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

**Ans: Ans : Python version 2 :**

* print statement is used as "print"
* Python 2 was released in the year 2000.Division of integers always returns integer
* strings are stored as ASCII by default.
* exceptions are enclosed in notations.
* has more complicated syntax than Python 3.
* A lot of libraries of Python 2 are not forward compatible.
* Division of integers always returns integer Ex: 5/2 =2
* More complex and difficult to interprete

**Python version 3 :**

* Python 3 it is used as "print()"
* Python 3 was released in the year 2008.
* string is unicode by default
* exceptions are enclosed in parentheses.
* has an easier syntax compared to Python 2.
* A lot of libraries are created in Python 3 to be strictly used
* Division of integer may result in float Ex: 5/2=2.5
* Readable and easily understandable

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

**Local Variable:**

* Local variables are declared within the functions.
* Local variables are kept on the stack.(Storage)
* For local variables, parameter passing is necessary.
* Changes in a local variable doesn't affect other functions of the program.
* If the local variable is not initialized, it defaults to a garbage value.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Global Variable**   * Global variables are declared outside the functions.  |  |  |  |  |  | | --- | --- | --- | --- | --- | | * Global variables are kept in a fixed location selected by the compiler.  |  | | --- | | * For global variables, parameter passing is not necessary. | | |  | | --- | | * Changes in a global variable is reflected throughout the code * The global variable takes zero by default if it is not initialized.. | |  | | | |  | |
|  |

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

**Ans:**

1). Data Type ==> (a). Integer (b). Float (c). String (d). Boolean

**Integer**  ==> 7,89,-1,-456   etc .

**Float** ==> 4.67 , 8.345, 64.234 etc

**String** ==> 'sam' , \"sam\" , '''sam'''

**Boolean** ==> True , False

**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, Booleans,Numbers,complex

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

**Ans:**

Loops are important in Python or in any other programming language as they help you to execute a block of code repeatedly.   
If you want to do something again and again then you can use the looping concept.

**Loops in Python  ==> (1). for loop   (2). while loop**

**Example:**

**(1). for loop**

**# index = 0 , length = 1**

1)for i in range(5):

     print(i)

output :

      0

      1

      2

      3

      4

2)for i in range(1,6):

        print(i)

output :

      1

      2

      3

      4

      5

3) for i in range(1,8,3):

        print(i)

Output:

      1

      4

      7

**# while loop ==>**

**(1)Initialization(2)Condition(3). Increment or decrement "**

**Example: 1**

i = 1

    n = 4

    while(i<=n):

 print(i)

        i = i + 1

output:

1

2

3

4

**2)**

i = 2

    n = int(input("Enter last number:"))

    while(i<=n):

        print(i)

        i = i + 2

**Output: final output ==> 2,4,6"**

       "Enter number:7

       "Sum of even numbers =  12

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

Ans:

With the help of functions, we can avoid rewriting the same logic or code again and again in a program

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

    print("Total product is = " , x(3,4))

  Output: Total product is = 12

**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 False.

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

--Ex ages = [5,12,18,30,56,78,90]

def myfunc(x):

  if(x<=18):

      return True

  else:

      return False

kid = list(filter(myfunc , ages))

kid

Output :

[5, 12, 18]

**3.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 : for i in range(len(a)):

  a[i] = int(a[i])

print(a)

Output: [67, 89, 345]

b = list(map(int , a))

b

Output: [67, 89, 345]

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

**Ans:**

**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.

**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

**1)index() ==>** This function is used to find the index of the object/element .

    "# This function returns the first index of the object

    "# if it is found otherwise it returns an exceptions showing that the element is not found .

**# syntax : list.index(object)**

**# Example ==>**

    a = ['sam' , 456 , 'mohit' , 98.67]

    print(a.index('mohit'))"

**Output** :  2

**2)insert (index, object) ==>** This function is used to insert an object at the given indexing.

**Syntax : list.insert(index , object)**

**"# Example ==>**

    a = ['sam' , 456 , 'mohit' , 98.67]

    print(a)

    a.insert(1 , "rahul")

    "print(a)

**Output**:

      ['sam', 456, 'mohit', 98.67]

      ['sam', 'rahul', 456, 'mohit', 98.67]

**3)reverse() ==>** This function is used to reverse an element present in the list

**"# syntax : list.reverse()**

**"# Example ==>**

    a = ['sam' , 456 , 'mohit' , 98.67]

    a.reverse()

    print(a)

**Output:** [98.67,’mohit’,456,’sam’]

**4)append() ==>** This function is used to add an element in the last indexing of container.

**# syntax : list.append(object**)

**# Example ==>**

    a = ['sam' , 456 , 'mohit' , 98.67]

    print(a)

    a.append("kriti")

    print(a)

**Output**:

       ['sam', 456, 'mohit', 98.67]

       ['sam', 456, 'mohit', 98.67, 'kriti']

**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 :**

* An ordered collection of key-value pairs.
* Curly braces {}, but with key-value pairs.
* Ordered (from Python 3.7 onwards).
* Does not allow duplicate keys.
* Mutable (can add, modify, or remove key-value pairs).
* Values are accessed using keys.
* To store related pieces of information.
* Exam: my\_dict = {“name”: “Alice”, “age”: 30}

**Set :**

* An unordered collection of unique elements
* Curly braces {}.
* Unordered.
* Does not allow duplicate elements.
* Mutable (can add or remove elements).
* Elements are accessed directly.
* To store unique elements.
* Exam: my\_set = {1, 2, 3}

The **keys()** method extracts the keys of the dictionary and returns the list of keys as a view object.

