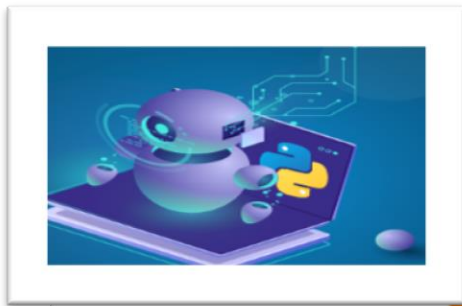


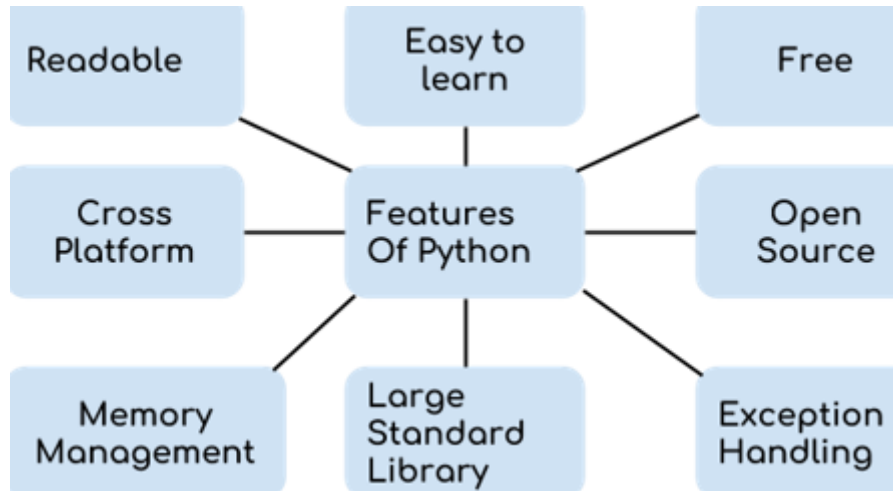
Data science and Machine learning with Python



Designed by Abdur Rahman Joy - MCSD, MCPC, MCSE, MCTS, OCJP, Sr. Technical Trainer for VFX at IDB BISW (Scholarship program), and C#.net, R, Scala, Kotlin, JAVA, Android/IOS/Windows Mobile Apps, SQL server, Azure, Oracle, SharePoint Development, AWS , CEH, KALI Linux, Python, Data Science, Machine Learning ,Software Testing, Graphics, Multimedia and Game Developer at Joy Infosys and other premises like BITM, SkillsJob, PNTL, Leads Training and New Horizon inc , Cell #: +880-1712587348, email: jspaonline@gmail.com. Web URL: <http://www.joyinfosys.com/me>.

PYTHON

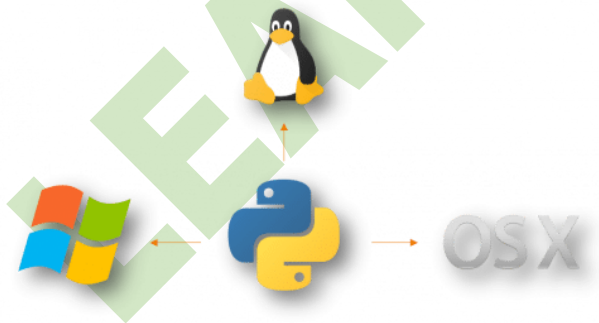
PYTHON IS A GENERAL-PURPOSE PROGRAMMING LANGUAGE THAT IS BECOMING EVER MORE POPULAR FOR DATA SCIENCE. COMPANIES WORLDWIDE ARE USING PYTHON TO HARVEST INSIGHTS FROM THEIR DATA AND GAIN A COMPETITIVE EDGE. UNLIKE OTHER PYTHON TUTORIALS, THIS COURSE FOCUSES ON PYTHON SPECIFICALLY FOR DATA SCIENCE. IN OUR INTRODUCTION TO PYTHON COURSE, YOU'LL LEARN ABOUT POWERFUL WAYS TO STORE AND MANIPULATE DATA, AND HELPFUL DATA SCIENCE TOOLS TO BEGIN CONDUCTING YOUR OWN ANALYSES.



