

Python Basics

Learning Outcome: To be able to effectively understand and apply the basics of Python such as variable declaration, print statement, taking user input, etc.

Program 1

Objective: To declare, initialise, and print a variable.

Code:

```
#Variable declaration and initialisation  
c=10
```

```
#Printing a value  
print(c)
```

Output:

```
1 #Variable declaration and initialisation  
2 c=10  
3  
4 #Printing a value  
5 print(c)
```

10

Program 2

Objective: To print the ASCII value of an integer.

Code:

```
#Printing ASCII value  
a=65  
print(chr(a))
```

Output:

```
1 #Printing ASCII value  
2 a=65  
3 print(chr(a))
```

A

Program 3

Objective: To print a string variable.

Code:

```
#String variable  
name="Ali"  
print(name)
```

Output:

```
1 #String variable  
2 name="Ali"  
3 print(name)
```

Ali

Program 4

Objective: To take user input and print it.

Code:

```
#Taking user input  
username = input("Enter your name ")  
print("Welcome",name)
```

Output:

```
1 #Taking user input  
2 username = input("Enter your name ")  
3 print("Welcome",name)
```

Enter your name Ali
Welcome Ali

Program 5

Objective: To print the datatype of a variable.

Code:

```
#Printing Datatype
a=10
b='Ali'
print("a is of type",type(a),"nb is of type",type(b))
```

Output:

```
1 #Printing Datatype
2 a=10
3 b='Ali'
4 print("a is of type",type(a),"nb is of type",type(b))
```

```
a is of type <class 'int'>
b is of type <class 'str'>
```

Program 6

Objective: To print a boolean datatype.

Code:

```
#Boolean Datatype
c=10
print(bool(c))
```

Output:

```
1 #Boolean Datatype
2 c=10
3 print(bool(c))
```

```
True
```

Program 7

Objective: To perform the division and modulo operations.

Code:

```
#Division and Modulo Operation
e=34
d=12
f=e/d
g=e%d
print(round(f,2),g)
```

Output:

```
1 #Division and Modulo Operation
2 e=34
3 d=12
4 f=e/d
5 g=e%d
6 print(round(f,2),g)
```

2.83 10

Program 8

Objective: To implement the len() function and print using string index.

Code:

```
#len() returns the length of a string
string="ABCDEFGH"
print(len(string),string[0],string[len(string)-1])
```

Output:

```
1 #len() returns the length of a string
2 string="ABCDEFGH"
3 print(len(string),string[0],string[len(string)-1])
```

8 A H

Program 9

Objective: To develop an application that can perform a multitude of functions such as - basic arithmetic calculations, ASCII calculations, Print statement, Datatypes, Strings, etc.

Code:

```
#Advanced Quick Function Calculator
print("Welcome to Quick Functions.")
while True :
    var=int(input("\nChoose your option from the list given below:\n1. Basic Calculations\n2. ASCII Calculations\n3. String\n4. Basic Print Statement\n5. Datatypes\n6. Exit\n"))
    if(var==1):
        calc1=int(input("\nWelcome to Basic Calculator. Choose your option from the list given below\n1. Add\n2. Subtract\n3. Multiply\n4. Divide\n5. Modulo\n"))
        a=int(input("Enter 1st number "))
        b=int(input("Enter 2nd number "))
        if(calc1==1):
            print("Sum is ",a+b)
        elif(calc1==2):
            print("Difference is ",a-b)
        elif(calc1==3):
            print("Product is ",a*b)
        elif(calc1==4):
            print("Quotient is ",a/b)
        elif(calc1==5):
            print("Remainder is ",a%b)
        else:
            print("Incorrect option.")
    elif(var==2):
        calc2=int(input("Welcome to ASCII Calculator. Choose your option from the list given below\n1. Character to ASCII\n2. ASCII to Character\n"))
        if(calc2==1):
            c=input("Input Character ")
            print(ord(c))
        elif(calc2==2):
            d=int(input("Input Integer "))
            print(chr(d))
        else:
            print("Incorrect option.")
    elif(var==3):
        string=input("Enter a string ")
        print("The length of the string is",len(string),"The first character is",string[0],"The last character is",string[len(string)-1])
    elif(var==4):
        name=input("Input your name ")
        print("Your name is ",name)
    elif(var==5):
        data1="String"
        data2=10
        data3=14.5
        print("The type of String is",type(data1),"The type of Integer is",type(data2),"The type of Float is",type(data3))
    elif(var==6):
        print("Bye")
        break
    else:
        print("Incorrect option.")
```

Output (Code):

```
#Advanced Quick Function Calculator
print("Welcome to Quick Functions.")
while True :
    var=int(input("\nChoose your option from the list given below:\n1. Basic Calculations\n2. ASCII Calculations\n3. String\n4. Basic Print Statement\n5. Datatypes\n6. Exit\n"))
    if(var==1):
        calc1=int(input("\nWelcome to Basic Calculator. Choose your option from the list given below\n1. Add\n2. Subtract\n3. Multiply\n4. Divide\n5. Modulo\n"))
        a=int(input("Enter 1st number "))
        b=int(input("Enter 2nd number "))
        if(calc1==1):
            print("Sum is ",a+b)
        elif(calc1==2):
            print("Difference is ",a-b)
        elif(calc1==3):
            print("Product is ",a*b)
        elif(calc1==4):
            print("Quotient is ",a/b)
        elif(calc1==5):
            print("Remainder is ",a%b)
        else:
            print("Incorrect option.")
    elif(var==2):
        calc2=int(input("Welcome to ASCII Calculator. Choose your option from the list given below\n1. Character to ASCII\n2. ASCII to Character\n"))
        if(calc2==1):
            c=input("Input Character ")
            print(ord(c))
        elif(calc2==2):
            d=int(input("Input Integer "))
            print(chr(d))
        else:
            print("Incorrect option.")
    elif(var==3):
        string=input("Enter a string ")
        print("The length of the string is",len(string),"The first character is",string[0],"The last character is",string[len(string)-1])
    elif(var==4):
        name=input("Input your name ")
        print("Your name is ",name)
    elif(var==5):
        data1="String"
        data2=10
        data3=14.5
        print("The type of String is",type(data1),"The type of Integer is",type(data2),"The type of Float is",type(data3))
    elif(var==6):
        print("Bye")
        break
    else:
        print("Incorrect option.")
```

Output (Final):

Welcome to Quick Functions.

Choose your option from the list given below:

1. Basic Calculations
2. ASCII Calculations
3. String
4. Basic Print Statement
5. Datatypes
6. Exit

5

The type of String is <class 'str'>

The type of Integer is <class 'int'>

The type of Float is <class 'float'>

Choose your option from the list given below:

1. Basic Calculations
2. ASCII Calculations
3. String
4. Basic Print Statement
5. Datatypes
6. Exit

2

Welcome to ASCII Calculator. Choose your option from the list given below

1. Character to ASCII
2. ASCII to Character

1

Input Character A

65

Choose your option from the list given below:

1. Basic Calculations
2. ASCII Calculations
3. String
4. Basic Print Statement
5. Datatypes
6. Exit

6

Bye

Conclusion: We have successfully implemented and understood the basics of Python such as variable declaration, initialisation, printing, datatypes, strings, and a few in-built functions such as len().

Strings in Python

Learning Outcome: To be able to effectively apply and perform any kinds of functions given a string or any type of string input.

Program 1

Objective: To print “hello world”, string concatenation, and access strings using index.

Program:

```
var1 = 'Hello World!'
var2 = "Python Programming"

#printing String
print(var1, " ", var2)

#Accessing string using index
print (var1[0])
print (var2[1:5])

#String Concatenation
print ("Updated String :- ", var1[0:6] + 'Ali')
```

Output:

```
1 #To print "hello world", string concatenation, and access strings using index
2 var1 = 'Hello World!'
3 var2 = "Python Programming"
4
5 #Printing String
6 print(var1, " ", var2)
7
8 #Accessing string using index
9 print (var1[0])
10 print (var2[1:5])
11
12 #String Concatenation
13 print ("Updated String :- ", var1[0:6] + 'Ali')

Hello World!  Python Programming
H
ytho
Updated String :-  Hello Ali
```

Program 2

Objective - To take user input of string and print it.

