

```
In [1]: # Aut@r: Susana Edith Barrientos Galicia
# Introduction Numpy

# import the numpy package as np
import numpy as np

# create 2 new list height and weight

personHeight = [5.2, 5.4, 4.4, 4.5, 5.6, 6]
personWeight = [81, 55, 70, 45, 44]

# create 2 numpy arrays from height and weight
personHeight = np.array(personHeight)
personWeight = np.array(personWeight)
```

```
In [2]: # print 'personHeight' array
```

```
print(personHeight)
print(personWeight)
print(type(personWeight))
```

```
[5.2 5.4 4.4 4.5 5.6 6. ]
[81 55 70 45 44]
<class 'numpy.ndarray'>
```

```
In [3]: # Aut@r: Susana Edith Barrientos Galicia
# Test Introduction Numpy
```

```
import numpy as np
array=np.array([4,9,16,25])
type(array)
```

```
Out[3]: numpy.ndarray
```

```
In [4]: # Aut@r: Susana Edith Barrientos Galicia
# Indexing an Array
```

```
# Indexing in 1 dimension

# given array
myArray = np.array([11,22,33,24,57,473])
print(myArray)
```

```
[ 11  22  33  24  57 473]
```

```
In [5]: # Aut@r: Susana Edith Barrientos Galicia
```

```
# get the 1st element
# indexing starts from '0'
print(myArray[0])
print(myArray[2])
```

```
11
```

```
33
```

```
In [6]: # Aut@r: Susana Edith Barrientos Galicia
# Indexing an Array

# Indexing in 2 dimension

array = np.array([
                    [101, 231, 321],
                    [412, 512, 622],
                    [712, 821, 912]
                ])
array
```

```
Out[6]: array([[101, 231, 321],
               [412, 512, 622],
               [712, 821, 912]])
```

```
In [7]: # get the element in 3rd row and 2nd column
print(array[2][1])
```

```
821
```

```
In [16]: # we can pass ith row and jth column in seeparate brackets ([])
print(array[2][1])
```

```
821
```

```
In [8]: # pick the second row from the array
print(array[2])
```

```
[712 821 912]
```

```
In [9]: # pick the secon column from the array
print(array[:,2])
```

```
[321 622 912]
```

```
In [10]: # Aut@r: Susana Edith Barrientos Galicia
# Test Indexing an Array

sample_array = np.array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20])

sample_array[[0, 4, 7]]
```

```
Out[10]: array([10, 14, 17])
```

```
In [11]: # Aut@r: Susana Edith Barrientos Galicia
# Test Indexing an Array

a = np.array([[1, 0, 1, 0, 2, 3], [1, 3, 0, 1, 2, 0], [0, 1, 0, 0, 1, 3]])

a[1, 2]
a[2, 1]
```

```
Out[11]: 1
```

```
In [12]: # Aut@r: Susana Edith Barrientos Galicia
# Slicing an Array
```

```
myArray = [101, 121, 112, 123, 114]
newArray = myArray[1:4]
newArray
```

```
Out[12]: [121, 112, 123]
```

```
In [13]: # first three elements
c = myArray[:3]
c
```

```
Out[13]: [101, 121, 112]
```

```
In [14]: # all the elements from 112
d = myArray[2:]
d
```

```
Out[14]: [112, 123, 114]
```

```
In [15]: # Aut@r: Susana Edith Barrientos Galicia
# Slicing an Array
# 2D array
```

```
array2 = np.array([
    [101, 131, 122, 113, 143],
    [145, 165, 137, 318, 193],
    [240, 241, 252, 253, 324],
    [225, 126, 727, 928, 129]
])
```

```
print(array2[1:,2:4])
```

```
[[137 318]
 [252 253]
 [727 928]]
```

```
In [16]: sample_array = np.array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20])
sample_array [0:4]
```

```
Out[16]: array([10, 11, 12, 13])
```

```
In [17]: import numpy as np
a = np.array([[1,0,1,0,2,3], [1,3,0,1,2,0], [0,1,0,0,1,3]])
b = a[:,1:3]
print(b)
```

```
[[0 1]
 [3 0]
 [1 0]]
```

```
In [18]: # Aut@r: Susana Edith Barrientos Galicia
# Operation on a Array
```

```
arrayOne = np.array([120, 230, 310, 410, 150])
arrayTwo = np.arange(5)
```

```
In [19]: # add the two arrays
arrayThree = arrayOne + arrayTwo
arrayThree
```

```
Out[19]: array([120, 231, 312, 413, 154])
```

```
In [20]: # multiply each elements in the array y 4
arrayOne*4
```

```
Out[20]: array([ 480,  920, 1240, 1640,  600])
```

```
In [43]: # get square of each elements
arrayOne**2
```

```
Out[43]: array([ 14400,  52900,  96100, 168100,  22500])
```

```
In [21]: # using numpy with comparison expressions
myArray = np.array([34, 45, 67, 45, 23])

# check with elements are greater than or equal to 40
# the comparison condition gives boolean output

newArray = myArray >= 40
newArray
```

```
Out[21]: array([False,  True,  True,  True, False])
```

```
In [22]: # elements greater than or equal to 40
myArray[newArray]
```

```
Out[22]: array([45, 67, 45])
```

```
In [23]: myArray
```

```
Out[23]: array([34, 45, 67, 45, 23])
```

```
In [24]: myArray[myArray >= 40]
```

```
Out[24]: array([45, 67, 45])
```

```
In [25]: # Aut@r: Susana Edith Barrientos Galicia
# Test Operation on a Array

import numpy as np
array1 = np.array([[0, 3, 2, 5], [1, 0, 2, -2]])
print(array1 >= 3)
```

```
[[False  True False  True]
 [False False False False]]
```

```
In [26]: # Aut@r: Susana Edith Barrientos Galicia
# Test Operation on a Array
```

```
import numpy as np
array2 = np.array([[1, 3, 5], [2, 4, 6]])
print((array2*2)**2)
```

```
[[ 4  36 100]
 [ 16  64 144]]
```

```
In [27]: # Aut@r: Susana Edith Barrientos Galicia
# Arithmetic Functioning in Numpy
```

```
# given array
array4 = np.array([5,7,8,2,4])
array4
```

```
Out[27]: array([5, 7, 8, 2, 4])
```

```
In [28]: # sum
# add all the elements of array
array4.sum()
```

```
Out[28]: 26
```

```
In [29]: # find minimum of array
array4.min()
```

```
Out[29]: 2
```

```
In [30]: # get cue of elemens of array
np.power(array4,3)
```

```
Out[30]: array([125, 343, 512,  8,  64])
```

```
In [31]: # Aut@r: Susana Edith Barrientos Galicia
# Test Arithmetic Functioning in Numpy
```

```
import numpy as np
a = np.array([0, 1, 2])
b = np.array([5, 5, 5])
c = a + b
c*5
```

```
Out[31]: array([25, 30, 35])
```

```
In [32]: # Aut@r: Susana Edith Barrientos Galicia
# Test Arithmetic Functioning in Numpy
```

```
import numpy as np
a = np.arange(10)
b = np.power(a,2)
b.min()
```

