In [1]:
```
# install numpy module
```

In [2]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\nishi\anaconda3\lib\site-pac
kages (1.24.3)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
pip list
```

```
websocket-client            0.58.0
Werkzeug                    2.2.3
whatthepatch                1.0.2
wheel                       0.38.4
widgetsnbextension          4.0.5
win-inet-pton               1.1.0
wrapt                       1.14.1
xarray                      2023.6.0
xlwings                     0.29.1
xxhash                      2.0.2
xyzservices                 2022.9.0
y-py                        0.5.9
yapf                        0.31.0
yarl                        1.8.1
ypy-websocket               0.8.2
zict                        2.2.0
zipp                        3.11.0
zope.interface              5.4.0
zstandard                   0.19.0
Note: you may need to restart the kernel to use updated packages.
```

In [4]:
```
pip show numpy
```

```
Name: numpy
Version: 1.24.3
Summary: Fundamental package for array computing in Python
Home-page: https://www.numpy.org (https://www.numpy.org)
Author: Travis E. Oliphant et al.
Author-email:
License: BSD-3-Clause
Location: C:\Users\nishi\anaconda3\Lib\site-packages
Requires:
Required-by: astropy, bokeh, Bottleneck, contourpy, daal4py, datasets, datash
ader, datashape, gensim, h5py, holoviews, hvplot, imagecodecs, imageio, imbal
anced-learn, matplotlib, mkl-fft, mkl-random, numba, numexpr, pandas, patsy,
pyarrow, pyerfa, PyWavelets, scikit-image, scikit-learn, scipy, seaborn, stat
smodels, tables, tifffile, transformers, xarray
Note: you may need to restart the kernel to use updated packages.
```

In [5]:
```
#Take the array of elements in python
import sys
lst1=[10,20,30,40]
print(lst1,type(lst1))
print("Total Memory of list object",sys.getsizeof(lst1))
```

```
[10, 20, 30, 40] <class 'list'>
Total Memory of list object 88
```

In [6]:
```
import numpy as np
print(np.__version__)
```

```
1.24.3
```

In [7]:
```
a=np.array(lst1)
print(a,type(a))
print("Total Memory of Array object",sys.getsizeof(a))
```

```
[10 20 30 40] <class 'numpy.ndarray'>
Total Memory of Array object 128
```

In [8]:
```
lst1=[10,20]
print(lst1,type(lst1))
print("Total Memory of list object",sys.getsizeof(lst1))
```

```
[10, 20] <class 'list'>
Total Memory of list object 72
```

In [9]:
```
a=np.array(lst1)
print(a,type(a))
print("Total Memory of Array object",sys.getsizeof(a))
```

```
[10 20] <class 'numpy.ndarray'>
Total Memory of Array object 120
```

In [10]:
```
lst1=[10,20,30,40,50,60,70,80,90]
print(lst1,type(lst1))
print("Total Memory of list object",sys.getsizeof(lst1))
```

```
[10, 20, 30, 40, 50, 60, 70, 80, 90] <class 'list'>
Total Memory of list object 136
```

In [11]:
```
a=np.array(lst1)
print(a,type(a))
print("Total Memory of Array object",sys.getsizeof(a))
```

```
[10 20 30 40 50 60 70 80 90] <class 'numpy.ndarray'>
Total Memory of Array object 148
```

In [12]:
```
lst1=[10,20,30,40,50,60,70,80,90,100,200,300,400,500,600,700,800,900]
print(lst1,type(lst1))
print("Total Memory of list object",sys.getsizeof(lst1))
```

```
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800,
900] <class 'list'>
Total Memory of list object 200
```

In [13]:
```
a=np.array(lst1)
print(a,type(a))
print("Total Memory of Array object",sys.getsizeof(a))
```

```
[ 10  20  30  40  50  60  70  80  90 100 200 300 400 500 600 700 800 900] <cl
ass 'numpy.ndarray'>
Total Memory of Array object 184
```

In [14]:
```
lst1=[10,20,30]
print(lst1,type(lst1))
```

```
[10, 20, 30] <class 'list'>
```

In [15]:
```
lst1=lst1+1 # vector based operations on list not possible
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[15], line 1
----> 1 lst1=lst1+1

