TITLE: Lists

OBJECTIVE: To perform List operations

OUTCOME: Tuple operations like Concatenation, Adding/Deleting elements, accessing elements, replacing elements, appending lists

CONCLUSION: Successfully implemented List operations

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

```
list1[0]:  1997
list2[1:5]:  [2, 3]
```

```
[ ] list = ['physics', 'chemistry', 1997, 2000];
    print ("Value available at index 2 : ")
    print (list[2])
    list[2] = 999999;
    print ("New value available at index 2 : ")
    print (list[2])
```

```
Value available at index 2 :
1997
New value available at index 2 :
999999
```

```
[ ] list1 = ['physics', 'chemistry', 1997, 2000];
    print (list1)
    del (list1[2]);
    print ("After deleting value at index 2 :")
    print (list1)
```

```
['physics', 'chemistry', 1997, 2000]
After deleting value at index 2 :
['physics', 'chemistry', 2000]
```

```
[ ] list1, list2 = [123, 'xyz', 'zara'], [456, 'abc']
    print ("First list length : ", len(list1))
    print ("Second list length : ", len(list2))
```

```
First list length :  3
Second list length :  2
```

```
[ ] aList = [123, 'xyz', 'zara', 'abc'];
    aList.append( 2009 );
    print ("Updated List : ", aList)
```

```
Updated List :  [123, 'xyz', 'zara', 'abc', 2009]
```

```
[ ] aList = [123, 'xyz', 'xyz', 'abc', 123];
    print ("Count for xyz : ", aList.count('xyz'))
    print ("Count for m : ", aList.count('zara'))
```

```
Count for xyz :  2
Count for m :  0
```

```
[ ] aList = [123, 'xyz', 'zara', 'abc', 123];
    bList = [2009, 'kru'];
    aList.extend(bList)
    print ("Extended List : ", aList)
```

```
Extended List :  [123, 'xyz', 'zara', 'abc', 123, 2009, 'kru']
```

```
[ ] aList=[1,'x','k','a','b']
    print("index for x ",aList.index('x'))
```

```
index for x  1
```

```
[ ] aList=[1,4,5,6,'kru']
    aList.insert(3,78)
    print("Final list : ",aList)
```

```
Final list :  [1, 4, 5, 78, 6, 'kru']
```

```
[ ] aList=[1,2,'kru','n']
    print(aList)
    print("List after popping :",aList.pop())
    print(aList)
    print("list after popping element from mentioned index : ",aList.pop(2))
    print(aList)
    aList.insert(2,'kk')
    print(aList)
```

```
[1, 2, 'kru', 'n']
List after popping : n
[1, 2, 'kru']
list after popping element from mentioned index :  kru
[1, 2]
[1, 2, 'kk']
```

```
[ ] aList=[1,2,'kru','n']
    aList.remove(1)
    print(aList)
    aList.remove('n')
    print(aList)
```

```
[2, 'kru', 'n']
[2, 'kru']
```

```
[ ] aList=[1,2,'kru','n']
    aList.reverse()
    print(aList)
```

```
['n', 'kru', 2, 1]
```

```
[ ] aList=['xyz','hi',' kru','n']
    aList.sort()
    print(aList)
```

```
[' kru', 'hi', 'n', 'xyz']
```

TITLE: Loops

OBJECTIVE: To perform Loops operations using break, continue

OUTCOME: Working with loops using continue break

CONCLUSION: Successfully implemented Loops operations

```
list = [10,30,23,43,65,12]
sum = 0
for i in list:
    sum = sum+i
print("The sum is:",sum)
```

```
The sum is: 183
```

```
for i in range(14):
    print(i,end=' ')
```

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13
```

```
n = int(input("Enter the number "))
for i in range(1,11):
    c = n*i
    print(n,"*",i,"=",c)
```

```
Enter the number 4
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
4 * 10 = 40
```

```
n = int(input("Enter the number "))
for i in range(2,n,2):
    print(i)
```

```
Enter the number 19
2
4
6
8
10
12
14
16
18
```

```
list = ['Peter','Shabnam','Ricky','Devansh']
for i in range(len(list)):
    print("Hello",list[i])
```

