

AUGUST 25 – 31, 2019



## المبادرة السعودية للمطورين

تعلم .. فكر .. حاول .. أبداع

## المبادرة السعودية للمطورين

مسار Python

مشرفي المسار:

عبدالله عوده – انتصار النصار – رؤى كردي – ليلى المصعبي



## ملاحظات قبل بدء الدروس:

- على المتدربين نشر كل يوم الجزئية التي تم كتابتها من النص البرمجي في الـ **Github** تحت **Topic** بعنوان **saudidev.org** كما تم توضيحه في دروس الـ **Github** سابقاً

- على المتدربين نشر كل يوم مقدار التقدم وصورة لما تم تعلمه وتطبيقه على **Twitter** تحت الهاشتاقات:  
#المبادرة\_السعودية\_للمطورين  
\_100#يوم\_برمجة  
#100DaysOfCode

تمنياتنا لك بالتوفيق  
المبادرة السعودية للمطورين

# اليوم السادس

# التحويل من نوع إلى نوع آخر في لغة البايثون

## Python Casting

- Casting in **python** is therefore done using constructor functions:

هذه الدالة موجودة جاهزة في بايثون، تقوم بتحويل القيمة التي نمررها لها، إلى عدد صحيح، القيمة التي يتم تمريرها تكون عبارة عن (عدد) أو (نص يمثل عدد صحيح)، أي يجب أن تمثل قيمة عددية.

**int()** - constructs an integer number from an integer literal, a float literal (by rounding down to the previous whole number), or a string literal (providing the string represents a whole number)

هذه الدالة موجودة جاهزة في بايثون، تقوم بتحويل القيمة الممررة لها، إلى عدد عشري، القيمة التي يتم تمريرها تكون عبارة عن (عدد) أو (نص يمثل عدد)، أي يجب أن تمثل قيمة عددية.

**float()** - constructs a float number from an integer literal, a float literal or a string literal (providing the string represents a float or an integer)

هذه الدالة موجودة جاهزة في بايثون، تقوم بتحويل القيمة التي يتم تمريرها لها من جميع الأنواع ، إلى نص/ سلسلة نصية.

**str()** - constructs a string from a wide variety of data types, including strings, integer literals and float literals

## مثال للتحويل إلى عدد صحيح

### Example

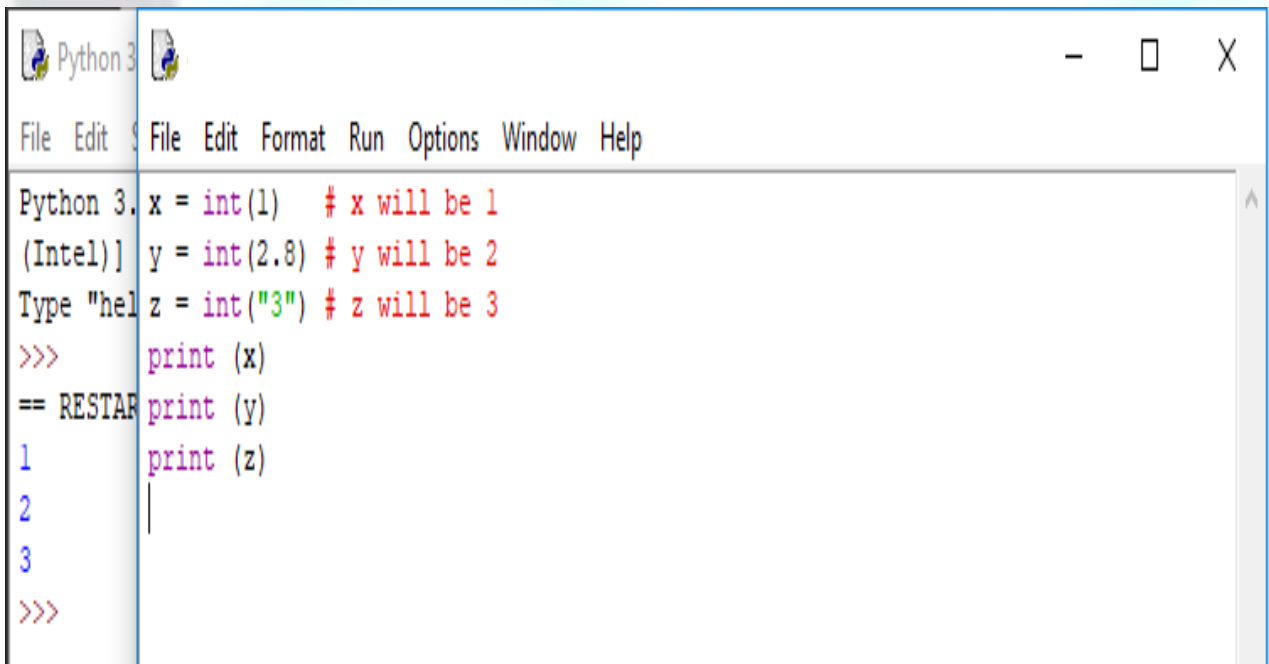
Integers:

```
x = int(1)    # x will be 1
y = int(2.8)  # y will be 2
z = int("3")  # z will be 3
```

تكتب النوع الذي تريد التحويل إليه ثم قوسين () وبداخل القوسين تضع القيمة المراد تحويلها.

### Run the example

نتيجة تشغيل الكود



```
Python 3.6.4 Shell
File Edit Shell File Edit Format Run Options Window Help
Python 3.6.4 Shell
(Intel) x = int(1) # x will be 1
Type "help()" for more
>>> print(x)
1
>>> y = int(2.8) # y will be 2
>>> print(y)
2
>>> z = int("3") # z will be 3
>>> print(z)
3
>>>
```

## مثال للتحويل إلى عدد عشري

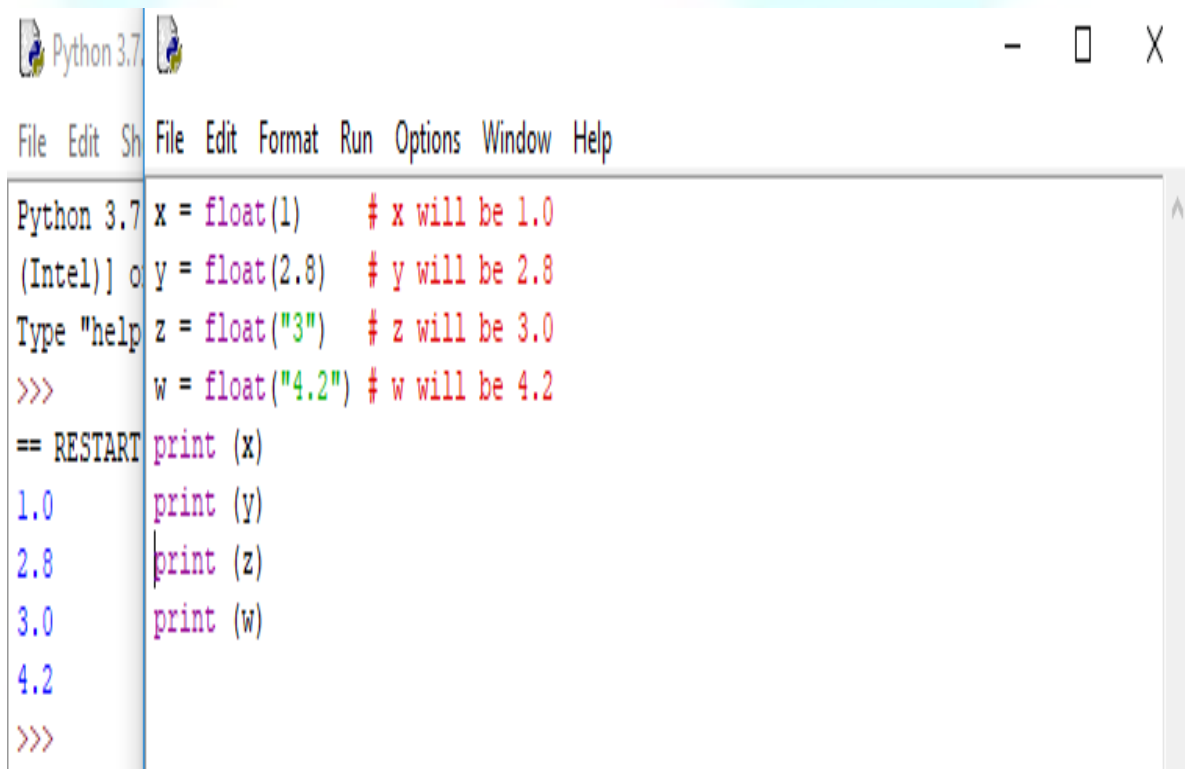
### Example

Floats:

```
x = float(1)      # x will be 1.0
y = float(2.8)    # y will be 2.8
z = float("3")    # z will be 3.0
w = float("4.2")  # w will be 4.2
```

## نتيجة تشغيل الكود

### Run the example



```
Python 3.7
File Edit Shell File Edit Format Run Options Window Help
Python 3.7 x = float(1)      # x will be 1.0
(Intel) o y = float(2.8)    # y will be 2.8
Type "help z = float("3")  # z will be 3.0
>>> w = float("4.2") # w will be 4.2
== RESTART print (x)
1.0 print (y)
2.8 print (z)
3.0 print (w)
4.2
>>>
```

