

AUGUST 20 – 24, 2019



المبادرة السعودية للمطورين

تعلم .. فكر .. حاول .. أبداع

المبادرة السعودية للمطورين

مسار Python

مشرفي المسار:

عبدالله عوده – انتصار النصار – رؤى كردي – ليلى المصعبي

python™

## 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 الأول

## What is Python?

ماهي لغة بايثون؟

تعتبر البايثون من أشهر لغات البرمجة إطلاقاً

**Python** is a popular programming language and It was created by Guido van Rossum and released in 1991.

## Used for:

استخدامات لغة البايثون

- Web Development (server-side).
- Software Development.
- Mathematics.
- System Scripting.

## What can Python do?

ماذا سأستفيد من لغة بايثون؟

- **Python** can be used on a server to create web applications.
- **Python** can be used alongside software to create workflows.
- **Python** can connect to database systems. It can also read and modify files.
- **Python** can be used to handle big data and perform complex mathematics.
- **Python** can be used for rapid prototyping, or for production-ready software development.

## Why Python?

لماذا لغة بايثون بالتحديد؟

- بايثون تعمل على أكثر من منصة وعلى أهم أنظمة التشغيل.
- بايثون سهلة التعلم ويمكن قراءة الكود والتعديل عليه بسهولة.
- بايثون أكوادها قصيرة ومختصرة ولا تأخذ عدد كبير من السطور.
- بايثون سريعة التنفيذ.
- بايثون كائنيه التوجه.

- **Python** works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- **Python** has a simple syntax similar to the English language.
- **Python** has syntax that allows developers to write programs with fewer lines than some other programming languages.
- **Python** runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- **Python** can be treated in a procedural way, an object-orientated way or a functional way.

## The Python Logo

شعار لغة البايثون



## Python download&Install

تنزيل وتنصيب لغة البايثون

التأكد من نوع البايثون الموجود على جهازك  
من خلال الأمر `python --version` على لوحة الأوامر Command Line (للـ ويندوز)  
أو التيرمينال Terminal (للـ ماك)

Many PCs and Macs will have **Python** already installed.

To check if you have python installed on a **Windows PC**, search in the start bar for **Python** or run the following on the **Command Line** (cmd.exe):

```
C:\Users\Your Name>python --version
```

To check if you have **Python** installed on a **Linux or Mac**, then on **Linux** open the **Command Line** r on **Mac** open the **Terminal** and type:

```
python --version
```

## لبداية البرمجة بلغة البايثون على جهازك تحتاج:

كما في الخطوة السابقة التأكد من نوع البايثون الموجود على جهازك. يفضل أن يكون ٣,٧ آخر إصدار

نستخدم الـ **Command Line** و **Terminal** فقط لاختبار أو طباعة دوال جاهزة أو لرؤية نتيجة البرامج التي تم كتابتها مسبقا بواسطة ايديتور **Editor** أو **IDE python shell** تقريبا يعمل كـ **Command Line** ولكن بطريقة أكثر فاعلية لأنه متخصص بالبايثون.

يفضل تحميل الايديتور **Editor** حيث أنه يعتبر بيئة بناء لأي برنامج بايثون، كما أنه يساعد أيضا في تنظيم الكود وتنزيل المكتبات اللازمة لعمل الدوال.

الايديتور **Editor** الذي ينصح به في هذه الدروس هو: فيجوال ستوديو كود  
برامج البايثون التي لا تتطلب الاتصال بأي واجهات برمجية أخرى.

بالإمكان استخدام مواقع البرمجة أونلاين للبايثون مثل:

- <https://www.python.org/shell/>
- <https://repl.it/languages/python3>
- [https://www.tutorialspoint.com/execute\\_python\\_online.php](https://www.tutorialspoint.com/execute_python_online.php)

You can download **Python** for free from the following website: <https://www.python.org/>

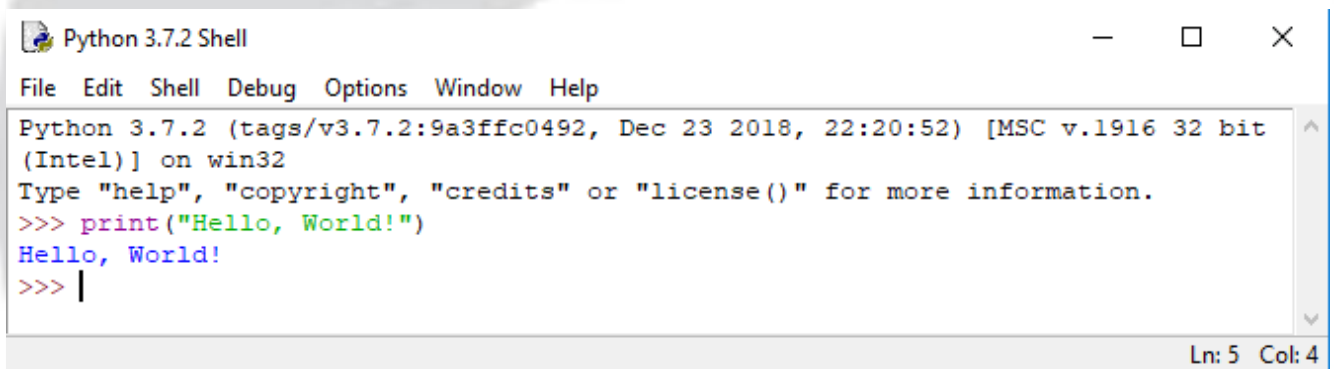
**Hint:** Make sure you installed the latest version.

## First Lesson:

الآن أنت مستعد للدرس الأول

- Let's write our first **Python** code:

```
print("Hello, World!")
```



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello, World!")
Hello, World!
>>> |
```

ببساطة قمنا بإنشاء برنامج مهمته طباعة جملة

هذا يعني أن **print()** دائما تطبع لي ما يكون بين علامتي الاقتباس

- As we can see print ("Hello, World") will give me **Hello World**. That's mean print ("") will always give me what's between parenthesis.

باستطاعتك بكل سهولة الخروج من واجهه الأوامر بواسطة **exit()**

- Whenever you are done in the **Python** command line, you can simply type the following to quit the **Python** command line interface:

```
exit()
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello, World!")
Hello, World!
>>> exit()
>>> |
```

المسافة البادئة في لغة بايثون مهمة جداً.

- Where in other programming languages the indentation in code is for readability only, in **Python** the indentation is very important **Python** uses indentation to indicate a block of code

```
if 5 > 2:
    print("Five is greater than two!")
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> if 5 > 2:
    print("Five is greater than two!")

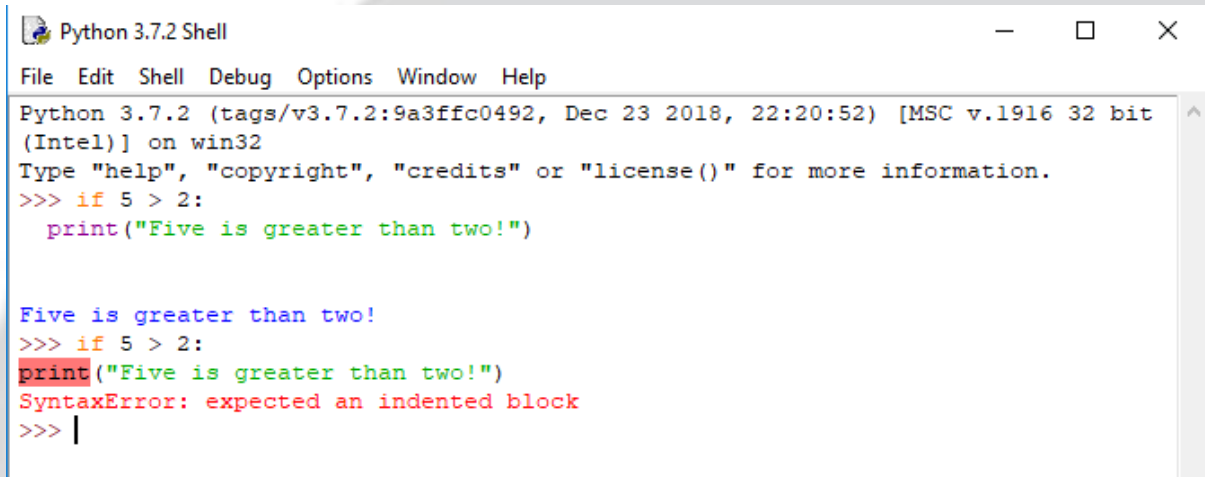
Five is greater than two!
>>> |
```

سيظهر لك خطأ لو قمت بتجاهل المسافة المبدئية.

- **Python** will give you an error if you skip the indentation.