Python is a free, open source programming language. So, all you have to do is install Python once and you can start working with it. And since Python is open source, you can contribute your own code to the community.



Python is also a cross platform compatible language. So, what does this mean? Well, you can install and run Python on several operating systems. So, whether you have a Windows, Mac or Linux, you can be rest assured that Python will work on all these operating systems.



Python is also a great visualization tool. Python provides libraries such as matplotlib, seaborn and bokeh to create stunning visualizations.



Python is the most popular language for implementing machine learning and deep learning tasks. Today, every top organization is investing in the field of machine learning and majority of these machine learning applications are implemented with python at back-end.



To install the Python on your operating system, go to this link: <https://www.python.org/downloads/>. You will see a screen like this.

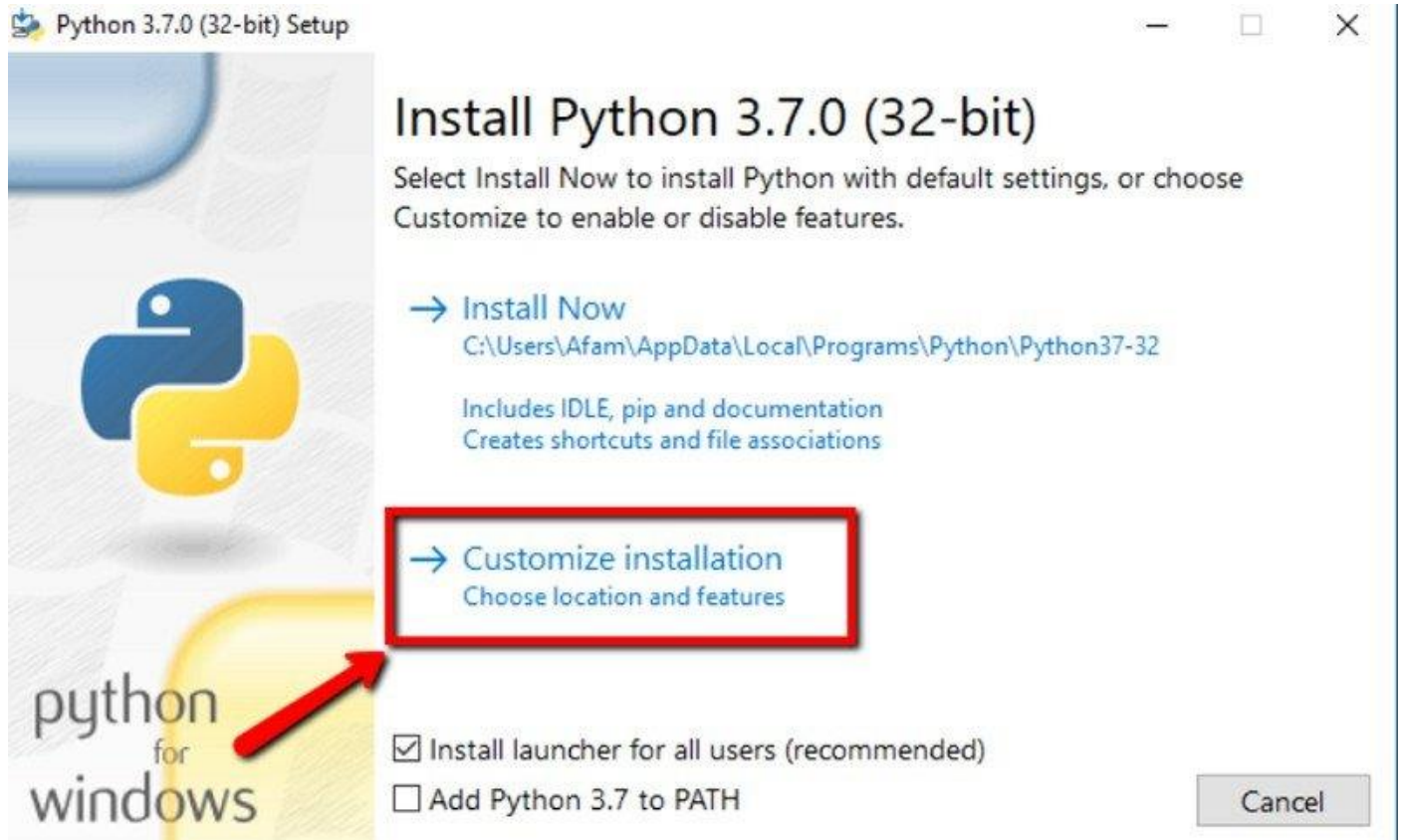


Designed by Abdur Rahman Joy - MCS D, MCPC, MCSE, MCTS, OCJP, Sr. Technical Trainer for VFX at IDB BISW (Scholarship program), and C#.net, R, Scala, Kotlin, JAVA, Android/IOS/Windows Mobile Apps, SQL server, Azure, Oracle, SharePoint Development, AWS , CEH, KALI Linux, Python, Data Science, Machine Learning ,Software Testing, Graphics, Multimedia and Game Developer at Joy Infosys and other premises like BITM, SkillsJob, PNTL, Leads Training and New Horizon inc , Cell #: +880-1712587348, email: jspaonline@gmail.com. Web URL: <http://www.joyinfosys.com/me>.

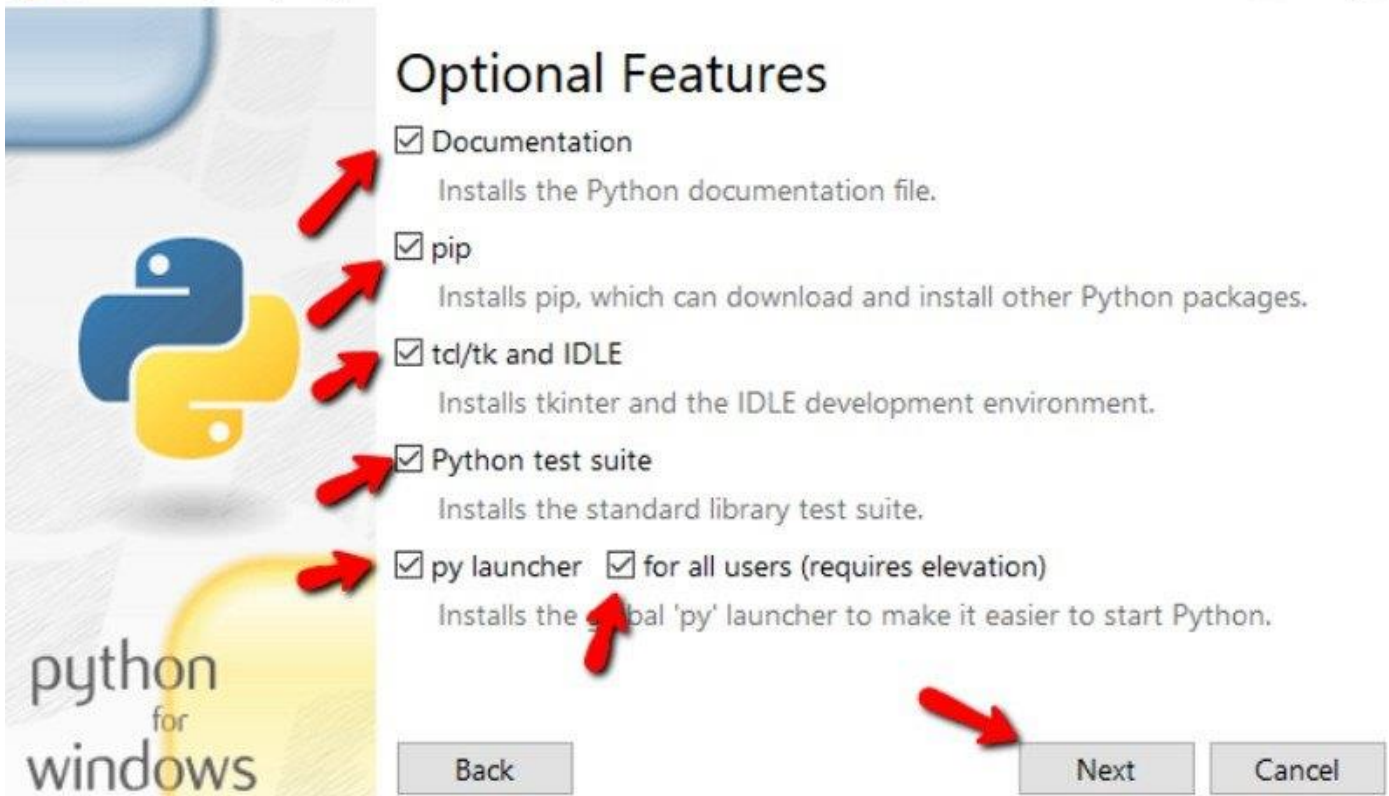
This is the **official Python website** and it will detect the operating system and based on that it would recommend you to download Python. Here I am using Mac OS X so it gave me the download options for Python 2 and Python 3 for Mac OS X. I would recommend you to download the **latest version of Python 3** (Python 3.6.4 in the screenshot).

