## Core Python Syllabus

1. **Python introduction**

* **What is python**

Guido van Rossum begin in 1980 and released the first version of 1991. Python is High level general purpose computer programming language. It is simple as like to learn but has vast application

* **Hello world in python**

print(‘Hello world’)

* **Type of object and advantages**

Two type of object user defined and built in object (it is also known function )

Built-in objects make programs easy to write

Built-in objects are components of extensions

Built-in objects are often more efficient than custom data structures

Built-in objects are a standard part of the language.

* **Indentation rules and comment**

Code depends on the how to write, don’t depends any {} is requires to way of execution

for I in range(100):

print(i)

* **Variable and data type**

Variable is container to store the data in python language and stored data is called value and is written in a to z any letter and by \_ but don’t start special character or number

A=110

A is variable and 110 is value

Variable is that thing though identity of value

* **Different data type of Python**

Int : 1,3,5

Float : 2.3

String : ‘itnesh’

Bool : True / False

Complex: 2+3j

List: [1,2,'itnesh']

Tuple: (1,2,'itnesh')

Sets: {1,2,'itnesh'}

Dictionary: {"name": "itnesh", "age":25}

* **Type conversion**

int(2.3) convert into integer =2

str(2) convert into string ='2'

float(5) convert into 5.0

tuple('s') convert into ('s',)

list('1') convert into ['1']

set('d') convert into {'d'}

dict([(1,'apple'), (2,'ball')]): {1: 'apple', 2: 'ball'}

* **Python operators**

**Arithmetic Operators:**

+ : addition: 2+3: 5

- : Subtraction: 5-2:3

\* :Multiplication: 5\*6: 30

/ :division: 4/5: 0.8

% :modulus: 5/2: 1

\*\* :Exponentiation: 2\*\*2:4

// :Floor division: 5//2: 2

**Assignment Operators:**

= x=5 assignment operators

+= x += 3 x = x + 3

-= x -= 3 x = x – 3

\*= x\*=3 x=x\*3

Etc

**Comparison Operators:**

== Equal x == y

!= Not equal x != y

Greater than x > y

< Less than x < y

>= Greater than or equal to x>=y

<= less than or equal to x<=y

**Logical Operators:**

and: both statement correct like then true : x>2 and x<9

or : at least one condition is true then true : x < 5 or x < 4

not : no any condition is correct then true: not(x < 5 and x < 10)

**Identity Operators:**

is : Returns True if both variables are the same object: x is y

is not: Returns True if both variables are not the same object: x is not y

Membership Operators

in : return true if x present in y: Y in x

not in : return true if x is not present in y: Y not in x

* **STRING MANIPULATION**

*a ="Hello World"* # this is string

**Accessing**

*letter=a[0]*

*print (letter*)

*H*

*Length*

*len(a)*

*11*

*print(a.count('l'))*

*3*

*print(a.find("H"))*

*0*

*print(a.index("World"))*

*6*

**Slicing**

*print(a[0:3])*

*Hel*

*print(a[:3])*

*Hel*

*print(a[3:])*

*lo World*

*print(a[:-3])*

*Hello Wo*

**Splitting**

*list(a.split(' '))*

# Split on whitespace

*['Hello', 'World']*

**Startswith / Endswith**

*a.startswith("H")*

*True*

*a.endswith("d")*

*True*

*a.endswith("w")*

*False*

**Changing Upper and Lower Case Strings**

*print(a.lower())*

*hello world*

*print(a.title())*

*Hello World*

*print(a.upper())*

*HELLO WORLD*

*print(a.swapcase())*

*hELLO wORLD*

**Replacing**

*a.replace("Hello", "Goodbye")*

*'Goodbye World*

**Strip**

*j=" kumar "*

*j.strip()*

*#removes from both ends*

*'kumar'*

**Concatenation:** addition

*"Hello" + "World"*

*"HelloWorld"*

**Join**

*print(":".join(a))*

# #add a : between every char

*H:e:l:l:o: :W:o:r:l:d*

1. **Data structure in python**

* **Mutable and Immutable**: mutable though can be change on the basis of requirement that data type but immutable made at single time (no any change)
* **Python list**