```
if 5 > 2:
print("Five is greater than two!")
```





```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> if 5 > 2:
    print("Five is greater than two!")

Five is greater than two!
>>> if 5 > 2:
print("Five is greater than two!")
SyntaxError: expected an indented block
>>> |
```

ممتاز، لقد أتممت درسك الأول بنجاح!

روابط مفيدة

See more,

- [04- Python Editors | محررات خاصة بايثون](#)
- [بايثون بالعربي - تنزيل البرنامج وكتابة أول كود - 01](#)
- [#2 Python Tutorial for Beginners | Python Installation | PyCharm](#)
- [How to install Python 3.7.3 on Windows 10](#)
- [Python Tutorial for Beginners 1: Install and Setup for Mac and Windows](#)
- [How to Start Programming in Python](#)
- [Get started using Python on Windows for beginners](#)
- [Using Python on a Macintosh](#)
- [Python IDEs and Code Editors \(Guide\)](#)

طبق ما تعلمته معنا في درسنا الأول  
ولا تنسى مشاركتنا أكوادك.

# اليوم الثاني

# التعليقات في لغة البايثون

## Python Comments

- غالبا نستخدم التعليقات لشرح ووضع ملاحظات حول الكود.
- نستخدم التعليقات أيضا لتسهيل عملية فهم وقراءة الكود، خصوصا لو أردنا مراجعته بعد وقت طويل.
- يمكنك استخدام التعليقات للأكواد أيضا في حين لا نريدها أن تنفذ أثناء عملية التشغيل/التنفيذ.
- لا تقلق فالتعليقات لا تؤثر أبداً على الكود المكتوب.

- Comments can be used to explain **Python** code.
- Comments can be used to make the code more readable.
- Comments can be used to prevent execution when testing code.

## إنشاء تعليق في سطر واحد

### ➤ Creating a Comment

Comments start with a **#**, and **Python** will render the rest of the line as a comment.

لكتابة التعليق، ضع الرمز **#** ثم أكتب بعده ما تريد

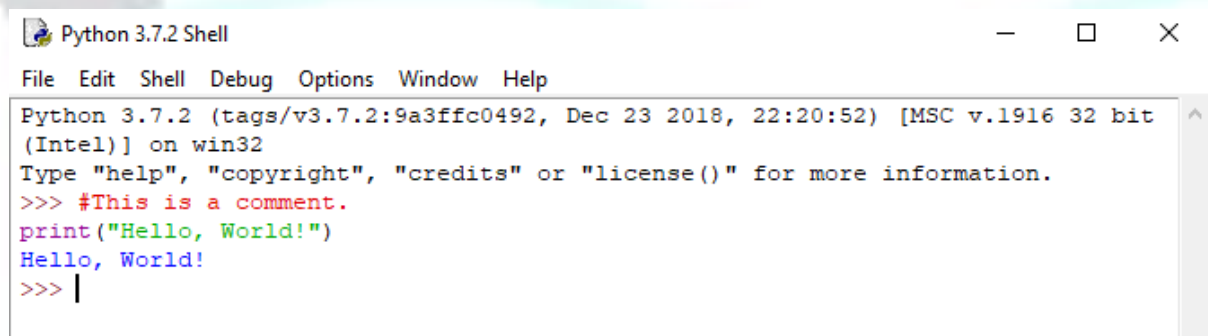
### Example

```
#This is a comment
print("Hello, World!")
```

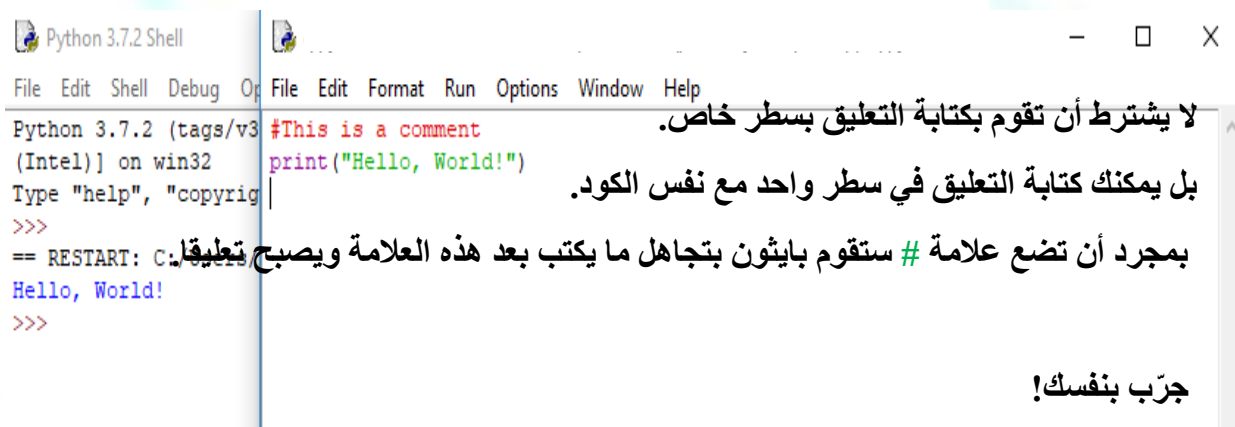
قم بتطبيق هذا المثال بنفسك لتتأكد من فهمك

### 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, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> #This is a comment.
print("Hello, World!")
Hello, World!
>>> |
```



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> #This is a comment
print("Hello, World!")
Hello, World!
>>> |
```

لا يشترط أن تقوم بكتابة التعليق بسطر خاص.

بل يمكنك كتابة التعليق في سطر واحد مع نفس الكود.

بمجرد أن تضع علامة **#** ستقوم بايثون بتجاهل ما يكتب بعد هذه العلامة ويصبح التعليق.

جرب بنفسك!

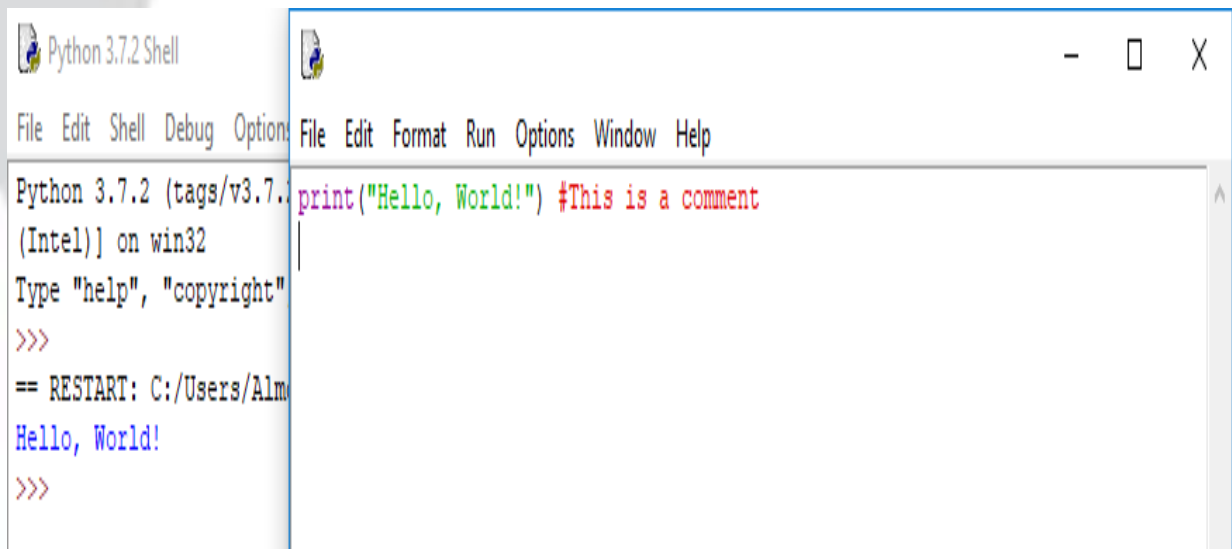
- Comments can be placed at the end of a line, and **Python** will ignore the rest of the line

## Example

```
print("Hello, World!") #This is a comment
```

نتيجة تشغيل الكود

## Run the example