Install Python from Installer

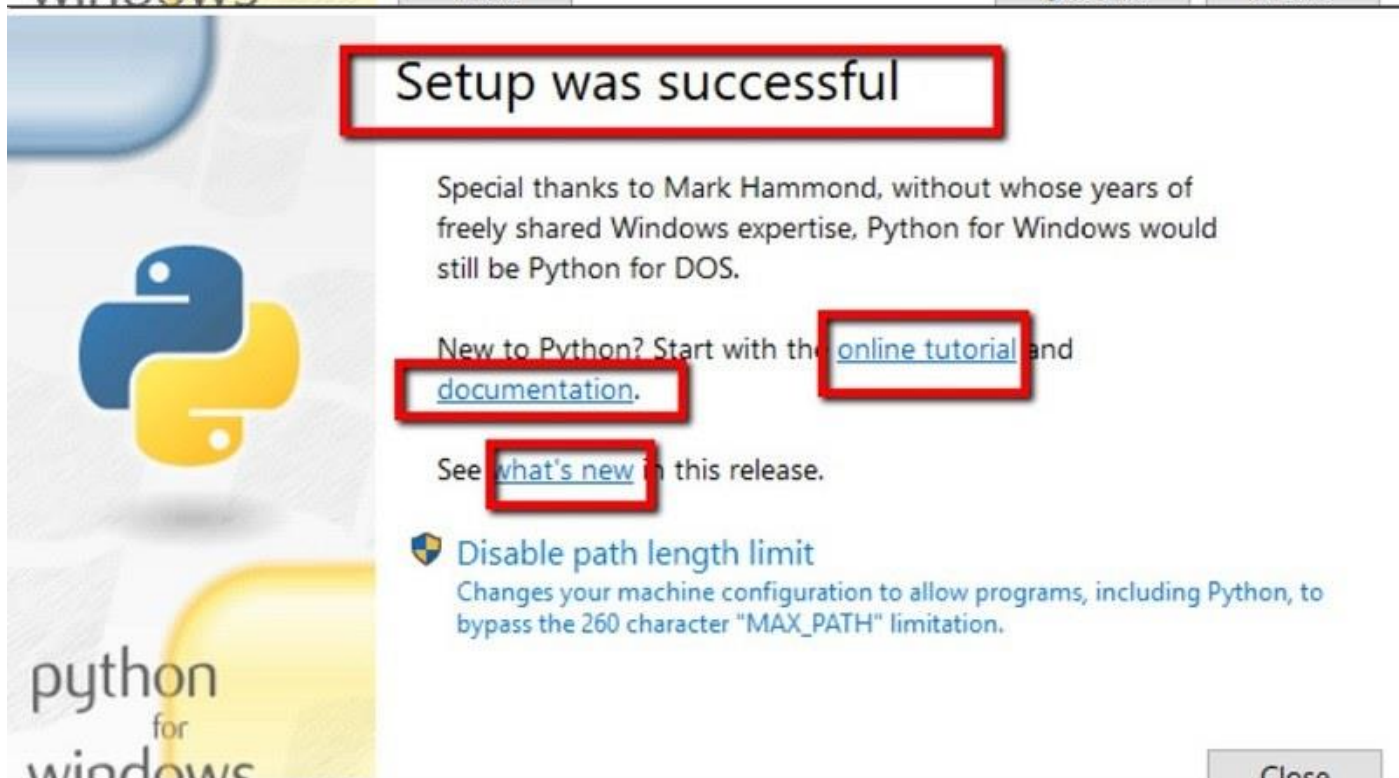
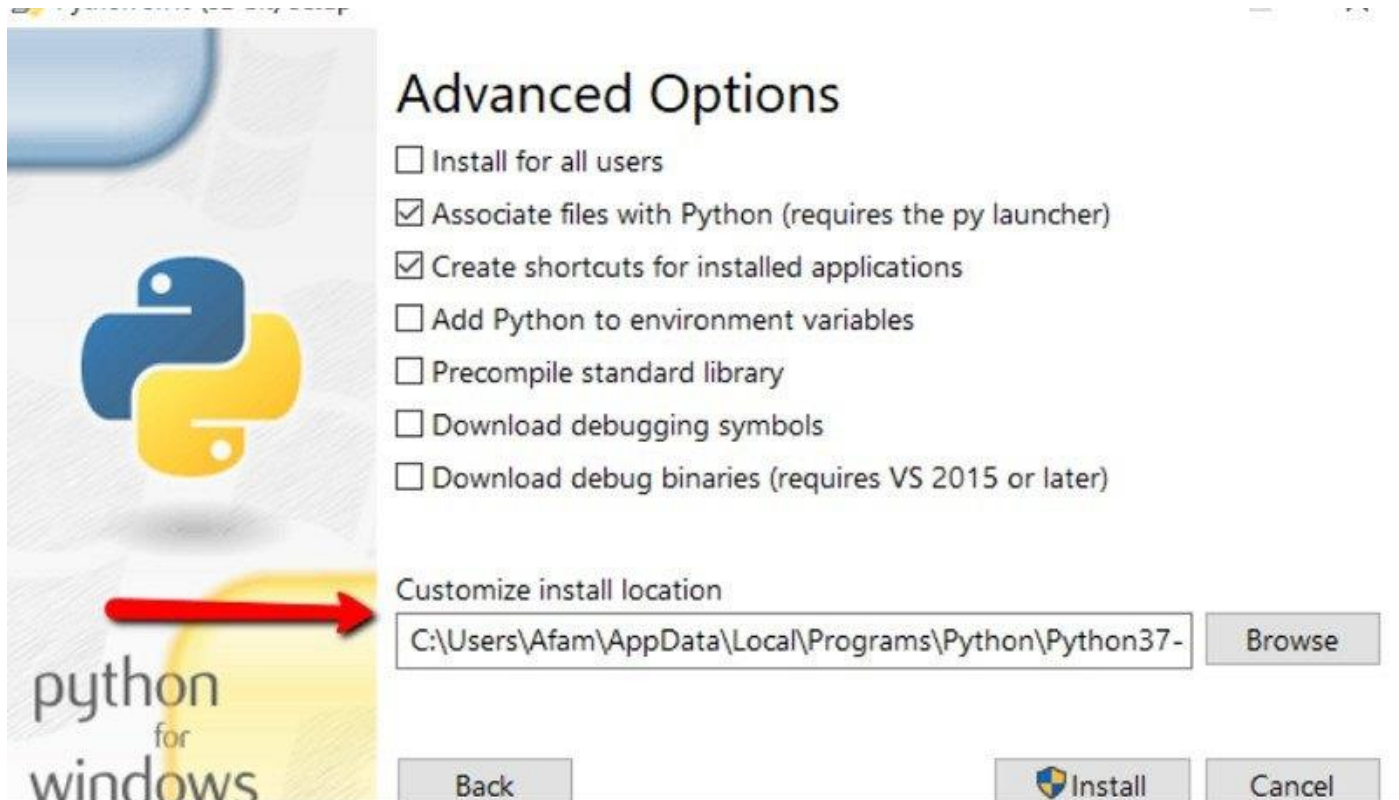
Right-click on the downloaded file, and click "Run as Administrator," which brings two options – choose "Customize Installation."



