**Test : 1**

**Q.1 Difference between python version 2 and version 3 ? how do you check current version of python ?**

**Ans : Python version 2 :**

* Print function brackets optional
* Prefix string with u to make unicode string
* Division of integers always returns integer Ex: 5/2 =2
* Raw\_input() reads string
* Input() evaluates data read
* Generator.next() used

**Python version 3 :**

* print()function brackets compalsory
* string is unicode by default
* Division of integer may result in float Ex: 5/2=2.5
* Raw\_input()not available
* Input()always reads string
* Next(generator) used

There are 3 ways to check python version **–**

**1.By using IDLE interactive shell—**

* Search->IDLE-->open-->interactive shell-->first line python version 3.9.2

**2.By using CMD(Command Prompt) –**

* Open CMD-->write python -->enter
* Open CMD-->write python -- version-->enter
* Open CMD-->write py -- version -->enter

**3.By using Writing script –**

* Python IDLE to create file -->

Import sys #sys module in python

Print(Sys.version) #version variable

Then , save and run

**Q.2 Difference between local variable and global variable ?exaples?**

**Ans :**

**Global Variable :**

-- Variables that are defined and declared outside a function and need to be used inside a function are called globle variable.

--These variable are accessible to all the function of the program.

--The values of these variable can be changed as they are accessible by various functions.

Ex : x= “global” #global variable

def hello():

print(“x is in side function”,x)

hello()

print(“x is out side function”,x)

**Output:** x is in side function global

x is out side function global

**Local Variable :**

--When a variable is declared inside the function body , it is called a loacl variable.

-- This variable are accessible only within the function they are declared.

--They are secure and reliable as they are accessible by limited function so the values can not be changed.

Ex : def hello():

X=’global’ #local variable

Print(‘x’)

hello()

Print(“ “)

Print(x) # error

**Output :** global

**Q.3 Waht are data types in python ?**

**Ans:** Datatype represents the different kinds of values that we stored on the variable.

There are 6 types of datatypes :

1.Numbers: int , float, complex number

2.String

3.Dictionary

4.set

5.List

6.Tuple

**1.Numbers: int , float, complex number :**Python built in datatypes includes Numeric types.Python numbers are the ones with numeric types.

--To store number types, such as integer,floats,etc.Python numbers includes datatypes,like the int,float,long and complex data type.let us see some examples before going through each datatype:

Ex: 1)Interger – a=5

Print(a,”is of type”,type(a))

Output – 5 is of type<class’int’>

2)Float--- a=2.0

Print(a,”is of type”,type(a))

Output – 2.0 is of type<class’float’>

3)Complex number --- a=1+2j

Print(a,”is of type”,type(a))

Output – 1+2j is of type<class’complex’>

**2.String :** A string is a collection of one or more characters put in a single quote,duoble quote or triple quote.

-- Multiline strings can be denited using triple quote , ‘’’ or ‘’’.

Ex : s=’This is a string ‘

Print(s,type(s))

Output – This is a string <class ‘str’>

**3.Dictionary :**Dictionary is a data structure in which we represent a group of object as key-value pair.

-- Syntax : dict\_name={key:value}

-- Indexing and slicing not work.

-- Hetrogeneous elements are allowed(like int,float,string etc.)

-- Mutable in nature.

-- Key must be unique but dublicates value are allowed.

Ex : a) dict={} # empty dictionary

b) dict={‘name’:’ankit’, ‘age’:29}

**4.set :** set is a data structure which is also called collection of items, in which we can represent a group of unique value as a single entity.

-- Syntax : set\_name={item1,item2,……………….itemn}

-- We write the items of set inside the curly brackets “ {}”.

-- Indexing order are not fixed.

-- Indexing and slicing not work.

-- Hetrogeneous elements are allowed(like int,float,string etc.)

-- Mutable in nature.

-- Dublicates value are not allowed.

Ex. a) var=set() # empty set

b)set={10,’ankit’,56.3,’true’}

**5.List** : List is a data structure which is also called collection of items in which we can store anything like string, float, int etc.

-- Syntax : list\_name=[item1,item2,……………….itemn]

-- We write the item of the list inside square bracket and each item is separeted by comma.

-- Dublicates value are allowed.

-- Mutable in nature.

Ex : a) list=[] #empty list

b) list=[ 10,’ankit’,56.3,’true’ ]

**6.Tuple:** Tuple is a data structure which is also called collection of items in which we can store anything like string, float, int,boolean etc.

-- Syntax : tuple\_name=(item1,item2,……………….itemn)

-- We write the item of the list inside round bracket and each item is separeted by comma.

-- Dublicates value are allowed.

-- Immutable in nature.