Out[32]: 0

```
In [33]: # Aut@r: Susana Edith Barrientos Galicia
# Concatenation of an Array

# conccatenate two 1D array

arrayX = np.array([11, 22, 13])
arrayY = np.array([23, 22, 12])
np.concatenate([arrayX, arrayY])

# You can also concatenate more than two arrays at once.

arrayZ = np.array([23,45])
print(np.concatenate([arrayX, arrayY, arrayZ]))

[11 22 13 23 22 12 23 45]
```

```
In [34]: # Aut@r: Susana Edith Barrientos Galicia
# Concatenation of an Array

# conccatenate two 2D array

array5 = np.array([
                [1,2,3],
                [4,5,6]
            ])

array5
```

Out[34]: array([[1, 2, 3],  
[4, 5, 6]])

```
In [35]: # by default conccateenate() is along 'axis = 0'
np.concatenate([array5, array5])
```

Out[35]: array([[1, 2, 3],  
[4, 5, 6],  
[1, 2, 3],  
[4, 5, 6]])

```
In [36]: # concatenate along the second axis
np.concatenate([array5, array5], axis=1)
```

Out[36]: array([[1, 2, 3, 1, 2, 3],  
[4, 5, 6, 4, 5, 6]])

```
In [37]: # Aut@r: Susana Edith Barrientos Galicia
# Test Concatenation of an Array

import numpy as np

x = np.array([[3], [5], [7]])
y = np.array([[5], [7], [9]])

np.concatenate([x, y], axis=1)
```

```
Out[37]: array([[3, 5],  
               [5, 7],  
               [7, 9]])
```

```
In [39]: # Aut@r: Susana Edith Barrientos Galicia  
         # Splitting of an Array
```

```
import numpy as np  
a = np.arange(24).reshape(4,6)  
np.vsplit(a,1)
```

```
Out[39]: [array([[ 0,  1,  2,  3,  4,  5],  
                [ 6,  7,  8,  9, 10, 11],  
                [12, 13, 14, 15, 16, 17],  
                [18, 19, 20, 21, 22, 23]])]
```

```
In [ ]:
```