Program:

```
str1 = input("Please Enter Your Own String : ")
#printing user input
print(str1)
```

Output:

```
[5] 1 #To take user input of string and print it
2 str1 = input("Please Enter Your Own String : ")
3
4 #Printing user input
5 print(str1)
```

```
Please Enter Your Own String : Ali
Ali
```

Program 3

Objective: To perform a few string operations using functions such as capitalize(), center(), count(), find().

Program:

```
str2 = "this is string example....wow!!!"
```

```
#capitalize() method returns a copy of the string with only its first character capitalized.
print ("str2.capitalize() : ", str2.capitalize())
```

```
#center() returns centered in a string of length width. Padding is done using the specified fillchar.
#Default filler is a space.
```

```
print ("str2.center(40, 'a') : ", str2.center(40, '*'))
```

```
#count() returns the number of occurrences of substring sub in the range [start, end].
```

```
sub = "i";
```

```
print ("str2.count(sub, 4, 40) : ", str2.count(sub, 4, 40))
```

```
#find() determines if string str occurs in string, or in a substring of string if starting
```

```
#index beg and ending index end are given.
```

```
str3 = "is";
```

```
print (str2.find(str3))
```

Output:

```
1 #String operations using functions such as capitalize(), center(), count(), find()
2 str2 = "this is string example....wow!!!"
3
4 #capitalize() method returns a copy of the string with only its first character capitalized.
5 print ("str2.capitalize() : ", str2.capitalize())
6
7 #center() returns centered in a string of length width. Padding is done using the specified fillchar.
8 #Default filler is a space.
9 print ("str2.center(40, 'a') : ", str2.center(40, '*'))
10
11 #count() returns the number of occurrences of substring sub in the range [start, end].
12 sub = "i";
13 print ("str2.count(sub, 4, 40) : ", str2.count(sub, 4, 40))
14
15 #find() determines if string str occurs in string, or in a substring of string if starting
16 #index beg and ending index end are given.
17 str3 = "is";
18 print (str2.find(str3))
```

```
str2.capitalize() : This is string example....wow!!!
str2.center(40, 'a') : ****this is string example....wow!!!****
str2.count(sub, 4, 40) : 2
2
```

Program 4

Objective: To perform string operation using more functions such as index(), isalnum(), isdigit(), isalpha(), islower(), isnumeric(), isspace(), istitle(), isupper(), join(), len().

Program:

#index() determines if string str occurs in string or in a substring of string if starting #index beg and ending index end are given.

```
str1 = "this is string example....wow!!!"
```

```
str2 = "exam"
```

```
print (str1.index(str2, 10,32))
```

#isalnum() checks whether the string consists of alphanumeric characters.

```
str = "this2009"; # No space in this string
```

```
print (str.isalnum())
```

#isalpha() checks whether the string consists of only alphabets.

```
str = "ShabnamSharma"; # No space & digit in this string
```

```
print (str.isalpha())
```

#isdigit() checks whether the string consists of only digits.

```
str = "123456"; # Only digit in this string
```

```
print (str.isdigit())
```

#islower() returns true if all characters are lowercase, else returns false

```
str = "this is string example....wow!!!";
```

```
print (str.islower())
```

#isnumeric() returns true if all characters are numbers, else returns false

```
str = "23443434";
```

```
print (str.isnumeric())
```

#isspace() returns true if all characters are whitespaces, else returns false

```
str = " ";
```

```
print (str.isspace())
```

#istitle() returns true if all words start with uppercase character, else returns false

```
str = "This Is String Example...Wow!!!";
```

```
print (str.istitle())
```

#isupper() returns true if all characters are uppercase, else returns false

```
str = "THIS IS STRING EXAMPLE....WOW!!!";
```

```
print (str.isupper())
```

#join() returns a string in which the string elements of sequence are joined by str separator.

```
s = " * * ";
```

```
seq = ("abc", "bttt", "cqweqe"); # This is sequence of strings.
```

```
print (s.join(seq))
```

#len() returns length of the given string

```
str = "this is string example....wow!!!";
```

```
print ("Length of the string: ", len(str))
```

Output:

```
1 #More functions such as index(), isalnum(), isdigit(), isalpha(), islower(), isnumeric()
2 #isspace(), istitle(), isupper(), join()
3
4 #index() determines if string str occurs in string or in a substring of string if starting
5 #index beg and ending index end are given.
6 str1 = "this is string example....wow!!!"
7 str2 = "exam"
8 print (str1.index(str2, 10,32))
9
10 #isalnum() checks whether the string consists of alphanumeric characters.
11 str = "this2009"; # No space in this string
12 print (str.isalnum())
13
14 #isalpha() checks whether the string consists of only alphabets.
15 str = "ShabnamSharma"; # No space & digit in this string
16 print (str.isalpha())
17
18 #isdigit() checks whether the string consists of only digits.
19 str = "123456"; # Only digit in this string
20 print (str.isdigit())
21
22 #islower() returns true if all characters are lowercase, else returns false
23 str = "this is string example....wow!!!";
24 print (str.islower())
25
26 #isnumeric() returns true if all characters are numbers, else returns false
27 str = "23443434";
28 print (str.isnumeric())
29
30 #isspace() returns true if all characters are whitespaces, else returns false
31 str = " ";
32 print (str.isspace())
33
34 #istitle() returns true if all words start with uppercase character, else returns false
35 str = "This Is String Example...Wow!!!";
36 print (str.istitle())
37
38 #isupper() returns true if all characters are uppercase, else returns false
39 str = "THIS IS STRING EXAMPLE....WOW!!!";
40 print (str.isupper())
41
42 #join() returns a string in which the string elements of sequence are joined by str separator.
43 s = " * * ";
44 seq = ("abc", "bttt", "cqweqe"); # This is sequence of strings.
45 print (s.join(seq))
46
47 #len() returns length of the given string
48 str = "this is string example....wow!!!";
49 print ("Length of the string: ", len(str))
```

```
15
True
True
True
True
True
True
True
True
abc * * bttt * * cqweqe
Length of the string: 32
```

Program 5

Objective: To perform string operation using even more functions such as lstrip(), lower(), upper(), max(), min(), replace(), swapcase(), startswith().

Program:

#lstrip() returns a copy of the string in which all chars have been stripped from
#the beginning of the string (default whitespace characters)

str = " this is string example....wow!!! "

print (str.lstrip())

str = "88888888this is string example....wow!!!999999";

print (str.lstrip('8'))

print (str.rstrip('9'))

#lower() returns copy of string with all lowercase characters

str = "THIS IS STRING EXAMPLE....WOW!!!"

print (str.lower())

#upper() returns copy of string with all uppercase characters

str = "this is string example....wow!!!"

print (str.upper())

#max() returns largest character

str = "check....wow!!!"

print ("Max character: " + max(str))

#min() returns smallest character

str = "example"

print ("Min character: " + min(str))

#replace() finds and replaces a string with the given input

#Syntax is str.replace(old, new[, max])

str = "this is string example....wow!!! this is really string"

print (str.replace("is", "was", 3))

#swapcase() swaps string from uppercase to lowercase and vice versa

str = "this is string example....wow!!!";

print (str.swapcase())

#startswith() returns true if string starts with given input

str = "this is string example....wow!!!";

print (str.startswith('is', 2, 4))

Output:

```
1 #Even more functions such as lstrip(), lower(), upper(), max(), min(), replace(),
2 #swapcase(), startswith()
3
4 #lstrip() returns a copy of the string in which all chars have been stripped from
5 #the beginning of the string (default whitespace characters)
6 str = "    this is string example....wow!!!"
7 print (str.lstrip())
8 str = "88888888this is string example....wow!!!999999";
9 print (str.lstrip('8'))
10 print (str.rstrip('9'))
11
12 #lower() returns copy of string with all lowercase characters
13 str = "THIS IS STRING EXAMPLE....WOW!!!"
14 print (str.lower())
15
16 #upper() returns copy of string with all uppercase characters
17 str = "this is string example....wow!!!"
18 print (str.upper())
19
20 #max() returns largest character
21 str = "check....wow!!!"
22 print ("Max character: " + max(str))
23
24 #min() returns smallest character
25 str = "example"
26 print ("Min character: " + min(str))
27
28 #replace() finds and replaces a string with the given input
29 #Syntax is str.replace(old, new[, max])
30 str = "this is string example....wow!!! this is really string"
31 print (str.replace("is", "was", 3))
32
33 #swapcase() swaps string from uppercase to lowercase and vice versa
34 str = "this is string example....wow!!!";
35 print (str.swapcase())
36
37 #startswith() returns true if string starts with given input
38 str = "this is string example....wow!!!";
39 print (str.startswith( 'is', 2, 4 ))
```

```
this is string example....wow!!!
this is string example....wow!!!999999
88888888this is string example....wow!!!
this is string example....wow!!!
THIS IS STRING EXAMPLE....WOW!!!
Max character: w
Min character: a
thwas was string example....wow!!! thwas is really string
THIS IS STRING EXAMPLE....WOW!!!
True
```

Conclusion: We have successfully implemented multiple string functions and operations.