On the next window that appears, under optional features check all the checkboxes and click "Next."



Under the “Advanced Options” window it is important to set the location for the installation. I will usually set it to my C drive, which in many cases will be the default location. Next, click on “Install.” Close the installer when the install finishes.



Python is a very simple language, and has a very straightforward syntax. It encourages programmers to program without boilerplate (prepared) code. The simplest directive in Python is the "print" directive - it simply prints out a line (and also includes a newline, unlike in C).

Designed by Abdur Rahman Joy - MCSO, MCPC, MCSE, MCTS, OCJP, Sr. Technical Trainer for VFX at IDB BISW (Scholarship program), and C#.net, R, Scala, Kotlin, JAVA, Android/iOS/Windows Mobile Apps, SQL server, Azure, Oracle, SharePoint Development, AWS, CEH, KALI Linux, Python, Data Science, Machine Learning, Software Testing, Graphics, Multimedia and Game Developer at Joy Infosys and other premises like BITM, SkillsJob, PNTL, Leads Training and New Horizon inc, Cell #: +880-1712587348, email: jsponline@gmail.com. Web URL: <http://www.joyinfosys.com/me>.

There are two major Python versions, Python 2 and Python 3. Python 2 and 3 are quite different. This tutorial uses Python 3, because it more semantically correct and supports newer features.

For example, one difference between Python 2 and 3 is the **print** statement. In Python 2, the "print" statement is not a function, and therefore it is invoked without parentheses. However, in Python 3, it is a function, and must be invoked with parentheses.

To print a string in Python 3, just write:

```
print("This line will be printed.")
```

Variables and Types

Python is completely object oriented, and not "statically typed". You do not need to declare variables before using them, or declare their type. Every variable in Python is an object.

This tutorial will go over a few basic types of variables.

Numbers

Python supports two types of numbers - integers and floatingpoint numbers. (It also supports complex numbers, which will not be explained in this tutorial).

To define an integer, use the following syntax:

```
myint = 7  
print(myint)
```

To define a floating point number, you may use one of the following notations:

```
myfloat = 7.0  
print(myfloat)  
myfloat = float(7)  
print(myfloat)
```

Strings

Strings are defined either with a single quote or a double quotes.

```
mystring = 'hello'  
print(mystring)  
mystring = "hello"  
print(mystring)
```

The difference between the two is that using double quotes makes it easy to include apostrophes (whereas these would terminate the string if using single quotes)

```
mystring = "Don't worry about apostrophes"  
print(mystring)
```

There are additional variations on defining strings that make it easier to include things such as carriage returns, backslashes and Unicode characters.

Simple operators can be executed on numbers and strings:

```
one = 1  
two = 2  
three = one + two  
print(three)
```

```
hello = "hello"  
world = "world"  
helloworld = hello + " " + world  
print(helloworld)
```

Assignments can be done on more than one variable "simultaneously" on the same line like this

```
a, b = 3, 4  
print(a,b)
```

Mixing operators between numbers and strings is not supported:

This will not work!

```
one = 1  
two = 2  
hello = "hello"  
print(one + two + hello)
```


Python Keywords

Keywords are the reserved words in Python.