## مثال للتحويل إلى نص

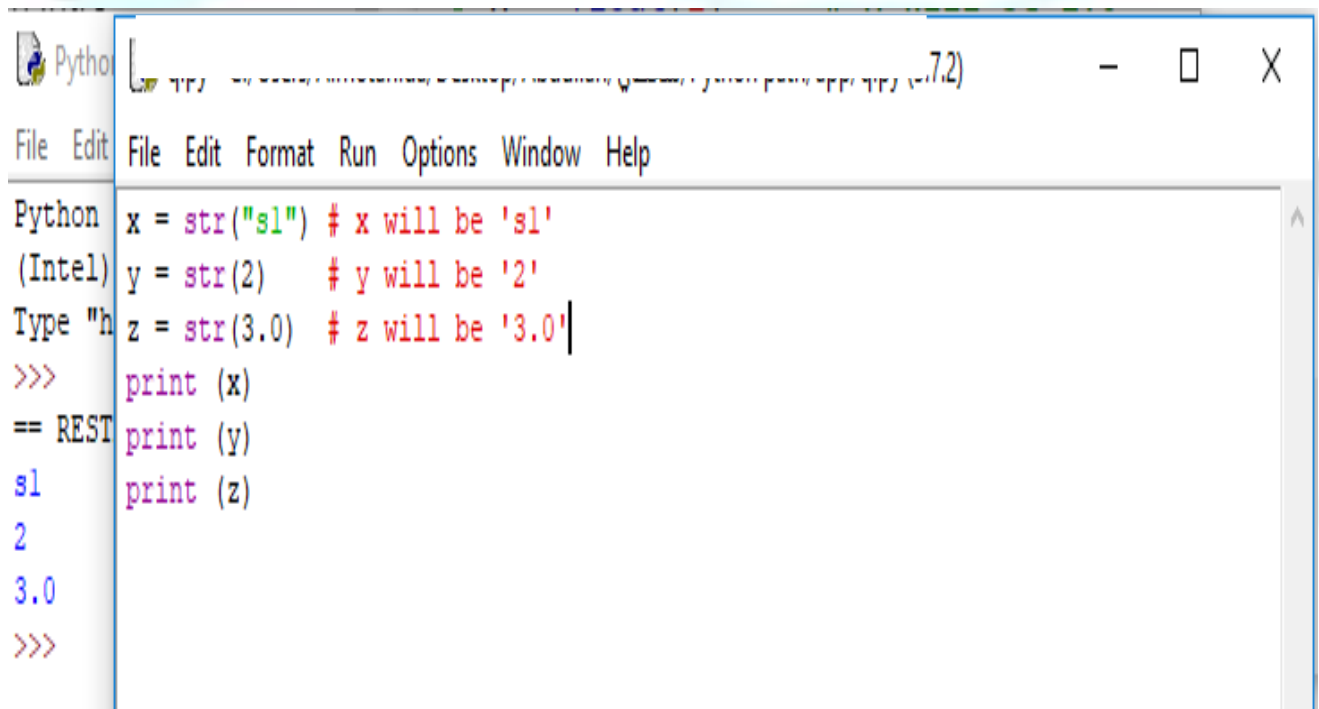
### Example

Strings:

```
x = str("s1") # x will be 's1'
y = str(2)    # y will be '2'
z = str(3.0)  # z will be '3.0'
```

نتيجة تشغيل الكود

### Run the example



```
Python 7.2
File Edit Format Run Options Window Help
Python x = str("s1") # x will be 's1'
(Intel) y = str(2)    # y will be '2'
Type "h z = str(3.0)  # z will be '3.0'|
>>> print (x)
== REST print (y)
s1 print (z)
2
3.0
>>>
```

تهانينا!  
أتممت درسك

روابط قد تفيدك

See more ..

- [Easy Tutorial Explaining Casting In Python](#)
- [Video Lesson - Casting in Python](#)

طبق ما تعلمته معنا في درسنا  
ولا تنسى مشاركتنا أكوادك



# اليوم السابع

# السلاسل النصية / النصوص في لغة البايثون

## Python Strings

لكتابة نص في لغة بايثون نستخدم علامة الاقتباس الفردية ' أو المزدوجة "

- String literals in python are surrounded by either single quotation marks, or double quotation marks.

'hello' is the same as "hello"

**print()**  
لعرض/طباعة القيمة النصية عن طريق دالة الطباعة

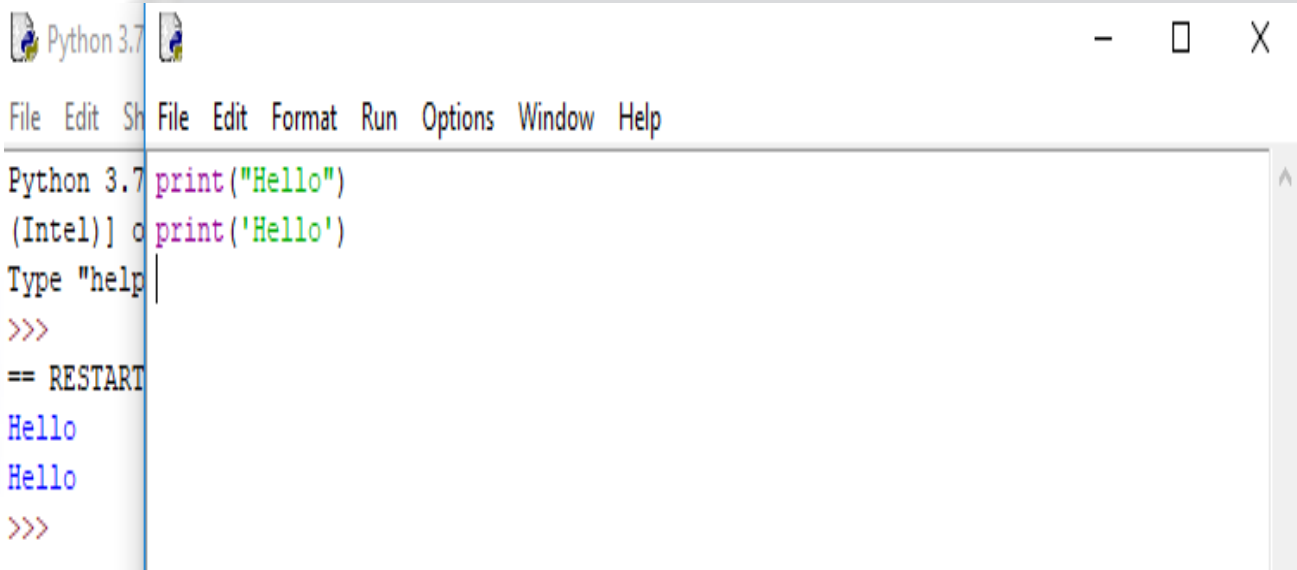
You can display a string literal with the **print()** function

### Example

```
print("Hello")  
print('Hello')
```

## نتيجة تشغيل الكود

## Run the example



```
Python 3.7
File Edit Shell File Edit Format Run Options Window Help
Python 3.7 print("Hello")
(Intel)] o print('Hello')
Type "help"
>>>
== RESTART ==
Hello
Hello
>>>
```

## ➤ Assign String to a Variable

## تعريف متغير يخزن فيه نص

لتخزين القيم النصية اكتب أولاً اسم المتغير متبوعاً بالعلامة = ثم اكتب النص أو الجملة بعدها.

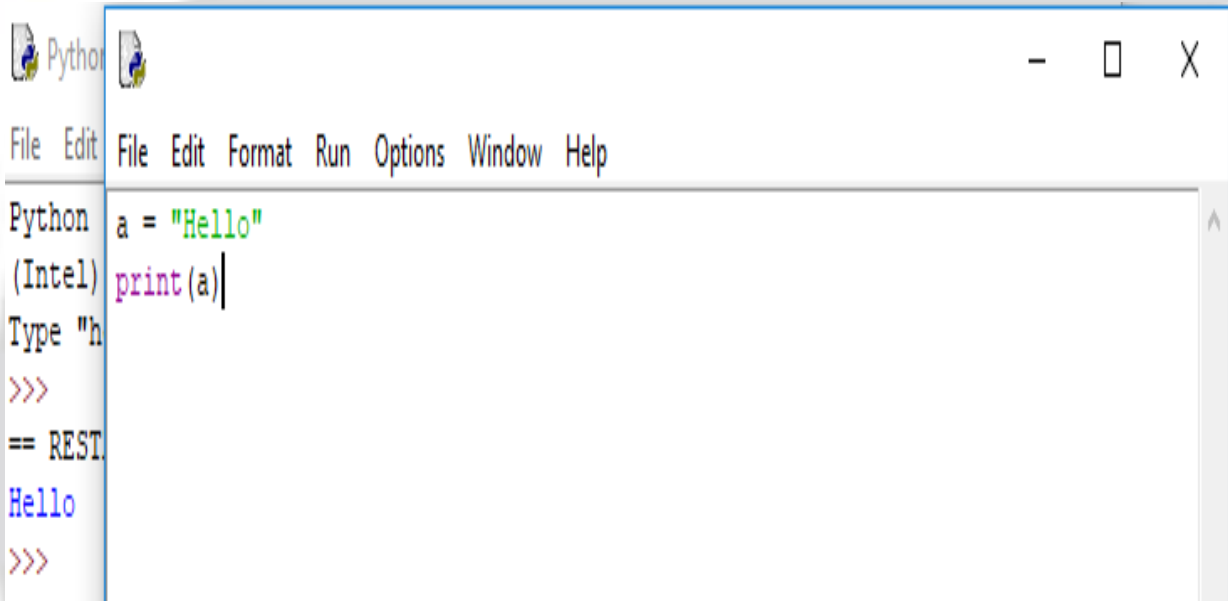
Assigning a string to a variable is done with the variable name followed by an equal sign and the string.

