**In-Built Function**

**print( ):**

To prints the given object to the screen.

print("python is programming language")  
x=56  
print("X=",x)

**output:**

python is programming language

X= 56

**input( ):**

It is used to get an input from the user.

val=input("enter a value:")  
print("you entered:",val)

**output:**

enter a value:10

you entered: 10

**type( ):**

It is used to find the datatype of variable.

a=10  
b="Hi python"  
c=10.5  
print(type(a))  
print(type(b))  
print(type(c))

**output:**

<class 'int'>

<class 'str'>

<class 'float'>

**Strings:-**

1. **str( ):**

It is used to convert a specified value into a string.

x=str('python programming language')  
print("the string is:",x)

**output:**

the string is: python programming language

1. **capitalize( ):**

In the string first character is change as capital.

str="python programming language"  
str1=str.capitalize()  
print("the string is :",str)  
print("after capitalize the string is :",str1)

**output:**

the string is : python programming language

after capitalize the string is : Python programming language

1. **casefold( ):**

It changes the whole string to lower case.

str="PYTHON PROGRAMMING"  
str1="PyThon PROGramming"  
str2=str.casefold()  
str3=str1.casefold()  
print("the string is :",str)  
print("after casefold the string is :",str2)  
print("the string is :",str1)  
print("after casefold the string is :",str3)

**output:**

the string is : PYTHON PROGRAMMING

after casefold the string is : python programming

the string is : PyThon PROGramming

after casefold the string is : python programming

1. **lower( ):**

It is used to convert into lowercase character.

str="PYThon"  
str1=str.lower()  
print("the string are :",str)  
print("after lower the string is :",str1)

**output:**

the string are : PYThon

after lower the string is : python

1. **swapcase( ):**

It convert the lowercase to uppercase or uppercase to lowercase.

str="PYTHON PROGRAMMING"  
str1="python programming"  
str2="PYthon PRogrAMMinG"  
str3=str.swapcase()  
str4=str1.swapcase()  
str5=str2.swapcase()  
print("the string is:",str)  
print("the string after swapcase is:",str3)  
print("the string is:",str1)  
print("the string after swapcase is:",str4)  
print("the string is:",str2)  
print("the string after swapcase is:",str5)

**output:**

the string is: PYTHON PROGRAMMING

the string after swapcase is: python programming

the string is: python programming

the string after swapcase is: PYTHON PROGRAMMING

the string is: PYthon PRogrAMMinG

the string after swapcase is: pyTHON programming

1. **center( ):**

**syntax:**

center(width, fillchar)

str="PYTHON PROGRAMMING"  
str1="python program"  
str2=str.center(25)  
str3=str1.center(20,'#')  
print("the string is :",str)  
print("after center the string is :",str2)  
print("the string is :",str1)  
print("after center the string is :",str3)

**output:**

the string is : PYTHON PROGRAMMING

after center the string is : PYTHON PROGRAMMING

the string is : python program

after center the string is : ###python program###

1. **count( ):**

It is used to count the number of character in the string.

**syntax:**

count(substring, start, end)

str="python programming"  
str1=str.count('m')  
print("the string is :",str)  
print("number of occurences string is :",str1)

**output:**

the string is : python programming

number of occurences string is : 2

str="abbcaddeedaddaabbcca"  
oc=str.count('a',3,12)  
print("the string is :",str)  
print("number of occurrences :",oc)

**output:**

the string is : abbcaddeedaddaabbcca

number of occurrences : 2

1. **find( ):**

**syntax:**

find(substring, start, end)

(It return -1, if the string does not match)

str="welcome to the python programming language"  
str1=str.find("the")  
str2=str.find("is")  
str3=str.find("m",25)  
str4=str.find("a",45)  
str5=str.find("p",20,25)  
print("the string are :",str)  
print("to find the",str1)  
print("to find is :",str2)  
print("to find m :",str3)  
print("to find e :",str4)  
print("to find p :",str5)