We cannot use a keyword as a variable name, function name or any other identifier. They are used to define the syntax and structure of the Python language.

In Python, keywords are case sensitive.

There are 33 keywords in Python 3.7. This number can vary slightly in the course of time.

All the keywords except `True`, `False` and `None` are in lowercase and they must be written as it is. The list of all the keywords is given below.

<code>False</code>	<code>class</code>	<code>finally</code>	<code>is</code>	<code>return</code>
<code>None</code>	<code>continue</code>	<code>for</code>	<code>lambda</code>	<code>try</code>
<code>True</code>	<code>def</code>	<code>from</code>	<code>nonlocal</code>	<code>while</code>
<code>and</code>	<code>del</code>	<code>global</code>	<code>not</code>	<code>with</code>
<code>as</code>	<code>elif</code>	<code>if</code>	<code>or</code>	<code>yield</code>
<code>assert</code>	<code>else</code>	<code>import</code>	<code>pass</code>	
<code>break</code>	<code>except</code>	<code>in</code>	<code>raise</code>	

Keywords in Python

Python Identifiers

An identifier is a name given to entities like class, functions, variables, etc. It helps to differentiate one entity from another.

Rules for writing identifiers

1. Identifiers can be a combination of letters in lowercase (**a to z**) or uppercase (**A to Z**) or digits (**0 to 9**) or an underscore `_`. Names like `myClass`, `var_1` and `print_this_to_screen`, all are valid example.
2. An identifier cannot start with a digit. `1variable` is invalid, but `variable1` is perfectly fine.
3. Keywords cannot be used as identifiers.

Python Statement

Instructions that a Python interpreter can execute are called statements. For example, `a = 1` is an assignment statement. If statement, `for` statement, `while` statement etc. are other kinds of statements which will be discussed later.

Multi-line statement

In Python, end of a statement is marked by a newline character. But we can make a statement extend over multiple lines with the line continuation character (`\`). For example:

```
a = 1 + 2 + 3 + \  
    4 + 5 + 6 + \  
    7 + 8 + 9
```

This is explicit line continuation. In Python, line continuation is implied inside parentheses (`()`), brackets [`[]`] and braces `{ }`. For instance, we can implement the above multi-line statement as

```
a = (1 + 2 + 3 +  
    4 + 5 + 6 +  
    7 + 8 + 9)
```

Here, the surrounding parentheses (`()`) do the line continuation implicitly. Same is the case with [`[]`] and `{ }`. For example:

```
colors = ['red',  
          'blue',  
          'green']
```

We could also put multiple statements in a single line using semicolons, as follows

```
a = 1; b = 2; c = 3
```

Python Indentation

Most of the programming languages like C, C++, Java use braces { } to define a block of code. Python uses indentation.

A code block (body of a function, loop etc.) starts with indentation and ends with the first unindented line. The amount of indentation is up to you, but it must be consistent throughout that block.

Generally four whitespaces are used for indentation and is preferred over tabs. Here is an example.

```
for i in range(1,11):  
    print(i)  
    if i == 5:  
        break
```

Python Comments

Comments are very important while writing a program. It describes what's going on inside a program so that a person looking at the source code does not have a hard time figuring it out. You might forget the key details of the program you just wrote in a month's time. So taking time to explain these concepts in form of comments is always fruitful.

In Python, we use the hash (#) symbol to start writing a comment.

It extends up to the newline character. Comments are for programmers for better understanding of a program. Python Interpreter ignores comment.

```
#This is a comment  
#print out Hello  
print('Hello')
```

Multi-line comments

If we have comments that extend multiple lines, one way of doing it is to use hash (#) in the beginning of each line. For example:

```
#This is a long comment  
#and it extends  
#to multiple lines
```

Another way of doing this is to use triple quotes, either ''' or """".

These triple quotes are generally used for multi-line strings. But they can be used as multi-line comment as well. Unless they are not docstrings, they do not generate any extra code.

```
""""This is also a  
perfect example of  
multi-line comments""""
```