If-Else in Python

Learning Outcome: To be able to effectively apply the if-else statement construct in Python.

Program 1

Objective: To check for even or odd number using if-else statement.

Program:

```
num = int(input("Enter a number: "))
if num%2 == 0:
    print("Number is even")
else:
    print("Number is odd")
```

Output:

```
1 #Check whether number is odd or even
2 num = int(input("Enter a number: "))
3 if num%2 == 0:
4     print("Number is even")
5 else:
6     print("Number is odd")
```

Enter a number: 22

Number is even

Program 2

Objective: To find largest of 3 numbers using if-else statement.

Program:

```
#Program to find largest of 3 numbers
a = int(input("Enter a: "))
b = int(input("Enter b: "))
c = int(input("Enter c: "))
if a>b and a>c:
    print("a is largest")
elif b>c:
    print("b is largest")
else:
    print("c is largest")
```

Output:

```
1 #Program to find largest of 3 numbers
2 a = int(input("Enter a: "))
3 b = int(input("Enter b: "))
4 c = int(input("Enter c: "))
5 if a>b and a>c:
6     print("a is largest")
7 elif b>c:
8     print("b is largest")
9 else:
10    print("c is largest")
```

Enter a: 20
Enter b: 45
Enter c: 2
b is largest

Program 3

Objective: To check eligibility for voting.

Program:

```
#Program to check eligibility for voting
age = int (input("Enter your age? "))
if age>=18:
    print("You are eligible to vote !!")
else:
    print("Sorry! you have to wait !!")
```

Output:

```
1 #Program to check eligibility for voting
2 age = int (input("Enter your age? "))
3 if age>=18:
4     print("You are eligible to vote !!")
5 else:
6     print("Sorry! you have to wait !!")
```

Enter your age? 32
You are eligible to vote !!

Program 4

Objective: To find grade of the student given the marks.

Program:

```
#Program to find grade of the student
marks = int(input("Enter the marks? "))
if marks > 85 and marks <= 100:
    print("Congrats ! you scored grade A ...")
elif marks > 60 and marks <= 85:
    print("You scored grade B + ...")
elif marks > 40 and marks <= 60:
    print("You scored grade B ...")
elif (marks > 30 and marks <= 40):
    print("You scored grade C ...")
else:
    print("Sorry you have failed")
```

Output:

```
1 #Program to find grade of the student
2 marks = int(input("Enter the marks? "))
3 if marks > 85 and marks <= 100:
4     print("Congrats ! you scored grade A ...")
5 elif marks > 60 and marks <= 85:
6     print("You scored grade B + ...")
7 elif marks > 40 and marks <= 60:
8     print("You scored grade B ...")
9 elif (marks > 30 and marks <= 40):
10    print("You scored grade C ...")
11 else:
12    print("Sorry you have failed")
```

```
Enter the marks? 22
Sorry you have failed
```

Conclusion: We have successfully implemented different types of the if-else statement construct in Python.

Lists in Python

Learning Outcome - To be able to effectively apply and perform different kinds of functions and operations on lists datatype.

Program 1

Objective: To print list using its index, update its value, and delete a value.

Program:

```
#Printing lists using their index
list1 = ['physics', 'chemistry', 1997, 2000];
list2 = [1, 2, 3, 4, 5, 6, 7 ];
print ("list1[0]: ", list1[0])
print ("list2[1:3]: ", list2[1:3])
```

```
#Updating a list value
print ("Value available at index 2 : ")
print (list1[2])
list1[2] = 999999;
print ("New value available at index 2 : ")
print (list1[2])
```

```
#Deleting a list value
del(list1[2])
print ("After deleting value at index 2 :")
print (list1)
```

Output:

```
1 #Printing lists using their index
2 list1 = ['physics', 'chemistry', 1997, 2000]
3 list2 = [1, 2, 3, 4, 5, 6, 7 ]
4 print ("list1[0]: ", list1[0])
5 print ("list2[1:3]: ", list2[1:3])
6
7 #Updating a list value
8 print ("Value available at index 2 : ")
9 print (list1[2])
10 list1[2] = 999999;
11 print ("New value available at index 2 : ")
12 print (list1[2])
13
14 #Deleting a list value
15 del(list1[2])
16 print ("After deleting value at index 2 :")
17 print (list1)
```

```
list1[0]: physics
list2[1:3]: [2, 3]
Value available at index 2 :
1997
New value available at index 2 :
999999
After deleting value at index 2 :
['physics', 'chemistry', 2000]
```

Program 2

Objective: To perform list operations using functions such as len(), append(), count(), extend(), index().

Program:

```
#len()
print ("First list length : ", len(list1),"\\nSecond list length : ",len(list2))

#append()
list2.append(2009);
print ("Updated List : ", list2)

#count
aList = [123, 'xyz', 'xyz', 'abc', 123]
print ("Count for xyz : ", aList.count('xyz'))

#extend()
list1.extend(list2)
print ("Extended List : ", list1)

#index()
print ("Index for chemistry : ", list1.index( 'chemistry' ))
```

Output:

```
1 #To perform list operations using functions such as
2 #len(), append(), count(), extend(), index()
3
4 #len()
5 print ("First list length : ", len(list1),"\\nSecond list length : ",len(list2))
6
7 #append()
8 list2.append(2009);
9 print ("Updated List : ", list2)
10
11 #count
12 aList = [123, 'xyz', 'xyz', 'abc', 123]
13 print ("Count for xyz : ", aList.count('xyz'))
14
15 #extend()
16 list1.extend(list2)
17 print ("Extended List : ", list1)
18
19 #index()
20 print ("Index for chemistry : ", list1.index( 'chemistry' ))
```

```
First list length : 3
Second list length : 7
Updated List : [1, 2, 3, 4, 5, 6, 7, 2009]
Count for xyz : 2
Extended List : ['physics', 'chemistry', 2000, 1, 2, 3, 4, 5, 6, 7, 2009]
Index for chemistry : 1
```

Program 3

Objective: To perform list operations using functions such as `insert()`, `pop()`, `remove()`, `reverse()`, `sort()`.

Program:

```
#insert()
aList = [123, 'xyz', 'zara', 'abc']
print("List before insertion : ",aList)
aList.insert( 3, 2009)
print ("Final List : ", aList)

#pop()
aList.pop()
print("List after popping last element : ",aList)
aList.pop(2)
print("List after popping element from mentioned index: ",aList)

#remove()
aList.remove('xyz');
print ("List : ", aList)

#reverse()
aList.reverse();
print ("List : ", aList)

#sort()
aList.sort();
print ("List : ", aList)
```

Output:

```
1 #To perform list operations using functions such as
2 #insert(), pop(), remove(), reverse(), sort()
3
4 #insert()
5 aList = [123, 'xyz', 'zara', 'abc']
6 print("List before insertion : ",aList)
7 aList.insert( 3, 2009)
8 print ("Final List : ", aList)
9
10 #pop()
11 aList.pop()
12 print("List after popping last element : ",aList)
13 aList.pop(2)
14 print("List after popping element from mentioned index: ",aList)
15
16 #remove()
17 aList.remove('xyz')
18 print ("List : ",aList)
19
20 #reverse()
21 aList.reverse()
22 print ("List : ",aList)
23
24 #sort()
25 aList.sort()
26 print ("List : ",aList)
```

```
List before insertion :  [123, 'xyz', 'zara', 'abc']
Final List :  [123, 'xyz', 'zara', 2009, 'abc']
List after popping last element :  [123, 'xyz', 'zara', 2009]
List after popping element from mentioned index:  [123, 'xyz', 2009]
List :  [123, 2009]
List :  [2009, 123]
List :  [123, 2009]
```

Conclusion: We have successfully implemented and performed different kinds of functions and operations on lists datatype.

