Python Overview and Installation

Python is an open source scripting language which is high-level, interpreted, interactive and object-oriented. It is designed to be highly readable. The syntax of Python language is easy to understand and uses English keywords frequently.

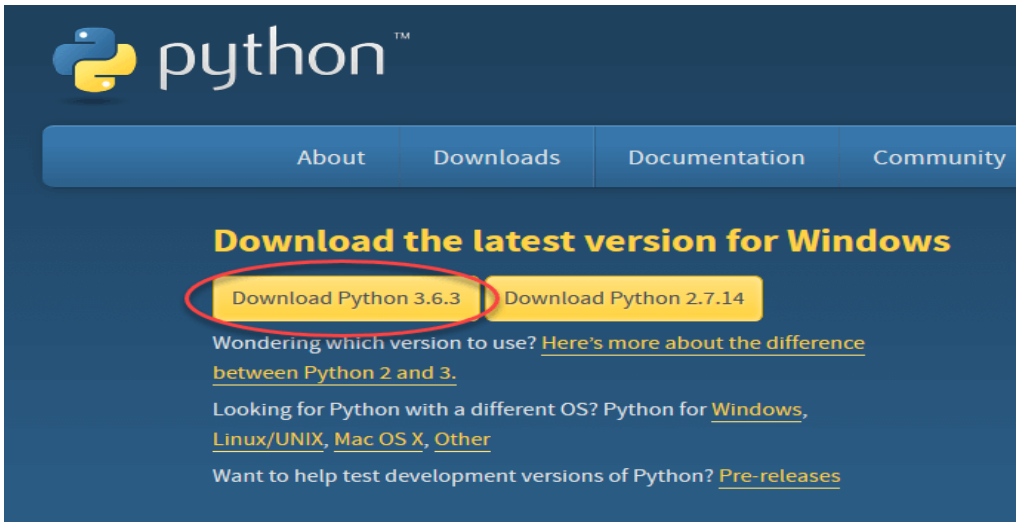
1. Install Python on Windows [Pycharm IDE]

PyCharm is a cross-platform editor developed by JetBrains. Pycharm provides all the tools you need for productive Python development.

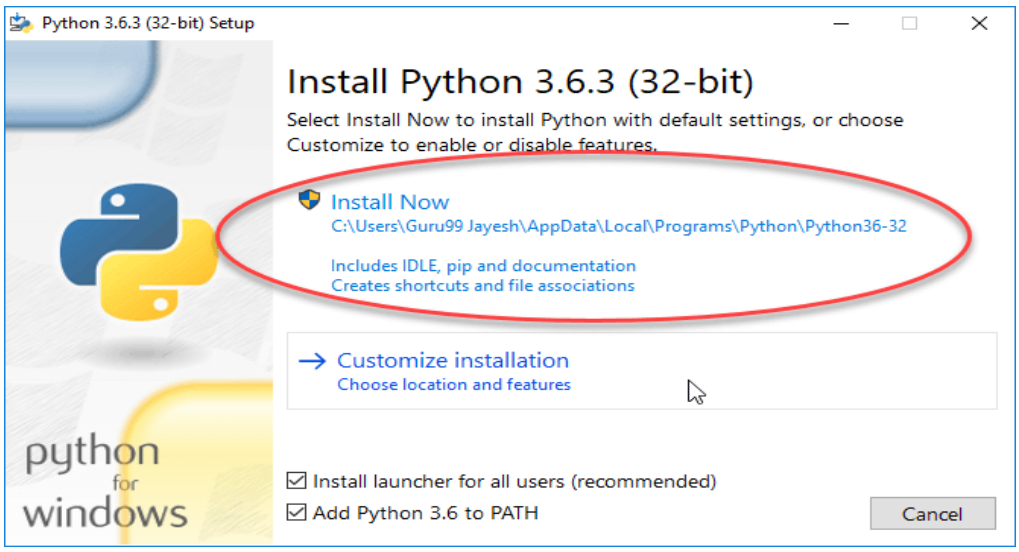
Below are the detailed steps for installing Python and PyCharm.