```
Python 3.7.2 Shell
File Edit Shell Debug Options
Python 3.7.2 (tags/v3.7.2:1, Dec 5 2019, 19:10:44) on win32
Type "help", "copyright"
>>>
== RESTART: C:/Users/Alm
Hello, World!
>>>
```

لا يشترط أن يكون التعليق فقط كتابة وشرح.

إذا أردت أن تجرب كودًا معينًا ثم لا تريده أن يعمل يمكنك كتابته وجعله كتعليق أيضًا، وبذلك لن يعمل هذا الكود أثناء التشغيل، ويمكنك الرجوع إليه في أي وقت.

جرب واحكم!

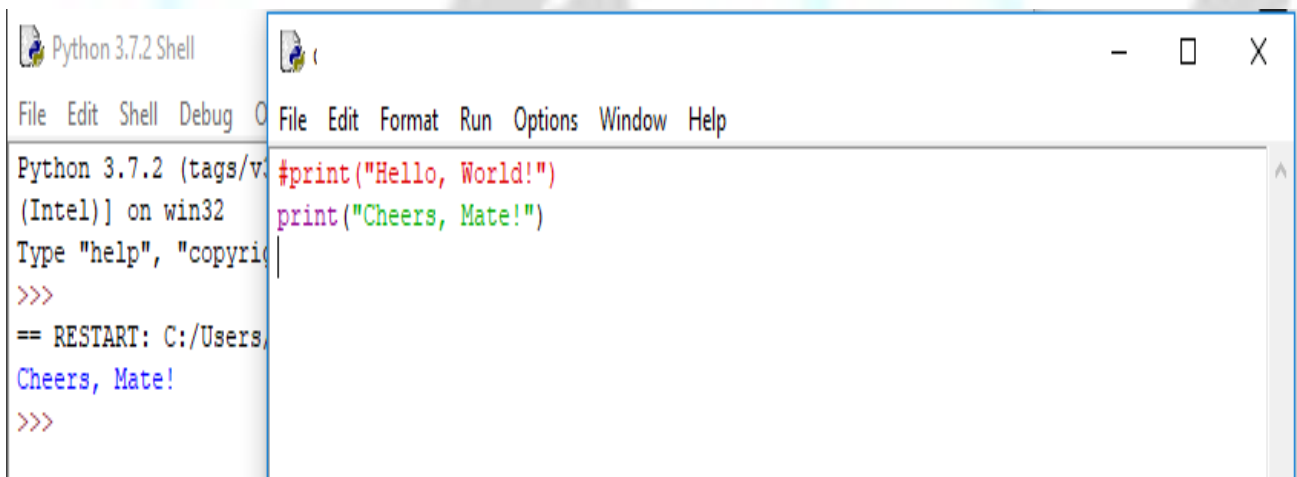
- Comments does not have to be text to explain the code, it can also be used to prevent **Python** from executing code

## Example

```
#print("Hello, World!")  
print("Cheers, Mate!")
```

نتيجة تشغيل الكود

## Run the example



```
Python 3.7.2 Shell  
File Edit Shell Debug O  
Python 3.7.2 (tags/v3.7.2:1, Dec 6, 2019) [AMD64] on win32  
Type "help", "copyright", "credits() or "license()">  
>>>  
== RESTART: C:/Users/...>  
Cheers, Mate!  
>>>
```

## كتابة التعليقات في أكثر من سطر

### ➤ Multi Line Comments

**Python** does not really have a syntax for multi-line comments.

To add a multiline comment, you could insert a **#** for each line.

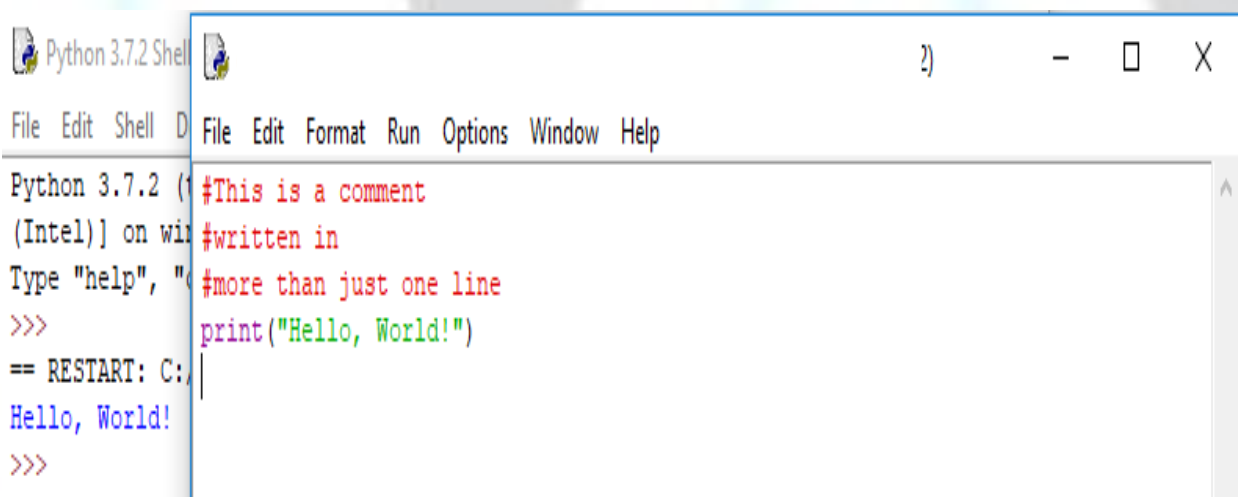
### Example

في بداية كل سطر قم بكتابة العلامة **#** قبل كل النص

```
#This is a comment
#written in
#more than just one line
print("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) on win32
Type "help", "copyright()", "credits()", or "quit()" for more
>>> #This is a comment
>>> #written in
>>> #more than just one line
>>> print("Hello, World!")
Hello, World!
>>>
```

## ➤ Multiline String

كتابة نص في أكثر من سطر

التعليقات التي تحتوي أكثر من سطر أو سلسلة نصية طويلة فيجب إحاطتها بثلاث علامات تنصيص (منفردة أو مزدوجة) في البداية والنهاية

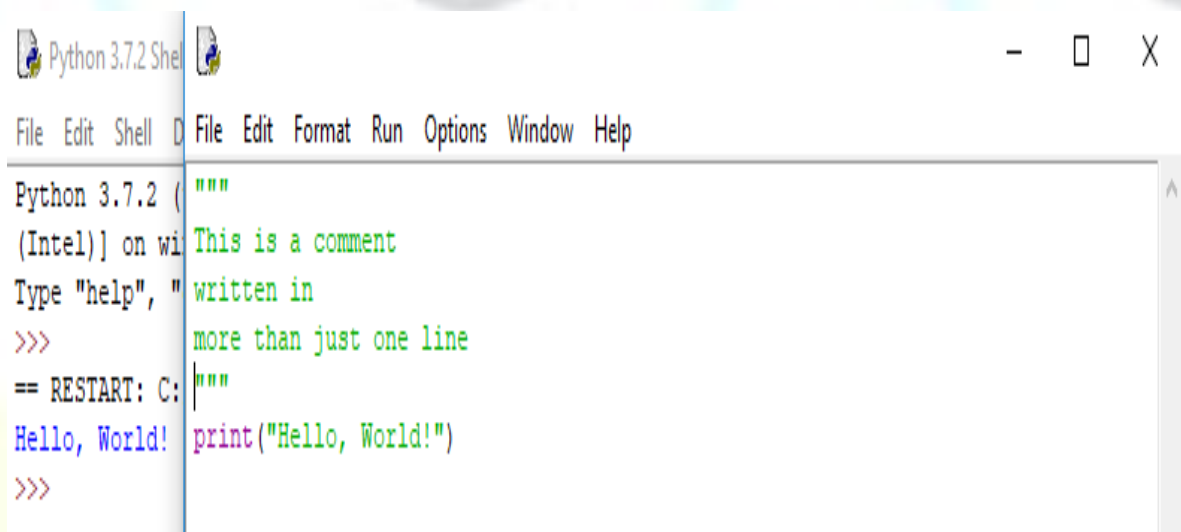
Since **Python** will ignore string literals that are not assigned to a variable, you can add a multiline string (triple quotes) in your code and place your comment inside it.

### Example

```
"""
This is a comment
written in
more than just one line
"""
print("Hello, World!")
```

نتيجة تشغيل الكود

### Run the example



