What is python?

As per the Python software Foundation, Python is a high level interactive, interpreted and object oriented scripting language.

What is the difference between scripting languages and programming languages?

Programming languages

1. Programming languages are meant for developing both system software as well as application software.
2. The programs which are supported to run the system is known as system software various application software is nothing but the programs which are used to achieve the specific requirements of the customer or the client.

3. in order to work with any programming language it is mandatory to require to install special software's called compiler for an example if you want to write and execute the C program then we have to install C compiler call the turbo c 2 similarly if you want to work with Java language then it is recommended to install a special software called JDK

4. After writing any program using programming languages it is mandatory to compile before execution.

5. Programming languages are heavyweight that is that taking more resources and more processing time

6. In programming languages there is a separation between source code and application program

Scripting languages:

1. Mostly scripting languages are meant for developing application software

2. the most of the scripting languages does not have the capability to execute independently that means the scripting languages required the support of system software.
Example for the scripting languages are JavaScript, Shell Script, python etc

3. In order to work with the scripting languages does not required to install special software's, but few scripting languages required to install software's if you want to work with JavaScript that is not required to install Special software whether your system should be loaded with the Notepad and the basic browser is enough similarly if you want execute Shell Script then your system should be loaded with any Unix based operating system

4. If we develop any program using scripting languages does not required to compile before execution rather that executed directly without any preliminary compilation

5. Scripting languages are lightweight visa taking less resources endless process

6. In the scripting languages there is no separate exe file rather the source file itself it is acting as an executable file

Note:
Python be inherited by taking the best powerful features of programming languages and scripting languages

a. object oriented programming capability from C + + and Java
b. procedure oriented capability from C language
c. scripting capability from JavaScript and Shell Script.
d. modularity from modular 3 and some other language support

<u>Applications of Python:</u>
1. Web application development
2. Automation applications
3. Testing tools

4. Gaming applications
5. Internet of things - IOT
6. Data Analytics and data science
7. Artificial intelligence ...etc

Advantages of python

1. High level language

a. if any language which enable you to develop any application by using regular English instructions such as capital alphabets, small alphabets, 0 to 9 digits and special characters

2. Portability:

Portability always depends upon the source code.
i.e no matters what operating system we are using the syntax and semantics of the language remains same.

3. easy to learn, write and maintain

Python script statements resembles as like a regular English instructions. Thus compared to other languages writing the python script statements or easy

4. Open source

Python is available to the world under GPL general public licence. Python software can be used by anyone without paying a paisa. And it is giving the facility we can also see the internals of the python if required we can also make the amendments in the language and we can also release the new flavours of python

5. Independent

Python script statement can be executed in irrespective of the operating system, by taking the support of python virtual machine

6. Interpreted

Python script statements or executed director without any preliminary compilation

7. Procedure oriented

In the procedure oriented programming the logic can be maintained and manipulate in the form of functions

8. Object oriented

In the object-oriented programming the logic can be maintained and manipulate in the form of class and object

9. Huge library support

In other languages we have the support of libraries
Library contains collection of variables, functions, macros ...
In Python we have the support of module. The module provides list of variables, functions, classes and object …

With the help of the models we can develop any type of application simple and easier

10. Dynamically typed

Other languages like c,c++, java are statically typed
Python is dynamically type that is, Python scripting by the time of declaring the variable, does not required to specify the data type and the type of the variable will be decided by the Python interpreter based on the value we are assigned to it.

11. Database connectivity

Python is giving the convenience developers to interact with any data bases to store the data permanently with an effect to security.

➢ *Batteries included*

- ✓ argpars For Command line arguments
- ✓ boto is Amzon web services library

- ✓ CherryPy Is Object Oriented for HTTP Frame Works
- ✓ Fiona read and write data from large files
- ✓ jellyFish For phonetic matching Strings
- ✓ mysql_connector_python is a driver written to connect to My SQL Database
- ✓ numpy is Multi-dimensional arrays
- ✓ pandas for data analytics and statistics
- ✓ Pillow for Image Processing
- ✓ scipy for scientific and engineering calculations