1.1 Installing Python

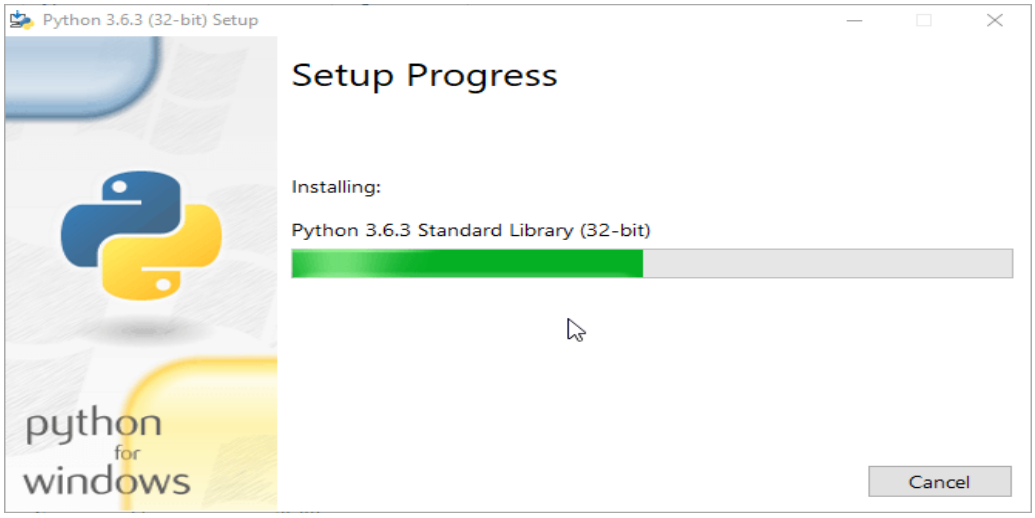
Step 1) To download and install Python visit the official website of Python http://www.python.org/downloads/ and choose your version. We have chosen Python version 3.6.3



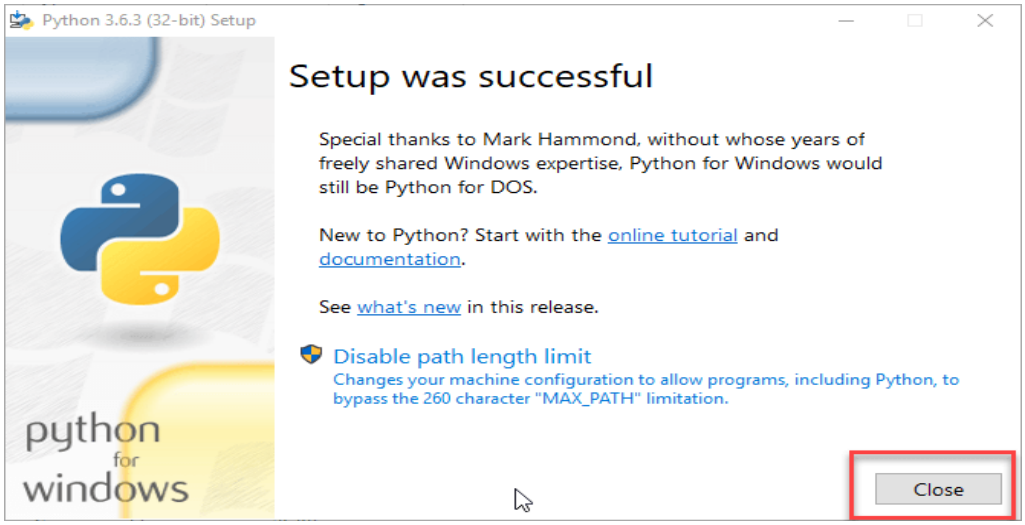
Step 2) Once the download is complete, run the exe for install Python. Now click on Install Now.