Ex : a)var=() # empty tuple

b)var=(1,2,55.6,’true’,10)

**Q.4 What is difference between mutable and immutable data types? How many data types are mutable or not ? please provide reason also with example ?**

**Ans :**

**Mutable :**

-- You can change content without changing identity.

-- Ex.List , Set , Dictionary

**Immutable :**

-- You can not change immutable objects once created.

--Ex. String , Tuple , int, float, complex

**Q.5 Why we use loops ? Please provide 3 examples of for loop and 3 exaples of while loops?**

**Ans :** Sometimes we need to run a block of code a number of times.

**There are two types of loops:**

**1.For loop :** (Definite loop) A loop that has set number of times to run.

**--Syntax** : for variable in range(start ,end, increment)

**--Ex : 1)** Write a program to number table print ?

A= int(input(“enter a number:”))

For number in range(1,11):

Print(A\*number)

Output : enter a number :5

5 10 15 20 25 30 35 40 45 50

**2)** for I in range(1,6):

Print(I,end=” “)

**Output :**1 2 3 4 5

**3)** a=[]

B=”Sam”

For i in b:

a.append()

**Output**: [‘s’ , ‘a’, ‘m’ ]

**2.while Loop** : (Indefinite loop ) A loop that has indefinite number of times to run.

**--Syntax** : while condition :

Statement

--**Ex**. 1) to print sqaure

i=1

while i<=10:

print(“sqaure of :”,i , “is”, i\*\*2)

i=i+1

**Output:** Square of 1 is: 1

Square of 2 is: 4

Square of 3 is: 9

…. Square of 10 is: 100

**2)** Write a program to print 1 to 4 ?

i= 4

n=4

while(i<=n):

print(i)

i=i+1

**Output :** 1 2 3 4

**3)** Write a user define program who print users according number ?

i=int(input(“enter the first no :”))

n=int(input(“enter last no:”))

p=int(input(“enter step no:”))

while(i<=n):

print(i,end= “ “)

i=i+p

**Output :** enter first no: 0

Enter last no:11

Enter step no:2

2 4 6 8 10

**Q.6 Why we use functions ? Difference between lambda , filter ,reduce , map function with example ?**

**Ans:** Function is a block of code or set of instruction assigned for a perticular task.

-- Function reduces code redundancy.

--function consumes less memory.

-- function reduces size of program.

Types of function :

* **Built in function :**

Ex. input() ,range(),count()

* **User define function:**

Ex : add() , sum()

-- def keyword is used to create a function

-- Systax : def functionname():

\\ Instruction

**--EX:** def add():

A=4

B=5

C=A+B

Print(“addition is :”, c)

add()

**Output:** addition is :9

**1.lambda() :** A lambda function is a function without any name , we can make it using lambda keyword.

-- A lambda function can take any number of arguments but can only have one expression.

**-- syntax:** lambda arguments: expression

--Ex: res=lambda x,y : x+y

Print(res(4,5)

Output :9

**2.filter() :** This method a sequence by a function.

-- The filter function returns an iterator where the items are filtered through a function to test.

-- If the item is accepted return True or Flase.

**--Syntax :** filter(function, sequence)

**3.reduce():** Reduce function is used to apply a function to all the element of a seqence.

-- The reduce function is used to apply a perticular function passed in its argument to all of the list elements mentioned in the sequence passed along.

-- the function is defined in ‘functools’ module.

-- At first step , first two element of sequence are picked and the result is obtained.

**--Syntax :** import functools

Functools.reduce(fun,seq)

**4.map():** This function returns the result after applying a function to each item of given sequence like tuple , list , set , dict etc.

-- Map function execute a specified function for each item in a iterable. The item is sent to the function.

**-- Syntax:** map(Fun, seq)

-- Ex : def fact(n):

F=1

for I in range(1,1+n):

F=f\*i

Return F

Li=[2,3,4,5]

Res=map(fact,Li)

Print(list,(Res))

Output: [2,6,24,120]

**Q.7 Difference between list and tuple ? please provide index() , insert() , append() , reverse() examples in list ?**

**Ans: List :**

-- List is a comma separted values in square brackets and square bracket is mandatory.

Ex. A=[‘jay’, 23,3000]

--List is mutable.

--List is slow in execution than tupe.List is less efficient in memory utilization than tuple.

--Comprehension concept is applicable only for list.

--List support packing but not support unpacking.

Ex: a) list =[1,5,66,’ankit’]

list.insert(3,’learn coding’)

Print(list)

Output :[1,5,66,’learn coding’,’ankit’] # Updation allowed in list

**Tuple :**

--Tuple is a comma separed values in round bracket Ex: a=(‘jay’ , 23, 3000)

--Tuple is immutable

--Tuple object takes less memory than list object for same data.

--Comprehension concept not applicable for tuple.

