

# Execute Python Syntax

Python syntax can be executed by writing directly in the Command Line.

```
In [1]: print("Hello, World!")
```

Hello, World!

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

```
C:\Users\Your Name>python myfile.py
```

## Python Indentation

**Indentation refers to the spaces at the beginning of a code line.**

Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.

Python uses indentation to indicate a block of code.

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

Five is greater than two!

```
In [3]: # Python gives you an error if you skip the indentation.

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

```
File "<ipython-input-3-e4c6c63ccc37>", line 4
    print("Five is greater than two!")
    ^
```

**IndentationError:** expected an indented block

```
In [4]: # The number of spaces is up to you as a programmer, but it has to be at least one.

if 5 > 2:
    print("Five is greater than two!")
if 6 > 2:
    print("Six is greater than two!")
```

Five is greater than two!  
Six is greater than two!

```
In [5]: # The same number of spaces have to be used in the same block of code, otherwise Pyt

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

```
File "<ipython-input-5-e811d4179bd9>", line 5
    print("Five is greater than two!")
```

^  
**IndentationError:** unexpected indent

```
In [6]: # No error.  
  
if 5 > 2:  
    print("Five is greater than two!")  
    print("Six is greater than two!")
```

Five is greater than two!  
Six is greater than two!

## Python Variables

In Python, variables are created when you assign a value to it.

```
In [7]: # Creating Variables.  
  
x = 5  
y = "Hello, World!"
```

Python has no command for declaring a variable.

## Comments

Python has commenting capability for the purpose of in-code documentation.

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

```
In [8]: # Comments in Python.  
  
#This is a comment.  
print("Hello, World!")
```

Hello, World!

## Python Built-in Functions

Python has several functions that are readily available for use. These functions are called built-in functions.

## Help Function

The Python help function is used to display the documentation of modules, functions, classes, keywords, etc.

The help function has the following syntax:

help([object])

```
In [9]: # Displays the documentation of the predefined print function in Python.  
  
help(print)
```

Help on built-in function print in module builtins:

```
print(...)
  print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

  Prints the values to a stream, or to sys.stdout by default.
  Optional keyword arguments:
  file: a file-like object (stream); defaults to the current sys.stdout.
  sep: string inserted between values, default a space.
  end: string appended after the last value, default a newline.
  flush: whether to forcibly flush the stream.
```

## Check GPU Version

In [10]: `!nvidia-smi`

Thu Mar 24 16:23:20 2022

NVIDIA-SMI 472.47										Driver Version: 472.47										CUDA Version: 11.4									
GPU Name				TCC/WDDM				Bus-Id				Disp.A				Volatile Uncorr. ECC													
Fan		Temp		Perf		Pwr:Usage/Cap		Memory-Usage				GPU-Util				Compute M. MIG M.													
=====										=====										=====									
0		NVIDIA GeForce ...		WDDM		00000000:01:00.0				Off				N/A															
N/A		60C		P8		N/A / N/A		119MiB / 2048MiB				0%				Default N/A													
-----										-----										-----									
Processes:																													
GPU		GI		CI		PID		Type		Process name										GPU Memory Usage									
		ID		ID																									
=====										=====										=====									
No running processes found																													
-----										-----										-----									

## Present Working Directory

In [11]: `pwd`

Out[11]: `'C:\\Users\\shrav'`

## Is Command

Particularly, the `ls` command lets you see the content of your current working directory.