**output:**

the string are : welcome to the python programming language

to find the 11

to find is : -1

to find m : 28

to find e : -1

to find p : 22

1. **format( ):**

str="java"  
str1="c#"  
str2="{} and {} both are programming language".format(str,str1)  
print("the string are :",str2)  
str3="{1} and {0} both are programming language".format(str,str1)  
print("the string are :",str3)

**output:**

the string are : java and c# both are programming language

the string are : c# and java both are programming language

val=10  
print("the value is:",val)  
print("decimal: {0:d}".format(val));  
print("binary: {0:b}".format(val));  
print("octal: {0:o}".format(val));  
print("hexadecimal: {0:x}".format(val));

**output:**

the value is: 10

decimal: 10

binary: 1010

octal: 12

hexadecimal: a

val=22/7  
print("decimal :{:}".format(val));  
print("after decimal point limit digit :{:.4}".format(val));  
print("after decimal point limit digit with percentage :{:.4%}".format(val));

**output:**

decimal :3.142857142857143

after decimal point limit digit :3.143

after decimal point limit digit with percentage :314.2857%

n=10  
n1=100000000  
print("decimal:",format(n,"d"))  
print("binary:",format(n,"b"))  
print("octal:",format(n,"o"))  
print("hexadecimal:",format(n,"x"))  
print("after decimal point:",format(n,".2f"))  
print("It separate the numbers to read:",format(n1,","))

**output:**

decimal: 10

binary: 1010

octal: 12

hexadecimal: a

after decimal point: 10.00

It separate the numbers to read: 100,000,000

1. **index( ):**

It is used to return the position of string.

**syntax:**

index(substring, start, end)

str="welcome to the python programming"  
str1=str.index("on")  
str2=str.index(("mat"))  
str3=str.index("p",14,20)  
print("the string is :",str)  
print("position of given string:",str1)  
print("position of given string:",str2)  
print("position of given string:",str3)

**output:**

the string is : welcome to the python programming

position of given string: 19

position of given string: 15

line 3, in <module>

str2=str.index(("mat"))

ValueError: substring not found

1. **join( ):**

str=":"  
l=['1','2','3']  
str1=str.join(l)  
print("the list is :",\*l)  
print("after using join in the string :",str1)

**output:**

the list is : 1 2 3

after using join in the string : 1:2:3

1. **len( ):**

To find the length of string.

str="python"  
str1=len(str)  
print("the string are :",str)  
print("the length of string is :",str1)

**output:**

the string are : python

the length of string is : 6

1. **replace( ):**

**syntax:**

replace(old, new, count)

str1="java is a programming language"  
str2=str1.replace("java","c")  
str3="java c c# java php python java"  
str4=str3.replace("java","c#")  
str5=str3.replace("java","c#",1)  
print("the string is :",str1)  
print("the string after replace is :",str2)  
print("the string is :",str3)  
print("the string after is :",str4)  
print("the string after is :",str5)

**output:**

the string is : java is a programming language

the string after replace is : c is a programming language

the string is : java c c# java php python java

the string after is : c# c c# c# php python c#

the string after is : c# c c# java php python java

1. **reversed( ):**

It is used to reverse the string.

str="python"  
str1=reversed(str)  
print("the string is:",str)  
print("reversed string is: ",end="")  
print("".join(str1))

**output:**

the string is: python

reversed string is: nohtyp

1. **split( ):**

**syntax:**

split(sep = none, maxsplit = -1)

str="python is a programming language"  
str1=str.split()  
print("the string is:",str)  
print("the string is splited:",str1)

**output:**

the string is: python is a programming language

the string is splited: ['python', 'is', 'a', 'programming', 'language']

str="python is a programming language"  
str1=str.split('a')  
print("the string is:",str)  
print("the string is splited:",str1)

**output:**

the string is: python is a programming language

the string is splited: ['python is ', ' progr', 'mming l', 'ngu', 'ge']

1. **zip( ):**

The**zip()** function returns a zip object, which is an iterator of tuples where thefirst item in each passed iterator is paired together, and then the second item in each passed iterator are paired together etc.

If the passed iterators have different lengths, the iterator with the least items decides the length of the new iterator.

