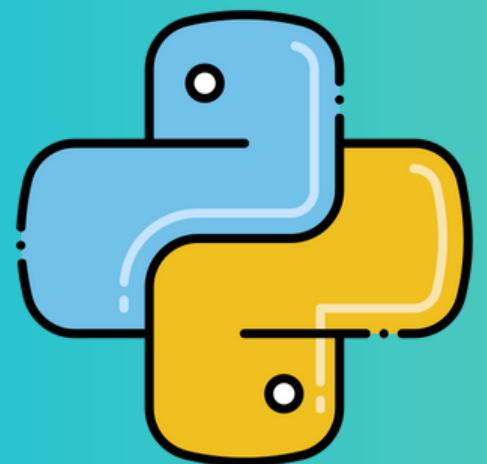
#### طريقة الكتابة

لاستدعاء الحلقة التكرارية نكتب **while** أول السطر ثم نكتب شرط ل بدأية الحلقة و وضع حد لايقاف الحلقة و كذلك لا تنسى النقطتين فوق بعض و ترك أربع مسافات في السطر الجديد

```
while True:
```

# P Y T H O N   C O U R S E

## أُسَاسِيَّات لغة البايُثُون



BY. PYQONSOLE

## 7 - الحلقات التكرارية

- في المثال التالي طريقة تعمل على إيقاف الشرط لأن كانت الحلقة تحمل بيانات رقمية

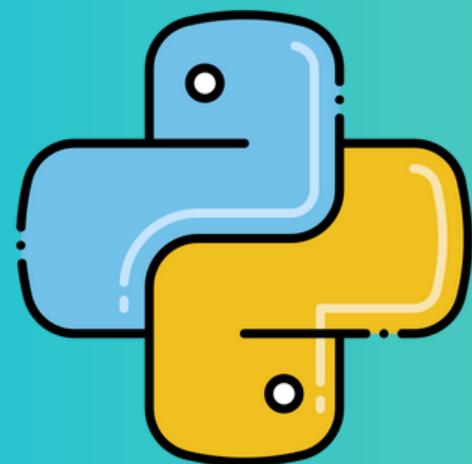
```
x = 1
x = x + 1 # x = 1 + 1
print(x) # output : 2
```

في هذا المثال أعطينا قيمة للx بـ 1 ثم فمنا  
بإضافة 1 للx بالعبارة  $x = x + 1$  و في لغة  
البايُثُون نستطيع اختصارها بالعبارة  $x += 1$  و  
الناتج واحد

```
x = 1
x += 1 # x = 1 + 1
print(x) # output : 2
```

# P Y T H O N   C O U R S E

## رُسَاسِيَّات لغة البايُثُون



BY. PYQONSOLE



## 7 - الحلقات التكرارية

### • مثال

```
limit = 5
while 1 < limit:
 print('PyQonsole')
 limit -= 1
```

PyQonsole  
PyQonsole  
PyQonsole  
PyQonsole

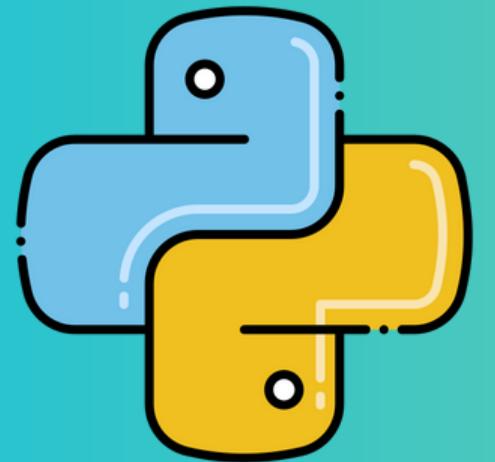
في هذا المثال ستتم طباعة  
أربع مرات إلى غاية  
توقف الحلقة

• ملاحظة : عبارة `limit -= 1` هي المساعدة  
على ايقاف الحلقة . تكتب هكذا للاختصار  
فقط . و هي تعني

`limit = limit - 1`

# PYTHON COURSE

## أساسيات لغة البايثون



## 7 - الحلقات التكرارية

- الحلقة التكرارية معلومة البداية و غير معلومة النهاية
- تنفذ على while فقط

### طريقة الكتابة

```
while True:
 print("it's true !!")
```

في هذا المثال ستم طباعة it's true دون توقف و يتطلب إيقافها تدخل بالكمبيورد بضغط زر ctrl + c