Step 3) You can see Python installing at this point.

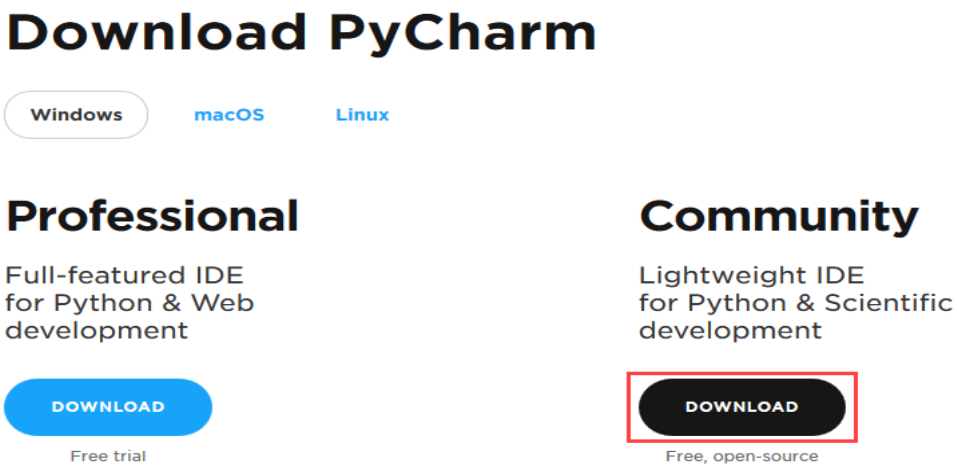


Step 4) When it finishes, you can see a screen that says the Setup was successful. Now click on "Close".