Loops in Python

Learning Outcome: To be able to effectively apply various looping constructs in Python.

Program 1

Objective: To print every element of a string, list, and use range() in for loop.

Program:

```
st = "Python"
ex=['hello','hi','hey']
for i in st:
    print(i)
print("\n")
for j in ex:
    print(j)
print("\n")
for i in range(11):
    print(i,end=' ')
```

Output:

```
1 #To print every element of a string/list using for loop
2 st = "Python"
3 ex=['hello','hi','hey']
4 for i in st:
5     print(i)
6     print("\n")
7     for j in ex:
8         print(j)
9         print("\n")
10    for i in range(11):
11        print(i,end=' ')
```

P
y
t
h
o
n

hello
hi
hey

0 1 2 3 4 5 6 7 8 9 10

Program 2

Objective: To print sum, multiplication table using loops.

Program:

```
list = [10,30,23,43,65,12]

#Sum
sum = 0
for i in list:
    sum = sum+i
print("The sum is:",sum)
print('\n')

#Multiplication Table
n = int(input("Enter the number "))
for i in range(1,6):
    c = n*i
    print(n,"*",i,"=",c)
```

Output:

```
1 #To print sum, multiplication table using loops
2 list = [10,30,23,43,65,12]
3
4 #Sum
5 sum = 0
6 for i in list:
7     sum = sum+i
8 print("The sum is:",sum)
9 print('\n')
10
11 #Multiplication Table
12 n = int(input("Enter the number "))
13 for i in range(1,6):
14     c = n*i
15     print(n,"*",i,"=",c)
```

The sum is: 183

Enter the number 3

```
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
```

Program 3

Objective: To implement break, continue, pass, for else, while loop.

Program:

```
#break - breaks the loop when a certain condition is met
```

```
i = 0
```

```
str1 = 'abba'
```

```
while i < len(str1):
```

```
    if str1[i] == 's':
```

```
        break
```

```
    print('Current Letter :', str1[i])
```

```
    i += 1
```

```
print("\n")
```

```
#continue - continues iterating the current loop
```

```
for j in range(1,7):
```

```
    print(j)
```

```
    continue;
```

```
    print("hello")
```

```
else:
```

```
    print("for loop is exhausted\n")
```

```
#pass - It is used to declare the empty loop.
```

```
#It is also used to define empty class or function.
```

```
str2 = 'abba'
```

```
k = 0
```

```
while k < len(str1):
```

```
    k += 1
```

```
    pass
```

```
print('Value of k :', k)
```

Output:

```
1 #break, continue, pass, for else, while loop
2
3 #break – breaks the loop when a certain condition is met
4 i = 0
5 str1 = 'abba'
6
7 while i < len(str1):
8     if str1[i] == 's':
9         break
10    print('Current Letter :', str1[i])
11    i += 1
12 print("\n")
13
14 #continue – continues iterating the current loop
15 for j in range(1,7):
16     print(j)
17     continue;
18     print("hello")
19 else:
20     print("for loop is exhausted\n")
21
22 #pass – It is used to declare the empty loop.
23 #It is also used to define empty class or function.
24 str2 = 'abba'
25 k = 0
26 while k < len(str1):
27     k += 1
28     pass
29 print('Value of k :', k)
```

Current Letter : a
Current Letter : b
Current Letter : b
Current Letter : a

1
2
3
4
5
6
for loop is exhausted

Program 4

Objective: To implement infinite loop, breaking while loop with if condition.

Program:

```
#infinite loop
while (1):
    print("Hi! we are inside the infinite while loop")
```

```
#breaking while loop with if condition
var = 1
print("loop will only stop when input is 2\n")
while(var != 2):
    var = int(input("Enter the number: "))
    print("Entered value is %d"%(var))
else:
    print("Bye")
```

Output:

```
1 #infinite loop, breaking while loop with if condition
2
3 #infinite loop
4 while (1):
5     print("Hi! we are inside the infinite while loop")
6
7 #breaking while loop with if condition
8 var = 1
9 print("loop will only stop when input is 2\n")
10 while(var != 2):
11     var = int(input("Enter the number: "))
12     print("Entered value is %d"%(var))
13 else:
14     print("Bye")
```

loop will only stop when input is 2

```
Enter the number: 3
Entered value is 3
Enter the number: 5
Entered value is 5
Enter the number: 2
Entered value is 2
Bye
```

Conclusion: We have successfully implemented various types of looping constructs in Python.

Python Applications

Learning Outcome: To be able to create different application using Python.

Program 1

Objective: To implement an inventory and stock checking system which allows the user to purchase or view the stock, and automatically replenishes the stock once it is depleted.

Program:

```
def show():
    s = [[str(e) for e in row] for row in h]
    lens = [max(map(len, col)) for col in zip(*s)]
    fmt = '\t'.join('{{:{}}}'.format(x) for x in lens)
    table = [fmt.format(*row) for row in s]
    print('\n'.join(table))

def checkInven():
    for i in range(1,len(h)):
        if(h[i][2]<=5):
            print("\nStock replinshed")
            h[i][2]=20

h=[['Name', 'Price', 'Quantity', 'Manufacturer'],['Mouse',500,20,'HP'],
['Keyboard',1000,15,'Logitech'],
,['Monitor',2500,10,'Asus'],['MS Office 2021',1000,25,'Microsoft']]
n='a'
cost=0
while(n!=='e'):
    checkInven()
    print("\nInventory is\n")
    show()
    n=input("\np : purchase\nc : count\ne : exit\n")
    if(n=='p'):
        a=int(input("Enter the index (1=Mouse, 2=Keyboard..) "))
        if(a>4):
            print("Incorrect option. Try again.")
            continue
        count=int(input("How many units you want to buy? "))
        if(h[a][2]<count):
            print("Sorry.. not enough stock")
        else:
            h[a][2]-=count
            print("Purchase successful")
            cost+=h[a][1]*count
    elif(n=='c'):
        b=int(input("Enter the index (1=Mouse, 2=Keyboard..) "))
        if(b>4):
            print("Incorrect option. Try again.")
            continue
        print("Stock left is ",h[b][2])
    elif(n=='e'):
        pass
    else:
        print("Incorrect option. Try again.")
print("\nYour total bill amount is ",cost,"\\nThank you for doing business with us.")
```

Output (Code):

```
1 def show():
2     s = [[str(e) for e in row] for row in h]
3     lens = [max(map(len, col)) for col in zip(*s)]
4     fmt = '\t'.join('{{:{}}}'.format(x) for x in lens)
5     table = [fmt.format(*row) for row in s]
6     print('\n'.join(table))
7
8 def checkInven():
9     for i in range(1,len(h)):
10        if(h[i][2]<=5):
11            print("\nStock replinished")
12            h[i][2]=20
13
14 h=[['Name','Price','Quantity', 'Manufacturer'], ['Mouse',500,20,'HP'], ['Keyboard',1000,15,'Logitech'],
15   ,['Monitor',2500,10,'Asus'], ['MS Office 2021',1000,25,'Microsoft']]
16 n='a'
17 cost=0
18 while(n!='e'):
19     checkInven()
20     print("\nInventory is\n")
21     show()
22     n=input("\np : purchase\n c : count\n e : exit\n")
23     if(n=='p'):
24         a=int(input("Enter the index (1=Mouse, 2=Keyboard..) "))
25         if(a>4):
26             print("Incorrect option. Try again.")
27             continue
28         count=int(input("How many units you want to buy? "))
29         if(h[a][2]<count):
30             print("Sorry.. not enough stock")
31         else:
32             h[a][2]-=count
33             print("Purchase successful")
34             cost+=h[a][1]*count
35     elif(n=='c'):
36         b=int(input("Enter the index (1=Mouse, 2=Keyboard..) "))
37         if(b>4):
38             print("Incorrect option. Try again.")
39             continue
40         print("Stock left is ",h[b][2])
41     elif(n=='e'):
42         pass
43     else:
44         print("Incorrect option. Try again.")
45 print("\nYour total bill amount is ",cost,"Thank you for doing business with us.")
```