**Indexing**

*L=[2,35,’rohan’,3]*

*Print(L[1])*

*35*

*O=[3,5,6,[‘man’,’sohan’]]*

*O[2][0]*

*Man*

*O[-1]*

*[‘man’,’sohan’]*

**Slicing**

*My\_list=[‘p’,’q’,’r’,’t’,’d’]*

*My\_list[1:3]*

*Q,r.t*

**Appending add in last in list**

*My\_list.append(2)*

*My\_list=[‘p’,’q’,’r’,’t’,’d’,7]*

**Extend**

*My\_list.extend([3,5,6])*

*[‘p’,’q’,’r’,’t’,’d’,7,3,5,6]*

**Concatenation :** addition of two data type except for number

*Odd=[7,51,3]*

*Even=[2,4,6]*

*Print(Odd+Even)*

*[7,51,3,2,4,6]*

**Insert**

*Odd.insert(1,3)*

*[7,3,51,3]*

**Delete**

*Del odd[51]*

*[7,3,3]*

**Remove**

*My\_list.remove(‘p’)*

*[‘q’,’r’,’t’,’d’,7]*

**Remove from last element pop**

*Print(My\_self.pop)*

*7*

.**count(), .sort(), .clear(),.reverse()**

* **Python tupple**

*tup1 = ('physics', 'chemistry', 1997, 2000)*

*print(tup1[0]): physics*

*del tup1*

**len, concatenation**

**repetation**

*(1,2,3)\*3=(1,2,3,1,2,3,1,2,3)*

*menbership1 in (1,2,3):*

*True*

**Iteration**

Calling the values one by one

*for x in (1,2,3):*

*print(x)*

*1*

*2*

*3*

**indexing, slicing,**

* **Python sets**

It gives the ouput in order form

*Days={"Mon","Tue","Wed","Thu","Fri","Sat"}*

*Days.add("Sun")*

*print(Days)*

*Days.discard('Tue')*

*{'Thu', 'Fri', 'Mon', 'Sun', 'Sat', 'Wed'}*

*{'Thu', 'Fri', 'Tue', 'Mon', 'Sun', 'Sat', 'Wed'}*

**UNION**

*s1={1,2,3,4,5}*

*s2={4,5,6,7,8}*

*s1.union(s2)*

*{1, 2, 3, 4, 5, 6, 7, 8}*

[*set.pop()*](https://www.tutorialsteacher.com/python/set-pop)*,* [*set.remove()*](https://www.tutorialsteacher.com/python/set-remove)

* **Python dictionaries**

It is form of keys and values pair

*dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}*

*dict['Name']*

*Zara*

*dict['Age'] = 8;* # update existing entry

*del dict['Name'];* # remove entry with key 'Name'

*dict.clear();* # remove all entries in dict

*pop\_ele = dict.pop(‘Name’)*

*{'Age': 7, 'Class': 'First'}*

*d = {'a': 10, 'b': 20, 'c': 30}*

*d.items()*

*dict\_items([('a', 10), ('b', 20), ('c', 30)])*

*list(d.keys())*

*['a', 'b', 'c']*

*list(d.values())*

*[10, 20, 30]*

*d1 = {'a': 10, 'b': 20, 'c': 30}*

*d2 = {'b': 200, 'd': 400}*

*d1.update(d2)*

*d1*

*{'a': 10, 'b': 200, 'c': 30, 'd': 400}*

1. **Python conditional statements**

* **if statement**

*food = 'spam'*

*if food == 'spam':*

*print('Ummmm, my favorite!')*

* **if else statement**

*if food == 'spam':*

*print('Ummmm, my favorite!')*

*else:*

*print("No, I won't have it. I want spam!")*

* **elif statement**

*var=input(‘Enter your char from a to c’)*

*if choice == 'a':*

*print("You chose 'a'.")*

*elif choice == 'b':*

*print("You chose 'b'.")*

*elif choice == 'c':*

*print("You chose 'c'.")*

*else:*

*print("Invalid choice.")*

**input function is used to get the values from user side**

* **Nested conditionals**

*if x < y:*

*STATEMENTS\_A*

*else:*

*if x > y:*

*STATEMENTS\_B*

*else:*

*STATEMENTS\_C*

* **Recursion** : repeat the same task until and unless the desired condition don’t meet