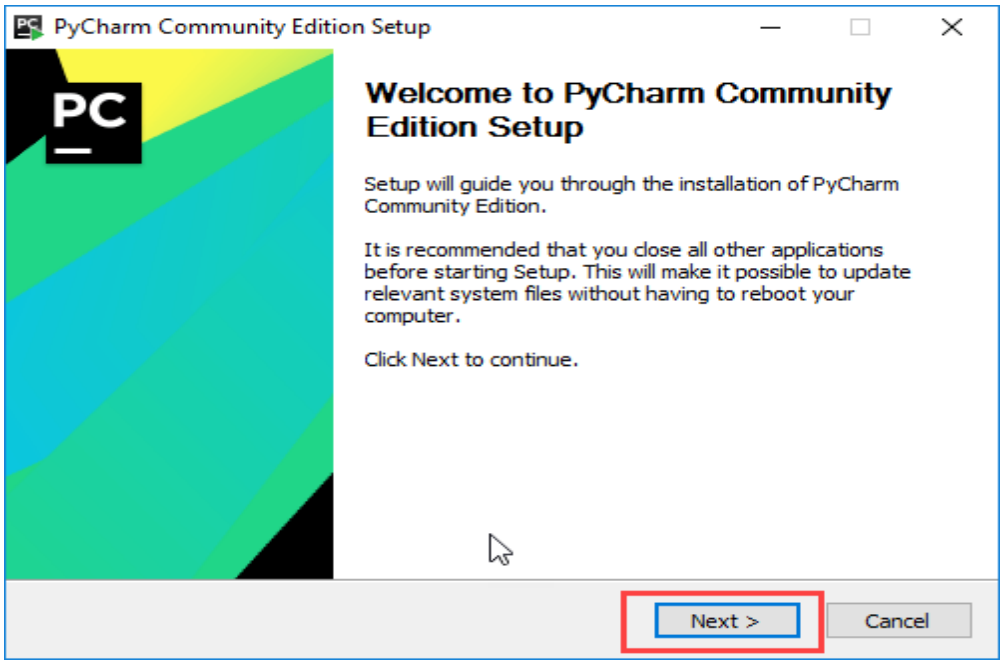


1.2 Installing Pycharm

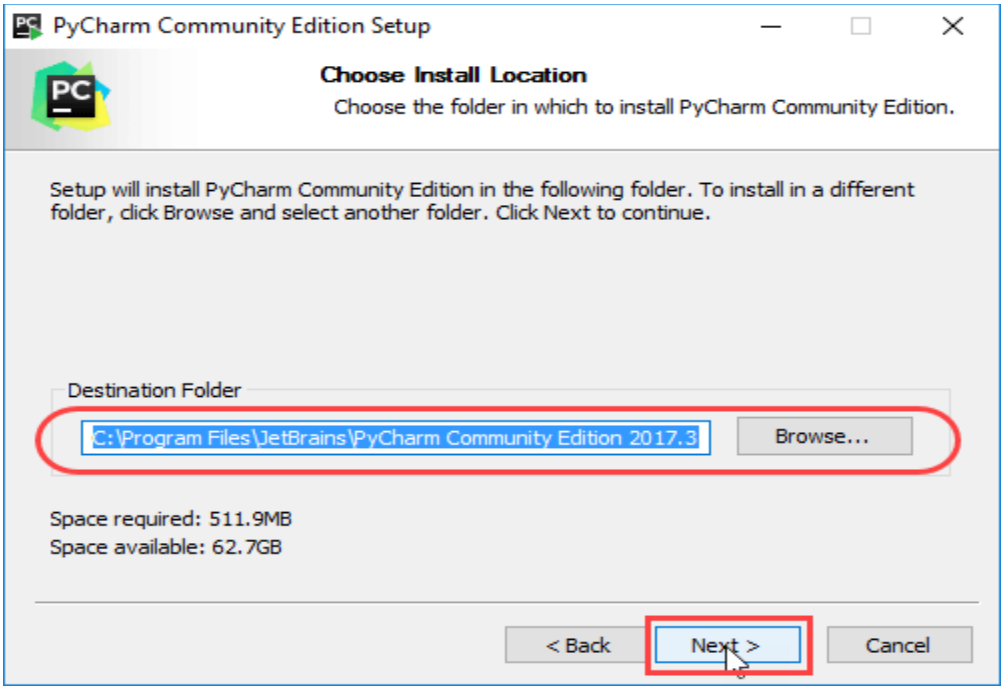
Step1) To download PyCharm visit the website https://www.jetbrains.com/pycharm/download/ and Click the "DOWNLOAD" link under the Community Section.



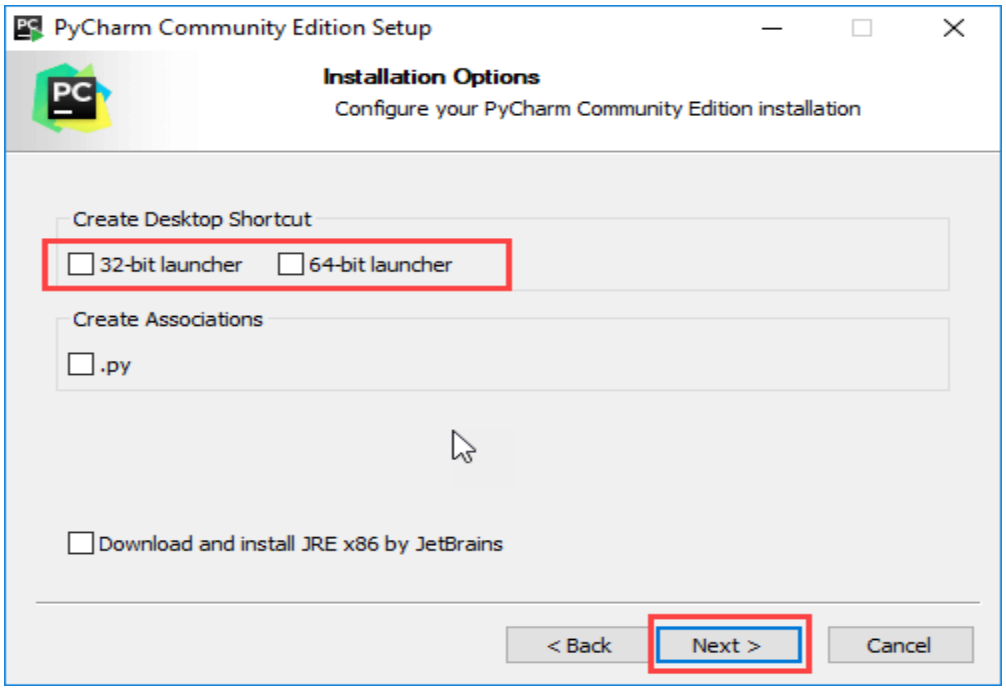
Step 2) Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click "Next".