Output (Final):

Inventory is

Name	Price	Quantity	Manufacturer
Mouse	500	20	HP
Keyboard	1000	15	Logitech
Monitor	2500	10	Asus
MS Office 2021	1000	25	Microsoft

p : purchase

c : count

e : exit

p

Enter the index (1=Mouse, 2=Keyboard..) 1

How many units you want to buy? 10

Purchase successful

Inventory is

Name	Price	Quantity	Manufacturer
Mouse	500	10	HP
Keyboard	1000	15	Logitech
Monitor	2500	10	Asus
MS Office 2021	1000	25	Microsoft

p : purchase

c : count

e : exit

p

Enter the index (1=Mouse, 2=Keyboard..) 2

How many units you want to buy? 10

Purchase successful

Stock replenished

Inventory is

Name	Price	Quantity	Manufacturer
Mouse	500	10	HP
Keyboard	1000	20	Logitech
Monitor	2500	10	Asus
MS Office 2021	1000	25	Microsoft

p : purchase

c : count

e : exit

e

Your total bill amount is 15000

Thank you for doing business with us.

Program 2

Objective: To implement a food delivery service which asks the customer what they wish to order and from which delivery service, and takes their address for delivery as well.

Code:

```
deli=['Zomato','Swiggy','Amazon Food']
foo=['Burger','Roll','Pizza']
pri=[120,100,200]
repeat='y'
while(repeat!='n'):
    user=int(input("1. Zomato\n2. Swiggy\n3. Amazon Food\nEnter your preferred delivery partner "))
    choice=int(input("\n1. Burger : 120/-\n2. Roll : 100/-\n3. Pizza : 200/-\nEnter your choice "))
    loc=input("\nEnter your address ")
    print("\nYour order for",foo[choice-1],"has been placed on",deli[user-1],"and will arrive at your address",loc)
    print("Total payment to be made is ",pri[choice-1])
    repeat=input("\nDo you wish to order again? (y/n)\n")
```

Output:

```
1  deli=['Zomato','Swiggy','Amazon Food']
2  foo=['Burger','Roll','Pizza']
3  pri=[120,100,200]
4  repeat='y'
5  while(repeat!='n'):
6      user=int(input("1. Zomato\n2. Swiggy\n3. Amazon Food\nEnter your preferred delivery partner "))
7      choice=int(input("\n1. Burger : 120/-\n2. Roll : 100/-\n3. Pizza : 200/-\nEnter your choice "))
8      loc=input("\nEnter your address ")
9      print("\nYour order for",foo[choice-1],"has been placed on",deli[user-1],"and will arrive at your address",loc)
10     print("Total payment to be made is ",pri[choice-1])
11     repeat=input("\nDo you wish to order again? (y/n)\n")
```

```
1. Zomato
2. Swiggy
3. Amazon Food
Enter your preferred delivery partner 2
```

```
1. Burger : 120/-
2. Roll : 100/-
3. Pizza : 200/-
Enter your choice 1
```

```
Enter your address 201, AK Apartments, Hennur
```

```
Your order for Burger has been placed on Swiggy and will arrive at your address 201, AK Apartments, Hennur
Total payment to be made is  120
```

```
Do you wish to order again? (y/n)
```

```
y
1. Zomato
2. Swiggy
3. Amazon Food
Enter your preferred delivery partner 3
```

```
1. Burger : 120/-
2. Roll : 100/-
3. Pizza : 200/-
Enter your choice 3
```

```
Enter your address 201, AK Apartments, Hennur
```

```
Your order for Pizza has been placed on Amazon Food and will arrive at your address 201, AK Apartments, Hennur
Total payment to be made is  200
```

```
Do you wish to order again? (y/n)
```

```
n
```

Program 3

Objective: To implement a landlord-tenant system which splits the bill with respect to the number of tenants the landlord has. Input should be all bills that need to be split along with no. of tenants.

Code:

```
utilities=input("Enter all bills to be shared separated with a comma\n")
l1=utilities.split(",")
l2=[]
sum=0
n=int(input("Enter no. of tenants "))
n+=1
for i in range(len(l1)):
    print("Enter total amount for",l1[i],"bill ")
    l2.append(float(input()))
for i in range(len(l1)):
    sum+=round(l2[i]/n,2)
    print("The",l1[i],"bill is split as",round(l2[i]/n,2),"per head")
print("\nThe total per head is",sum)
```

Output:

```
1 utilities=input("Enter all bills to be shared separated with a comma\n")
2 l1=utilities.split(",")
3 l2=[]
4 sum=0
5 n=int(input("Enter no. of tenants "))
6 n+=1
7 for i in range(len(l1)):
8     print("Enter total amount for",l1[i],"bill ")
9     l2.append(float(input()))
10    for i in range(len(l1)):
11        sum+=round(l2[i]/n,2)
12        print("The",l1[i],"bill is split as",round(l2[i]/n,2),"per head")
13    print("\nThe total per head is",sum)
```

```
Enter all bills to be shared separated with a comma
water,electricity,internet
Enter no. of tenants 3
Enter total amount for water bill
400
Enter total amount for electricity bill
3200
Enter total amount for internet bill
1000
The water bill is split as 100.0 per head
The electricity bill is split as 800.0 per head
The internet bill is split as 250.0 per head

The total per head is 1150.0
```

Program 4

Objective: To implement a recharge service for your SIM provider. Input should be 10 digit mobile number, State of Service, and choice of recharge.

Code:

```
import sys
v=[['SI no.', 'Talktime(mins)', 'Price', 'Validity'],[1,200,100,28],[2,400,180,56],[3,500,220,56], [4,999,599,84]]
d=[['SI no.', 'Data(Gb)', 'Price', 'Validity'],[1,5,100,28],[2,10,180,56],[3,15,220,56],[4,100,599,84]]
inter=[['SI no.', 'Talktime(mins)', 'Price', 'Validity'],[1,200,599,14],[2,500,999,28]]
tu=[['SI no.', 'Talktime(mins)', 'Price', 'Validity'],[1,100,120,84],[2,200,240,84],[3,300,360,84], [4,500,550,84]]

def show(x):
    print('\n')
    s = [[str(e) for e in row] for row in x]
    lens = [max(map(len, col)) for col in zip(*s)]
    fmt = '\t'.join('{{:{}}}'.format(x) for x in lens)
    table = [fmt.format(*row) for row in s]
    print('\n'.join(table))

print("Welcome to AI Recharge Services")
no=int(input("Enter 10 digit mobile no. "))
x=str(no)
if(len(x)!=10):
    sys.exit("Invalid number. Recharge cancelled")
state=input("Enter your State of Service ")
choice=10
while(choice!=5):
    choice=int(input("\n1. Voice\n2. Data\n3. International Roaming\n4. Top up\n5. Exit\nEnter your choice "))
    if choice==1:
        show(v)
        pack=int(input("Enter the pack you wish to select "))
        print("Please make payment of",v[pack][2],"on the payment screen.")
    elif choice==2:
        show(d)
        pack=int(input("Enter the pack you wish to select "))
        print("Please make payment of",d[pack][2],"on the payment screen.")
    elif choice==3:
        show(inter)
        pack=int(input("Enter the pack you wish to select "))
        print("Please make payment of",inter[pack][2],"on the payment screen.")
    elif choice==4:
        show(du)
        pack=int(input("Enter the pack you wish to select "))
        print("Please make payment of",du[pack][2],"on the payment screen.")
    elif choice==5:
        print("Thank you")
    else:
        print("Invalid choice. Recharge cancelled.")
```