## Example

```
a = "Hello"
print(a)
```

## Run the example

## نتيجة تشغيل الكود



```
Python
File Edit Format Run Options Window Help
Python a = "Hello"
(Intel) print(a)
Type "h
>>>
== REST.
Hello
>>>
```

## ➤ Multiline Strings

## تعريف نص يتألف من عدة أسطر

باستخدام علامة الاقتباس الثلاثية يمكنك تعريف/كتابة نص كبير يتألف من عدة سطور

You can assign a multiline string to a variable by using three quotes.

## Example

يمكنك استخدام علامة الاقتباس الثلاثية المزدوجة `"""` أو الثلاثية المفردة `'''`

You can use three double quotes (`"""`) Or three single quotes (`'''`).

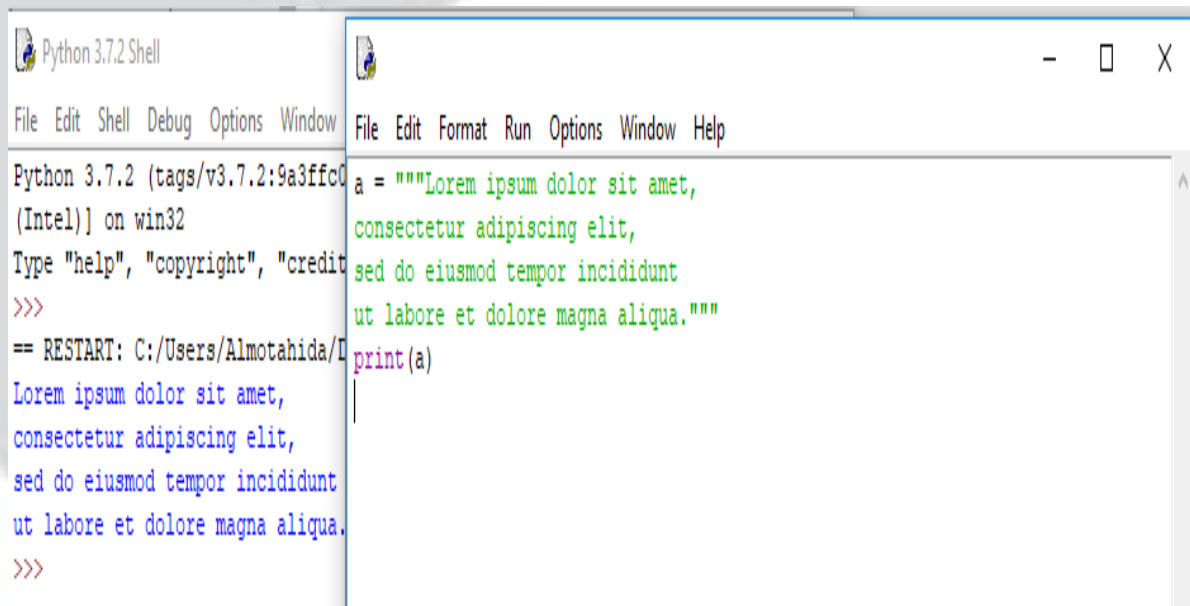
(you can write whatever you want between the quotes).

```
a = """Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua."""
print(a)
```

المثال التالي يوضح لك الفكرة

## Run the example

نتيجة تشغيل الكود



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window
Python 3.7.2 (tags/v3.7.2:9a3ffc0, Dec 11 2019) on win32
Type "help", "copyright", "credits()" or "quit()" for more
>>>
== RESTART: C:/Users/Almotahida/Desktop >>>
Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.
>>>
```

السلاسل النصية هي مصفوفات

## ➤ Strings are Arrays

السلاسل النصية في بايثون تعرف بأنها مصفوفة من الحروف.

النوع string يعتمد على نظام Unicode وهذا يعني أنك تستطيع التعامل مع النصوص العربية والانجليزية والفرنسية .. إلخ بدون مشاكل.

Strings in **Python** are arrays of bytes representing Unicode characters.

يتم تخزين النصوص في الذاكرة حرفاً حرفاً، و للوصول لأحرف النص نستخدم الأقواس المربعة [ ]

**Python** does not have a character data type, a single character is simply a string with a length of 1. Square brackets can be used to access elements of the string.

في هذا المثال نريد الوصول إلى الحرف الثاني من السلسلة النصية **Hello World**

## Example

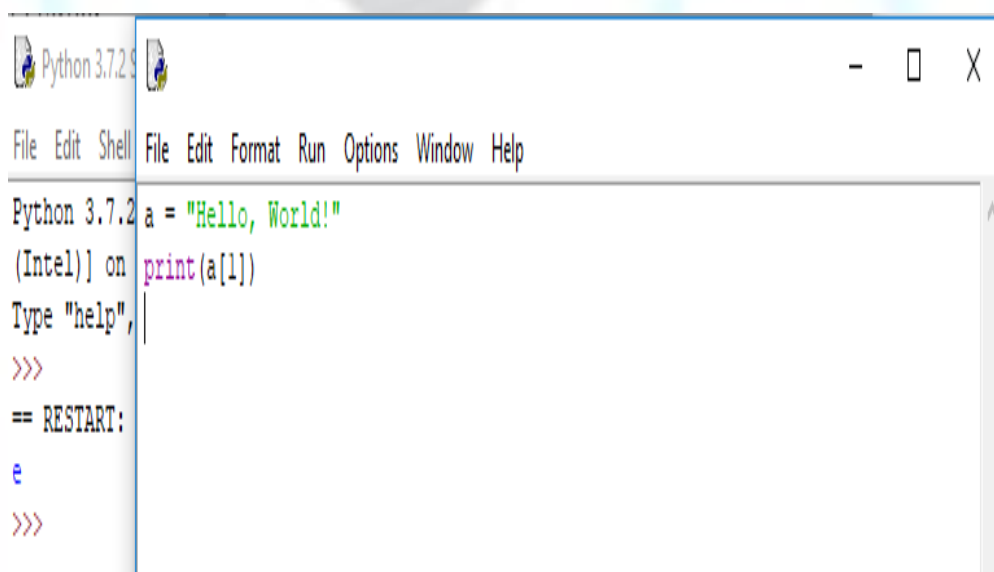
Get the character at position 1

(remember that the first character has the position 0). ترقيم المصفوفة في البايثون يبدأ من صفر

```
a = "Hello, World!"  
print(a[1])
```

## Run the example

نتيجة تشغيل الكود



```
Python 3.7.2 Shell
File Edit Shell File Edit Format Run Options Window Help
Python 3.7.2 a = "Hello, World!"
(Intel)] on print(a[1])
Type "help",
>>>
== RESTART:
e
>>>
```

تجزئ النص، لأخذ أي جزء من النص اعتمد على أرقام الخانات لكل حرف في السلسلة النصية.

في المثال التالي نريد الجزء **llo** من النص المخزن في المتغير **b**

## Example

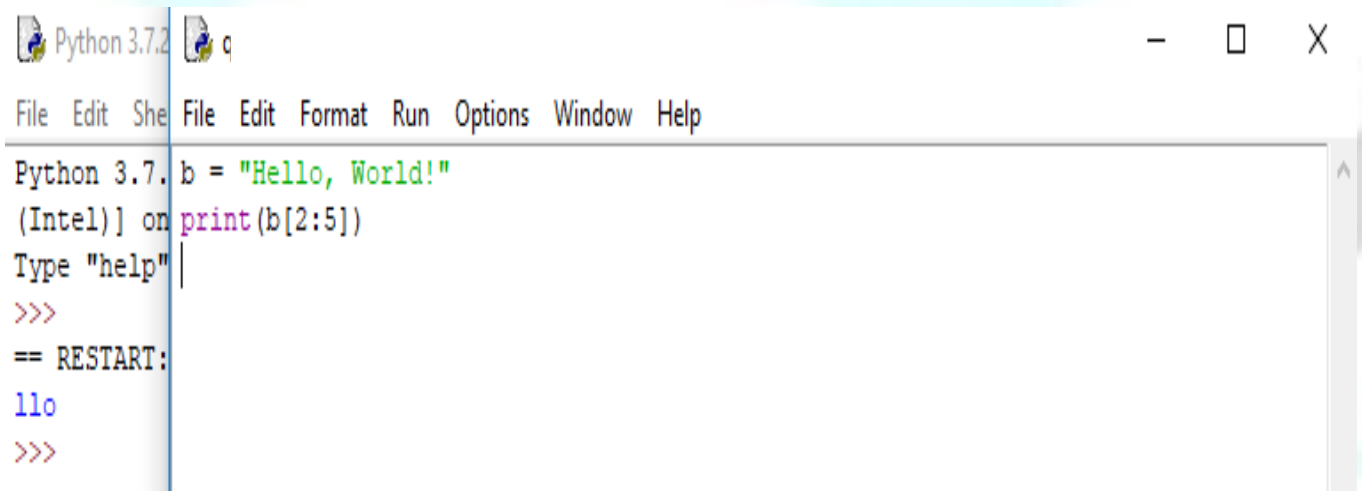
Substring. Get the characters from position 2 to position 5 (not included)