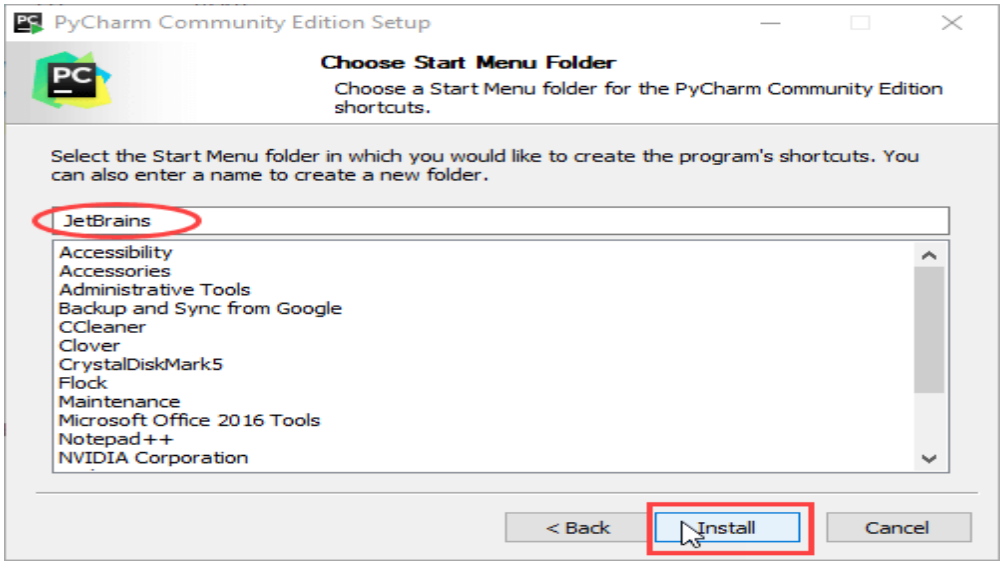
Step 3) On the next screen, Change the installation path if required. Click "Next".



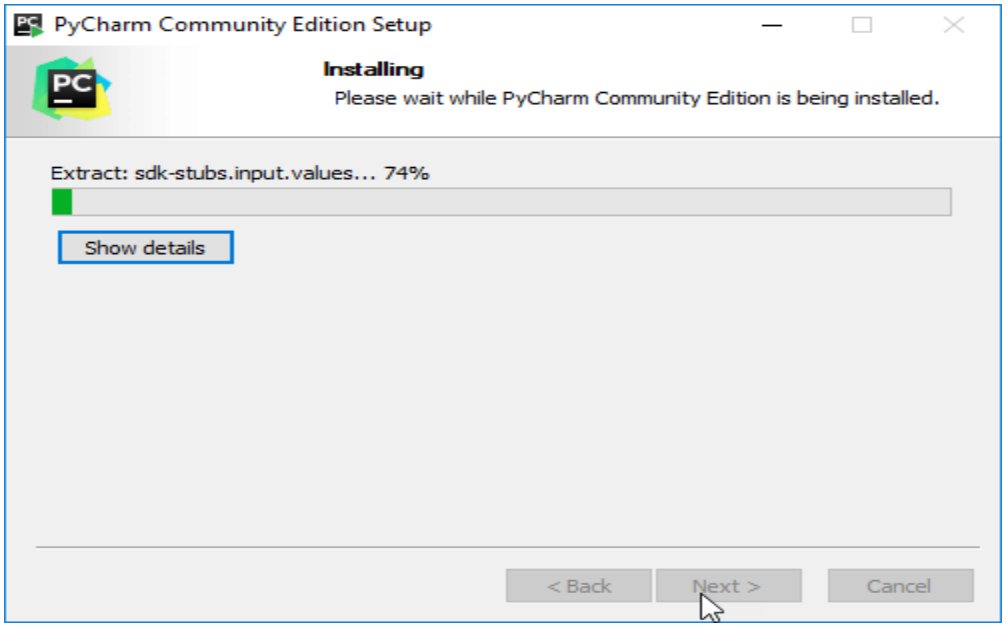
Step 4) On the next screen, you can create a desktop shortcut if you want and click on "Next".