--Tuple supports both packing and unpacking

Ex : a) var =(55,10,20,’True’)

Var.append(‘kirti’)

Print(var)

Output : Error , because immutable nature ,updation are not allowed in tuple.

**1)index()** – This function is used to find the index of the element.This function return the first index of the object if it is found otherwise it return as exception showing that element is not found. Index() function is used to indicate the position of an item in a list.

Syntax – list.index(element)

**Ex**. L1= [11,50,2.12,66]

Print(L1.index(50))

**Output** -- 1

**2)insert() –** This function is used to insert an object at the given indexing.

**Syntax** – list.insert(index\_no,element)

**Ex.** List=[‘sam’,89,’rahul’]

a.insert(0, ‘pooja’)

a

**Output** – [‘pooja’ , ‘sam’ , 89, ‘rahul’]

**3)append() –** This function is used to add an element in the last indexing of the list.

**Ex.** a =[‘sam’ ,’preeti’ ,51,4.22]

a.append(666)

a

**Output ---** [‘sam’ , ‘preeti’ , 51 , 4.22 , 666]

**4)reverse() –** This function is used to reverse an element present in the list.

**Syntax** – list.reverse()

**Ex.** a=[‘ram’ , ‘sham’ ,25, ‘riya’ ]

a.reverse()

a

**Output –** [‘riya’ ,25, ‘sham’ , ‘ram’ ]

**Q.8 Difference between dictionary and set ? Please provide keys() , values() , items() example of dictionary and union() , intersection() in sets ? Is frozenset is mutable or not ?**

**Ans: Dictionary :**

-- Dictionary in python is an unordered collection of data values , used to store data values like a map, duplicates allowed.

-- Identification : { key: value }

-- Ex. Dict ={ ‘Name’:’Dipa’, ‘Phoneno’:’123545’}

-- In dictionary keys also unique and elements are accessed using the keys.

**1) keys() –** This function is used to prints the keys in the dictionary.

**Ex .** dict ={ ‘name’: ‘sham’ , ‘age’:30 ,’location’:’pune’}

Print(dict.keys())

**Output** – name, age , location

**2) values() —** This function is used to prints the values in the dictionary.

Ex. dict ={ ‘name’: ‘sham’ , ‘age’:30 ,’location’:’pune’}

Print(dict. values())

**Output** – sham , 30 , pune

**3)items() --** This function is used to prints the keys and values in the dictionary.

**Ex .** dict ={ ‘name’: ‘sham’ , ‘age’:30 ,’location’:’pune’}

Print(dict. items())

**Output –** name : sham

Age : 30

Location : pune

**Set :**

-- Set is an unordered collection of data type that is iterable ,mutable and has no duplicate elements.

-- Identification ={ }

-- EX. Set={1, 5,’true’,55.2,3,4}

**1)union() –** The union of two sets contains all the elements contained in either set (or bothsets)

**Ex**. a={ ‘mohit’ , ‘sam’ , ‘raj ‘}

b = {‘kirti’ , ‘sam ‘}

c= a.union(b ) // c= a| b

c

**Output –** {‘ kirti’ , ‘mohit’ ,’raj’ , ‘sam’}

**2) intersection() – The intersection contains all the elements in the both sets.**

**Ex .** a={ ‘mohit’ , ‘sam’ , ‘raj ‘}

b = {‘kirti’ , ‘sam ‘}

c= a.intersection(b ) // c= a & b

c

**Output –** { ‘sam’}

**Frozen set :**

-- Frozen set ia an immutable version of python set.

-- while elements of a set can be modified at any time , elements of the frozen set remain the same after creation.

-- By using frozenset() inbuilt function.

-- **Syntax :** frozenset(items)

--There are no methods like add , update , remove , for frozenset like set.

-- frozen:unchangeble

**--Ex.** Vowels={‘a’,’e’,’o’,’u’,’I’}

Vowels.add(‘f’)

Vowels

**Output**: error

**Q.9 What are decorators ? Please provide examples ?**

**Ans .** Function which takes other functions as input , add additional functionalities and return it.

Function ---------- > Decorator ----------------- > Function

It is a callable python object which modifies other functions / class.

**Ex .** def décor(func):

def inner():

func() # existing functionality

# add new functionality

Print(“welcome”)

return inner

def printer():

print(“welcome”)

print(“welcome”)

printer=décor(printer)

printer()

**Output –** welcome

Welcome

Welcome

**Q.10 What are list comprehension ? Please provide examples ?**

**Ans .** List comprehension is used for to create a new list .

List comprehension is an elegant way to define and create lists based on existing list.

List comprehension is generally more compact and faster than normal functions and loops for creating list.