```
Python 3.7.2 Shell
File Edit Shell D File Edit Format Run Options Window Help
Python 3.7.2 (
(Intel)] on wi
Type "help", "
>>>
== RESTART: C:
Hello, World!
>>>
```



رائع!  
أتممت درسك الثاني بنجاح

روابط قد تفيدك

For more information See below:

- [#66 Python Tutorial for Beginners | Comments](#)
- [Python tutorial | | Commets](#) التعليقات في البايثون
- [7- Python Comments](#) التعليقات

طبّق ما تعلمته معنا في درسنا الثاني  
ولا تنسى مشاركتنا أكوادك

# اليوم الثالث

# المتغيرات في لغة البايثون

## Python Variables

### ➤ Creating Variables

### إنشاء المتغيرات

نستخدم المتغيرات لتخزين القيم/البيانات فيها.

تعريف المتغيرات في لغة بايثون تختلف عن اللغات الأخرى بأنها ليست بحاجة للتعريف/للإعلان عن نوع المعرف المستخدم، المبرمج لا يحتاج إلى تحديد أو تعريف نوع المتغير، سيقوم مفسر لغة بايثون بتحديد نوع المتغير تلقائياً فور إسناد قيمة إليه، يتم إنشاء وحجز جزء من الذاكرة للمتغير فور إسناد قيمة إليه.

Variables are containers for storing data values.

Unlike other programming languages, **Python** has no command for declaring a variable.

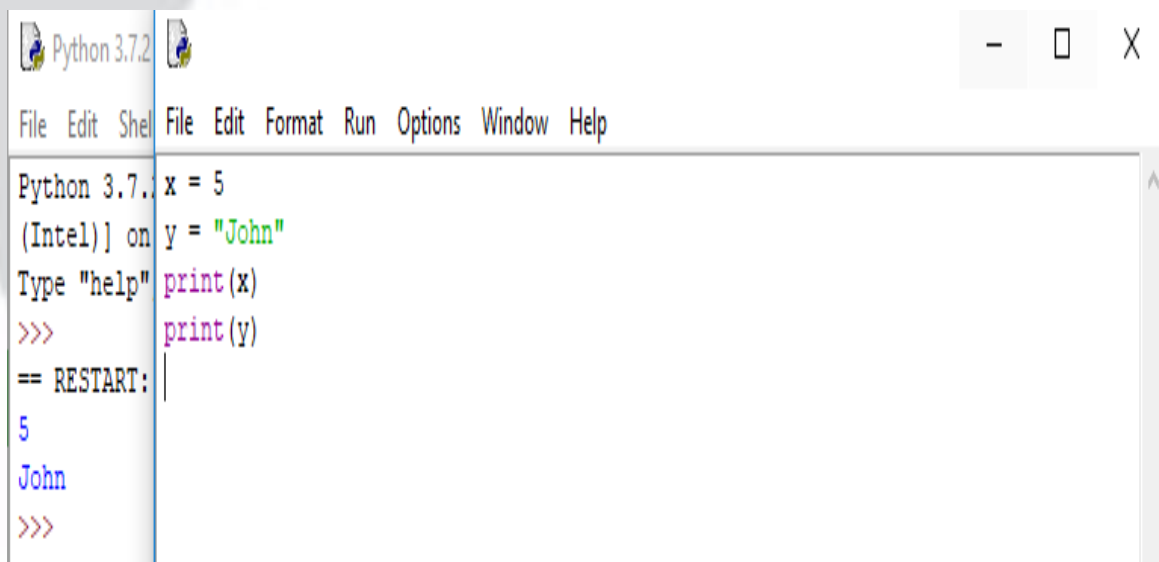
A variable is created the moment you first assign a value to it.

## Example

```
x = 5
y = "John"
print(x)
print(y)
```

## Run the example

نتيجة تشغيل الكود



```
Python 3.7.2
File Edit Shell File Edit Format Run Options Window Help
Python 3.7.2: x = 5
(Intel) on y = "John"
Type "help" print(x)
>>> print(y)
== RESTART: |
5
John
>>>
```

يمكن تغيير نوع المتغير حتى لو تم إنشائه، ولذلك بإسناد قيمة جديدة إليه بنوع مختلف.

- Variables do not need to be declared with any particular type and can even change type after they have been set.

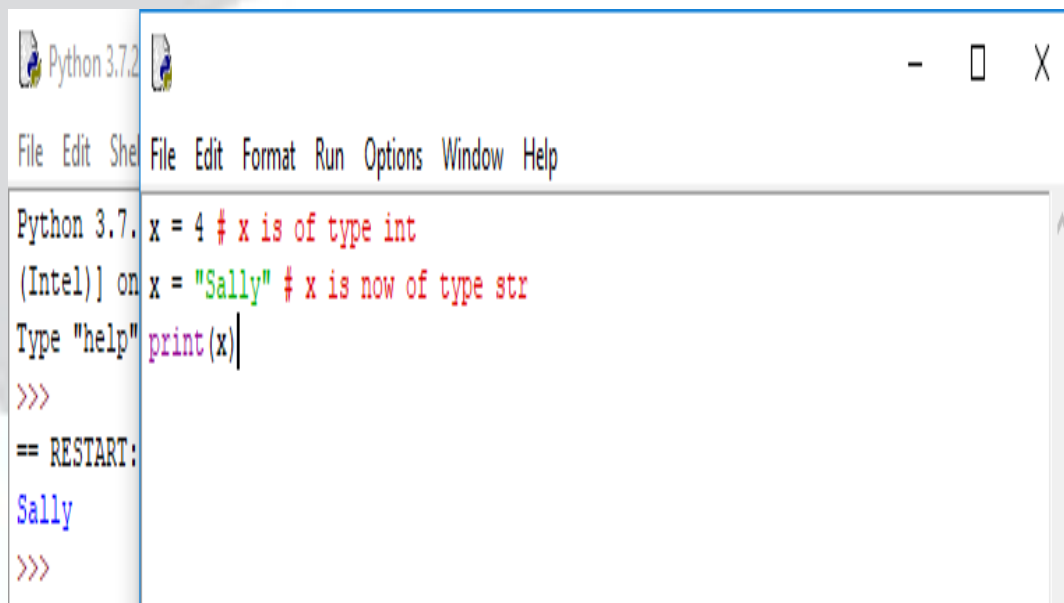
## Example

قم بكتابة و تجربة هذا المثال ..

```
x = 4 # x is of type int
x = "Sally" # x is now of type str
print(x)
```

## Run the example

نتيجة تشغيل الكود



```
Python 3.7.2
File Edit Shell File Edit Format Run Options Window Help
Python 3.7.2 (Intel) on
Type "help()" for more
>>> x = 4 # x is of type int
>>> x = "Sally" # x is now of type str
>>> print(x)
Sally
>>>
```

عند إسنادك نصوص/حروف للمتغير يجب وضعها بين علامتي تنصيص

- String variables can be declared either by using single or double quotes.

## Example

```
x = "John"
# is the same as
x = 'John'
```

## Run the example

## نتيجة تشغيل الكود

```
x = "John"
print(x)
#double quotes are the same as single quotes:
x = 'John'
print(x)
```

```
C:\Users\My Name>python demo_variables7.py
John
John
```

## Variable Names

## تسمية المتغيرات

أسماء المتغيرات قد تكون قصيرة كالحروف، وأيضا تكون أسماء ذات معنى وموصوفة.

A variable can have a short name (like x and y) or a more descriptive name (age, car name, total volume).

## Rules for Python variables

### قواعد تسمية المتغيرات

- A variable name must start with a letter or the underscore character.

يجب أن تبدأ بالحروف الأبجدية الصغيرة أو الكبيرة أو الشرطة المنخفضة.

- A variable name cannot start with a number.

المتغيرات لا تبدأ بالأرقام.

- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_).

بقية حروف المتغير ممكن أن تتكون من حروف أبجدية كبيرة/صغيرة، الشرطة المنخفضة، أرقام.

- Variable names are case-sensitive (age, Age and AGE are three different variables).

أسماء المتغيرات حساسة لحالة الأحرف في الكتابة

### ➤ Assign Value to Multiple Variables

تعريف أكثر من متغير في سطر

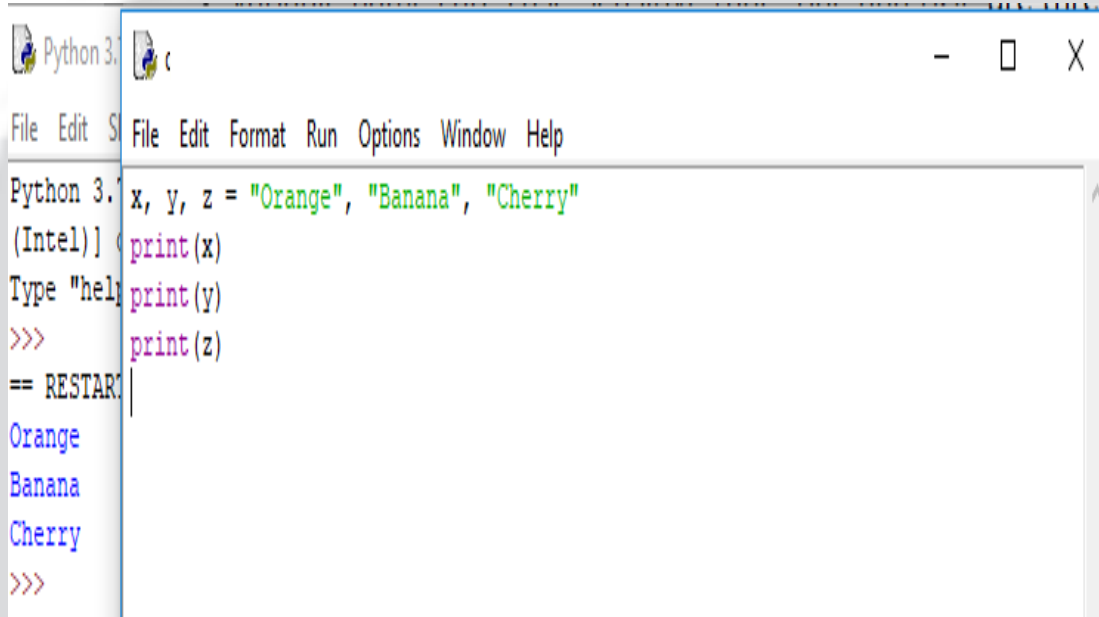
**Python** allows you to assign values to multiple variables in one line.

### Example