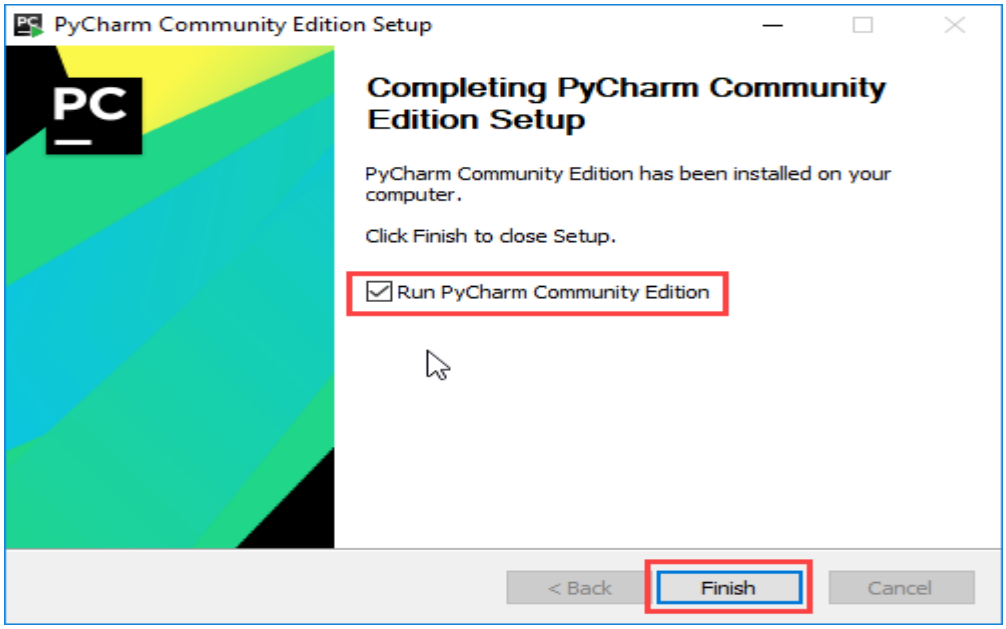
Step 5) Choose the start menu folder. Keep selected JetBrains and click on "Install".