TypeError: can only concatenate list (not "int") to list
```

In [16]:
```
#convert lst1 into ndarray object
a=np.array(lst1)
print(a,type(a))
```

```
[10 20 30] <class 'numpy.ndarray'>
```

In [17]:
```
a=a+1 # vector based operations are possible on ndarray object
a
```

Out[17]:
```
array([11, 21, 31])
```

In [18]:
```
# data retrieval from list object
lst1=[10,20,30,40,50,60,70,80,90]
print(lst1,type(lst1),id(lst1))
```

```
[10, 20, 30, 40, 50, 60, 70, 80, 90] <class 'list'> 2188386534336
```

In [19]:
```
for val in lst1:
    print(val,id(val),id(lst1))
```

```
10 140716838786120 2188386534336
20 140716838786440 2188386534336
30 140716838786760 2188386534336
40 140716838787080 2188386534336
50 140716838787400 2188386534336
60 140716838787720 2188386534336
70 140716838788040 2188386534336
80 140716838788360 2188386534336
90 140716838788680 2188386534336
```

In [21]:
```
# Data retrieval from ndarray object
lst1=[10,20,30,40,50,60,70,80,90]
a=np.array(lst1)
print(a,type(a),id(a))
```

```
[10 20 30 40 50 60 70 80 90] <class 'numpy.ndarray'> 2188381512752
```

In [22]:
```
for val in a:
    print(val,id(val),id(a))
```

```
10 2188400212432 2188381512752
20 2188400211600 2188381512752
30 2188400212720 2188381512752
40 2188400212432 2188381512752
50 2188400211600 2188381512752
60 2188400212720 2188381512752
70 2188400212432 2188381512752
80 2188400211600 2188381512752
90 2188400212720 2188381512752
```

In [23]:
```
lst1=[10,20,30,40,50,60,70,80,90]
print(lst1,type(lst1),id(lst1))
```

```
[10, 20, 30, 40, 50, 60, 70, 80, 90] <class 'list'> 2188381875264
```

In [24]:
```
lst1.reshape(3,3)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[24], line 1
----> 1 lst1.reshape(3,3)

AttributeError: 'list' object has no attribute 'reshape'
```

In [25]:
```
lst1=[10,20,30,40,50,60,70,80,90]
print(lst1,type(lst1))
a=np.array(lst1)
print(a,type(a))
```

```
[10, 20, 30, 40, 50, 60, 70, 80, 90] <class 'list'>
[10 20 30 40 50 60 70 80 90] <class 'numpy.ndarray'>
```

In [26]:
```
a.reshape(3,3)
```

Out[26]:
```
array([[10, 20, 30],
       [40, 50, 60],
       [70, 80, 90]])
```

In [27]:
```
lst1=[10,20,30,40,50,60,70,80,90,15,25,35]
print(lst1,type(lst1))
a=np.array(lst1)
print(a,type(a))
```

```
[10, 20, 30, 40, 50, 60, 70, 80, 90, 15, 25, 35] <class 'list'>
[10 20 30 40 50 60 70 80 90 15 25 35] <class 'numpy.ndarray'>
```

In [28]:
```
a.reshape(4,3)
```

Out[28]:
```
array([[10, 20, 30],
       [40, 50, 60],
       [70, 80, 90],
       [15, 25, 35]])
```

In [29]:
```
a.reshape(3,4)
```

Out[29]:
```
array([[10, 20, 30, 40],
       [50, 60, 70, 80],
       [90, 15, 25, 35]])
```

In [30]:
```
a.reshape(6,2)
```

Out[30]:
```
array([[10, 20],
       [30, 40],
       [50, 60],
       [70, 80],
       [90, 15],
       [25, 35]])
```

In [31]:
```
a.reshape(2,6)
```

Out[31]:
```
array([[10, 20, 30, 40, 50, 60],
       [70, 80, 90, 15, 25, 35]])
```

In [32]:
```
a.reshape(2,3,2)
```

Out[32]:
```
array([[[10, 20],
        [30, 40],
        [50, 60]],

       [[70, 80],
        [90, 15],
        [25, 35]]])
```

In [33]:
```
a.reshape(2,2,3)
```

Out[33]:
```
array([[[10, 20, 30],
        [40, 50, 60]],