Output (Code):

```
1 import sys
2 v=[ ['Sl no.', 'Talktime(mins)', 'Price', 'Validity'],[1,200,100,28],[2,400,180,56],[3,500,220,56],[4,999,599,84]]
3 d=[['Sl no.', 'Data(Gb)', 'Price', 'Validity'],[1,5,100,28],[2,10,180,56],[3,15,220,56],[4,100,599,84]]
4 inter=[['Sl no.', 'Talktime(mins)', 'Price', 'Validity'],[1,200,599,14],[2,500,999,28]]
5 tu=[['Sl no.', 'Talktime(mins)', 'Price', 'Validity'],[1,100,120,84],[2,200,240,84],[3,300,360,84],[4,500,550,84]]
6
7 def show(x):
8     print('\n')
9     s = [[str(e) for e in row] for row in x]
10    lens = [max(map(len, col)) for col in zip(*s)]
11    fmt = '\t'.join('{{:{}}}'.format(x) for x in lens)
12    table = [fmt.format(*row) for row in s]
13    print('\n'.join(table))
14
15 print("Welcome to Al Recharge Services")
16 no=int(input("Enter 10 digit mobile no. "))
17 x=str(no)
18 if(len(x)!=10):
19     sys.exit("Invalid number. Recharge cancelled")
20 state=input("Enter your State of Service ")
21 choice=10
22 while(choice!=5):
23     choice=int(input("\n1. Voice\n2. Data\n3. International Roaming\n4. Top up\n5. Exit\nEnter your choice "))
24     if choice==1:
25         show(v)
26         pack=int(input("Enter the pack you wish to select "))
27         print("Please make payment of",v[pack][2],"on the payment screen.")
28     elif choice==2:
29         show(d)
30         pack=int(input("Enter the pack you wish to select "))
31         print("Please make payment of",d[pack][2],"on the payment screen.")
32     elif choice==3:
33         show(inter)
34         pack=int(input("Enter the pack you wish to select "))
35         print("Please make payment of",inter[pack][2],"on the payment screen.")
36     elif choice==4:
37         show(du)
38         pack=int(input("Enter the pack you wish to select "))
39         print("Please make payment of",du[pack][2],"on the payment screen.")
40     elif choice==5:
41         print("Thank you")
42     else:
43         print("Invalid choice. Recharge cancelled.")
```

Output (Final):

Welcome to Al Recharge Services
Enter 10 digit mobile no. 9845312340
Enter your State of Service KA

1. Voice
 2. Data
 3. International Roaming
 4. Top up
 5. Exit
- Enter your choice 1

Sl no.	Talktime(mins)	Price	Validity
1	200	100	28
2	400	180	56
3	500	220	56
4	999	599	84

Enter the pack you wish to select 4
Please make payment of 599 on the payment screen.

1. Voice
 2. Data
 3. International Roaming
 4. Top up
 5. Exit
- Enter your choice 2

Sl no.	Data(Gb)	Price	Validity
1	5	100	28
2	10	180	56
3	15	220	56
4	100	599	84

Enter the pack you wish to select 2
Please make payment of 180 on the payment screen.

1. Voice
 2. Data
 3. International Roaming
 4. Top up
 5. Exit
- Enter your choice 5
Thank you

Conclusion: We have successfully implemented various application using Python.

Files in Python

Learning Outcome: To be able to perform different kinds of functions and operations on Files in Python.

Program 1

Objective: Opening, closing, appending, reading, and displaying the properties of a file.

Code:

```
# Open file in write mode  
fo = open("dap.txt", "w")
```

```
# Write to a file  
fo.write("ali")
```

```
# Close file  
fo.close()
```

```
# Open file in append mode  
fo = open("dap.txt", "a")  
fo.write("abbas")  
fo.close()
```

```
# Open and read the file after the appending:  
fo = open("dap.txt", "r")  
print(fo.read())
```

```
# Print properties of a file  
print ("Name of the file: ", fo.name)  
print ("Closed or not : ", fo.closed)  
print ("Opening mode : ", fo.mode)
```

```
fo.close()
```

Output:

```
1 # Opening, closing, appending, reading, and
2 #displaying the properties of a file
3
4 # Open file in write mode
5 fo = open("dap.txt", "w")
6
7 # Write to a file
8 fo.write("ali")
9
10 # Close file
11 fo.close()
12
13 # Open file in append mode
14 fo = open("dap.txt", "a")
15 fo.write("abbas")
16 fo.close()
17
18 # Open and read the file after the appending:
19 fo = open("dap.txt", "r")
20 print(fo.read())
21
22 # Print properties of a file
23 print ("Name of the file: ", fo.name)
24 print ("Closed or not : ", fo.closed)
25 print ("Opening mode : ", fo.mode)
26
27 fo.close()
```

aliabbas

Name of the file: dap.txt

Closed or not : False

Opening mode : r

Program 2

Objective: Checking current pointer position and seeking to a new position.

Code:

```
fo = open("dap.txt", "r")
str = fo.read(4)
print ("Read String is : ", str)

# Check current position
position = fo.tell()
print ("Current file position : ", position)

# Reposition pointer at the beginning once again
position = fo.seek(1, 0);
str = fo.read(3)
print ("Again read String is : ", str)

fo.close()
```

Output:

```
1 # Checking current pointer position and
2 # seeking to a new position
3
4 fo = open("dap.txt", "r")
5 str = fo.read(4)
6 print ("Read String is : ", str)
7
8 # Check current position
9 position = fo.tell()
10 print ("Current file position : ", position)
11
12 # Reposition pointer at the beginning once again
13 position = fo.seek(1, 0);
14 str = fo.read(3)
15 print ("Again read String is : ", str)
16
17 fo.close()
```

```
Read String is : alia
Current file position : 4
Again read String is : lia
```

Program 3

Objective: To implement os functions such as renaming, removing a file creating, checking, changing, and removing directories.

Code:

```
import os

# Rename a file from dap.txt to test.txt
os.rename("dap.txt", "test.txt")

# Delete file test.txt
os.remove("test.txt")

# This would give location of the current directory
print(os.getcwd())

# Create a directory "test"
os.mkdir("test")

# Changing a directory to "/home/newdir"
os.chdir("/content/test")

# This would remove "/tmp/test" directory.
os.rmdir( "/content/test" )
```

Output:

```
1 # os functions such as renaming, removing a file
2 # creating, checking, changing, removing directories
3
4 import os
5
6 # Rename a file from dap.txt to test.txt
7 os.rename("dap.txt", "test.txt")
8
9 # Delete file test.txt
10 os.remove("test.txt")
11
12 # This would give location of the current directory
13 print(os.getcwd())
14
15 # Create a directory "test"
16 os.mkdir("test")
17
18 # Changing a directory to "/home/newdir"
19 os.chdir("/content/test")
20
21 # This would remove "/tmp/test" directory.
22 os.rmdir( "/content/test" )
```

/content

Conclusion: We have successfully performed different kinds of functions and operations on Files in Python.

Tuples in Python

Learning Outcome: To be able to effectively apply and perform different kinds of functions and operations on tuples datatype.

Program 1

Objective: Creating, printing, print with index and deletion of tuples.

Code:

```
# Creation of tuple
tup1= ('cse','it','mech','ece','electrical')
tup2=(1,23,4,5,5,6)

# Printing tuple
print(tup1)
print(tup2)

# Printing with index
print(tup1[0])
print(tup1[2])
print(tup1[2:5])

# Deletion of tuple
del(tup1)
del(tup2)
```

Output:

```
1 # Creating, printing, print with index
2 # and deletion of tuples
3
4 # Creation of tuple
5 tup1= ('cse','it','mech','ece','electrical')
6 tup2=(1,23,4,5,5,6)
7
8 # Printing tuple
9 print(tup1)
10 print(tup2)
11
12 # Printing with index
13 print(tup1[0])
14 print(tup1[2])
15 print(tup1[2:5])
16
17 # Deletion of tuple
18 del(tup1)
19 del(tup2)
```

```
('cse', 'it', 'mech', 'ece', 'electrical')
(1, 23, 4, 5, 5, 6)
cse
mech
('mech', 'ece', 'electrical')
```

Program 2

Objective: Concatenation, finding length, and max element in a tuple.