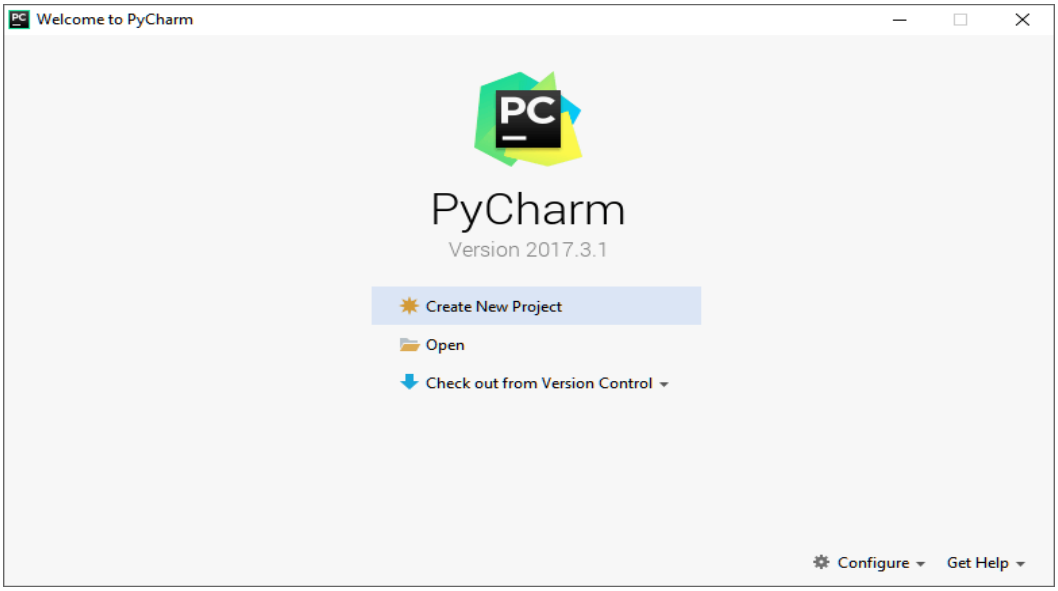
Step 6) Wait for the installation to finish.



Step 7) Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the "Run PyCharm Community Edition" box first and click "Finish".



Step 8) After you click on "Finish," the Following screen will appear.



Python Numbers

Number data type is used to store numeric values. Numbers are immutable data types, therefore changing the value of a number data type results in a newly allocated object.

```
#Integer
>>>a=5
>>>type(a)
<type 'int'>

#Floating Point
>>>b=2.5
>>>type(b)
<type 'float'>

#Long
>>>x=9898878787676L
>>>type(x)
<type 'long'>

#Complex
>>>y=2+5j
>>>y
(2+5j)
>>>type(y)
<type 'complex'>
>>>y.real
2
>>>y.imag
5
```

```
#Addition
>>>c=a+b
>>>c
7.5
>>>type(c)
<type 'float'>

#Subtraction
>>>d=a-b
>>>d
2.5
>>>type(d)
<type 'float'>

#Multiplication
>>>e=a*b
>>>e
12.5
>>>type(e)
<type 'float'>
```

```
#Division
>>>f=b/a
>>>f
0.5
>>>type(f)
<type float'>

#Power
>>>g=a**2
>>>g
25
```

Python Strings

A string is simply a list of characters in order. There are no limits to the number of characters you can have in a string.

```
#Create string
>>>s="Hello World!"
>>>type(s)
<type 'str'>

#String concatenation
>>>t="This is sample program."
>>>r = s+t
>>>r
'Hello World!This is sample program.'

#Get length of string
>>>len(s)
12

#Convert string to integer
>>>x="100"
>>>type(s)
<type 'str'>
>>>y=int(x)
>>>y
100
```

```
#Print string
>>>print s
Hello World!

#Formatting output
>>>print "The string (The string (Hello World!)
has 12 characters

#Convert to upper/lower case
>>>s.upper()
'HELLO WORLD!'
>>>s.lower()
'hello world!'

#Accessing sub-strings
>>>s[0]
'H'
>>>s[6:]
'World!'
>>>s[6:-1]
'World'
```