```
Hello Peter
Hello Shabnam
Hello Ricky
Hello Devansh
```

```
for i in range(0,7):
    print(i)
else:
    print("for loop completely exhausted, since there is no break.")
```

```
0
1
2
3
4
5
6
for loop completely exhausted, since there is no break.
```

```
for i in range(0,7):
    print(i)
    print("bye")
    continue;
    print("hello")
else:print("for loop is exhausted");
print("The loop is broken due to break statement...came out of the loop")
```

```
0
bye
1
bye
2
bye
3
bye
4
bye
5
bye
6
bye
for loop is exhausted
The loop is broken due to break statement...came out of the loop
```

```
i = 0
str1 = 'shabnam'
print(str1)
while i < len(str1):
    print('entered while loop before if statement')
    if str1[i] == 'a' or str1[i] == 'u':
        print('entered if statement')
        i += 1
        print('i incremented')
        continue
        print('after continue')
    print('Current Letter :', str1[i])
    i += 1
    print('going back to starting of while loop')
```

```
shabnam
entered while loop before if statement
Current Letter : s
going back to starting of while loop
entered while loop before if statement
Current Letter : h
going back to starting of while loop
entered while loop before if statement
entered if statement
i incremented
entered while loop before if statement
Current Letter : b
going back to starting of while loop
entered while loop before if statement
Current Letter : n
going back to starting of while loop
entered while loop before if statement
entered if statement
i incremented
entered while loop before if statement
```

```
i = 0
str1 = 'shabnam'

while i < len(str1):
    if str1[i] == 'n':
        i += 1
        break
    print('Current Letter :', str1[i])
    i += 1
```

```
Current Letter : s
Current Letter : h
Current Letter : a
Current Letter : b
Current Letter : a
Current Letter : m
```

```
    str1 = 'shabnam'
    i = 0

    while i < len(str1):
        i += 1
        pass
    print('Value of i :', i)
```

```
    Value of i : 7
```

```
    i=1
    #The while loop will iterate until condition becomes false.
    while(i<=10):
        print(i)
        i=i+1
```

```
1
2
3
4
5
6
7
8
9
10
```

```
    i=1
    number = int(input("Enter the number:"))
    while i<=10:
        print("%d X %d = %d \n"%(number,i,number*i))
        i = i+1
```

```
Enter the number:3
3 X 1 = 3

3 X 2 = 6

3 X 3 = 9

3 X 4 = 12

3 X 5 = 15

3 X 6 = 18

3 X 7 = 21

3 X 8 = 24

3 X 9 = 27

3 X 10 = 30
```

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

```
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
Hi! we are inside the infinite while loop
```

```
i=1
while(i<=5):
    print(i)
    i=i+1
else:
    print("The while loop exhausted")
```

```
1
2
3
4
5
The while loop exhausted
```

```
i=1
while(i<=5):
    print(i)
    i=i+1
    if(i==3):
        break
else:
    print("The while loop exhausted")
print("bye bye")
```

```
1
2
bye bye
```

```
list =[1,2,3,4]
i=1;
count = 1;
for i in list:
    if i == 4:
        print("item matched")
        count = count + 1;
        break
print("found at",count,"location");
```

```
item matched
found at 2 location
```

```
str = "python"
for i in str:
    if i == 'o':
        break
    print(i);
```

```
p
y
t
h
```

```
n=2
while 1:
    i=1;
    while i<=10:
        print("%d X %d = %d\n"%(n,i,n*i));
        i = i+1;
    choice = int(input("Do you want to continue printing the table, press 0 for no?"))
    if choice == 0:
        break;
    n=n+1
```

```
2 X 1 = 2

2 X 2 = 4

2 X 3 = 6

2 X 4 = 8

2 X 5 = 10

2 X 6 = 12

2 X 7 = 14

2 X 8 = 16

2 X 9 = 18

2 X 10 = 20

Do you want to continue printing the table, press 0 for no?1
3 X 1 = 3

3 X 2 = 6
```

```
i = 0
while(i < 10):
   i = i+1
   if(i == 5):
      continue
   print(i)
```

```
1
2
3
4
6
7
8
9
10
```

```
str = "Shabnam"
for i in str:
    if(i == 'n'):
        continue
    print(i)
```