**Syntax --** [ expression for item in list ]

Ex. 1) n= [ m for m in range(1,101) ]

Print(n)

Output – [ 1, 2 , 3 , …………………………….. , 100 ]

2) n= [m for m in range (1,101) if m%2 == 0 ]

Print(n)

Output – [2,4,6,8,10,…………………………..,100 ]

**Q.11 How do you create an array from a user defined list ?**

**Ans.** a = [ ]

size = int(input(“enter size :”))

for i in range(size):

val= int(input(“enter number :”))

a.append(val)

print(a)

b=np.array(a)

b

**Output --** Enter size : 3

Enter number : 22

Enter number : 33

Enter number : 44

Array([22 ,33 ,44])

**Q.12 Difference between hstack() and vstack() with example?**

**Ans. hstack() --** The hstack() function is used to stack the sequence of input arrays horizontally to make a single array .

**Ex.** a= np.array([1,2,3,4])

b = np.array([5,6,7,8])

np.hstack((a,b))

**Output –** array ( [1,2,3,4,5,6,7,8 ])

**Vstack() --** The vstack() function is used to stack the sequence of input arrays vertically to make a single array .

Ex. a= np.array([1,2,3,4])

b = np.array([5,6,7,8])

np.vstack((a,b))

**Output –** array ( [1,2,3,4] ,

[5,6,7,8 ] )

**Q.13 What is zeros() , ones() , eye() , diag() , randint() , rand() , seed() , linspace() , unique() in NumPy ? Please provide 1-1 examples of these functions ?**

**Ans.**

**zeros() –** Using this function we can create an array with all the values of zeros.

Ex. a=np.zeros(4)

a

Output – array([0.,0.,0.,0.])

**Ones( ) --** By this function we can create an array in which all the values are one.

**Ex.** a = np.ones(2 , 2)

a

**Output –** array([1., 1.],

[1.,1.])

**eye( ) --** In this function we create our diagonal position elements with one and rest all are zero**.**

**Ex**. a =np.eye(3,3)

A

Output – array([1.,0.,0.],

[0.,1.,0.],

[0.,0.,1.])

**diag ( ) –** Using this function we can pass our custom array in diagonal position.

Ex. a=np.array([ 1,34,67])

a

np.diag(a)

Output --- array([1, 0, 0],

[0, 34, 0],

[0, 0, 67])

**randint( )—**It will generate random number between given range.

**Syntax --** np.random.randint(min\_number,max\_num,total\_num)

**Ex.** a =np.random.randint(1,21,3)

a

Output – array([10,4,6])

**rand()** – It will create random number between 0 to 1.

**Ex.** a= np.random.rand(3)

a

**Output –** array([0.5603, 0.277 , 0.6023])

**seed() –** It will fix random numbers.

**Ex.** np.random.seed(32)

a = np . random . randint(1 , 10 , 3)

a

**Output –** array([8,6,7 ])

**linspace**()— This function return value between a given range and with a same gap between consecutive element.

**Ex.** a = np.linspace(1,2,5)

a

**Output ---** array([1,1.25, 1.75,1.5,2,])

**unique( )—**The array with unique values. The array with respective index values. The array with counting of frequency.

Syntax – np.unique (arr , return\_index =True , return\_counts=True )

Ex. a = np.array([10,20,30,40,50,10,20,30,30,30,30])

np.unique (a , return\_index =True , return\_counts=True )

Output—(array([10,20,30,40,50]),

array([0,1,2,5,6],dtype=int64),

array([3,3,7,1,1],dtype=int64))

**Q.14 How can you reshape your array in numpy ? Please provide example also ?**

**Ans. We can shape the numpy arrays in two ways :**

**1)reshape ( ) 2) resize( )**

**1)reshape ( ) ---** This reshape function gives new shape as an array withaout changing its data.

**Syntax** – np.reshape(array ,shape , order )

Returns: ndarray with mentioned shape

Returnd ndarray elements may be copy of the original elements or view of original array elements.

Ex. a=np.arrange(6)

a

>> array([0,1,2,3,4,5])

a.shape( )

>> (6,)

b = np.reshape(a,(2,3))

b

>> array([0,1,2],

[3,4,5])

b.shape()

>> (2,3)

**2) resize ( ) --** This function return new array with specified shapes.this function isnused to the change of shape the array.

Syntax – np.array(array name, shape)

Ex. a=np.arrange(5)

a

>> array([1,2,3,4])

np . resize(a,(2 , 3) )

>> array([0,1,2],

[3,4,0])

**Q.15 what is argmin() and argmax() in numpy ? please exaplain using an example ?**

**Ans.**

**argmin( ) –** This function is indicates the index at which the minimum number is located.

Ex. a= np. Array([1,0,2,5,8])

np.argmin(a)

Output – 1

**argmax ( ) --** This function is indicates the index at which the maximum number is located.

Ex. a= np. array([1,0,2,5,8])

np.argmax(a)

Output – 4