```
b = "Hello, World!"  
print(b[2:5])
```

[ 2:5 ] تعني أنك تريد جميع الأحرف الموجودة من خانة رقم ٢ وصولاً إلى ما قبل خانة ٥

## Run the example

نتيجة تشغيل الكود



```
Python 3.7.2 [q]  
File Edit Shell File Edit Format Run Options Window Help  
Python 3.7.1 b = "Hello, World!"  
(Intel)] on print(b[2:5])  
Type "help" |  
>>>  
== RESTART:  
llo  
>>>
```

لقد أتممت درسك بنجاح!

روابط قد تساعدك

For more information See below:

- [السلسلة النصية Python tutorial || Strings indexed](#)

طبّق ما تعلمته معنا في درسنا

و شاركنا أكوادك



# اليوم الثامن

# السلاسل النصية / النصوص في لغة البايثون

## Python Strings 2

بايثون تزودنا بكثير من الدوال التي تتعامل مع السلاسل النصية  
سنناقش بعضها في هذا الدرس

لحذف أي مسافات فارغة في بداية ونهاية السلسلة النصية استخدم الدالة `strip()`

The `strip()` method removes any whitespace from the beginning or the end.

## Example

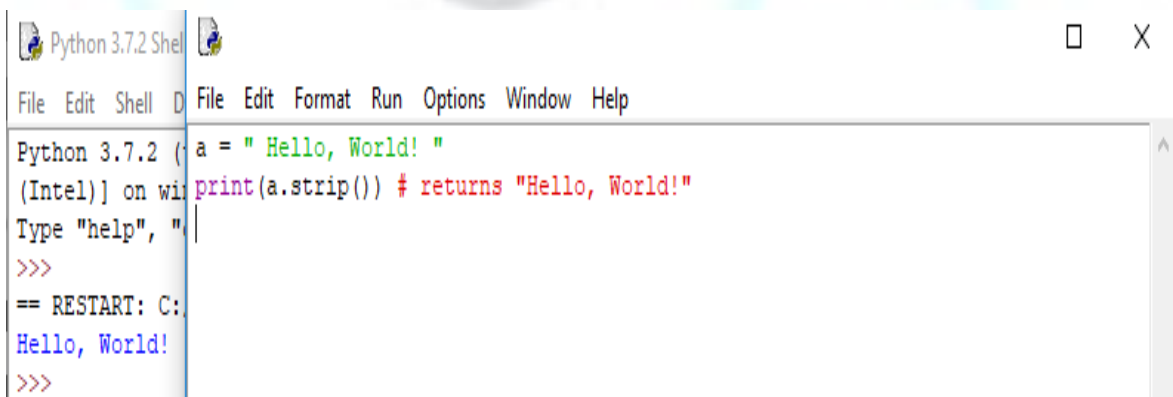
نقوم بكتابة اسم المتغير المخزن فيه النص

في مثالنا هنا `a` ثم نضع نقطة . بعدها نستدعي الدالة `strip()`

```
a = " Hello, World! "  
print(a.strip()) # returns "Hello, World!"
```

## Run the example

نتيجة تشغيل الكود



```
Python 3.7.2 Shell  
File Edit Shell D File Edit Format Run Options Window Help  
Python 3.7.2 (tags/3.7.2:20200202) [AMD64] on win32  
> a = " Hello, World! "  
> print(a.strip()) # returns "Hello, World!"  
>  
>>>  
== RESTART: C:\Python37\Shell ==  
> a = " Hello, World! "  
> print(a.strip()) # returns "Hello, World!"  
>  
>>>  
Hello, World!  
>>>
```

لمعرفة عدد أحرف السلسلة النصية استخدم الدالة `len()`  
المسافات الفارغة يتم حسابها أيضا.

The `len()` method returns the length of a string.

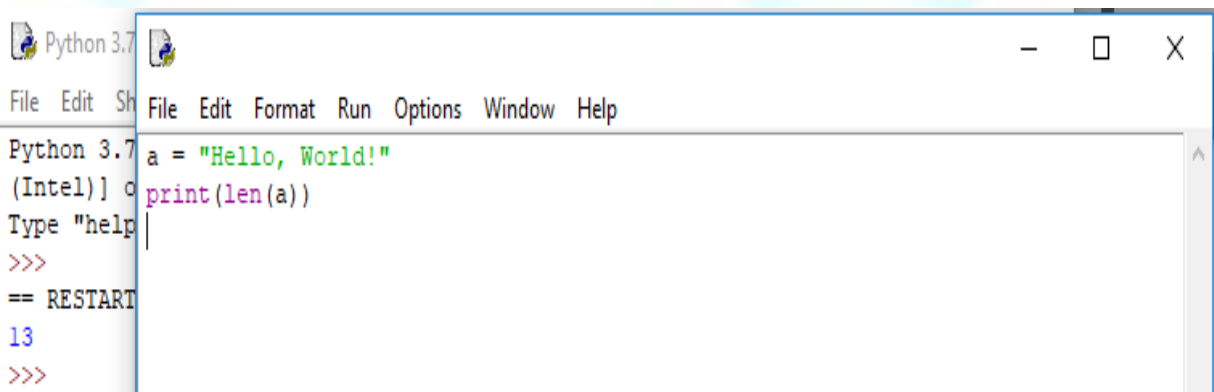
## Example

قمنا بتعريف متغير `a` وتخزين قيمة نصية بداخله

قمنا بطباعة عدد الأحرف (أو خانات) المتغير النصي الذي سترجعه الدالة `len()`  
`a = "Hello, World!"`  
`print(len(a))`

نتيجة تشغيل الكود

## Run the example



```
Python 3.7
File Edit Shell File Edit Format Run Options Window Help
Python 3.7 a = "Hello, World!"
(Intel) C print(len(a))
Type "help"
>>>
== RESTART ==
13
>>>
```

الدالة `lower()` تقوم بترجيع النص بأحرف صغيرة.

The **`lower()`** method returns the string in lower case.

## Example

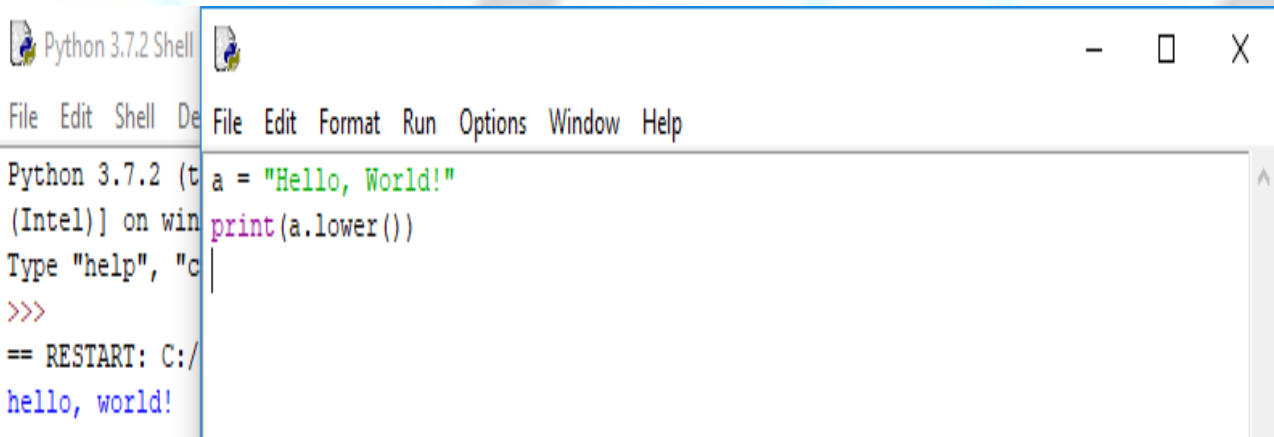
```
a = "Hello, World!"  
print(a.lower())
```

هنا قمنا بتعريف متغير `a` وتخزين قيمة نصية فيه

هنا قمنا بطباعة النص الذي سترجعه الدالة `lower()` عن طريق دالة الطباعة

## Run the example

نتيجة تشغيل الكود