```
S
h
a
b
a
m
```

```
list = [1,2,3,4,5]
flag = 0
for i in list:
    print("Current element:",i,end=" ");
    if i==3:
        pass
        print("\nWe are inside pass block\n");
        flag = 1
    if flag==1:
        print("\nCame out of pass\n");
        flag=0
```

```
Current element: 1 Current element: 2 Current element: 3
We are inside pass block


Came out of pass

Current element: 4 Current element: 5
```

```
str = "Python"
for i in str:
    print(i)
```

```
P
y
t
h
o
n
```

```
[ ]  list = [1,2,3,4,5,6,7,8,9,10]
     n = 6
     for i in list:
         c = n*i
         print(n," *",i, " =", c)
```

```
6  * 1  = 6
6  * 2  = 12
6  * 3  = 18
6  * 4  = 24
6  * 5  = 30
6  * 6  = 36
6  * 7  = 42
6  * 8  = 48
6  * 9  = 54
6  * 10  = 60
```

TITLE: If-else

OBJECTIVE: To perform If-else loop

OUTCOME: Check whether number is even or odd, print grade, vote eligibility using if-else loop

CONCLUSION: Successfully implemented If-else loop

```
num = int(input("enter the number?"))
if num%2 == 0:
    print("Number is even")
print("bye")
```

```
enter the number?111
bye
```

```
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");
if b>a and b>c:
    print("b is largest");
if c>a and c>b:
    print("c is largest");
```

```
Enter a- 10
Enter b- 20
Enter c- 15
b is largest
```

```
age = int (input("Enter your age? "))
if age>=18:
    print("You are eligible to vote !!");
else:
    print("Sorry! you have to wait !!");
```

```
Enter your age? 23
You are eligible to vote !!
```

```
num = int(input("enter the number?"))
if num%2 == 0:
    print("Number is even...")
else:
    print("Number is odd...")
```

```
enter the number?23
Number is odd...
```

```
number = int(input("Enter the number?"))
if number==10:
    print("number is equals to 10")
elif number==50:
    print("number is equal to 50");
elif number==100:
    print("number is equal to 100");
else:
    print("number is not equal to 10, 50 or 100");
```

```
Enter the number?3435
number is not equal to 10, 50 or 100
```

```
number = input("Enter the number?")
if number==10:
    print("number is equals to 10")
elif number==50:
    print("number is equal to 50");
elif number==100:
    print("number is equal to 100");
else:
    print("number is not equal to 10, 50 or 100");
```

```
Enter the number?22
number is not equal to 10, 50 or 100
```

```
    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 are fail ?")
```

```
Enter the marks? 78
You scored grade B + ...
```

TITLE: Strings

OBJECTIVE: To perform String operations using break, continue

OUTCOME: String operations like Concatenation, Append.

CONCLUSION: Successfully implemented String operations

```
var1 = 'Hello World!'
var2 = "Python Programming"
print(var1," ",var2)
```

```
Hello World!   Python Programming
```

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

print (var1[0])



print (var2[1:5])
```

```
H
ytho
```

```
var1 = 'Hello World!'
print ("Updated String :- ", var1[0:6] + 'Shabnam')
```

```
Updated String :-  Hello Shabnam
```

```
str1 = input("Please Enter Your Own String : ")

str2 = str1
str3 = str1[:]
str4 = str1[2:6]

print("The Final String : Str2  = ", str2)
print("The Final String : Str3  = = ", str3)
print("The Final String : Str4  = = ", str4)
```

```
Please Enter Your Own String : hello class
The Final String : Str2  =  hello class
The Final String : Str3  = =  hello class
The Final String : Str4  = =  llo
```

```
#Python String capitalize() method returns a copy of the string with only its first character capitalized.
str = "this is string example....wow!!!";
print ("str.capitalize() : ", str.capitalize())
```

```
str.capitalize() :  This is string example....wow!!!
```

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


str = "this is string example....wow!!!"
print ("str.center(40, 'a') : ", str.center(40, '*'))
```

```
str.center(40, 'a') :  ****this is string example....wow!!!****
```

```
str1 = "this is string example....wow!!!";
str2 = "exam";