**Syntax**

zip(iterator1, iterator2, iterator3 *...*)

**Adding of list value of same position:**

a=[1,2,3,4,5]  
b=[10,11,12,13,14]  
result=[]  
for first,second in zip(a,b):  
 result.append(first + second)  
print(result)

**output:**

[11, 13, 15, 17, 19]

1. **chr( ):**

It is used to get a string representing a character which points to a Unicode code integer.

print("the alphabet are:",)  
for i in range(97,123):  
 print(chr(i),end=" ")

**output:**

the alphabet are:

a b c d e f g h i j k l m n o p q r s t u v w x y z

1. **ord( ):**

It return an integer representing Unicode code.

str="abcdefghijklmnopqrstuvwxyz:;,.!@#$%^&\*()-\_+=}{[]|?+\*-/"  
print("the string is:",str)  
print("ascii code is:")  
for i in str:  
 print(ord(i),end=" ")

**output:**

the string is: abcdefghijklmnopqrstuvwxyz:;,.!@#$%^&\*()-\_+=}{[]|?+\*-/

ascii code is:

97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 58 59 44 46 33 64 35 36 37 94 38 42 40 41 45 95 43 61 125 123 91 93 124 63 43 42 45 47

**Check string:**

1. **isalnum( ):**

To check whether the string is alphanumeric or not.

str="123456"  
str1="welcome123"  
str2="welcome 1234"  
str3=str.isalnum()  
str4=str1.isalnum()  
str5=str2.isalnum()  
print(str3)  
print(str4)  
print(str5)

**output:**

True

True

False

1. **isalpha( ):**

To check whether the string is alphabetic or not.

str="python"  
str1="welcome123"  
str2="1234"  
str3=str.isalpha()  
str4=str1.isalpha()  
str5=str2.isalpha()  
print(str3)  
print(str4)  
print(str5)

**output:**

True

False

False

1. **isdigit( ):**

To check whether the string is digit or not.

str="1234567"  
str1="12-345-678"  
str2="123!@#"  
str3=str.isdigit()  
str4=str1.isdigit()  
str5=str2.isdigit()  
print(str3)  
print(str4)  
print(str5)

**output:**

True

False

False

1. **isdecimal( ):**

To check whether the string is decimal or not.

str1="learn python"  
str2="1234"  
str3="2.50"  
str4="@#$%^"  
str5=str1.isdecimal()  
str6=str2.isdecimal()  
str7=str3.isdecimal()  
str8=str4.isdecimal()  
print(str5)  
print(str6)  
print(str7)  
print(str8)

**output:**

False

True

False

False

1. **isidentifier( ):**

To check whether the string is identifier or not.

str="abcdef"  
str1="20xyz"  
str2="$abra"  
str3=str.isidentifier()  
str4=str1.isidentifier()  
str5=str2.isidentifier()  
print(str3)  
print(str4)  
print(str5)

**output:**

True

False

False

1. **islower( ):**

To check whether the string is lowercase character or not.

str="abcdef"  
str1="Xyz"  
str3=str.islower()  
str4=str1.islower()  
print(str3)  
print(str4)

**output:**

True

False

1. **isupper( ):**

To check whether the string is uppercase character or not.

str="WELCOME TO PYTHON"  
str1="learn python"  
str3=str.isupper()  
str4=str1.isupper()  
print(str3)  
print(str4)

**output:**

True

False

1. **isspace( ):**

It is used to check space in the string.

str=" "  
str1=str.isspace()  
print(str1)

**output:**

True

**List:**

1. **list( ):**

n="01234567"  
n1=list(n)  
print("the string is:",n)  
print("the string to list:",n1)

**output:**

the string is: 01234567

the string to list: ['0', '1', '2', '3', '4', '5', '6', '7']

1. **append( ):**

adding element to the list.

l=[]  
n=int(input("enter the number:"))  
for i in range(0,n):  
 l.append(input("enter:"))  
print("printing the list items")  
for i in l:  
 print(i,end=" ")

