* It is easy to learn and understand, simpler code
* World is moving towards AI with Machine learning, Python plays a key role in those

**Installation of Python:**

* Down load Exe file from <https://www.python.org/downloads/>
* Install file
* And Set Environment Variables Path for Python folder and Script folder (for pip file).

**PIP:**

Pip is the standard package manager for python. It allows you to install and manages additional packages that are not part of the python standard library.

**Selenium installs with Python:**

* Go to <https://pypi.org/project/selenium/>
* Copy Command
* Go to Command Prompt and paste and enter
* After Installation, to cross check , in CMD “pip show selenium”

**PyCharm Installation:**

* Down load Pycharm <https://www.jetbrains.com/pycharm/download/>
* Install and open
* Create a file
* And go to **Interpreter settings** make sure the Project Interpreter to Local Python Package, Since Pycharm provide run time Python, but it may not have all external pips, which we got from PIP.

**Python Basics:**

# : Comments

Print(“TT”) : Print something

Data Types: No need to define data type for variables (It’s like “Groovy”)

Variables: We can define variables at a time like b,c,d = 5,6.4,**"Great”**

To Print “String” and “Integer” (two different data types) together have to use “format” method

EX: **"{}{}"**.format(**"Value is"**, 5)

To check variable data type use “type”

EX: type(variable)

**Python Data Types:**

* **Numeric**
  + Integer
  + Long
  + Float
  + Complex (Holds complex numbers)
* **String** : Can keep text in “ ” or ‘ ‘
  + str = **"Prathap Bagala"**str1= **"Reddy"**str3= **"Prathap"**print(str[1])  
    printstr(str[0:5])  
    print(str+str1)
  + To Check **contains**(no contains key required, used **in**)
    - **str3 in str**
  + To **Split** a string, use “split” keyword
    - var = str.split(**" "**) : Split with space
    - If we have to split where we have . **str.split(".")**
  + **Trim**: To remove white space, use **Strip**
    - **Var.strip()**
    - **Val.lstrip()** : to remove only left side white space
    - **Val.rstrip()** : to remove only right side white space
* **List**: It’s almost like array in other languages. :: **It’s Non- Immutable**
  + EX: values = [1, 2, "Prathap", 4, 5]
  + **values[-1]:** To get last value from the list ; here 5
  + **values[1:3]** : To get values between index 1 and 3, it print 2, Prathap
  + **Insert**: To insert a value in existed list.
    - EX: Values.insert(3, “Reddy”)
    - Then new List becomes [1, 2, 'Prathap', 'Reddy', 4, 5]
  + **Append(“ “)** : To insert value at end
  + **To Update existing Value: values[2]=”Jeevan”** : It replace “Prathap”
  + **Del : To delete particular index;** Ex: del values[0] :
* **Tuple** :: **It’s Immutable**
  + EX: val = (1, 2, **"Pathap"**, 4.5)
* **Dictionary :** It’s same like Hash Map in java, Key Value pair.
  + EX: dict = {**"a"**:2, 4:**"bcd"**, **"c"**:**"Hello"**}

**If Else Condition:**

* There are no {} in Python
* Have to use **:** after if and else
* Always right statement after a tab from second line in side condition like below

if name == "Reddy":

print("It's False")  
else:  
 print("It's True")

* **Pass:** If we write pass after if condition it go inside but it don’t do anything

**For Loop:**

* Have to declare like below
  + **for** i **in** val:  
     print(i\*3)
* When need to get some range of numbers have to use “**range**” command
  + sum = 0  
    **for** j **in** range(1, 6):  
     sum = sum + j  
    print(sum)
* When have to Jump two indexes (i++🡪 i+2), below example print 1,3,5,7,9 (1 to 9(K-1))
  + **for** k **in** range(1,10,2):   
     print(k)
* When have to get from 0 to particular number, skip first index python consider 0 as first index. Below ex print 0 to 9(m-1)
  + **for** m **in** range(10):  
     print(m)

**While Loop:**

* To check a condition, it execute until given condition get false. Conditions true it keep on execute. There is no end like for look.
* Below example it always true so it will be infinite, so add some condition to end

it = 4  
**while** it>1:  
 print(it)

it = it-1 condition to end loop

* **Break:** To Break loop abruptly at particular condition. It totally come out of loop, it won’t execute any further steps
* **Continue:** To skip particular condition, it skip that specific condition and continue executing remaining iterations.