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

```
15
15
```

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

str = "this2009";  # No space in this string
print (str.isalnum())

str = "this is string example....wow!!!";
print (str.isalnum())
```

```
True
False
```

```
str = "ShabnamSharma";  # No space & digit in this string
print (str.isalpha())

str = "this is string example....wow!!!";
print (str.isalpha())
```

```
True
False
```

```
str = "  ";
print (str.isspace())

str = "This is string example....wow!!!";
print (str.isspace())
```

```
True
False
```

```
str = "This Is String Example...Wow!!!";
print (str.istitle())

str = "This is string example....wow!!!";
print (str.istitle())
```

```
True
False
```

```
str = "THIS IS STRING EXAMPLE....WOW!!!";
print (str.isupper())

str = "THIS is string example....wow!!!";
print (str.isupper())
```

```
True
False
```

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

```
abc  *  *  bttt  *  *  cqweqe
```

```
str = "this is string example....wow!!!";
print ("Length of the string: ", len(str))
```

```
Length of the string:  32
```

```
#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'))
```

```
this is string example....wow!!!
this is string example....wow!!!999999
88888888this is string example....wow!!!
```

```
str = "THIS IS STRING EXAMPLE....WOW!!!";
print (str.lower())
```

```
this is string example....wow!!!
```

```
str = "check....wow!!!";
print ("Max character: " + max(str))

str = "shabnam....!!!";
print ("Max character: " + max(str))
```

```
Max character: w
Max character: s
```

```
str = "this-is-real-string-example....wow!!!";
print ("Min character: " + min(str))

str = "this-is-a-string-example....wow!!!";
print ("Min character: " + min(str))
```

```
Min character: !
Min character: !
```

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

str = "this is string example....wow!!! this is really string"
print (str.replace("is", "was"))
print (str.replace("is", "was", 3))
```

```
thwas was string example....wow!!! thwas was really string
thwas was string example....wow!!! thwas is really string
```

```
str = "123456";  # Only digit in this string
print (str.isdigit())

str = "this is string example....wow!!!";
print (str.isdigit())
```

```
True
False
```

```
str = "THIS is string example....wow!!!";
print (str.islower())

str = "this is string example....wow!!!";
print (str.islower())
```

```
False
True
```

```
str = "this2009";
print (str.isnumeric())

str = "23443434";
print (str.isnumeric())
```

```
False
True
```

TITLE: Tuples

OBJECTIVE: To perform Tuple operations

OUTCOME: Tuple operations like Concatenation, Adding/Deleting elements, Accessing elements, Replacing elements

CONCLUSION: Successfully implemented Tuple operations

```
tup= ('cse','it','mech')
print(tup)
tup1=(1,23,4,5,5,6)
print(tup1)
tup2="abc","def"
print(tup2)
```

```
('cse', 'it', 'mech')
(1, 23, 4, 5, 5, 6)
('abc', 'def')
```

```
tup1 = ()
print(tup1)
```

```
()
```

```
tup1 = (24)
print(tup1)
```

```
24
```

```
tup= ('cse','it','mech','ece','electrical')
print(tup[0])
print(tup[2])
print(tup[2:5])
```

```
cse
mech
('mech', 'ece', 'electrical')
```

```
tup1= ('cse','it','mech','ece','electrical')
print(tup1)
tup2= ('shabnam','nishitha','roy')
print(tup2)
a=len(tup1)
b=len(tup2)
print(a)
print(b)

print(len(tup1))
```

```
('cse', 'it', 'mech', 'ece', 'electrical')
('shabnam', 'nishitha', 'roy')
5
3
5
```

```
tup1= ('cse','it','mech','ece','electrical')
print(tup1)
tup2= ('shabnam','nishitha','roy')
print(tup2)



print(max(tup1))
```

```
('cse', 'it', 'mech', 'ece', 'electrical')
('shabnam', 'nishitha', 'roy')
mech
```

TITLE: Dictionaries

OBJECTIVE: To perform Dictionaries operations

OUTCOME: Tuple operations like Adding/Deleting elements, Accessing elements, Replacing elements