*def factorial(n):*

*if n in [0,1]:*

*return 1*

*else:*

*return n\*factorial(n-1)*

* **The for loop**

*for friend in ['Margot', 'Kathryn', 'Prisila']:*

*print(friend)*

*Margot*

*Kathryn*

*Prisila*

* **The while statement**

*i=1*

*While i<6:*

*Print (i)*

*i+=1*

* **The break statement**

To break the statement at condition, after that no any execution will be proceed

*for i in [12, 16, 17, 24, 29]:*

*if i % 2 == 1: # if the number is odd*

*break*

* **The continue statement**

It skip that value given in condition

*for i in [12, 16, 17, 24, 29, 30]:*

*if i % 2 == 1: # if the number is odd, don't process it*

*continue*

*print(i)*

* **List comprehensions**

*numbers = [1, 2, 3, 4]*

*[x\*\*2 for x in numbers]*

*[1, 4, 9, 16]*

* **Range function**

range(3) : range(0,1) I can iterate this value by looping

*for i in range(2,5):*

*print(i)*

*2*

*3*

*4*

1. **File and exception handling**

* **File**

**File open**

*open(filename, mode)*

*“ r “,* for reading.

*“ w “,* for writing.

*“ a “,* for appending.: it append the text from last time cursor is present in file

*“ r+ “,* for both reading and writing

**read**

*file = open("file.text", "r")*

*print (file.read())*

**write**

*file = open('file.txt','w')*

*file.write("This is the write command")*

*file.write("It allows us to write in a particular file")*

*file.close()*

**append the text**

*file = open('file.txt','a')*

*file.write("This will add this line")*

*file.close()*

* **Os module**

*import os*

# Get the current working

*cwd = os.getcwd()*

**Change**

*os.chdir('path')*

**Present directory**

*pwd()*

**Directory making**

*os.makedirs(path)*

**Directory listing**

*os.listdir(path)*

**Remove the Directory**

*os.rmdir(path)*

**Remove the file**

*os.remove(path)*

* **Exception handling**

**Error**

1. ZeroDivisionError: Occurs when a number is divided by zero.
2. NameError: It occurs when a name is not found. It may be local or global.
3. IndentationError: If incorrect indentation is given.
4. IOError: It occurs when Input Output operation fails.
5. EOFError: It occurs when the end of the file is reached, and yet operations are being performed

**Exception**

*try:*

*a = int(input("Enter a:"))*

*b = int(input("Enter b:"))*

*c = a/b*

*except:*

*print("Can't divide with zero")*

1. **Object oriented concepts**

* **Oops concepts**

***Class***

*class Employee:*

*def \_\_init\_\_(self, name, id):*

*self.name = name*

*self.id = id*

**Inheritance ( object )**

*r=Employee(‘ram’,abc)*

* **Generators**

It is used for memory efficiency

*def infinite\_sequence():*

*num = 0*

*while True:*

*yield num*

*num += 1*

*gen = infinite\_sequence()*

*next(gen)*

*0*

And so on get upto want, withough gen it will go upto infinfinite and system will crass.

* **Inheritance**

It is the process to make another type of class by using of another class , that class is called parent class

*class Dog:*

*species = "Canis familiaris"*

*def \_\_init\_\_(self, name, age):*

*self.name = name*

*self.age = age*

*def \_\_str\_\_(self):*

*return f"{self.name} is {self.age} years old"*

*parenting the class*

*class Bulldog(Dog):*

*pass*

**instantiate class**

*jim = Bulldog("Jim", 5)*

*inherit*

*jim.speak("Woof")*

*'Jim says Woof'*

* **Polymorphism**

same names but carrying different functionalities

*class Audi:*

*def description(self):*

*print("This the description function of class AUDI.")*

*class BMW:*

*def description(self):*

*print("This the description function of class BMW.")*

*audi = Audi()*

*bmw = BMW()*

*for car in (audi,bmw):*

*car.description()*

This the description function of class AUDI

This the description function of class BM

* **Lambda functions**

*x = lambda a, b : a \* b*

*print(x(5, 6))*

*30*

* **Debugging of code**

import pdb

*def functionB(first\_val=23, last\_val=72):*

*pdb.set\_trace()*

*response = funcA(first\_val, last\_vale)*

*result = response \* first\_val / 7*

*return result*

* **Python module and user define module**

Like “os module” is python module but when made a function sum of two number in “test.py” then we want to use in different programming we just “import test” in that file.

1. **Tk inter library used for make GUI.**
2. **Beatifulsoup, tk inter and reauests libraruy**