**output:**

enter the number:6

enter:8

enter:4

enter:0

enter:2

enter:7

enter:1

printing the list items

8 4 0 2 7 1

1. **remove( ):**

It is used to remove the element.

l=[0,1,2,3,4]  
print("printing original list:")  
for i in l:  
 print(i,end=" ")  
l.remove(2)  
print()  
print("After removal of element")  
for i in l:  
 print(i,end=" ")

**output:**

printing original list:

0 1 2 3 4

After removal of element

0 1 3 4

1. **len( ):**

It is used to calculate the length of the list.

l=[0,1,2,3,4]  
print("the list is:",l)  
print("the length of list:")  
print(len(l))

output;

the list is: [0, 1, 2, 3, 4]

the length of list:

5

1. **max( ):**

It returns the maximum element of the list.

l=[0,1,2,3,4]  
print("the list is:",l)  
print("the maximum of list:")  
print(max(l))

output:

the list is: [0, 1, 2, 3, 4]

the maximum of list:

4

1. **min( ):**

It returns the minimum element of the list.

l=[0,1,2,3,4]  
print("the list is:",l)  
print("the minimum of list:")  
print(min(l))

**output:**

the list is: [0, 1, 2, 3, 4]

the minimum of list:

0

1. **sum( ):**

It returns the sum element of the list.

l=[0,1,2,3,4]  
print("the list is:",l)  
print("the sum of list:")  
print(sum(l))

**output:**

the list is: [0, 1, 2, 3, 4]

the sum of list:

10

1. **sorted( ):**

l=[4,8,1,7,3,0,2,9]  
print("the list is:",l)  
print("the sorted of list:")  
print(sorted(l))

**output:**

the list is: [4, 8, 1, 7, 3, 0, 2, 9]

the sorted of list:

[0, 1, 2, 3, 4, 7, 8, 9]

1. **insert( ):**

**syntax:**

insert(index,obj)

l=[12,'xyz','abc',67]  
l.insert(3,56)  
print("the list is:",l)  
print("after insertion the list:",l)

output:

the list is: [12, 'xyz', 'abc', 56, 67]

after insertion the list: [12, 'xyz', 'abc', 56, 67]

1. **pop( ):**

It will delete the last element.

**Syntax:**

pop(obj=list[-1])

l=[12,'xyz','abc',67]  
print("the list is:",l)  
l.pop()  
print("after pop the list:",l)

**output:**

the list is: [12, 'xyz', 'abc', 67]

after pop the list: [12, 'xyz', 'abc']

1. **map( ):**

It is used to return a lists of results after applying a given function to each item of an iterable.

def addition(n):  
 return n+n  
num=[1,2,3,4]  
result=map(addition,num)  
print("the list is:",num)  
print("after adding list:",list(result))

**output:**

the list is: [1, 2, 3, 4]

after adding list: [2, 4, 6, 8]

1. **enumerate( ):**

It returns an enumerated object. It takes two parameters , first is a sequence of elements and the second is the start index of the sequence. We can get the elements in sequence either through a loop or next() method.

l=[1,2,3,4,5,6]  
print("the list is: ",l)  
r=enumerate(l)  
print("after enumerate the list:")  
print(list(r))

**output:**

the list is: [1, 2, 3, 4, 5, 6]

after enumerate the list:

[(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6)]

**Dictionary:**

1. **dict( ):**

d=dict(name='hari',course='biology',email='hari@gmail.com')  
print("the dict item are:",d)

**output:**

the dict item are: {'name': 'hari', 'course': 'biology', 'email': 'hari@gmail.com'}

1. **items( ):**

stud={'name':'hari','course':'maths','email':'hari@gmail.com'}  
x=stud.items()  
print("the dict item are :",x)

**output:**

the dict item are : dict\_items([('name', 'hari'), ('course', 'maths'), ('email', 'hari@gmail.com')])

stud={'name':'hari','course':'biology','email':'hari@gmail.com'}  
for x,y in stud.items():  
 print(x,y)

**output:**

name hari

course biology

email [hari@gmail.com](mailto:hari@gmail.com)