```
Python 3.7.2 Shell  
File Edit Shell De File Edit Format Run Options Window Help  
Python 3.7.2 (tags/v3.7.2:1a7ff05, Dec 16 2019, 23:01:20) [AMD64] on win32  
Type "help", "copyright()", "credits()" and "quit()" for more  
>>> a = "Hello, World!"  
>>> print(a.lower())  
hello, world!  
>>>
```

الدالة upper() تقوم بترجيع النص بأحرف كبيرة.

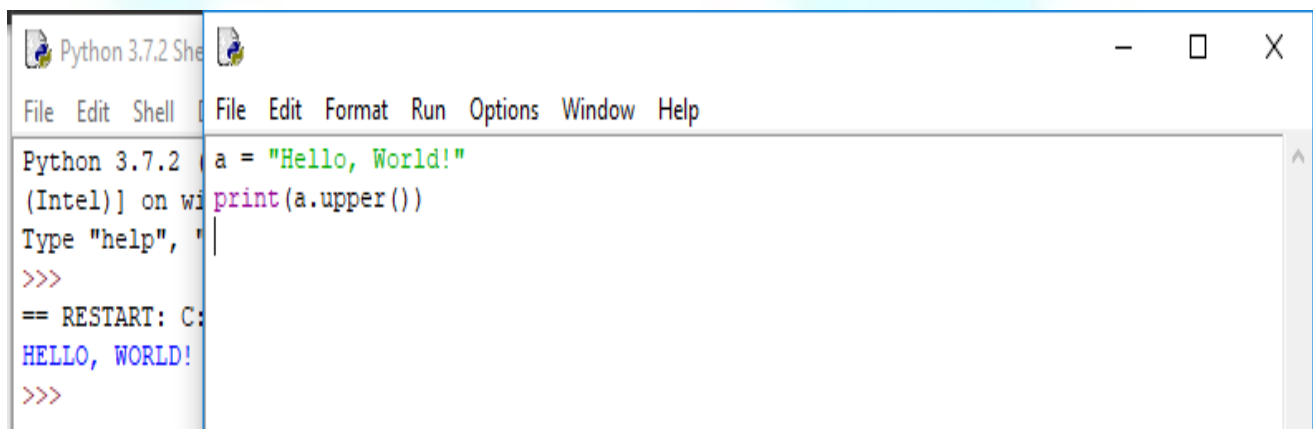
The **upper()** method returns the string in upper case.

### Example

```
a = "Hello, World!"  
print(a.upper())
```

### Run the example

نتيجة تشغيل الكود



```
Python 3.7.2 Shell [Python 3.7.2 Shell]
File Edit Shell File Edit Format Run Options Window Help
Python 3.7.2 (Intel) on win32: a = "Hello, World!"
print(a.upper())
Type "help", or press "q" to quit
>>>
== RESTART: C:\Python37\Python37.2\Python37.2 Shell ==
HELLO, WORLD!
>>>
```

لا تستسلم .. استمر

الدالة `replace()` تقوم بتبديل جزئية نصية بجزئية أخرى.  
نمرر النص الذي نريد استبداله، ثم نكتب النص الذي نريد أن يحل محل النص الأصلي  
`replace ( النص الجديد , النص الأصلي القديم )`

The `replace()` method replaces a string with another string.

## Example

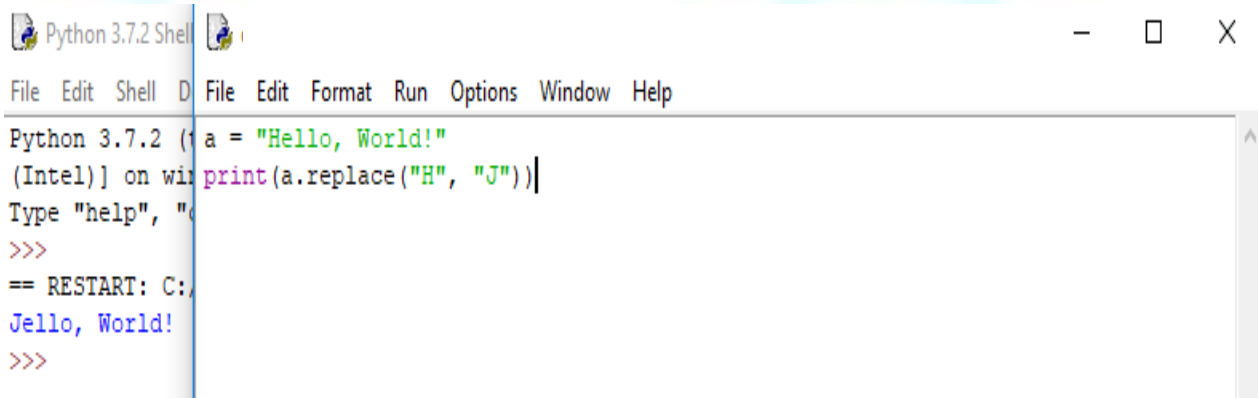
المثال سيوضح لك الفكرة

```
a = "Hello, World!"  
print(a.replace("H", "J"))
```

قمنا بتبديل H في كلمة Hello بالحرف J

## Run the example

نتيجة تشغيل الكود



```
Python 3.7.2 Shell  
File Edit Shell D File Edit Format Run Options Window Help  
Python 3.7.2 (a = "Hello, World!"  
(Intel)] on win print(a.replace("H", "J"))  
Type "help", "  
>>>  
== RESTART: C:  
Jello, World!  
>>>
```

الدالة `split()` تقوم بتقسيم النص على حسب الطريقة التي على أساسها تم استدعائها، ثم يتم ترجيع النص على شكل مصفوفة نصوص.

The **`split()`** method splits the string into substrings if it finds instances of the separator.

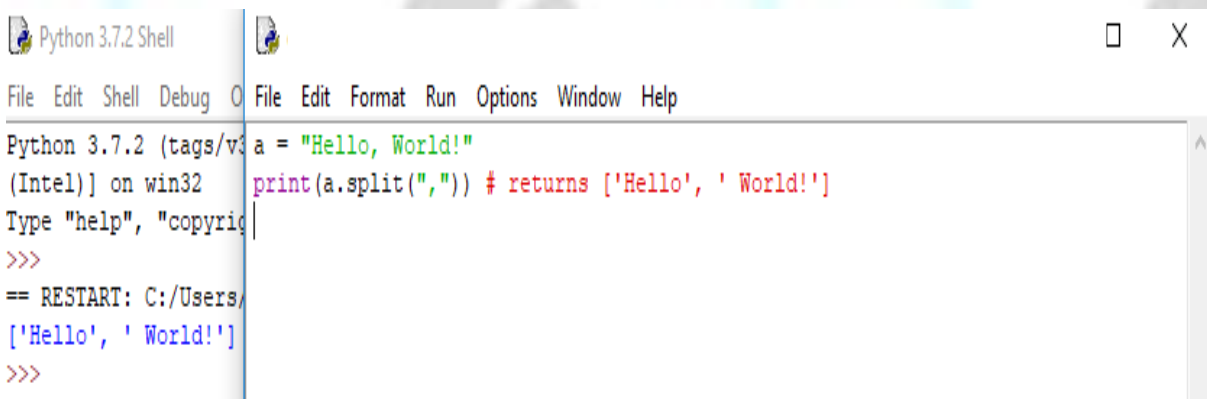
## Example

نمرر النص الذي على أساسه سيتم تجزئة النص الموجود في المتغير `a` في مثالنا هنا سيتم قص النص على أساس الفاصلة `,`

```
a = "Hello, World!"  
print(a.split(",")) # returns ['Hello', ' World!']
```

## Run the example

نتيجة تشغيل الكود



```
Python 3.7.2 Shell  
File Edit Shell Debug O File Edit Format Run Options Window Help  
Python 3.7.2 (tags/v3.7.2:1a3f604, Dec 14 2019) on win32  
Type "help", "copyright", "credits() or "license()" for more  
>>> a = "Hello, World!"  
>>> print(a.split(",")) # returns ['Hello', ' World!']  
>>> == RESTART: C:/Users/...>>>  
['Hello', ' World!']  
>>>
```



ممتاز

لقد أتممت درسك بنجاح!

روابط قد تساعدك

For more information See below:

- [How to Use Strings in Python - Python Tutorial for Beginners](#)
- [النصوص و الدوال Python 20](#)
- [replace استبدال النص Python in Arabic شرح بايثون بالعربي - #44 Learn Python in Arabic](#)
- [split string تقسيم النص Python in Arabic شرح بايثون بالعربي - #45 Learn Python in Arabic](#)

طبق ما تعلمته معنا في درسنا

ولا تنسى مشاركتنا أكوادك

# اليوم التاسع

# تنظيم/تنسيق النصوص String Format

كما تعلمنا سابقا، لا يمكن دمج المتغير النصي مع المتغير الرقمي

- As we learned in the **Python** Variables chapter, we cannot combine strings and numbers like this.

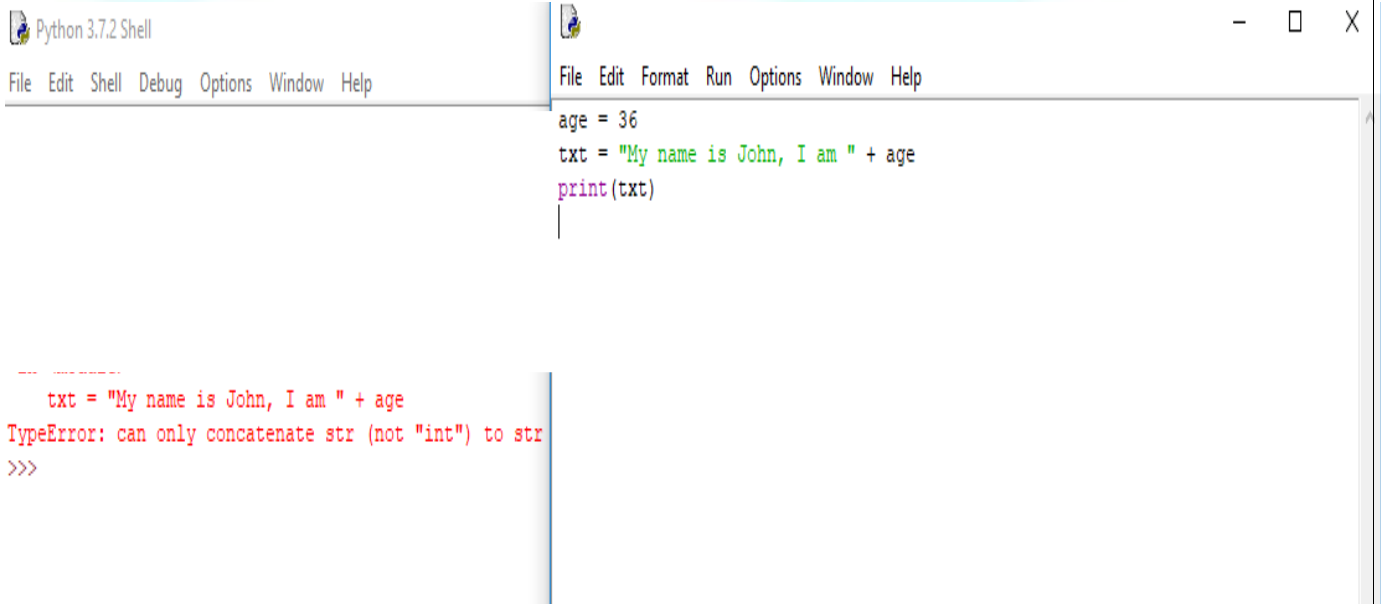
## Example

كما في مثالنا هنا

