

# **Python from Scratch**

## **Lesson 2**

- **The Python Command Line**
- **Python Syntax**
- **Python Indentation**
- **Python Variables**
- **Python Comments**
- **Creating a Comment**
- **Multi Line Comments**

## 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:

### Example

Comments in Python:

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

## 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 start 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.