

# Python From Scratch

## Getting Started With Python From here

### *Lesson 1 Content*

- What is Python?
- What can Python do?
- Why Python?
- Good to know
- Python Syntax compared to other programming languages
- Python Install
- Python QuickStart
- The Python Command Line
- Python Syntax
- Python Indentation
- Python Variables
- Python Comments
- Creating a Comment
- Multi Line Comments

## What is Python?

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

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

## What can Python do?

- 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?

- 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-oriented way or a functional way.

## Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

## Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

# Python Install

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 or on Mac open the Terminal and type:

```
python -version
```

- If you find that you do not have Python installed on your computer, then you can download it for free from the following website: <https://www.python.org/>
- 

## Python Quickstart

Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.

- The way to run a python file is like this on the command line:

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

Where "helloworld.py" is the name of your python file.

- Let's write our first Python file, called helloworld.py, which can be done in any text editor.

```
helloworld.py  
print("Hello, World!")
```

- Simple as that. Save your file. Open your command line, navigate to the directory where you saved your file, and run:

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

- The output should read:

```
Hello, World!
```

**Congratulations, you have written and executed your first Python program.**

---

## The Python Command Line

To test a short amount of code in python sometimes it is quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.

Type the following on the Windows, Mac or Linux command line:

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

Or, if the "python" command did not work, you can try "py":

```
C:\Users\Your Name>py
```

- From there you can write any python, including our hello world example from earlier in the tutorial:

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

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

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

- Which will write "Hello, World!" in the command line:

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

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

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

```
Hello, World!
```

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

```
exit()
```

---

## Python Syntax

As we learned, Python syntax can be executed by writing directly in the Command Line:

```
>>> 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.

### Example

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

- Python will give you an error if you skip the indentation:

Syntax Error:

- The number of spaces is up to you as a programmer, the most common use is four, but it has to be at least one.

### Example

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

- You have to use the same number of spaces in the same block of code, otherwise Python will give you an error:

### Example

Syntax Error:

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

---

## Python Variables

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

### Example

Variables in Python:

```
x = 5
y = "Hello, World!"
```

Python has no command for declaring a variable.

You will learn more about variables in the Python Variables chapter.

---

## 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:

## Python Comments

- 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.

## Creating a Comment

- Comments starts with a `#`, and Python will ignore them:

### Example

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

- 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`

- A comment does not have to be text that explains the code, it can also be used to prevent Python from executing code:

### Example

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

## 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

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

Or, not quite as intended, you can use a multiline string.

- 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!")
```

- As long as the string is not assigned to a variable, Python will read the code, but then ignore it, and you have made a multiline comment.