Flavours of Python

Flavours of Python is referred to different types of the Python Compilers

- ➢ ***Cpython***
  - ➢ It is Standard Python compiler implemented in c
  - ➢ Compiled Code by python is byte code. It can run on interpreter existed on PVM created on C-Language
  - ➢ By Using CPython we can also execute c and c++ functions and Programs
- ➢ ***Jython***
  - ➢ Initially it is called Jpython
  - ➢ It is implementation of python programming language which is designed to run on jvm
  - ➢ Jython contains libraries which are useful for both python and Java programs
- ➢ ***IronPython***
  - ➢ It is another implementation of Python For .Net Frame work . It is written in C#
  - ➢ When Python Programs are Compiled an Intermediate Language Instructions are Created But These Instructions Can Be Understand by CLR
  - ➢ It is Providing Various Libraries for Both python and .NET
- ➢ ***PyPy***

- Python Implementation Using Python, Actually Written in Rpython which is created by using python.
- Rpython is good for creating language interpreters .
- PyPy Programs Run Very fast. Because JIT Compiler is Add to Python Interpreter

- ***Pythonxy***
  - It is Also Python Implementation
  - we get after adding scientific and engineering related Packages

- ***Anaconda***
  - Handling Large scale of Data processing
  - Data Analytics and Scientific computing called Anaconda

Cptyhon -- http://www.python.org/downloads
Jython – http://www.jython.org
IronPython – http://ironpython.net//
Pypy : -- http://pypy.org/download.html
Pythonxy : -- https://python-xy.github.io/downloads.html
Anaconda : -- https://www.continuum.io/downloads

Who is using Python :

Youtube, google, NASA , Dropbox, facebook and yahoo …..

- Youtube : The popular YouTube videos Sharing services are mostly Written in Python

- Google Makes extensive use of python in its web search System

- NASA , LOS Alamos, Fermi lab, JPL, and other are Using Python For Scientific Programming

History of Python:

- ➢ Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.
- ➢ Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and Unix shell and other scripting languages.
- ➢ Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).
- ➢ **Python Name Was Taken From TV Show, Monty Python's Flying Circus**

2.Installation
www.python.org/downloads

3.Ways For Executing Python Script Statements

1.Using Python Shell
2.Using Python IDLE
3.Using Script Mode
4.Using Os Prompt or Windows Prompt

Python Shell :

1.It is CUI [Character User Interface] Based
[Where can perform operations using keyboard]
2. Doesn't have any look and feel
3. Doesn't provide any onscreen tip [syntax help]

cls,date,time commands from OS.

inorder to execute any os commands then we have to use, system( ) from OS Module

exit() used for to quit from python shell

Python IDLE

1. Python IDLE Provides Look and Feel
2. GUI [Graphical user interface] Based
3. It is Providing Menu Support
4. In Python IDLE we can't clear the screen Using os.system('cls')
5. In Python IDLE, It doesn't execute any OS Commands Rather internally it will open OS Shell and execute OS Command there only.

3. Using Python Script Mode

* In Python Shell and IDLE where we can execute only operation at a time, but if you want execute some group of operations at a time then we have to go script mode

Steps :

1. Open IDLE
2. Select New File from the File Menu
3. Define required Script Statements
4. Select Save From the file menu and the file with any name.py [Eg: first.py]
5. Select Run module [f5] From Run Menu

4. Executing Python Script Statements Using OS Prompt

Steps:

1.Open command prompt

```
      [Window+R --> cmd --> Ok ]

C:\Users\class>e:

E:\>cd Python_Online9

E:\Python_Online9>py second.py
101
<class 'int'>
140725914903328
```

Fundamentals of the Python:

Token is nothing but smallest individual component in the program

Python Tokens Are :

1.Character SET

Set of characters which are supported by Python Language interpreter

* Python will support Unicode char SET
* Unicode = ASCII and Non ASCII
* Non Ascii = Other Language Characters [National | International]
* Range : 0 to 65535
* We can develop Language Friendly Application

2.Variables

* It is a space to store the data or
* It is named container which enable you to store the data temporally during the program execution

Java :

Syn:[modifiers]b<datatype>b<identifiers>[=value];

Actually here b is nothing a space

Python:

Syn:<identifier>=<value>[;]

```
eno=10
ename="Ramesh";
```

3.Datatypes

4.Operators

5.Identifiers

Are nothing but all the names which are declared by us for our programming requirements, Such as : Variable names, Function names, Class names.....