Code:

```
tup1= ('cse','it','mech','ece','electrical')
print(tup1)
tup2= ('ali','roro','james')
print(tup2)

# Concatenation
tup3=tup1+tup2
print(tup3)

# Finding length
print("Length of first tuple is ",len(tup1))
print("Length of second tuple is ",len(tup2))

#Finding max element
print("Max element in first tuple is",max(tup1))
```

Output:

```
1 # Concatenation, finding length, and max element
2
3 tup1= ('cse','it','mech','ece','electrical')
4 print(tup1)
5 tup2= ('ali','roro','james')
6 print(tup2)
7
8 # Concatenation
9 tup3=tup1+tup2
10 print(tup3)
11
12 # Finding length
13 print("Length of first tuple is ",len(tup1))
14 print("Length of second tuple is ",len(tup2))
15
16 #Finding max element
17 print("Max element in first tuple is",max(tup1))
```

```
('cse', 'it', 'mech', 'ece', 'electrical')
('ali', 'roro', 'james')
('cse', 'it', 'mech', 'ece', 'electrical', 'ali', 'roro', 'james')
Length of first tuple is 5
Length of second tuple is 3
Max element in first tuple is mech
```

Conclusion: We have successfully implemented and performed different kinds of functions and operations on tuples datatype.

Dictionaries in Python

Learning Outcome: To be able to effectively apply and perform different kinds of functions and operations on dictionaries datatype.

Program 1

Objective: Creating, printing, updating, appending, deleting a key, clearing/deleting the dictionary.

Code:

```
# Creation of a dictionary
dict={'name':'ali','designation':'student','qual':'2nd PU','univ':'cmr'}
```

```
# Printing of dictionary elements
print(dict['name'])
print(dict['qual'])
print(dict)
```

```
# Updating a value
dict['name']='abbas'
print("After updating name-",dict['name'])
```

```
# Appending a value
dict['class']='cse7sem'
print("After appending class-",dict['class'])
```

```
# Deleting a key
del dict['designation']
print("After removing designation-",dict)
```

```
# Clearing the dictionary
dict.clear()
print("After clearing dictionary-",dict)
```

```
# Deleting the dictionary
del dict
```

Output:

```
1 # Creating, printing, updating, appending
2 # Deleting a key, clearing/deleting the dictionary
3
4 # Creation of a dictionary
5 dict={'name':'ali','designation':'student','qual':'2nd PU','univ':'cmr'}
6
7 # Printing of dictionary elements
8 print(dict['name'])
9 print(dict['qual'])
10 print(dict)
11
12 # Updating a value
13 dict['name']='abbas'
14 print("After updating name-",dict['name'])
15
16 # Appending a value
17 dict['class']='cse7sem'
18 print("After appending class-",dict['class'])
19
20 # Deleting a key
21 del dict['designation']
22 print("After removing designation-",dict)
23
24 # Clearing the dictionary
25 dict.clear()
26 print("After clearing dictionary-",dict)
27
28 # Deleting the dictionary
29 del dict
```

```
ali
2nd PU
{'name': 'ali', 'designation': 'student', 'qual': '2nd PU', 'univ': 'cmr'}
After updating name- abbas
After appending class- cse7sem
After removing designation- {'name': 'abbas', 'qual': '2nd PU', 'univ': 'cmr', 'class': 'cse7sem'}
After clearing dictionary- {}
```

Program 2

Objective: To print the properties of dictionaries.

Code:

```
dict={'name':'ali','designation':'student','qual':'2nd PU','univ':'cmr', 'name':'abbas'}  
# In case same key is given 2 values, latest will be considered  
print(dict['name'])  
  
# Length of dictionary  
print(len(dict))  
  
# Returns only the values  
print(dict.values())  
  
# Returns only the keys  
print(dict.keys())  
  
# Returns respective key-value pairs  
print(dict.items())
```

Output:

```
1 # Printing properties of dictionaries  
2  
3 dict={'name':'ali','designation':'student','qual':'2nd PU','univ':'cmr', 'name':'abbas'}  
4  
5 # In case same key is given 2 values, latest will be considered  
6 print(dict['name'])  
7  
8 # Length of dictionary  
9 print(len(dict))  
10  
11 # Returns only the values  
12 print(dict.values())  
13  
14 # Returns only the keys  
15 print(dict.keys())  
16  
17 # Returns respective key-value pairs  
18 print(dict.items())  
  
abbas  
4  
dict_values(['abbas', 'student', '2nd PU', 'cmr'])  
dict_keys(['name', 'designation', 'qual', 'univ'])  
dict_items([('name', 'abbas'), ('designation', 'student'), ('qual', '2nd PU'), ('univ', 'cmr'))]
```

Program 3

Objective: To implement get(), copy(), and update() methods.

Code:

```
dict={'name':'ali','designation':'student','qual':'2nd PU','univ':'cmr', 'name':'abbas'}

# The get() method returns the value for the given key, if present in the dictionary.
# If not, then it will return None (if get() is used with only one argument).

print(dict.get('name'))
print(dict.get('edu'))
print(dict.get('edu',"nothing"))

# copy() method is used to copy a dictionary
dict1=dict.copy()
print(dict1)

# update() method appends to a dictionary
dict2={'edu':'btech','semester':'7sem'}
dict.update(dict2)
print(dict)
```

Output:

```
1 # get(), copy(), and update() methods
2
3 dict={'name':'ali','designation':'student','qual':'2nd PU','univ':'cmr', 'name':'abbas'}
4
5 # The get() method returns the value for the given key, if present in the dictionary.
6 # If not, then it will return None (if get() is used with only one argument).
7
8 print(dict.get('name'))
9 print(dict.get('edu'))
10 print(dict.get('edu',"nothing"))
11
12 # copy() method is used to copy a dictionary
13 dict1=dict.copy()
14 print(dict1)
15
16 # update() method appends to a dictionary
17 dict2={'edu':'btech','semester':'7sem'}
18 dict.update(dict2)
19 print(dict)

abbas
None
nothing
{'name': 'abbas', 'designation': 'student', 'qual': '2nd PU', 'univ': 'cmr'}
{'name': 'abbas', 'designation': 'student', 'qual': '2nd PU', 'univ': 'cmr', 'edu': 'btech', 'semester': '7sem'}
```

Conclusion: We have successfully implemented and performed different kinds of functions and operations on dictionaries datatype.

NumPy Basics in Python

Learning Outcome: To be able to effectively perform basic NumPy functions and operations in Python.