1. **values( ):**

stud={'name':'hari','course':'maths','email':'hari@gmail.com'}  
x=stud.values()  
print("the dict values are :",x)

**output:**

the dict values are : dict\_values(['hari', 'maths', 'hari@gmail.com'])

stud={'name':'hari','course':'maths','email':'hari@gmail.com'}  
for i in stud:  
 print(stud[i])

**output:**

hari

maths

hari@gmail.com

1. **keys( ):**

stud={'name':'hari','course':'maths','email':'hari@gmail.com'}  
x=stud.keys()  
print("the dict keys are :",x)

**output:**

the dict keys are : dict\_keys(['name', 'course', 'email'])

stud={'name':'hari','course':'maths','email':'hari@gmail.com'}  
for i in stud:  
 print(i)

**output:**

name

course

email

1. **update( ):**

stud={'name':'hari','course':'maths','email':'hari@gmail.com'}  
print("the dict item are :",stud)  
stud.update({'rollno':'28'})  
print("the dict after insertion:",stud)

**output:**

the dict item are : {'name': 'hari', 'course': 'maths', 'email': 'hari@gmail.com'}

the dict after insertion: {'name': 'hari', 'course': 'maths', 'email': 'hari@gmail.com', 'rollno': '28'}

1. **popitem( ):**

stud={'name':'hari','course':'maths','email':'hari@gmail.com'}  
print("the dict item are :",stud)  
stud.popitem()  
print("the dict after pop item:",stud)

**output:**

the dict item are : {'name': 'hari', 'course': 'maths', 'email': 'hari@gmail.com'}

the dict after pop item: {'name': 'hari', 'course': 'maths'}

**set:**

1. **set( ):**

n=[1,2,3,4,5,6,7,8]  
s=set(n)  
print("the set is:",s)  
print(type(s))

**output:**

the set is: {1, 2, 3, 4, 5, 6, 7, 8}

<class 'set'>

1. **add( ):**

month={"jan","feb","mar","apr","may"}  
print("the set is:",month)  
month.add('june')  
month.add('july')  
print("after insertion the set is:",month)

**output:**

the set is: {'feb', 'mar', 'apr', 'may', 'jan'}

after insertion the set is: {'feb', 'mar', 'july', 'june', 'apr', 'may', 'jan'}

1. **update( ):**

month={"jan","feb","mar","apr","may"}  
print("the set is:",month)  
month.update("june")  
month.update("july")  
print("after insertion the set is:",month)

**output:**

the set is: {'apr', 'may', 'feb', 'mar', 'jan'}

after insertion the set is: {'feb', 'jan', 'u', 'mar', 'apr', 'may', 'n', 'y', 'e', 'j', 'l'}

1. **discard( ):**

month={"jan","feb","mar","apr","may"}  
print("the set is:",month)  
month.discard('mar')  
month.discard('jan')  
print("after deletion the set is:",month)

**output:**

the set is: {'may', 'jan', 'feb', 'mar', 'apr'}

after deletion the set is: {'may', 'feb', 'apr'}

1. **remove( ):**

month={"jan","feb","mar","apr","may"}  
print("the set is:",month)  
month.remove('mar')  
month.remove('jan')  
print("after deletion the set is:",month)

**output:**

the set is: {'may', 'mar', 'feb', 'apr', 'jan'}

after deletion the set is: {'may', 'feb', 'apr'}

month={"jan","feb","mar","april","may"}  
print("the set is:",month)  
month.discard('apr')  
month.remove('apr')  
print("after deletion the set is:",month)

**output:**

Traceback (most recent call last):

File "C:\Users\HARI\PycharmProjects\pythonProject\demo2.py", line 4, in <module>

month.remove('apr')

KeyError: 'apr'

the set is: {'mar', 'may', 'feb', 'april', 'jan'}

1. **union( ):**

month1={"jan","feb","mar","april","may"}  
month2={"dec","nov","sep"}  
u=set(month1|month2)  
print("the first set:",month1)  
print("the second set:",month2)  
print("union of to set is :",u)