```
age = 36
txt = "My name is John, I am " + age
print(txt)
```

نتيجة تشغيل الكود

## Run the example



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help

age = 36
txt = "My name is John, I am " + age
print(txt)

-----
txt = "My name is John, I am " + age
TypeError: can only concatenate str (not "int") to str
>>>
```

لكن يمكنك الدمج بينهم بواسطة دالة `format()` فهي تستخدم لإجراء عملية التنسيق على النص

- We can combine strings and numbers by using the `format()` method!

The `format()` method takes the passed arguments, formats them, and places them in the string where the placeholders `{ }` are.

### Example

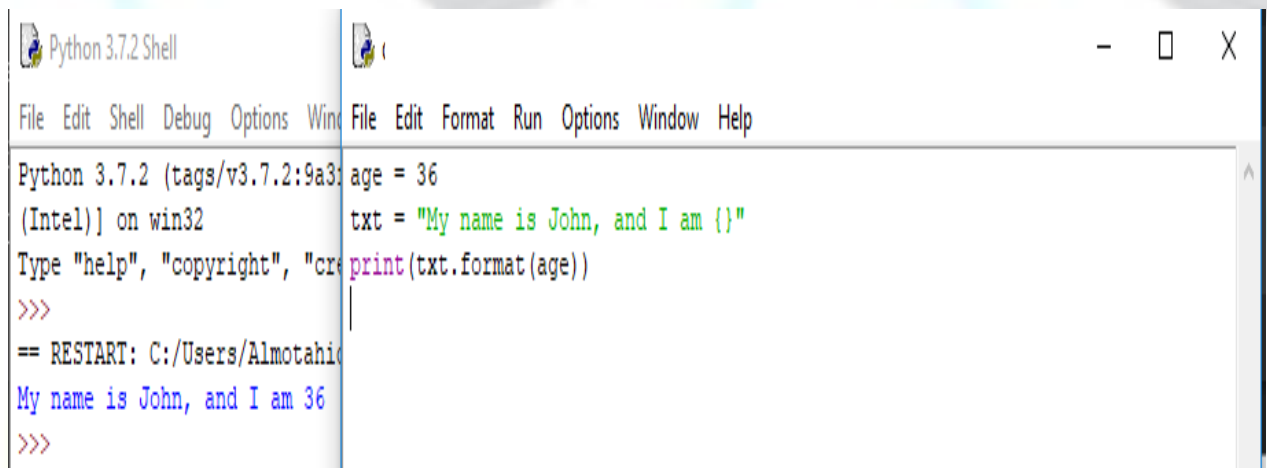
دالة `format()` تأخذ القيم التي تم تمريرها، في مثالنا هنا قيمة `age` وهي 36  
لنضعها في السلسلة النصية `txt` في أماكن/مواضع كتابة `{ }`

Use the `format()` method to insert numbers into strings.

```
age = 36
txt = "My name is John, and I am {}"
print(txt.format(age))
```

### Run the example

نتيجة تشغيل الكود



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3b333, Dec 14 2019) on win32
Type "help", "copyright", "credits()" or "quit()" for more
>>> age = 36
>>> txt = "My name is John, and I am {}"
>>> print(txt.format(age))
My name is John, and I am 36
>>>
```

يمكن تمرير عدد غير محدود من arguments لوضعها مكان {}

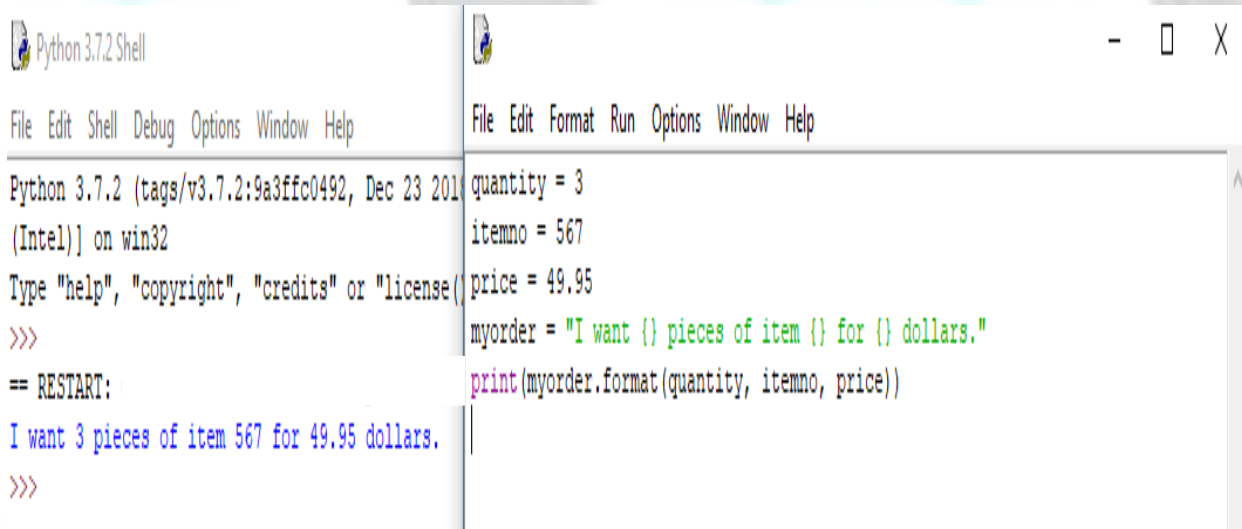
- The **format()** method takes unlimited number of arguments, and are placed into the respective placeholders.

## Example

```
quantity = 3
itemno = 567
price = 49.95
myorder = "I want {} pieces of item {} for {} dollars."
print(myorder.format(quantity, itemno, price))
```

## Run the example

نتيجة تشغيل الكود



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018) on win32
Type "help", "copyright", "credits" or "license()"
>>>
== RESTART: ==
I want 3 pieces of item 567 for 49.95 dollars.
>>>
```

يمكنك عند استخدام دالة `format()` ترقيم المدخلات حسب ترتيبها واستخدامها في تنسيق النص أكثر من مرة

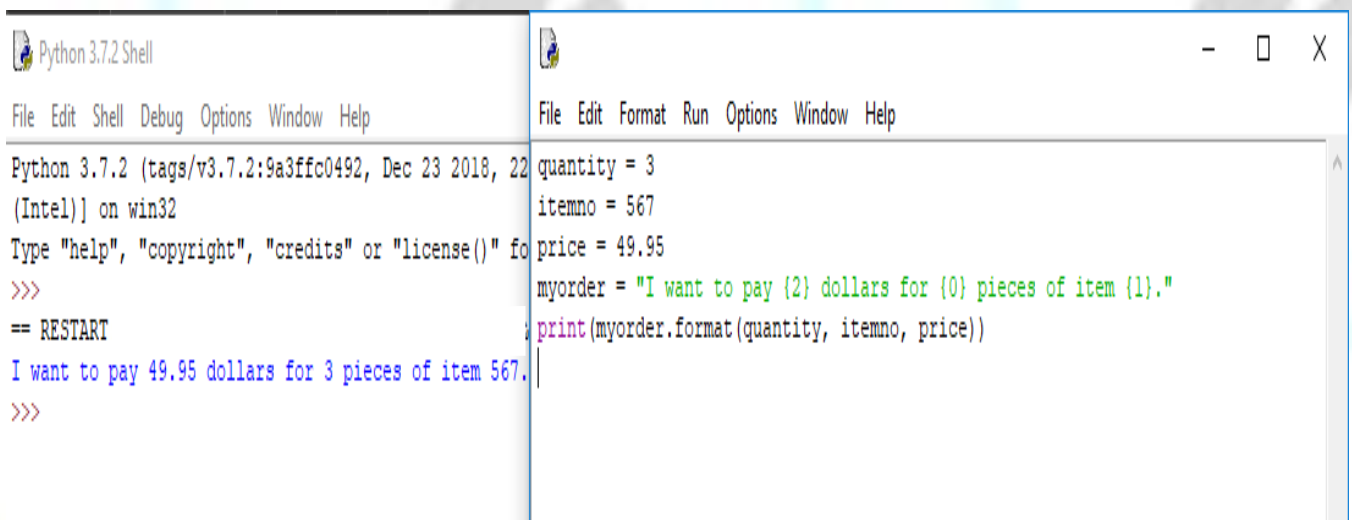
- You can use index numbers `{0}` to be sure the arguments are placed in the correct placeholders