Python Lists

List a compound data type used to group together other values. List items need not all have the same type. A list contains items separated by commas and enclosed within square brackets.

```
#Create List
>>>fruits=['apple','orange','banana','mango']
>>>type(fruits)
<type 'list'>

#Get Length of List
>>>len(fruits)
4

#Access List Elements
>>>fruits[1]
'orange'
>>>fruits[1:3]
['orange', 'banana']
>>>fruits[1:]
['orange', 'banana', 'mango']

#Appending an item to a list
>>>fruits.append('pear')
>>>fruits
['apple', 'orange', 'banana', 'mango', 'pear']
```

```
#Removing an item from a list
>>>fruits.remove('mango')
>>>fruits
['apple', 'orange', 'banana', 'pear']

#Inserting an item to a list
>>>fruits.insert(1,'mango')
>>>fruits
['apple', 'mango', 'orange', 'banana', 'pear']

#Combining lists
>>>vegetables=['potato','carrot','onion','beans','radish']
>>>vegetables
['potato', 'carrot', 'onion', 'beans', 'radish']

>>>eatables=fruits+vegetables
>>>eatables
```

```
#Mixed data types in a list
>>>mixed=['data',5,100.1,8287398L]
>>>type(mixed)
<type 'list'>
>>>type(mixed[0])
<type 'str'>
>>>type(mixed[1])
<type 'int'>
>>>type(mixed[2])
<type 'float'>
>>>type(mixed[3])
<type 'long'>
```

Python Tuples

A tuple is a sequence data type that is similar to the list. A tuple consists of a number of values separated by commas and enclosed within parentheses. Unlike lists, the elements of tuples cannot be changed, so tuples can be thought of as read-only lists.

```
#Create a Tuple
>>>fruits=("apple","mango","banana","pineapple")
>>>fruits
('apple', 'mango', 'banana', 'pineapple')

>>>type(fruits)
<type 'tuple'>

#Get length of tuple
>>>len(fruits)
4
```

```
#Get an element from a tuple
>>>fruits[0]
'apple'
>>>fruits[:2]
('apple', 'mango')

#Combining tuples
>>>vegetables=('potato','carrot','onion','radish')
>>>eatables=fruits+vegetables
>>>eatables
('apple', 'mango', 'banana', 'pineapple', 'potato', 'carrot', 'onion', 'radish')
```

If Statement

The *if* statement in Python is similar to the *if* statement in other languages.

```
>>>if a>10000:
        if a<1000000:
                print "Between 10k and 100k"
        else:
                print "More than 100k"
        elif a==10000:
                print "Equal to 10k"
        else:
                print "Less than 10k"

More than 100k
```

```
>>>s="Hello World"
>>>if "World" in s:
        s=s+"!"
print s

Hello World!
```

For Statement

The *for* statement in Python iterates over items of any sequence (list, string, etc.) in the order in which they appear in the sequence.

```
#Looping over characters in a string

helloString = "Hello World"

for c in helloString:
        print c
```

```
#Looping over items in a list

fruits=['apple','orange','banana','mango']

i=0
for item in fruits:
        print "Fruit-%d: %s" % (i,item)
        i=i+1
```

While Statement

The while statement in Python executes the statements within the while loop as long as the while condition is true.

```
#Prints even numbers upto 100

>>> i = 0

>>> while i<=100:
if i%2 == 0:
    print i
i = i+1
```

Functions

A function is a block of code that takes information in (in the form of parameters), does some computation, and returns a new piece of information based on the parameter information.

```
students = { '1': {'name': 'Bob', 'grade': 2.5},
             '2': {'name': 'Mary', 'grade': 3.5},
             '3': {'name': 'David', 'grade': 4.2},
             '4': {'name': 'John', 'grade': 4.1},
             '5': {'name': 'Alex', 'grade': 3.8}}


def averageGrade(students):
        "This function computes the average grade"
        sum = 0.0
        for key in students:
                sum = sum + students[key]['grade']
                average = sum/len(students)
        return average


avg = averageGrade(students)
print "The average garde is: %0.2f" % (avg)
```

Read More...

- **https://www.w3schools.com/python/**
- Code Academy Python Tutorial, http://www.codecademy.com/tracks/python
- Google's Python Class, https://developers.google.com/edu/python/
- Python Quick Reference Cheat Sheet, http://www.addedbytes.com/cheat-sheets/python-cheat-sheet/
- PyCharm Python IDE, http://www.jetbrains.com/pycharm/