```
x, y, z = "Orange", "Banana", "Cherry"
print(x)
print(y)
print(z)
```

## Run the example

نتيجة تشغيل الكود



```
Python 3.7.4 Shell
File Edit Shell Format Run Options Window Help
Python 3.7.4 Shell (Intel)
Type "help()" for help
>>> x, y, z = "Orange", "Banana", "Cherry"
>>> print(x)
Orange
>>> print(y)
Banana
>>> print(z)
Cherry
>>>
```

يمكنك تعريف عدة متغيرات لقيمة واحدة/لنفس القيمة

- And you can assign the same value to multiple variables in one line.

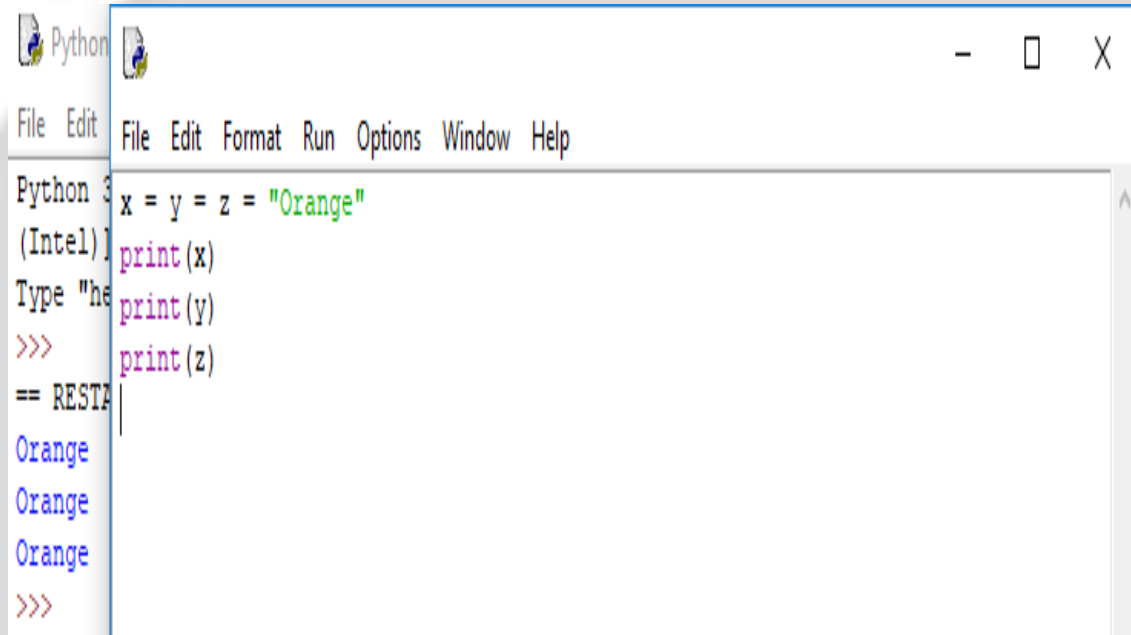
### Example

```
x = y = z = "Orange"
print(x)
print(y)
print(z)
```



## نتيجة تشغيل الكود

## Run the example



```
Python 3
File Edit Format Run Options Window Help
x = y = z = "Orange"
print(x)
print(y)
print(z)
>>>
Orange
Orange
Orange
>>>
```

## ➤ Output Variables

إخراج/طباعة المتغيرات

لطباعة القيم التي قمنا بتخزينها في المتغير نستخدم عبارة **print** ونستخدم علامة + الزائد أثناء الطباعة لدمج النص مع قيمة المتغير

The **Python print** statement is often used to output variables.

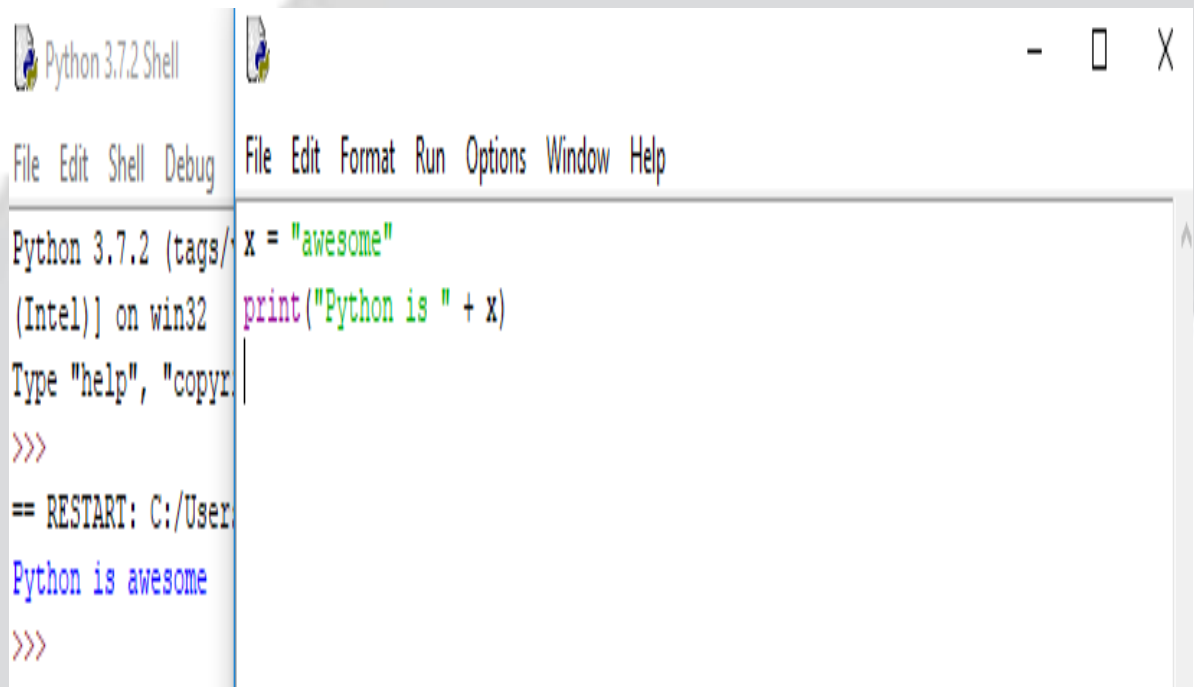
To combine both text and a variable, **Python** uses the + character.

## Example

```
x = "awesome"
print("Python is " + x)
```

نتيجة تشغيل الكود

## Run the example



```
Python 3.7.2 Shell
File Edit Shell Debug
Python 3.7.2 (tags/
(Intel)) on win32
Type "help", "copyr
>>>
== RESTART: C:/User
Python is awesome
>>>
```

يمكنك الجمع بين متغيرين بواسطة العلامة +

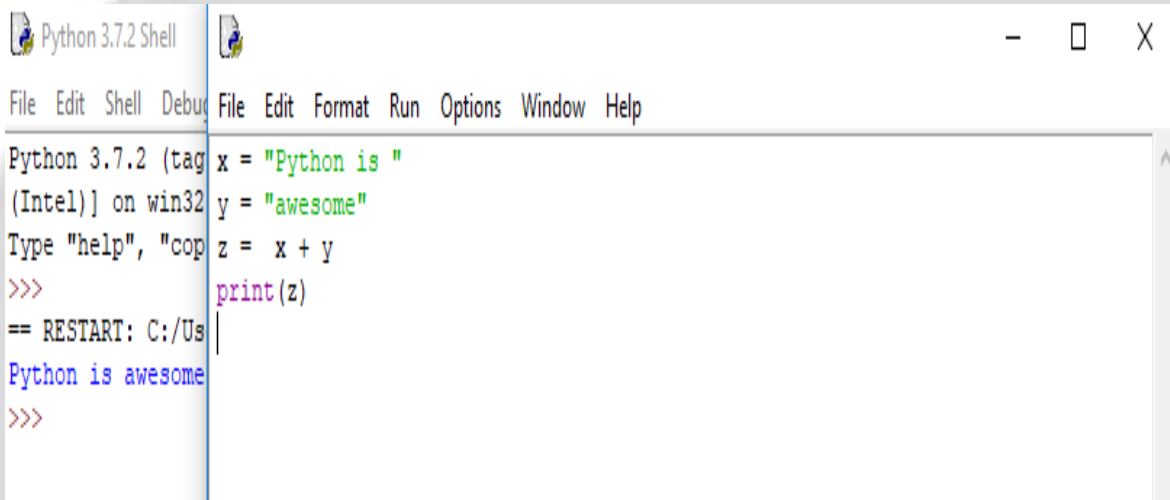
- You can also use the + character to add a variable to another variable.

### Example