## Example

```
quantity = 3
itemno = 567
price = 49.95
myorder = "I want to pay {2} dollars for {0} pieces of item {1}."
print(myorder.format(quantity, itemno, price))
```

## Run the example

## نتيجة تشغيل الكود



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:04:41) on win32
Type "help", "copyright", "credits" or "license()" for more
>>>
== RESTART
I want to pay 49.95 dollars for 3 pieces of item 567.
>>>
```

أتمت الدرس بنجاح!

روابط قد تفيدك

For more information See below:

- [تنظيم النصوص - String Formatting 23 - تعلم البرمجة بلغة بايثون](#)

طبّق ما تعلمته معنا في درسا  
ولا تنسى مشاركتنا أكوادك



# اليوم العاشر

# العوامل في لغة البايثون

## Python Operators

العوامل تستخدم لتنفيذ العمليات على المتغيرات وعلى القيم.  
والعامل عبارة عن رمز له معنى محدد ووظيفة محددة في البايثون.

Operators are used to perform operations on variables and values.

العوامل في البايثون تصنف إلى ٧ مجموعات:

**Python** divides the operators in the following groups:

- Arithmetic operators.
- Assignment operators.
- Comparison operators.
- Logical operators.
- Identity operators.
- Membership operators.
- Bitwise operators.

سنناقش بعضها معك في هذا الدرس ، والبقية في الدرس القادم بإذن الله

## العوامل في العمليات الحسابية/الرياضية

### ➤ Python Arithmetic Operators

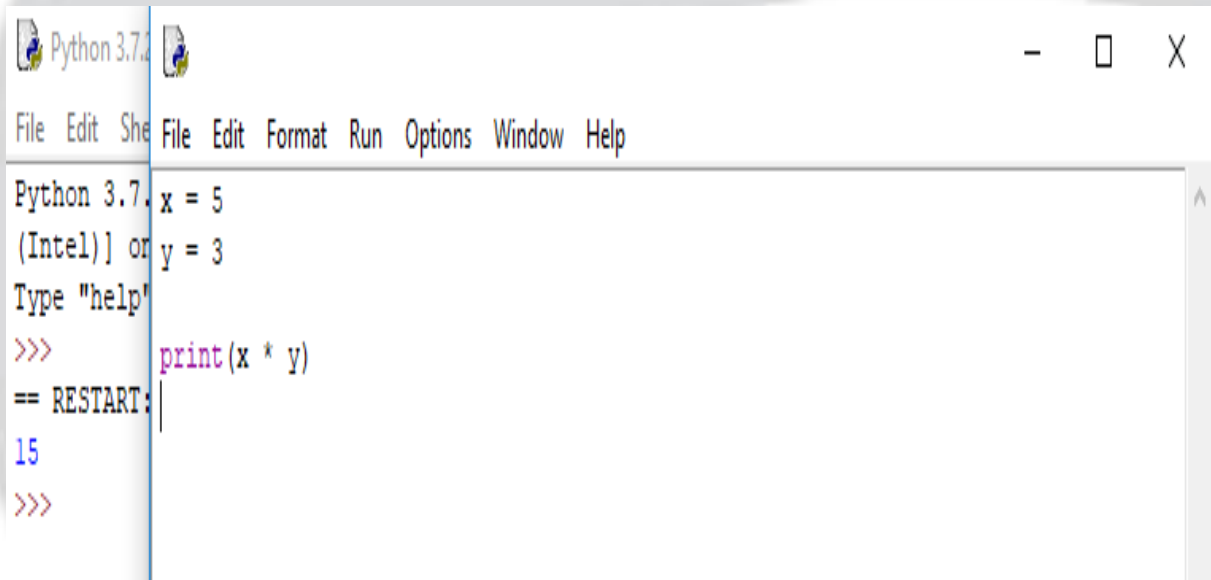
هي العوامل التي تستخدم قيم رقمية لتنفيذ العمليات الرياضية/الحسابية المعروفة

Arithmetic operators are used with numeric values to perform common mathematical operations.

Operator	Name	Example
+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	$x / y$
%	Modulus	$x \% y$
**	Exponentiation	$x ** y$
//	Floor division	$x // y$

اخترنا لك هذا المثال من الجدول السابق لتنفيذه  
يستخدم العامل \* لضرب قيمة بقيمة أخرى.

## Example



```
Python 3.7.2
File Edit Shell Format Run Options Window Help
Python 3.7.2 (Intel) on
Type "help()" for more
>>> x = 5
>>> y = 3
>>> print(x * y)
15
>>>
```

قم بتطبيق أمثلة أخرى على العمليات الحسابية من الجدول السابق، واستمتع!

## عوامل إعطاء قيم للمتغيرات/ التخصيص

### ➤ Python Assignment Operators

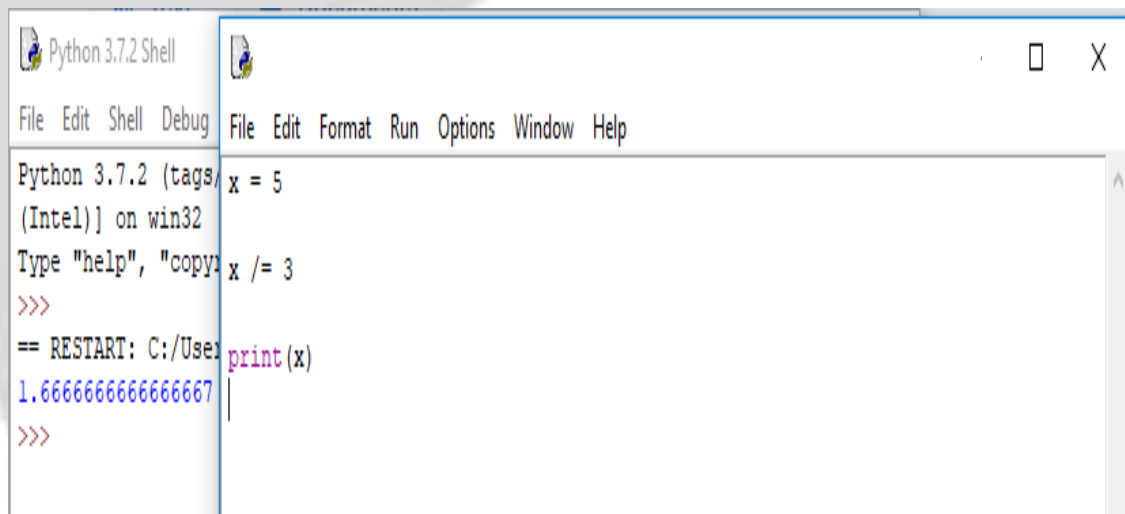
يمكنك استخدام هذه العوامل عند إسناد قيم للمتغيرات

Assignment operators are used to assign values to variables.

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
=	x  = 3	x = x   3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

اخترنا لك هذا المثال من الجدول السابق لتنفيذه  
يستخدم العامل /= لقسمة قيمة متغير على قيمة أخرى.

## Example