Rules :

1. It must starts with an alphabet or _
2. It may be in Lower | Upper | Mixed cases
3. No Limit in the length of identifiers
4. It May Have digits
   Eg: e00no; ena98me; [valid]
5. It may have a Special character [i.e. : _]
6. It should not be Python Keyword

6.Keywords

* These are nothing but reserved words
* Every keyword is having its importance in the program
* The meaning of the keyword can't be changed
* To know the keywords existed in the python then we have to use

kwlist [predefined Variable of type <class 'list'>]
Existed in keyword module

import keyword
keyword.kwlist   [shell and idle]
In script Mode --> print(keyword.kwlist)

```
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

```
>>> type(keyword.kwlist)
<class 'list'>
>>> len(keyword.kwlist)   #35
```

iskeyword() :
It returns True if the given String is a keyword else it will return False

```
Eg:   >>> keyword.iskeyword('Roja') #False
       >>>keyword.iskeyword('as')      #True
```

Print( ) :

- It an output functions from builtins module
- It used to display the result on the console

```
x=10

print(x)  #10

print("welcome")  #welcome
```

Eg2:

```
x=10
y=3.14
z="welcome"

print(x) #10
print(y) #3.14
print(z) #welcome

print(value,value,value,.....)
print(x,y,z)   #10 3.14 welcome
```

Note : While printing multiple values using print function, then each value is separated by a space, if you want provide any literal then we have to use sep attribute in print( )

Eg3: Using sep attribute

```
x=10
y=3.14
z="welcome"

print(x,y,z)    #if u want any literal as sep
                  #sep attribute in print( )
                  #default value for sep attribute is space [ sep=' ' ]

print(x,y,z,sep=' ') # 10 3.14 welcome
print(x,y,z,sep='-') # 10-3.14-welcome
print(x,y,z,sep=',') # 10,3.14,welcome
print(x,y,z,sep='') #103.14welcome
```

Note: While printing the values using print (), it will print the result on the screen and throws the cursor to the newline. Just because the default value of “end” attribute is ‘\n’. if you want print in result in same line then we have to use “end” attribute with end=’’.

Eg 4: Using end attribute in print( )

```
x=10
y=3.14
z="welcome"

print(x)   # print( ) -> printf( ) + "\n" [In C]
print(y)   # end attribute in print( )
print(z)   # the default value for end attribute is \n

print(x,end='\n')
print(y,end='\n')
print(z,end='\n')
print(x,end='')
print(y,end='')
print(z,end='')  # print(x,y,z,sep='')
```

We can use the Esc Sequence Char in both end or sep attribute of print( ) if required

\n	***New Line***
\a	***Bell sound***
\t	***Horizontal .Tab(4)***
\b	***Backspace***
\	***It will print \ symbol***
\'	***It will print ' symbol***
\"	***It will print " Symbol ...***

```
x=10
y=3.14
z="welcome"
```

```
print(x,y,z,sep=' ')
print(x,y,z,sep='\t')


print(x,end='\t')
print(y,end='\t')
print(z,end='\t')
```

Printing the value of variables using format specifier's

Format Specifies : These are used to specify what type values to be formatted during output.

```
  int --> %d
  float --> %f
  string -> %s

 Syn: print("formatspecifier" %variable)
         print("formatspecifiers" %(list of variables) )


x=10                    #int
y=3.14                 #float
z="welcome"      #str

print("%d" %x)
print("%f" %y)

print("x val is : %d " %x)
print("y val is : %f " %y)

print("%d %f %s" %(x,y,z))
print("x:%d y:%f z:%s " %(x,y,z))
          #x:10 y:3.14000 z:welcome
```

print( ) using replacement fields { }

- while working with replacement field then we have to use format( ) from the class : <class 'str'>.

Syn: print("msg | values ".format(variables))

Replacement Fields Can Be
- Index based
- Non Index Based
- Name based

```
name="sudha"
age=27

#Non Index Based
print("name :{}".format(name))  # name: sudha
print("name is :{} and age is :{}".format(name,age))

#IndexBased                              0        1 –index values
print("name is :{1} and age is:{0} ".format(age,name))

#NameBased
print("Name is :{} and age is :{}".format(name,age))
print("Name is :Mr|Mrs.{n} and age is :{a},{n} is from Hyd".
format(n=name,a=age))

#Output:
#Name is :Mr|Mrs.sudha and age is :27,sudha is from Hyd
```