```
x = "Python is "
y = "awesome"
z = x + y
print(z)
```

## Run the example

## نتيجة تشغيل الكود



```
Python 3.7.2 Shell
File Edit Shell Debug
Python 3.7.2 (tags/v3.7.2:1b3cc51e2, Oct 24 2019) [AMD64] on win32
Type "help", "copyright()", "credits()" and "quit()" for more
>>> x = "Python is "
>>> y = "awesome"
>>> z = x + y
>>> print(z)
Python is awesome
>>>
```

قم بتطبيق المثال على جهازك

أما بالنسبة للأرقام، يعمل **+** كعلامة زائد كما في العمليات الرياضية

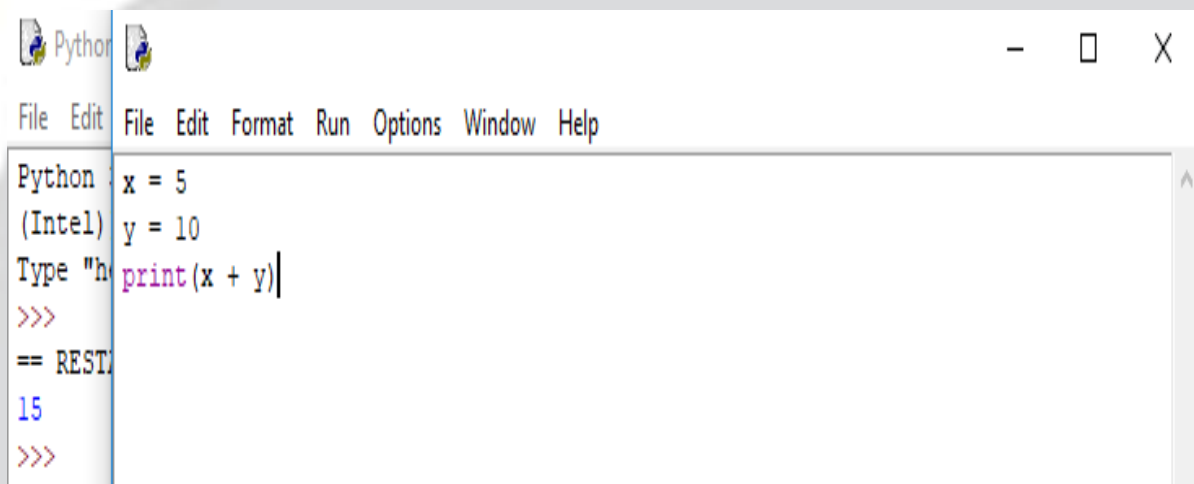
- For numbers, the **+** character works as a mathematical operator.

## Example

```
x = 5
y = 10
print(x + y)
```

## Run the example

نتيجة تشغيل الكود



```
Python
File Edit Format Run Options Window Help
Python x = 5
(Intel) y = 10
Type "h print(x + y)|
>>>
== REST
15
>>>
```

بايثون لا تسمح لك بالجمع بين نص `string` و رقم

- If you try to combine a string and a number, **Python** will give you an error.

## Example

طبّق المثال لترى النتيجة

```
x = 5
y = "John"
print(x + y)
```

ومبروك عليك أول **error** في بايثون!

## نتيجة تشغيل الكود

## Run the example

```
x = 5
y = "John"
print(x + y)
```

```
C:\Users\My Name>python demo_variables_test.py
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

أتممت درسك الثالث يا بطل!

روابط قد تفيدك

Check the links below:

- [Python 3 Programming Tutorial: Variables](#)
- [Python Programming #2 - Variables and Multiple Assignment](#)
- [Variables المتغيرات Python in Arabic شرح بايثون بالعربي 7 - 7](#)

طبق ما تعلمته معنا في درسنا الثالث  
ولا تنسى مشاركتنا أكوادك.

# اليوم الرابع

صديقنا الرائع، لقد وصلت لدرسك الرابع.. استمتع

## الأعداد في لغة البايثون

### Python Numbers

أنواع الأعداد في لغة بايثون

There are three numeric types in **Python**:

- Int. الأعداد الصحيحة
- Float. الأعداد العشرية ( أي التي تحتوي على فاصلة )
- Complex. الأعداد المركبة ( للعمليات الحسابية المعقدة )

نقوم بإنشاء المتغير العددي عندما نقوم بإسناد قيمة عديدة إليه، عند عملية الإسناد نكون قد قمنا بحجز مكان في الذاكرة لهذه القيمة.

➤ Variables of numeric types are created when you assign a value to them.

### Example

```
x = 1      # int
y = 2.8    # float
z = 1j     # complex
```

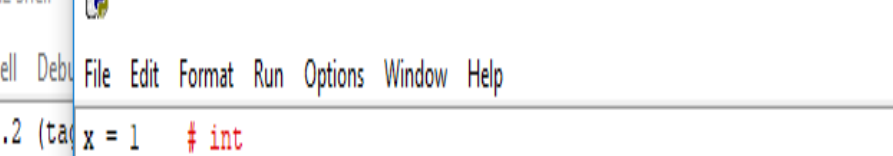
لعرض أو معرفة نوع المتغير في لغة البايثون نستخدم دالة **type()**

## Example

```
print (type(x))
print (type(y))
print (type(z))
```

## Run the example

## نتيجة تشغيل الكود



The screenshot shows a Python 3.7.2 Shell window with the following content:

```
Python 3.7.2 (tags/b3d93935ba/2019-09-17 00:00:00) on win32
Type "help", "copyright()", "credits()", "license()", "quit()", or "exit()"
>>> x = 1 # int
>>> y = 2.8 # float
>>> z = 1j # complex
>>> print(type(x))
<class 'int'>
>>> print(type(y))
<class 'float'>
>>> print(type(z))
<class 'complex'>
>>>
```



## الأعداد الصحيحة

## ➤ Int

هي الأعداد الموجبة والسالبة، والأعداد الصحيحة الطويلة، لا تحتوي على الفاصلة العشرية.

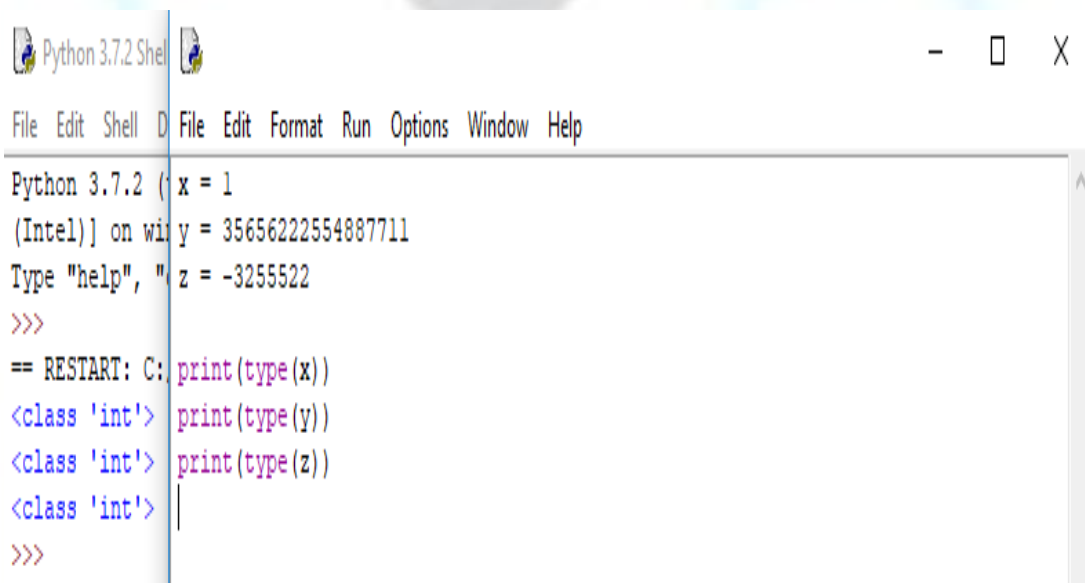
Int, or integer, is a whole number, positive or negative, without decimals, of unlimited length.