```
Python 3.7.2 Shell
File Edit Shell Debug
Python 3.7.2 (tags/
(Intel)] on win32
Type "help", "copy x = 5
>>> x /= 3
== RESTART: C:/Use>>> print(x)
1.6666666666666667
>>>
```

جرب أمثلة أخرى

## ➤ Python Comparison Operators

يمكنك استخدام هذه العوامل للمقارنة بين قيمتين

Comparison operators are used to compare two values.

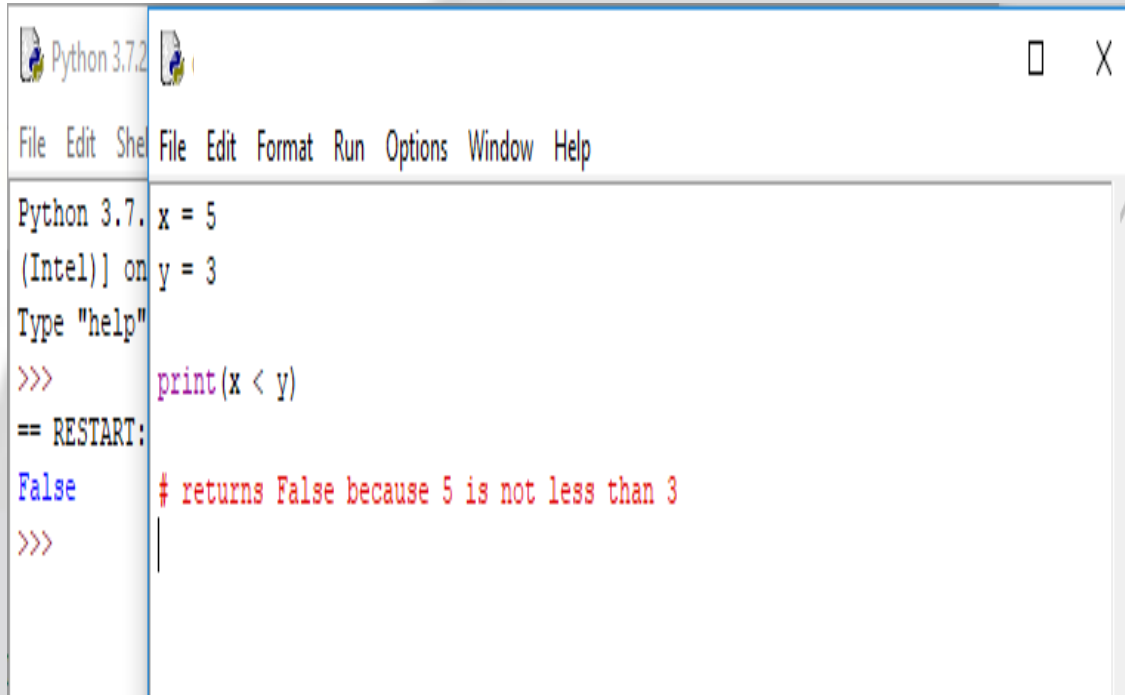
Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

اخترنا لك هذا المثال من الجدول السابق لتنفيذه

يُستخدم هذا العامل لتنفيذ كود محدد إذا كانت قيمة المتغير الأول أصغر من قيمة المتغير الثاني.

**Example**

الإجابات ستكون بـ True أو False



```
Python 3.7.2
File Edit Shell File Edit Format Run Options Window Help
Python 3.7.2 (Intel) on
Type "help"
>>> print(x < y)
== RESTART:
False # returns False because 5 is not less than 3
>>>
```

طبّق الأمثلة الأخرى على جهازك



## أتممت الدرس بنجاح!

### روابط ربما تهتمك!

For more information, check the links below.

- [المعاملات في بايثون | 07- Python Operators](#)
- [#11 Python Tutorial for Beginners | Operators in Python](#)
- [Python in Arabic #6 شرح بايثون بالعربي - العمليات الحسابية Arithmetic Operators](#)
- [Assignment Operators in Python - #14](#)
- [Python in Arabic #13 شرح بايثون بالعربي - عوامل التخصيص Assignment Operators](#)
- [Python in Arabic #15 شرح بايثون بالعربي - عوامل المقارنة Comparison Operators](#)
- [Comparison Operators in Python - #15](#)

طبق ما تعلمته معنا في درسنا

ولا تنسى مشاركتنا أكوادك

# اليوم الحادي عشر

# العوامل في لغة البايثون

## Python Operators 2

الأنواع المتبقية من العوامل

## عوامل الشروط المنطقية

### ➤ Python Logical Operators

تستخدم لوضع شروط منطقية وتقوم بترجيع قيمتي **True or False**

Logical operators are used to combine conditional statements.

Operator	Description	Example
and	Returns True if both statements are true	<code>x &lt; 5 and x &lt; 10</code>
or	Returns True if one of the statements is true	<code>x &lt; 5 or x &lt; 4</code>
not	Reverse the result, returns False if the result is true	<code>not(x &lt; 5 and x &lt; 10)</code>

في هذا المثال استخدمنا العامل **or** لتنفيذ كود محدد

فإذا تحقق على الأقل أحد الشروط سترجع لنا القيمة بـ **True**

### Example

```

Python 3.7.2 Shell
File Edit Shell Deb File Edit Format Run Options Window Help
Python 3.7.2 (tags/3.7.2:1) on win32
Type "help", "copyright()", "credits()", "license()", "quit()", or "exit()" for more help
>>> x = 5
>>> print(x > 3 or x < 4)
True
# returns True because one of the conditions are true (5 is greater than 3,
# but 5 is not less than 4)
>>>

```

العوامل التي تستخدم لمعرفة هل الكائنين يشيران لكائن واحد في الذاكرة أم لا

## ➤ Python Identity Operators

تستخدم لمقارنة الكائنات، ليس أن الكائنين متساويين في القيمة بل المقارنة تكون هل هما نفس الكائن ويشيران لنفس المكان في الذاكرة أم لا؟ عن طريق معرفة عنوان المتغير في الذاكرة في لغة البايثون

Identity operators are used to compare the objects, not if they are equal, but if they are actually the same object, with the same memory location.

Operator	Description	Example
is	Returns true if both variables are the same object	x is y
is not	Returns true if both variables are not the same object	x is not y

### العامل Is

يستخدم لمعرفة هل الكائنين يشيران لكائن واحد في الذاكرة أم لا؟  
ترجع القيمة بـ **True** إذا كان الكائنين **نعم** يشيران لنفس الكائن.  
وترجع القيمة بـ **False** إذا كان الكائنين **لا** يشيران لنفس الكائن.

### العامل Is not

يستخدم لمعرفة هل الكائنين **لا** يشيران لكائن واحد في الذاكرة أم لا؟  
ترجع القيمة بـ **True** إذا كان الكائنين **لا** يشيران لنفس الكائن.  
وترجع القيمة بـ **False** إذا كان الكائنين **نعم** يشيران لنفس الكائن.

المثال التالي سيوضح لك الفكرة أكثر ..

## Example

```
Python 3.7
File Edit Shell File Edit Format Run Options Window Help
Python 3.7 x = ["apple", "banana"]
(Intel) or y = ["apple", "banana"]
Type "help" z = x
>>>
== RESTART == print(x is not z)
False
True # returns False because z is the same object as x
False
>>> print(x is not y)

# returns True because x is not the same object as y,
# even if they have the same content

print(x != y)

# to demonstrate the difference between "is not" and "!=":
# this comparison returns False because x is equal to y
```

## Python Membership Operators

عوامل البحث للمصفوفات

Membership operators are used to test if a sequence is presented in an object.

Operator	Description	Example
in	Returns True if a sequence with the specified value is present in the object	x in y
not in	Returns True if a sequence with the specified value is not present in the object	x not in y

## العامل in

يستخدم لمعرفة هل المصفوفة تحتوي على قيمة معينة أم لا؟

ترجع القيمة بـ **True** إذا كان **نعم** يوجد عنصر في المصفوفة نفس القيمة المراد البحث عنها.  
وترجع القيمة بـ **False** إذا كان **لا** يوجد أي عنصر في المصفوفة لديه نفس القيمة المراد البحث عنها.

## العامل not in

يستخدم لمعرفة ما إذا كانت المصفوفة **لا** تحتوي على هذه القيمة المعينة أم لا؟

ترجع القيمة بـ **True** إذا كان **لا** يوجد أي عنصر في المصفوفة عنده نفس القيمة المراد البحث عنها.  
وترجع القيمة بـ **False** إذا كان **نعم** يوجد عنصر في المصفوفة عنده نفس القيمة المراد البحث عنها.

المثال التالي سيوضح لك الفكرة أكثر ..

## Example

```
Python 3.7.4 Shell
File Edit Shell File Edit Format Run Options Window Help
Python 3.7 x = ["apple", "banana"]
(Intel) on
Type "help" print("banana" in x)
>>>
== RESTART # returns True because a sequence with the value "banana" is in the list
True
>>>
```

## ➤ Python Bitwise Operators

Bitwise operators are used to compare (binary) numbers.

Operator	Name	Description
&	AND	Sets each bit to 1 if both bits are 1
	OR	Sets each bit to 1 if one of two bits is 1
^	XOR	Sets each bit to 1 if only one of two bits is 1
~	NOT	Inverts all the bits
<<	Zero fill left shift	Shift left by pushing zeros in from the right and let the leftmost bits fall off
>>	Signed right shift	Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off



أتممت الدرس بنجاح!

روابط ربما تهتمك!

For more information, check the links below.

- [Logical العوامل المنطقية Python in Arabic شرح بايثون بالعربي - 14 # Learn Python in Arabic Operators](#)
- [22 Python for non programmers Logical and Bitwise Operators](#)
- [Python Programming Tutorial - 10: Bitwise & Logical Operators](#)
- [Python Programming Tutorial - 51 - Bitwise Operators](#)
- [#15 Python Tutorial for Beginners | Python BitWise Operators](#)
- [Python Programming Tutorial - 11: Membership & Identity Operators](#)

طبق ما تعلمته معنا في درسنا  
ولا تنسى مشاركتنا أكوادك

# اليوم الثاني عشر

“ Please, I want {} sandwichs and {} donutes “

المطلوب طباعة الجملة أعلاه مع تطبيق النقاط التالية:

١- استبدال حرف الضمير **i** بحرف **we** باستخدام دالة الاستبدال التي سبق أخذها في أحد الدروس.

٢- في الأقواس الخالية وباستخدام الدالة المناسبة قم بمليء الفراغ الأول بـ العدد 5 والفراغ الثاني بالعدد 7.

٣- حوّل كل حرف “a” إلى كابتال (حرف كبير) Capital Letter

ثم اطبع الجملة النهائية بعد كل هذه التعديلات **ولا تنس** الاحتفاظ بالكود للرجوع إليه فيما بعد.