CONCLUSION: Successfully implemented Dictionaries operations

```
dict={'name':'shabnam','designation':'AsP','qual':'phd','univ':'cmr'}
print(dict['name'])


print(dict['qual'])
```

```
shabnam
phd
```

```
dict={'name':'shabnam','designation':'AsP','qual':'phd','univ':'cmr'}

dict['name']='sharma'

print(dict['name'])

dict['class']='cse7sem'

print(dict['class'])
```

```
sharma
cse7sem
```

```
dict={'name':'shabnam','designation':'AsP','qual':'phd','univ':'cmr'}


del dict['designation']

#print(dict['designation'])

dict.clear()

#print(dict['name'])

del dict
```

```
dict={'name':'shabnam','designation':'AsP','qual':'phd','univ':'cmr', 'name':'Nishitha'}


print(dict['name'])

print(len(dict))
```

```
Nishitha
4
```

```
dict={'name':'shabnam','designation':'AsP','qual':'phd','univ':'cmr', 'name':'Nishitha'}


print(dict.values())

print(dict.items())

print(dict.keys())
```

```
dict_values(['Nishitha', 'AsP', 'phd', 'cmr'])
dict_items([('name', 'Nishitha'), ('designation', 'AsP'), ('qual', 'phd'), ('univ', 'cmr')])
dict_keys(['name', 'designation', 'qual', 'univ'])
```

```
dict={'name':'shabnam','designation':'AsP','qual':'phd','univ':'cmr', 'name':'Nishitha'}


print(dict['name'])

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

```
Nishitha
nothing
```

TITLE: Files

OBJECTIVE: To perform File operations

OUTCOME: File operations like Reading, Writing files

CONCLUSION: Successfully implemented operations

```
print ("Python is really a great language,", "isn't it?")
```

```
Python is really a great language, isn't it?
```

```
str = input("Enter your input: ")
print ("Received input is : ", str)
```

```
Enter your input: hello
Received input is :  hello
```

```
# Open a file
fo = open("cse7sem.txt", "wb")
print ("Name of the file: ", fo.name)

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

```
Name of the file:  cse7sem.txt
```

```
f = open("cse7sem1.txt", "a")
f.write("kruthika")
f.close()

#open and read the file after the appending:
f = open("cse7sem1.txt", "r")
print(f.read())
```

```
kruthika
```

```
# Open a file
fo = open("cse7sem1.txt", "r+")

str = fo.read(10)
print ("Read String is : ", str) #shabnamsha---output

# 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(7)
print ("Again read String is : ", str)

ptr=fo.seek(0,1)
str1=fo.read(3)
print("my new string= ", str1)

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

```
Read String is :  shabnamsha
Current file position :  10
Again read String is :  habnams
my new string=  har
```

TITLE: Numpy

OBJECTIVE: To perform Numpy operations

OUTCOME: Numpy operations like Addition, Power, Reciprocal, Multiplication, Modulus, Division

CONCLUSION: Successfully implemented Numpy operations

```
import numpy as npa
npa= np.array([1, 2, 3])
print(npa)
```

```
[1 2 3]
```

```
import numpy as npp
a = npp.array([[1 , 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
print(a)
```

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

```
import numpy as np
a=np.zeros(2)
print(a)
```

```
[0. 0.]
```

```
import numpy as np
a=np.ones(4)
print(a)
```

```
[1. 1. 1. 1.]
```

```
import numpy as np
ab=np.empty(4)
print(ab)
```

```
[1. 1. 1. 1.]
```

```
import numpy as np
a=np.arange(7)
print(a)
```

```
[0 1 2 3 4 5 6]
```

```
import numpy as np
a=np.arange(2,9,2)
print(a)
```

```
[2 4 6 8]
```

```
y = np.ones(2, dtype=int)
print(y)
y
```

```
[1 1]
array([1, 1])
```

```
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8])
print(arr)
```

```
[1 2 3 4 5 6 7 8]
```

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


a=np.delete(a, 1)
print(a)
```

```
[1 2 3 4 5 6 7 8]
[1 3 4 5 6 7 8]
```

```
a = np.array([19, 22, 34, 14, 55, 76, 47, 8])
print(a)


a=np.sort(a)
print(a)
```

```
[19 22 34 14 55 76 47  8]
[ 8 14 19 22 34 47 55 76]
```

```
import numpy as np

arr = np.array([[1, 2, 3], [4, 5, 6]])
a=arr.ndim
print(arr)
print("dimensions = ", a)
```

```
[[1 2 3]
 [4 5 6]]
dimensions =  2
```

```
import numpy as np

arr = np.array([[[1, 2, 3], [4, 5, 6]], [[11, 22, 33], [44, 55, 66]]])
a=arr.ndim
b=arr.size
print(arr)
print("dimensions = ", a)
print("size = ", b)
```