**Example**

```
x = 1
y = 35656222554887711
z = -3255522

print(type(x))
print(type(y))
print(type(z))
```

## نتيجة تشغيل الكود

**Run the example**

```
Python 3.7.2 Shell
File Edit Shell D File Edit Format Run Options Window Help
Python 3.7.2 (x = 1
(Intel)] on win y = 35656222554887711
Type "help", " z = -3255522
>>>
== RESTART: C: print(type(x))
<class 'int'> print(type(y))
<class 'int'> print(type(z))
<class 'int'> |
>>>
```

## ➤ Float

## الأعداد العشرية

هي الأعداد الموجبة أو السالبة التي تحتوي على الفاصلة العشرية.

Float, or "floating point number" is a number, positive or negative, containing one or more decimals.

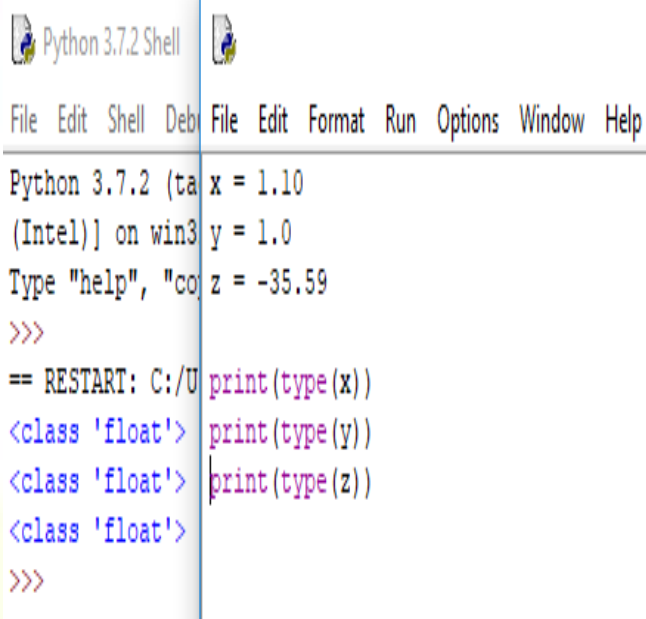
### Example

```
x = 1.10
y = 1.0
z = -35.59
```

```
print(type(x))
print(type(y))
print(type(z))
```

### Run the example

نتيجة تشغيل الكود



```
Python 3.7.2 (tags/beta/3.7.2:20200228) on win32
File Edit Shell Debug File Edit Format Run Options Window Help
x = 1.10
y = 1.0
z = -35.59
Type "help", "copyright()", or "credits()" for more
>>>
== RESTART: C:/Users/.../Python372/Python Shell ==
print(type(x))
<class 'float'>
print(type(y))
<class 'float'>
print(type(z))
<class 'float'>
>>>
```

## الأعداد العشرية تضمن أيضا الأعداد المرفوعة للقوى ١٠

- Float can also be scientific numbers with an "e" to indicate the power of 10.

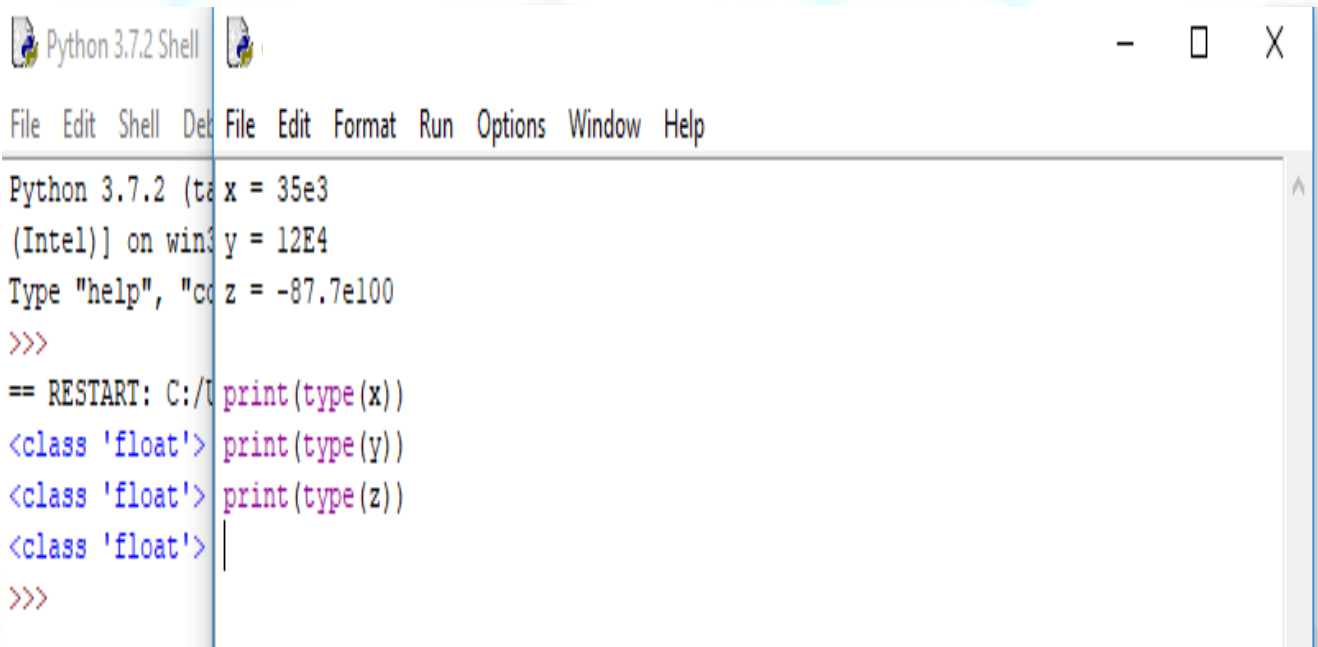
### Example

```
x = 35e3
y = 12E4
z = -87.7e100

print(type(x))
print(type(y))
print(type(z))
```

### نتيجة تشغيل الكود

### Run the example



```
Python 3.7.2 Shell
File Edit Shell Debug File Edit Format Run Options Window Help
Python 3.7.2 (tags/beta/0) on win32 [AMD64]
Type "help", "copyright()", "credits()", or "quit()" for more
>>> x = 35e3
>>> y = 12E4
>>> z = -87.7e100
>>>
== RESTART: C:\Users\user\AppData\Local\Programs\Python\Python37-32\python.exe ==
>>> print(type(x))
<class 'float'>
>>> print(type(y))
<class 'float'>
>>> print(type(z))
<class 'float'>
>>>
```

## ➤ Complex

## الأعداد المركبة

لغة بايثون تزودك بهذا النوع وهو يُكتب كالتالي ( الجزء الحقيقي + الجزء التخيلي ) وينتهي بحرف j/J مباشرة بعد العدد حتى يتم تمييزه بأنه عدد مركب وليس عدد عادي.

Complex numbers are written with a "j" as the imaginary part.

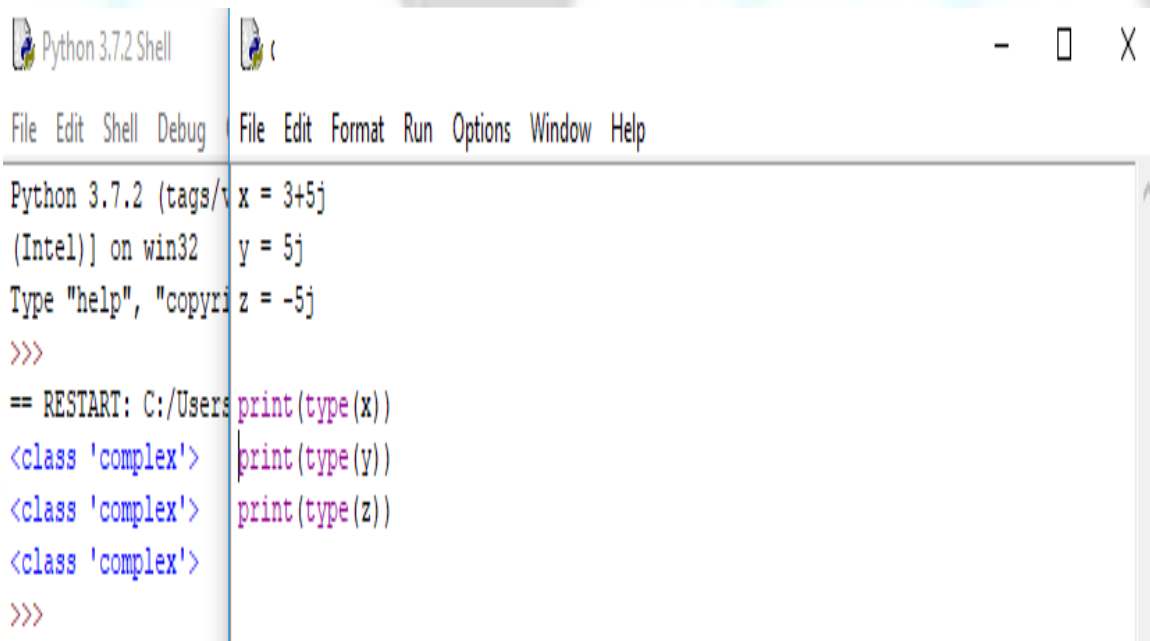
### Example

```
x = 3+5j
y = 5j
z = -5j

print(type(x))
print(type(y))
print(type(z))
```

