**Week 3**

**Task :**

**Functions –**

**Q 1. Built-in Functions:**

1. print( ) function
2. type( ) function
3. input( ) function
4. abs( ) function
5. pow( ) function
6. dir( ) function
7. sorted( ) function
8. max( ) function
9. round( ) function
10. divmod( ) function
11. id( ) function
12. ord( ) function
13. len( ) function
14. sum( ) function
15. help( ) function
16. **Print() Function :**

The print() function prints the specified message to the screen or another standard output device. The message that wants to print can be a string or any other object. This function converts the object into a string before written to the screen.

Eg. print(“Hello Python Users, From Celebal Technology !!”)

Output : Hello Python Users, From Celebal Technology !!

1. **Type() Function :**

The type() function returns the type of the specified object.

Eg. var1 = "Celebal Technology"

var2 = 200

print(type(var1))

print(type(var2))

Output : <class 'str'>

<class 'int'>

1. **Input() Function :**

The input() function allows taking the input from the user.

Eg. a = input('Enter your name: ')

print(“Hello, ” + a + “ From Sanjivani.”)

Output : Enter your name: Mansi

Hello, Mansi From Sanjivani.

1. **Abs() Function :**

The abs() function returns the absolute value of the specified number.

Eg. positive\_no=99

print(abs(positive\_no))

complex\_no=4+8j

print(abs(b))

Output : 99

8.94427190999916

1. **pow( ) Function :**

The pow() function returns the calculated value of x to the power of y i.e xy. If a third parameter is present in this function, then it returns x to the power of y, modulus z.

Eg. x = pow(8, 4)

print(“Power of X: ”, x)

Output : Power of X: 4096

1. **dir( ) Function :**

The dir() function returns all the properties and methods of the specified object, without the values. dir() tries to return a valid list of attributes of the object it is called upon.

For Class Objects, it returns a list of names of all the valid attributes and base attributes as well.

For Modules/Library objects, it tries to return a list of names of all the attributes, contained in that module.

If no parameters are passed it returns a list of names in the current local scope.

Eg.

class Student:

def \_\_dir\_\_(self):

return['student\_name', 'student\_rollno', 'student\_marks', 'student\_admission\_date', 'student\_address']

my\_obj = Student()

print(dir(my\_obj))

Output : ['student\_address', 'student\_admission\_date', 'student\_marks', 'student\_name', 'student\_rollno']

1. **sorted( ) function :**

The sorted() function returns a sorted list of the specified iterable object. You can specify the order to be either ascending or descending. In this function, Strings are sorted alphabetically, and numbers are sorted numerically.

Eg. tuple = ("C", "E", "L", "E", "B", "A", "L")

print(sorted(tuple))

Output : ['A', 'B', 'C', 'E', 'E', 'L', 'L']

1. **Max() Function :**

The max() function returns the item with the highest value, or the item with the highest value in an iterable.

Eg. max\_val=(2,5,7,10,15,1,8)

print(max(max\_val))

Output : 15

1. **Round( ) function:**

The round() function returns a floating-point number that is a rounded version of the specified number, with the specified number of decimals. The default number of decimals is 0, meaning that the function will return the nearest integer.

Eg. round\_no = 33.57

print(round(round\_no))

Output : 34

1. **len( ) function :**

The len() function returns the count of items present in a specified object. When the object is a string, then the len() function returns the number of characters present in that string.

Eg. str = “ Celebal ”

print(“Length of string : ” , len(str))

Output : Length of string : 7

**Q 2. Membership Operator in Python.**

Ans :

1. **In Operator :-**

Evaluates to true if it finds a variable in the specified sequence and false otherwise.

Eg. x in y, here in results in a 1 if x is a member of sequence y.

1. **Not In Operator :**

Evaluates to true if it does not finds a variable in the specified sequence and false otherwise.

Eg. x not in y, here not in results in a 1 if x is not a member of sequence y.

Example :

a = 10

b = 20

list = [1, 2, 3, 4, 5 ];

if ( a in list ):

print "Line 1 - a is available in the given list"

else:

print "Line 1 - a is not available in the given list"

if ( b not in list ):

print "Line 2 - b is not available in the given list"

else:

print "Line 2 - b is available in the given list"

**Q 3. Difference Between LIST, TUPLE, SET, DICTIONARY.**

**Ans :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.No** | **List** | **Tuple** | **Set** | **Dictionary** |
| 1. | List is a non-homogeneous data structure that stores the elements in single row and multiple rows and columns | Tuple is also a non-homogeneous data structure that stores single row and multiple rows and columns | Set data structure is also non-homogeneous data structure but stores in single row | Dictionary is also a non-homogeneous data structure which stores key value pairs |
| 2. | List can be represented by [ ] | Tuple can be represented by ( ) | Set can be represented by { } | Dictionary can be represented by { } |
| 3. | List allows duplicate elements | Tuple allows duplicate elements | Set will not allow duplicate elements | Set will not allow duplicate elements and dictionary doesn’t allow duplicate keys. |
| 4. | List can use nested among all | Tuple can use nested among all | Set can use nested among all | Dictionary can use nested among all |
| 5. | Ex: [1, 2, 3, 4, 5] | Ex: (1, 2, 3, 4, 5) | Ex: {1, 2, 3, 4, 5} | Ex: {1, 2, 3, 4, 5} |
| 6. | List can be created using list() function | Tuple can be created using tuple() function. | Set can be created using set() function | Dictionary can be created using dict() function. |
| 7. | List is mutable i.e we can make any changes in list. | Tuple is immutable i.e we can not make any changes in tuple | Set is mutable i.e we can make any changes in set. But elements are not duplicated. | Dictionary is mutable. But Keys are not duplicated. |
| 8. | List is ordered | Tuple is ordered | Set is unordered | Dictionary is ordered |
| 9. | Creating an empty list l=[] | Creating an empty tuple t=() | Creating an empty set a=set() | Creating an empty dictionary d={} |