In [12]: `ls`

Volume in drive C is Windows  
Volume Serial Number is B8C0-2582

Directory of C:\Users\shrav

24-03-2022	16:21	<DIR>	.
27-10-2021	00:42	<DIR>	..
12-01-2022	20:23	<DIR>	.astropy
12-03-2022	20:25		1,154 .bash_history
15-02-2022	10:13	<DIR>	.conda
09-09-2021	12:08		25 .condarc

```

09-09-2021 12:08 <DIR> .continuum
18-03-2022 02:29 <DIR> .cufflinks
14-02-2022 23:28 1,294 .dbshell
15-02-2022 11:20 <DIR> .docker
12-03-2022 15:35 288 .gitconfig
20-11-2021 19:35 <DIR> .idlerc
24-03-2022 16:21 <DIR> .ipynb_checkpoints
08-09-2021 01:12 <DIR> .ipython
06-12-2021 23:51 <DIR> .jupyter
09-09-2021 12:27 <DIR> .matplotlib
11-02-2022 10:19 0 .mongorc.js
24-03-2022 16:11 <DIR> .VirtualBox
08-09-2021 00:59 <DIR> .vscode
05-12-2021 12:51 3,717,524 000 - Ineuron 1001.ipynb
23-12-2021 08:52 275,714 000_Ineuron_1001.ipynb
24-03-2022 16:21 69,667 003_Python Syntax.ipynb
07-09-2021 15:08 <DIR> 3D Objects
20-02-2022 23:34 <DIR> anaconda3
27-01-2022 02:25 6,706 Assignment 1 INURON.ipynb
05-01-2022 13:06 3,696 Aviation.ipynb
27-10-2021 01:42 <DIR> Contacts
31-12-2021 16:10 23,579 D$B.ipynb
14-02-2022 01:07 <DIR> Documents
31-12-2021 16:05 12,742 Dot_and_Box_Game_using_Python.ipynb
25-12-2021 12:30 3,290 dots and line.ipynb
24-03-2022 16:21 <DIR> Downloads
11-10-2021 13:36 <DIR> Dropbox
27-10-2021 01:42 <DIR> Favorites
25-12-2021 12:43 10,677 grid.ipynb
04-01-2022 19:23 5,247 lecture5.ipynb
27-10-2021 01:42 <DIR> Links
27-10-2021 01:42 <DIR> Music
08-02-2022 14:12 92,260 NUMPY INURON .ipynb
26-01-2022 16:34 576 NUMPY INURON -Copy1.ipynb
24-03-2022 10:03 <DIR> OneDrive
16-02-2022 01:22 <DIR> Postman
21-12-2021 18:02 <DIR> PycharmProjects
27-10-2021 01:42 <DIR> Saved Games
12-03-2022 23:27 <DIR> seaborn-data
27-10-2021 01:42 <DIR> Searches
14-11-2021 02:11 9,037 Test.xlsx
14-11-2021 02:41 65 test1.txt
13-11-2021 17:29 576 Untitled.ipynb
27-11-2021 21:53 72 Untitled1.ipynb
28-11-2021 01:08 3,123 Untitled2.ipynb
28-11-2021 01:28 1,135 Untitled3.ipynb
31-12-2021 15:58 72 Untitled4.ipynb
26-01-2022 16:32 72 Untitled5.ipynb
27-01-2022 02:02 72 Untitled6.ipynb
30-10-2021 22:06 <DIR> Videos
22-03-2022 09:48 <DIR> VirtualBox VMs
26 File(s) 4,238,663 bytes
31 Dir(s) 264,318,640,128 bytes free

```

## Python print() Function

The `print()` function prints the specified message to the screen, or other standard output device.

The message can be a string, or any other object, the object will be converted into a string before written to the screen.

## Syntax

```
print(object(s), sep=separator, end=end, file=file, flush=flush)
```

## Parameter Values

Parameter	Description
<i>object(s)</i>	Any object, and as many as you like. Will be converted to string before printed
<i>sep='separator'</i>	Optional. Specify how to separate the objects, if there is more than one. Default is ' '
<i>end='end'</i>	Optional. Specify what to print at the end. Default is '\n' (line feed)
<i>file</i>	Optional. An object with a write method. Default is sys.stdout
<i>flush</i>	Optional. A Boolean, specifying if the output is flushed (True) or buffered (False). Default is False

```
In [13]: help(print)
```

Help on built-in function print in module builtins:

```
print(...)
  print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

  Prints the values to a stream, or to sys.stdout by default.
  Optional keyword arguments:
  file: a file-like object (stream); defaults to the current sys.stdout.
  sep:   string inserted between values, default a space.
  end:   string appended after the last value, default a newline.
  flush: whether to forcibly flush the stream.
```

```
In [14]: # No error.
print("Hello World")
```

Hello World

```
In [15]: # Closing " is missing. Hence, Python gives you an error.
print("Hello World)
```

```
File "<ipython-input-15-6a48aad13a41>", line 2
    print("Hello World)
          ^
```

**SyntaxError:** EOL while scanning string literal

```
In [16]: # No error.
```

```
print('Hello World')
```

Hello World

```
In [17]: # This code is valid in Python 2.x, however it will not work in Python 3.x version.

print "This is my first program"
```

```
File "<ipython-input-17-96839505bd2e>", line 3
```

```
    print "This is my first program"
```

^

**SyntaxError:** Missing parentheses in call to 'print'. Did you mean print("This is my first program")?

```
In [18]: # Print more than one object.

print("Hello", "how are you?")
```

Hello how are you?

```
In [19]: # Print a tuple.

x = ("apple", "banana", "cherry")
print(x)
```

('apple', 'banana', 'cherry')

```
In [20]: # Print two messages, and specify the separator.

print("Hello", "how are you?", sep="---")
```

Hello---how are you?