Program 1

Objective: To create and print 1D and 2D arrays using NumPy

Code:

```
import numpy as np

a = np.array([1, 2, 3])
b = np.array([[1 , 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
print(a)
print(b)
```

Output:

```
1 #Creating and printing 1D and 2D array
2 import numpy as np
3
4 a = np.array([1, 2, 3])
5 b = np.array([[1 , 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
6 print(a)
7 print(b)
```

```
[1 2 3]
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

Program 2

Objective: To implement a few built in functions such as zeros(), ones(), empty(), arange(), delete(), and sort()

Code:

```
#Returns a new array of given shape and type with zeros  
c=np.zeros(2)  
print(c)  
  
#Returns a new array of given shape and type with ones  
d=np.ones(4)  
print(d)  
  
e = np.ones(2, dtype=int)  
print(e)  
  
#Returns a new array of given shape and type without initializing entries  
f=np.empty([4])  
print(f)  
  
#Returns an array with evenly spaced elements as per the interval  
g=np.arange(7)  
print(g)  
  
h=np.arange(2,9,2)  
print(h)  
  
#Returns a new array with the deletion of sub-arrays  
i=np.delete(h, 2)  
print(i)  
  
#Returns a sorted array in ascending order  
z = np.array([19, 22, 34, 14, 55, 76, 47, 8])  
print(z)  
j=np.sort(z)  
print(j)
```

Output:

```
1 #Few in built numpy functions
2
3 #Returns a new array of given shape and type with zeros
4 c=np.zeros(2)
5 print(c)
6
7 #Returns a new array of given shape and type with ones
8 d=np.ones(4)
9 print(d)
10
11 e = np.ones(2, dtype=int)
12 print(e)
13
14 #Returns a new array of given shape and type without initializing entries
15 f=np.empty([4])
16 print(f)
17
18 #Returns an array with evenly spaced elements as per the interval
19 g=np.arange(7)
20 print(g)
21
22 h=np.arange(2,9,2)
23 print(h)
24
25 #Returns a new array with the deletion of sub-arrays
26 i=np.delete(h, 2)
27 print(i)
28
29 #Returns a sorted array in ascending order
30 z = np.array([19, 22, 34, 14, 55, 76, 47, 8])
31 print(z)
32 j=np.sort(z)
33 print(j)
```

```
[0. 0.]
[1. 1. 1. 1.]
[1 1]
[1. 1. 1. 1.]
[0 1 2 3 4 5 6]
[2 4 6 8]
[2 4 8]
[19 22 34 14 55 76 47 8]
[ 8 14 19 22 34 47 55 76]
```

Program 3

Objective: To return the number of dimensions, size and shape of an array

Code:

```
#Return the number of dimensions, size and shape of an array
arr = np.array([[1, 2, 3], [4, 5, 6]])
a=arr.ndim #dimension
b=arr.size #size
c=arr.shape #shape
print(arr)
print("dimensions = ",a,"\\nsize = ",b,"\\nshape =",c)

#Shapes an array without changing the data of the array.
y = arr.reshape(3,2)
print("\\nAfter reshaping\\n",y)
```

Output:

```
1 #Return the number of dimensions, size and shape of an array
2 arr = np.array([[1, 2, 3], [4, 5, 6]])
3 a=arr.ndim #dimension
4 b=arr.size #size
5 c=arr.shape #shape
6 print(arr)
7 print("dimensions = ",a,"\\nsize = ",b,"\\nshape =",c)
8
9 #Shapes an array without changing the data of the array.
10 y = arr.reshape(3,2)
11 print("\\nAfter reshaping\\n",y)
```

```
[[1 2 3]
 [4 5 6]]
dimensions =  2
size =  6
shape = (2, 3)
```

```
After reshaping
[[1 2]
 [3 4]
 [5 6]]
```

Conclusion: We have successfully implemented basic NumPy functions and operations in Python.

NumPy Arithmetic Operations

Learning Outcome: To be able to effectively perform NumPy arithmetic operations in Python.

Program 1

Objective: To perform addition, subtraction, multiplication, division, power, and reciprocal operations.

Code:

```
a=np.array(5)
print("A= ",a)
b=np.array([10,20,30])
print("B= ", b)

#Addition
addition= np.add(a,b)
print("\nAfter Addition =", addition)

#Subtraction
sub=np.subtract(b,a)
print("After Subtraction =",sub)

#Multiplication
mul=np.multiply (a,b)
print("After Multiplcation =",mul)

#Division
div=np.divide (b,a)
print("After Division =",div)

div1=np.divide (b,10)
print("After Division =",div1)

#Reciprocal
rec=np.reciprocal(div)
print(rec)

#Power
pow=np.power(a,2)
print("after ^2 = ", pow)
```

Output:

```
1 #Arithematic Operations
2 a=np.array(5)
3 print("A= ",a)
4 b=np.array([10,20,30])
5 print("B= ", b)
6
7 #Addition
8 addition= np.add(a,b)
9 print("\nAfter Addition =", addition)
10
11 #Subtraction
12 sub=np.subtract(b,a)
13 print("After Subtraction =",sub)
14
15 #Multiplication
16 mul=np.multiply (a,b)
17 print("After Multiplcation =",mul)
18
19 #Division
20 div=np.divide (b,a)
21 print("After Division =",div)
22
23 div1=np.divide (b,10)
24 print("After Division =",div1)
25
26 #Reciprocal
27 rec=np.reciprocal(div)
28 print(rec)
29
30 #Power
31 pow=np.power(a,2)
32 print("after ^2 = ", pow)
```

```
A= 5
B= [10 20 30]

After Addition = [15 25 35]
After Subtraction = [ 5 15 25]
After Multiplcation = [ 50 100 150]
After Division = [2. 4. 6.]
After Division = [1. 2. 3.]
[0.5 0.25 0.16666667]
after ^2 = 25
```

Program 2

Objective: To perform mod, remainder, real, imaginary, and conjugate operations.

Code:

```
#Mod and Remainder return the same result
```

```
#In fact, mod is an alias for remainder
```

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

```
b= np.array ([3,5,7])
```

```
print("values of A =", a)
```

```
print("values of B =", b)
```

```
m=np.mod(a,b)
```

```
r=np.remainder(a,b)
```

```
print("\nvalues of MOD =", m)
```

```
print("values of REMAINDER =", r)
```

```
#Real and Imaginary numbers
```

```
c=np.array([-5.6j, 0.2j,11, 1+1j])
```

```
print("\n",c)
```

```
#Returns only real part
```

```
print("\nreal=",np.real(c))
```

```
#Returns only imaginary part
```

```
print("imaginary=",np.imag(c))
```

```
#Returns the conjugate
```

```
print("Conjugate=", np.conj(c))
```

Output:

```
1 #Mod and Remainder return the same result
2 #In fact, mod is an alias for remainder
3 a= np.array ([10,20,30])
4 b= np.array ([3,5,7])
5
6 print("values of A =", a)
7 print("values of B =", b)
8
9 m=np.mod(a,b)
10 r=np.remainder(a,b)
11 print("\nvalues of MOD =", m)
12 print("values of REMAINDER =", r)
13
14 #Real and Imaginary numbers
15 c=np.array([-5.6j, 0.2j,11, 1+1j])
16 print("\n",c)
17
18 #Returns only real part
19 print("\nreal=",np.real(c))
20
21 #Returns only imaginary part
22 print("imaginary=",np.imag(c))
23
24 #Returns the conjugate
25 print("Conjugate=", np.conj(c))
```

values of A = [10 20 30]

values of B = [3 5 7]

values of MOD = [1 0 2]

values of REMAINDER = [1 0 2]

[-0.-5.6j 0.+0.2j 11.+0.j 1.+1.j]

real= [-0. 0. 11. 1.]

imaginary= [-5.6 0.2 0. 1.]

Conjugate= [-0.+5.6j 0.-0.2j 11.-0.j 1.-1.j]

Program 3

Objective: To implement around(), ceil(), and floor() functions.

Code:

```
d=np.array ([1.1,-5.32,123,-0.56,25.5,32])
print("\n",d)
print("\nafter rounding up\n", np.around(d))
print("\nafter rounding up to 1st decimal value\n", np.around(d,decimals=1))

#Floor round off
print("\nFloor round off")
print(np.floor(d))

#Ceiling round off
print("\nCeiling round off")
print(np.ceil(d))
```

Output:

```
1 #Rounding off values
2 d=np.array ([1.1,-5.32,123,-0.56,25.5,32])
3 print("\n",d)
4 print("\nafter rounding up\n", np.around(d))
5 print("\nafter rounding up to 1st decimal value\n", np.around(d,decimals=1))
6
7 #Floor round off
8 print("\nFloor round off")
9 print(np.floor(d))
10
11 #Ceiling round off
12 print("\nCeiling round off")
13 print(np.ceil(d))
```

[1.1 -5.32 123. -0.56 25.5 32.]

after rounding up

[1. -5. 123. -1. 26. 32.]

after rounding up to 1st decimal value

[1.1 -5.3 123. -0.6 25.5 32.]

Floor round off

[1. -6. 123. -1. 25. 32.]

Ceiling round off

[2. -5. 123. -0. 26. 32.]

Conclusion: We have successfully implemented NumPy arithmetic operations in Python.