In [ ]:

```
# Maha Ebrahim mohammed al juhdali
# 4051350
# IA8G
# lab 4 : Data Preparation-NumPy Library
```

In [1]:

```
import numpy as np
a = np.arange(15).reshape(3,5)
a
```

Out[1]:

```
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14]])
```

In [3]:

```
import numpy as np
print(np.array([8,4,6,0,2]))
```

```
[8 4 6 0 2]
```

In [4]:

```
print('Create a 2-D array by passing a list of lists into array().')
A = np.array( [ [1, 2, 3],[4, 5, 6] ] )
print(A)

print('Access elements of the array with brackets.')
print(A[0, 1], A[1, 2])

print('The elements of a 2-D array are 1-D arrays.')
A[0]
```

```
Create a 2-D array by passing a list of lists into array().
[[1 2 3]
 [4 5 6]]
Access elements of the array with brackets.
2 6
The elements of a 2-D array are 1-D arrays.
```

Out[4]:

```
array([1, 2, 3])
```

In [13]:

```
a = np.array([[3, -1, 4],[1, 5, -9]])
print(a)
b=np.array([[2, 4, -5, 6],[-1, 7, 9, 3],[3, 2, -7, -2]])
print(b)
np.dot(a,b)
```

```
[[ 3 -1  4]
 [ 1  5 -9]]
[[ 2  4 -5  6]
 [-1  7  9  3]
 [ 3  2 -7 -2]]
```

Out[13]:

```
array([[ 19,  13, -52,   7],
       [-30,  21, 103,  39]])
```

In [5]:

```
print('Addition concatenates lists together.')
print([1, 2, 3] + [4, 5, 6])

print('Mutliplication concatenates a list with itself a given number of times.')
print([1, 2, 3] * 4)
```

```
Addition concatenates lists together.
[1, 2, 3, 4, 5, 6]
Mutliplication concatenates a list with itself a given number of times.
[1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]
```

In [29]:

```
x=np.array([3, -4, 1])
print(x)
y=np.array([5, 2, 3])
print(y)
print(x + 10)
print(y * 4)
print(x + y)
print(x * y)
```

```
[ 3 -4  1]
[5 2 3]
[13  6 11]
[20  8 12]
[ 8 -2  4]
[15 -8  3]
```

In [16]:

```
A= np.array([[1, 2, 3],[4, 5, 6]])
print(A)
print(A.ndim)
print(A.shape)
print(A.size)
print(A.dtype)
```

```
[[1 2 3]
 [4 5 6]]
2
(2, 3)
6
int32
```

In [17]:

```
A= np.array([[1, 2, 3],[4, 5, 6]])
print(A)
print(A.dtype)
#Change Atype to float64
b=A.astype('float64')
print(b)
print(b.dtype)
```

```
[[1 2 3]
 [4 5 6]]
int32
[[1. 2. 3.]
 [4. 5. 6.]]
float64
```

In [18]:

```
x = np.arange(10)
print(x)
print(x[3])
print(x[:4])
print(x[4:])
print(x[4:8])
```

```
[0 1 2 3 4 5 6 7 8 9]
3
[0 1 2 3]
[4 5 6 7 8 9]
[4 5 6 7]
```

In [24]:

```
A = np.array([[0,1,2,3,4],[5,6,7,8,9]])
print(A)
print(A[1, 2])
print( A[:, 2:])
```

```
[[0 1 2 3 4]
 [5 6 7 8 9]]
7
[[2 3 4]
 [7 8 9]]
```

In [6]:

```
x = np.arange(0, 50, 10)
index = np.array([3, 1, 4])
print(x[index])

# A boolean array extracts the elements of 'x' at the same places as 'True'.
mask = np.array([True, False, False, True, False])
x[mask]
```

```
[30 10 40]
```

Out[6]:

```
array([ 0, 30])
```

In [7]:

```
y = np.arange(10, 20, 2)
print(y)
mask = y > 15
print(mask)
print(y[mask])

# Change the values of 'y' that are larger than 15 to 100.
y[mask] = 100
print(y)
```

```
[10 12 14 16 18]
[False False False  True  True]
[16 18]
[ 10  12  14 100 100]
```

In [31]:

```
from sklearn import datasets

iris = datasets.load_iris()
print(iris.filename)
```

```
C:\anaconda3\lib\site-packages\sklearn\datasets\data\iris.csv
```

In [39]:

```
import numpy as np
iris_data = np.genfromtxt('C:\\anaconda3\\lib\\site-packages\\sklearn\\datasets\\data\\iris
                          delimiter=",", skip_header=1)
print(iris_data)
```

```
[[5.1 3.5 1.4 0.2 0. ]
 [4.9 3.  1.4 0.2 0. ]
 [4.7 3.2 1.3 0.2 0. ]
 [4.6 3.1 1.5 0.2 0. ]
 [5.  3.6 1.4 0.2 0. ]
 [5.4 3.9 1.7 0.4 0. ]
 [4.6 3.4 1.4 0.3 0. ]
 [5.  3.4 1.5 0.2 0. ]
 [4.4 2.9 1.4 0.2 0. ]
 [4.9 3.1 1.5 0.1 0. ]
 [5.4 3.7 1.5 0.2 0. ]
 [4.8 3.4 1.6 0.2 0. ]
 [4.8 3.  1.4 0.1 0. ]
 [4.3 3.  1.1 0.1 0. ]
 [5.8 4.  1.2 0.2 0. ]
 [5.7 4.4 1.5 0.4 0. ]
 [5.4 3.9 1.3 0.4 0. ]
 [5.1 3.5 1.4 0.3 0. ]
 [5.7 3.8 1.7 0.3 0. ]
```

In [49]:

```
import numpy as np
iris_data = np.genfromtxt('C:\\anaconda3\\lib\\site-packages\\sklearn\\datasets\\data\\iris
                          delimiter=",", skip_header=1)
print("mean of {} is {}".format(iris.feature_names[0], iris_data[:,0].mean()))
print("standard deviation of {} is {}".format(iris.feature_names[0], iris_data[:,0].std()))
print("variance of {} is {}".format(iris.feature_names[0], iris_data[:,0].var()))
print("maximum element of {} is {}".format(iris.feature_names[0], iris_data[:,0].max()))
print("minimum element of {} is {}".format(iris.feature_names[0], iris_data[:,0].min()))
print("sum of {} is {}".format(iris.feature_names[0], iris_data[:,0].sum()))
```

```
mean of sepal length (cm) is 5.843333333333334
standard deviation of sepal length (cm) is 0.8253012917851409
variance of sepal length (cm) is 0.6811222222222223
maximum element of sepal length (cm) is 7.9
minimum element of sepal length (cm) is 4.3
sum of sepal length (cm) is 876.5
```

In [ ]:

```

```