Syntax: dict.keys()

**Example:**

**{1: ‘One’ , 2: ‘tow’ , 3: ‘Three’}**

**Dictionarykeys = numberskeys()**

**Print(dictionarykeys)**

**Output:**

**Dict\_keys([1,2,3])**

The .**values()** method returns all of its values of a Python dictionary in a view object that will reflect any changes to the dictionary values. It takes no arguments.

Syntax: dictionary.values()

Sales : { ‘apple’ :2 , ‘orange’ : 3 , ‘grapes’ : 4 }

Print(sales.value())

Output:

Dict\_values([2,3,4])

**items()**

The **items()** method returns a view object that displays a list of dictionary's (key, value) tuple pairs.

**Example**: {‘physics’ : 57 , ‘Mathes’ : 80}

Print(marke.items())

Output:

Dict\_items([(‘physics’ : 57),( ‘Mathes’ : 80)])

The Python set **union()** method returns a new set with distinct elements from all the sets.

**Frozen set :**

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

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

[**Decorators**](https://www.geeksforgeeks.org/function-decorators-in-python-set-1-introduction/) are a very powerful and useful tool in Python since it allows programmers to modify the behaviour of a function or class.

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

List comprehension offers a shorter syntax when you want to create a new list based on the values of an existing list.

**Example:**

letters = list(map(lambda x: x, ‘human’))

print(letters)

Output:

[‘h’ , ‘u’, ‘m’ , ‘a’ , ‘n’]

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

a = []

size = int(input("Enter size :"))

for i in range(size):

  val = int(input("Enter number :"))

  a.append(val)

b = np.array(a)

b

**Output:**

**Output:**

Enter size :5

Enter number :45

Enter number :12

Enter number :85

Enter number :15

Enter number :5656

array([  45,   12,   85,   15, 5656])

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

**Ans :**

HStack positions views in a horizontal line,

VStack positions them in a vertical line

**Exp:**

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

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

c = np.array([9,10,11,12])

np.hstack((a ,b))

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

**Exp:**

np.vstack((a,b,c))

Output:

array ([[ 1, 2, 3, 4], [ 5, 6, 7, 8], [ 9, 10, 11, 12]])

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

**Ans:**

**(1). ZEROS() ==> This function creates an array(either one dimensional or multi-dimensional) and fill all the values with zeros .**

**Exam:**

import numpy as np

a = np.zeros(5)    #

a

Output: array([0., 0., 0., 0., 0.])

**2). Ones() ==>** **This function creates an array(either one dimensional or multi-dimensional) and fill all the values with one .**

**Exam:**

import numpy as np

a = np.ones(5)

a

**Output:** array([1., 1., 1., 1., 1.])

**(3). eye() ==>** **This function creates an array with all the digonal elements as 1 and rest all are 0."**

**#Square Matrix ==> n(rows)=n(columns) ==> digonal elements[(0,0),(1,1),(2,2)..(n,n)]**

**Example:**

a = np.eye(3,3) # jisme no of row no of colume ke equal hote h squre matrix

a

Output: array([[1., 0., 0.], [0., 1., 0.], [0., 0., 1.]])

**(4). Diag() ==> This function creates a 2-Dimensional array with all the digonal elements as the given value and rest all are zero . "**

**Example:**

 = np.diag([1,4,8,90])   #list se array me convert karna

 a

Output:

array([[ 1, 0, 0, 0], [ 0, 4, 0, 0], [ 0, 0, 8, 0], [ 0, 0, 0, 90]])

**5)Randint() ==> This function is used to generate a random number between a given range .**

**"# syntax : randint(min\_value , max\_value ,total\_numbers)**

**Example:**

a = np.random.randint(1,10,4)

a

**Output:** array([8, 8, 3, 9])

**(6). Rand() ==> This function is used to generate random number between 0 to 1**

**"# syntax : rand(number\_of\_value)**

    a = np.random.rand(5)

    a

**Output:**

   array([0.23485556, 0.66067748, 0.67975315, 0.34019032, 0.50015791])

**7)Seed Function ==>**

**"# Random Numbers ==> Refresh ==> Random Numbers update ==> model accuracy 80% , model accuracy**

**"# Seed Function ==> Call the seed() ==>pass the  random number ==> id's fix in memory location ==> not  update"**

**Exp :**

  a = np.random.randint(1,50,12)  # random me hum jab bhi enter karenge bar bar output value change hoke aayegi

  a

Output:

array([39, 39, 19, 11, 21, 21, 36, 31, 18, 21, 29, 25])

**8)LinSpace() ==> This function returns values between a given range and with a same gap between consicutive elements . "**

**Exp:**

import numpy as np

a = np.linspace(1,2,5) ## 1.25 - 1 = 0.25 , 1.50-1.25 = 0.25 , 1.75

a

Output:

array([1. , 1.25, 1.5 , 1.75, 2. ])

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

import numpy as np

a = [1,2,34]

b = np.array(a)

b

**Ans:** array ([ 1, 2, 34])

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

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

np.min(a)

**Output :**

10

**Exp:**

np.argmin(a)

**Output :**

0

**Exp:**

np.argmax(a)

**Output :**

4