**output:**

the first set: {'feb', 'jan', 'may', 'mar', 'april'}

the second set: {'sep', 'dec', 'nov'}

union of to set is : {'feb', 'dec', 'mar', 'april', 'sep', 'jan', 'may', 'nov'}

month1={"jan","feb","mar","april","may"}  
month2={"dec","nov","sep"}  
u=set(month1.union(month2))  
print("the first set:",month1)  
print("the second set:",month2)  
print("union of to set is :",u)

**output:**

the first set: {'april', 'mar', 'feb', 'jan', 'may'}

the second set: {'sep', 'nov', 'dec'}

union of to set is : {'mar', 'feb', 'dec', 'april', 'sep', 'jan', 'nov', 'may'}

1. **intersection( ):**

month1={"jan","feb","mar","april","may"}  
month2={"dec","feb","may"}  
i=set(month1&month2)  
print("the first set:",month1)  
print("the second set:",month2)  
print("intersection of to set is :",i)

**output:**

the first set: {'feb', 'jan', 'mar', 'may', 'april'}

the second set: {'may', 'feb', 'dec'}

intersection of to set is : {'may', 'feb'}

month1={"jan","feb","mar","april","may"}  
month2={"dec","feb","may"}  
i=set(month1.intersection(month2))  
print("the first set:",month1)  
print("the second set:",month2)  
print("intersection of to set is :",i)

**output:**

the first set: {'jan', 'may', 'april', 'feb', 'mar'}

the second set: {'may', 'dec', 'feb'}

intersection of to set is : {'may', 'feb'}

1. **intersection\_update( ):**

month1={"jan","feb","mar","april","may"}  
month2={"dec","feb","may"}  
month3={"may","feb"}  
print("the first set is:",month1)  
print("the second set is:",month2)  
print("the third set is:",month3)  
month1.intersection\_update(month2,month3)  
print("the intersection is:",month1)

**output:**

the first set is: {'mar', 'feb', 'may', 'april', 'jan'}

the second set is: {'may', 'feb', 'dec'}

the third set is: {'may', 'feb'}

the intersection is: {'may', 'feb'}

1. **difference between two sets:**

month1={"jan","feb","mar","april","may"}  
month2={"dec","feb","may"}  
print("the first set is:",month1)  
print("the second set is:",month2)  
diff=month1-month2  
print("the difference between two sets is:",diff)

**output:**

the first set is: {'jan', 'feb', 'may', 'mar', 'april'}

the second set is: {'feb', 'dec', 'may'}

the difference between two sets is: {'jan', 'mar', 'april'}

month1={"jan","feb","mar","april","may"}  
month2={"dec","feb","may"}  
print("the first set is:",month1)  
print("the second set is:",month2)  
diff=month1.difference(month2)  
diff1=month2.difference(month1)  
print("the difference between two sets is:",diff)  
print("the difference between two sets is:",diff1)

**output:**

the first set is: {'jan', 'feb', 'may', 'april', 'mar'}

the second set is: {'feb', 'may', 'dec'}

the difference between two sets is: {'april', 'jan', 'mar'}

the difference between two sets is: {'dec'}

1. **symmetric difference of two sets:**

x={1,2,3,4,5,6}  
y={1,2,9,8,10}  
print("the first set are:",x)  
print("the second set are:",y)  
z=x^y  
print("the symmetric difference between two sets are:",z)

**output:**

the first set are: {1, 2, 3, 4, 5, 6}

the second set are: {1, 2, 8, 9, 10}

the symmetric difference between two sets are: {3, 4, 5, 6, 8, 9, 10}

x={1,2,3,4,5,6}  
y={1,2,9,8,10}  
print("the first set are:",x)  
print("the second set are:",y)  
z=x.symmetric\_difference(y)  
print("the symmetric difference between two sets are:",z)

**output:**

the first set are: {1, 2, 3, 4, 5, 6}

the second set are: {1, 2, 8, 9, 10}

the symmetric difference between two sets are: {3, 4, 5, 6, 8, 9, 10}