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

 [[11 22 33]
  [44 55 66]]]
dimensions =  3
size =  12
```

```
import numpy as np

a = np.array(42)
b = np.array([1, 2, 3, 4, 5])
c = np.array([[1, 2, 3], [4, 5, 6]])
d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])

print(a)
print("dimension= ", a.ndim)
print(b)
print("dimension= ", b.ndim)
print(c)
print("dimension= ", c.ndim)
print(d)
print("dimension= ", d.ndim)
```

```
42
dimension=  0
[1 2 3 4 5]
dimension=  1
[[1 2 3]
 [4 5 6]]
dimension=  2
[[[1 2 3]
  [4 5 6]]

 [[1 2 3]
  [4 5 6]]]
dimension=  3
```

```
import numpy as np

arr = np.array([[1, 2, 3, 4], [5, 6, 7, 8]])

print(arr.shape)
```

```
(2, 4)
```

```
a = np.arange(6)
print(a)

b = a.reshape(3,2)
print(b)
```

```
[0 1 2 3 4 5]
[[0 1]
 [2 3]
 [4 5]]
```

```
import numpy as np
a=np.array(9)
print("Print A= ",a)
b=np.array([10,20,30])
print("Print B= ", b)
addition= np.add(a,b)
print("After Addition =", addition)
sub=np.subtract(b,a)
print("After Subtraction =",sub)
mul=np.multiply (a,b)
print("After Multiplcation =",mul)
div=np.divide (b,a)
print("After Division =",div)
div1=np.divide (b,5)
print("After Division =",div1)
```

```
Print A=  9
Print B=  [10 20 30]
After Addition = [19 29 39]
After Subtraction = [ 1 11 21]
After Multiplcation = [ 90 180 270]
After Division = [1.11111111 2.22222222 3.33333333]
After Division = [2. 4. 6.]
```

```
#Reciprocal

a= np.array([0.25,1.33,1,111])
print(a)
rec=np.reciprocal(a)
print(rec)
```

```
[  0.25   1.33   1.   111.  ]
[4.         0.7518797  1.         0.00900901]
```

```
a=np.array([10,100,1000])
print(a)
pow=np.power(a,2)
print("after ^2 = ", pow)
b=np.array([2,3,1])
pow1=np.power(a,b)
print("after b array elements as ^ = ", pow1)
```

```
[  10  100 1000]
after ^2 =  [    100   10000 1000000]
after b array elements as ^ =  [    100 1000000    1000]
```

```
a= np.array ([10,20,30])
b= np.array ([3,5,7])
print("values of A=", a)
print("values of B=", b)
mm=np.mod(a,b)
rm=np.remainder(a,b)
print("values of MOD=", mm)
print("values of REMAINDER=", rm)
```

```
values of A= [10 20 30]
values of B= [3 5 7]
values of MOD= [1 0 2]
values of REMAINDER= [1 0 2]
```

```
a=np.array([-5.6j, 0.2j,11, 1+1j])
print(a)
print("real=",np.real(a))
print("imaginary=",np.imag(a))
print("Conjugate=", np.conj(a))
```

```
[-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 ]
```

```
a=np.array ([1.0,5.3,123,0.56,25.5,32])
print(a)
print("after rounding up=", np.around(a))
print("after rounding up to 1st decimal value =", np.around(a,decimals=1))
```

```
[  1.     5.3  123.     0.56  25.5   32.  ]
after rounding up= [  1.   5. 123.   1.  26.  32.]
after rounding up to 1st decimal value = [  1.    5.3 123.    0.6  25.5  32. ]
```

```
a=np.array ([1.0,-5.3,123,-0.56,25.5,32])
print(a)
print('\n')
print(np.floor(a))
print(np.ceil(a))
```

```
[  1.    -5.3  123.    -0.56  25.5   32.  ]


[  1.  -6. 123.  -1.  25.  32.]
[  1.  -5. 123.  -0.  26.  32.]
```