**Functions:**

* Function is a group of related statements that perform a specific task.
* “def” is command to create function in python
  + *#Function Declaration***def** GreetMe(name):  
     print(**"Hello "** + name)  
    *#Function Call*GreetMe(**"Prathap"**)
  + **def** AddIntegers(a,b):  
     **return** a+b  
    print(AddIntegers(2,3))

**Classes:** Are user defined blue print or prototype

* It contains methods, class variables, instance variables, Constructors. Define like below and use code intend
  + **Ex: class Calculator:**
  + To create object for a class, just use class name and assign to a variable. There is no “new” keyword like Java.
    - **EX:** **obj = Calculator()**
    - To call Object**; obj.getData();** (Accessing methods from that class object)

**Constructor:**

* Always Constructor with “init”
  + **def\_\_init\_\_(self)**
* When we have class object with parameters, we should have same number of parameters in constructor.

**Class Variables:**

* The variables which we define immediately inside class. These are same for all objects of class.

**Instance Variable:**

* These are differs for every object of class
* We have to call instance variable with self. Variable
  + **EX: self.num1=a**

**Inheritance:**

* Here no keyword(EX: Implements) required like java
* When Need to inherit a class, Just get it by that class name like below
  + **class oopsDemo (Calculator)**
  + Here I’m inheriting Calculator(Parent) methods to **oopsDemo** (Child)
* If we have any constructer declarations (except default) in parent class, have to create constructor in child class and call parent class constructor.
  + **def \_\_init\_\_(self):  
     Calculator.\_\_init\_\_(self,2,10)**

**Read External File:**

* Have to open file first to read or write, use **“Open ”**method.
  + **file = open('Text.txt')**
  + **with** open(**"Text,txt"**,**'r'**) **as** file: #r is for mode of operation, here “read”
  + #it opens file and close after done with operations. More efficient way, no need to write close at last.
* To read a file, use “**read**” method
  + **file.read()**
  + file.read(5) // to get first 5 chars
* To read files line by line(Single line at a time), have to use “**readline**” method
  + ***file.readline()***
* Using **Loops** we can read all lines of a file at a time
* **ReadLines**: It’s store data of a file like list, so we can loop it and use data
  + **for line in file.readlines():  
     print(line)**
* if we don’t use **with open(Text.txt) as file:.**Always use “**Close**” to close the file after done with operation to avoid data leak and corrupt of file.
  + **file.close()**

**Write External File:**

* Use “write” method to write into file
  + **file.write(line)**

**Exception Handling:**

* We can use “**raise**” keyword to make test fail when condition didn’t meet.
  + **if** ItemsinCart !=2:  
     **raise** Exception(**"Didn't match"**)
* Also we can use “**Assert**”

**Try Except Mechanism:**

* It’s same like try catch in java, but here we use **“except**” instead of catch keyword.
* **Try**: Code try first this block of code, if it success it won’t go to **except** block, else it goes to except block and executes that part of code.
* We can use 2 methods here, one is with our own error message, other with syaytem error message
  + *#Method 1: with Customized exception message***try**:  
     **with** open(**"filelog.txt"**, **'r'**) **as** reader:  
     reader.read()  
    **except**:  
     print(**"Failed with Try"**)
  + *#Method2: with System exception message***try**:  
     **with** open(**"filelog.txt"**, **'r'**) **as** reader:  
     reader.read()  
    **except** Exception **as** e:  
     print(e)

**Finally:**

* We use this keyword along with Try Exception mechanism
* It execute at any cost if we have failure in try or in except, doesn’t matter.