       [[70, 80, 90],
        [15, 25, 35]]])
```

In [34]:
```
a.reshape(4,4) #cannot reshape array of size 12 into shape (4,4)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[34], line 1
----> 1 a.reshape(4,4)

ValueError: cannot reshape array of size 12 into shape (4,4)
```

In [35]:
```
print(a,type(a))
```

```
[10 20 30 40 50 60 70 80 90 15 25 35] <class 'numpy.ndarray'>
```

In [36]:
```
a.ndim
```

Out[36]:
```
1
```

In [37]:
```
a.shape
```

Out[37]:
```
(12,)
```

In [38]:
```
a.shape(3,4)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[38], line 1
----> 1 a.shape(3,4)

TypeError: 'tuple' object is not callable
```

In [39]:
```
a.shape=(3,4)
```

In [40]:
```
a
```

Out[40]:
```
array([[10, 20, 30, 40],
       [50, 60, 70, 80],
       [90, 15, 25, 35]])
```

In [41]:
```
a.ndim
```

Out[41]:
```
2
```

In [42]:
```
a.shape
```

Out[42]:
```
(3, 4)
```

In [43]:
```
a.shape(2,3,2)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[43], line 1
----> 1 a.shape(2,3,2)

TypeError: 'tuple' object is not callable
```

In [44]:
```
a.shape=(2,3,2)
```

In [45]:
```
a
```

Out[45]:
```
array([[[10, 20],
        [30, 40],
        [50, 60]],

       [[70, 80],
        [90, 15],
        [25, 35]]])
```

In [46]:
```
a.ndim
```

Out[46]:
```
3
```

In [47]:
```
a.shape
```

Out[47]:
```
(2, 3, 2)
```

In [48]:
```
lst=[10,"Rossum",23.45,True]
print(lst,type(lst))
```

```
[10, 'Rossum', 23.45, True] <class 'list'>
```

In [49]:
```
a=np.array(lst)
print(a,type(a))
```

```
['10' 'Rossum' '23.45' 'True'] <class 'numpy.ndarray'>
```

In [50]:
```
print(lst,type(lst))
```

```
[10, 'Rossum', 23.45, True] <class 'list'>
```

In [51]:
```
a=np.array(lst,object)
print(a,type(a))
```

```
[10 'Rossum' 23.45 True] <class 'numpy.ndarray'>
```

In [52]:
```
a.dtype
```

Out[52]:
```
dtype('O')
```

In [53]:
```
print(a.dtype)
```

```
object
```

In [54]:
```
a.shape=(2,2)
```

In [55]:
```
a
```

Out[55]:
```
array([[10, 'Rossum'],
       [23.45, True]], dtype=object)
```

In [56]:
```
lst=[10,"Rossum",23.45,True]
print(lst,type(lst))
```

```
[10, 'Rossum', 23.45, True] <class 'list'>
```

In [57]:
```
lst[0]=100
```

In [58]:
```
print(lst,type(lst))
```

```
[100, 'Rossum', 23.45, True] <class 'list'>
```

In [59]:
```
a=np.array(lst)
print(a,type(a))
```

```
['100' 'Rossum' '23.45' 'True'] <class 'numpy.ndarray'>
```

In [60]:
```
l1=[[10,20,30],[40,50,60],[70,80,90]]
l2=[[1,2,3],[4,5,6],[7,8,9]]
print(l1,type(l1))
print(l2,type(l2))
```

```
[[10, 20, 30], [40, 50, 60], [70, 80, 90]] <class 'list'>
[[1, 2, 3], [4, 5, 6], [7, 8, 9]] <class 'list'>
```

In [61]:
```
a=np.array(l1)
b=np.array(l2)
print(a,type(a))
print(b,type(b))
```

```
[[10 20 30]
 [40 50 60]
 [70 80 90]] <class 'numpy.ndarray'>
[[1 2 3]
 [4 5 6]
 [7 8 9]] <class 'numpy.ndarray'>
```

In [62]:
```
c=a+b
print(c,type(c))
```

```
[[11 22 33]
 [44 55 66]
 [77 88 99]] <class 'numpy.ndarray'>
```

In [ ]: