Python Getting Started

Dr. Labeed Al-Saad

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):

n 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

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.

When you want to that in python editor, you hae to write this python command:

In [2]:
```
print("Hello, World!")
```

```
Hello, World!
```

Python Version

To check the Python version of the editor, you can find it by importing the sys module:

Example:

Check the Python version of the editor:

In [4]:
```
import sys

print(sys.version)
```

```
3.11.7 | packaged by Anaconda, Inc. | (main, Dec 15 2023, 18:05:47) [MSC v.1916 6
4 bit (AMD64)]
```

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

Execute Python Syntax

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

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

Hello, World!

**You creating a python file, 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:

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

```
Five is greater than two!
```

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

Example:

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

```
  Cell In[6], line 2
    print("Five is greater than two!")
    ^
IndentationError: expected an indented block after 'if' statement on line 1
```

**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:

In [7]:
```
if 5 > 2:
 print("Five is greater than two!")
if 5 > 2:
        print("Five is greater than two!")
```

```
Five is greater than two!
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:

In [8]:
```
if 5 > 2:
 print("Five is greater than two!")
        print("Five is greater than two!")
```

```
  Cell In[8], line 3
    print("Five is greater than two!")
    ^
IndentationError: unexpected indent
```

Python Variables

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

Example:

Variables in Python:

In [9]:
```
x = 5
y = "Hello, World!"

print(x)
print(y)
```

```
5
Hello, World!
```

**Python has no command for declaring a variable.

Python Comments

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

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.

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

Example:

Comments in Python:

In [10]:
```
#This is a comment.
print("Hello, World!")
```

```
Hello, World!
```

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

Example

In [11]:
```
print("Hello, World!") #This is a comment
```

```
Hello, World!
```

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

Example:

In [12]:
```
#print("Hello, World!")
print("Cheers, Mate!")
```

```
Cheers, Mate!
```

Multiline Comments

Python does not really have a syntax for multiline comments.

To add a multiline comment you could insert a # for each line:

Example:

In [13]:
```
#This is a comment
#written in
#more than just one line
print("Hello, World!")
```

```
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:

In [14]:
```
"""
This is a comment
written in
```

```
more than just one line
"""
print("Hello, World!")
```

```
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.