### Run the example

### نتيجة تشغيل الكود



```
Python 3.7.2 Shell
File Edit Shell Debug File Edit Format Run Options Window Help
Python 3.7.2 (tags/win32) on win32
Type "help", "copyright()", "credits()", or "quit()" for more
>>> x = 3+5j
>>> y = 5j
>>> z = -5j
>>>
>>> print(type(x))
<class 'complex'>
>>> print(type(y))
<class 'complex'>
>>> print(type(z))
<class 'complex'>
>>>
```

## دالة تحويل أنواع الأعداد

### ➤ Type Conversion

You can convert from one type to another with the

`int()`, `float()`, and `complex()` methods.

يمكنك التحويل بين أنواع الأعداد عن طريق هذه الدوال

### Example

ركز في مثالنا الرائع!

ولا تنسى تجربته على جهازك أيضا

```
x = 1 # int
y = 2.8 # float
z = 1j # complex

#convert from int to float:
a = float(x)

#convert from float to int:
b = int(y)

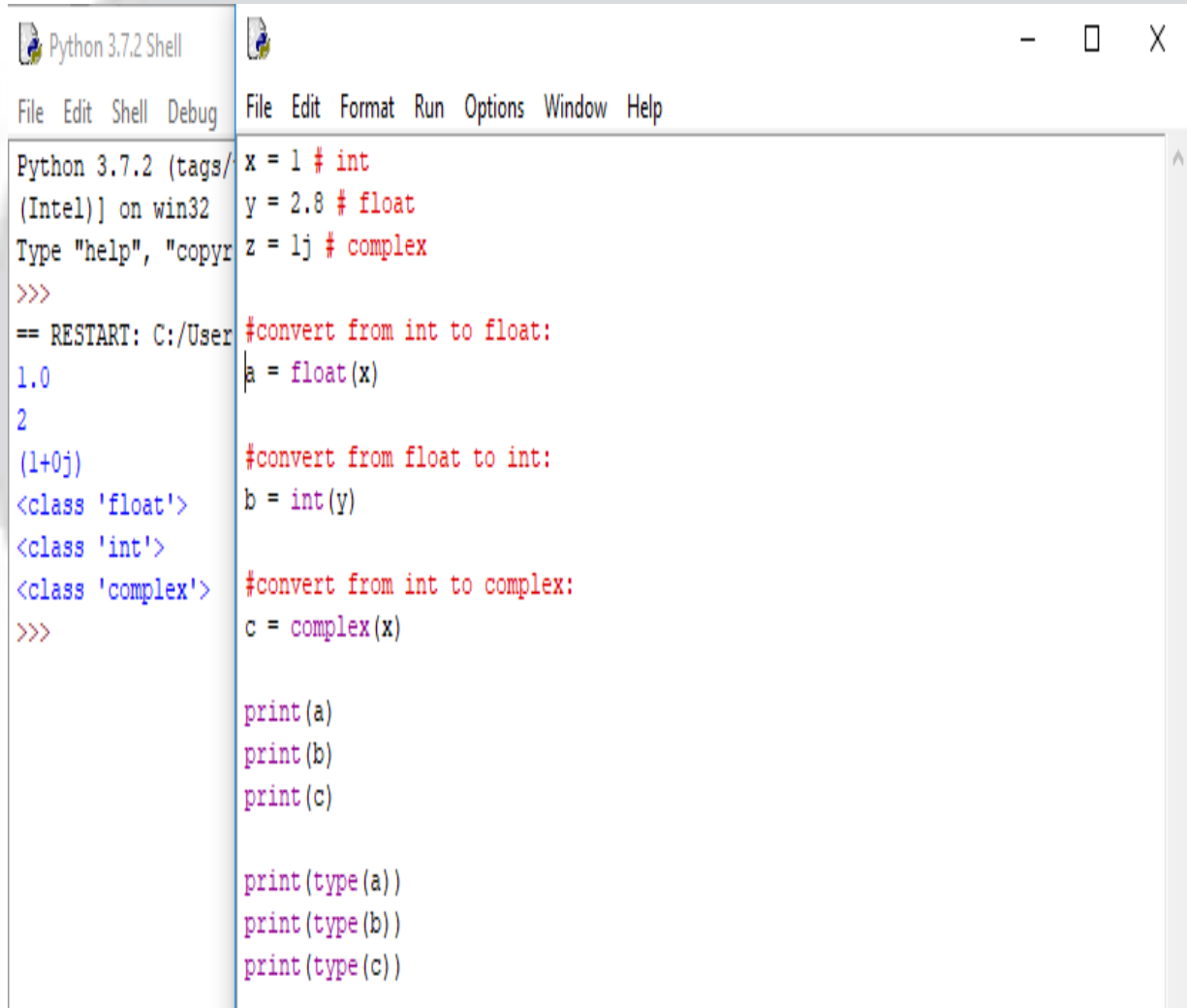
#convert from int to complex:
c = complex(x)

print(a)
print(b)
print(c)

print(type(a))
print(type(b))
print(type(c))
```

## نتيجة تشغيل الكود

## Run the example



```
Python 3.7.2 Shell
File Edit Shell Debug
Python 3.7.2 (tags/
(Intel)] on win32
Type "help", "copyr
>>>
== RESTART: C:/User
1.0
2
(1+0j)
<class 'float'>
<class 'int'>
<class 'complex'>
>>>

x = 1 # int
y = 2.8 # float
z = 1j # complex

#convert from int to float:
a = float(x)

#convert from float to int:
b = int(y)

#convert from int to complex:
c = complex(x)

print(a)
print(b)
print(c)

print(type(a))
print(type(b))
print(type(c))
```

**Note:**

You cannot convert complex numbers into another number type.

ملاحظة : لا يمكنك تحويل الأعداد المركبة إلى الأنواع العددية الأخرى

## العدد العشوائي

### ➤ Random Number

**random** هو موديل جاهز في لغة بايثون يحتوي على دوال تستخدم في العمليات الحسابية و الهندسية ومن هذه الدوال دالة **random()** تقوم بترجيع عدد عشوائي.

**Python** does not have a **random()** function to make a random number, but **Python** has a built-in module called **random** that can be used to make random numbers.

### Example

Import the random module, and display a random number between 1 and 9

```
import random  
  
print(random.randrange(1,10))
```

### نتيجة تشغيل الكود

### Run the example

I ran the example 4 times.



```
Python 3.7.2 Shell
File Edit Shell File Edit Format Run Options Window Help
Python 3.7.2 (Intel) on
Type "help" for help
>>> import random
>>> print(random.randrange(1,10))
7
>>>
== RESTART: ==
4
>>>
== RESTART: ==
1
>>>
== RESTART: ==
2
>>>
```

وصلت إلى نهاية الدرس الرابع  
درسك القادم سيكون أكثر متعة

ألق نظرة سريعة على هذه الروابط

For more information See below:

- [Python Programming Tutorial for Beginners - Numeric Data Types](#)
- [Learn Python in Arabic #12 - 12 import and random randint والاستيراد و الارقام العشوائية](#)

طبق ما تعلمته معنا في درسنا الرابع  
ولا تنسى مشاركتنا أكوادك



# اليوم الخامس

```
basket = x + y + z
```

اكتب كود يقوم بعمل المعادلة التالية:

```
x= apple , y = orange , z= limon
```

علمًا أن قيمة كل المتغيرات كالتالي:

ثم اطبع النتيجة ؟

معرفة إضافية

هناك طرق لفصل الـ **string** عبر دالة **len()** بالإمكان الاطلاع عليها في الرابط التالي:

<https://bit.ly/2N996uB>