In [1]:

```
import numpy as np
```

In [4]:

```
#Creating sample array
arr = np.arange(1,11)
```

In [5]:

```
#Show
arr
```

Out[5]:

```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

In [6]:

```
#Get a value at an index
arr[8]
```

Out[6]:

```
9
```

In [7]:

```
#Get values in a range
arr[1:5]
```

Out[7]:

```
array([2, 3, 4, 5])
```

In [8]:

```
#Get values in a range
arr[0:5]
```

Out[8]:

```
array([1, 2, 3, 4, 5])
```

In [9]:

```
#Setting a value with index range (Broadcasting)
arr[0:5]=100

#Show
arr
```

Out[9]:

```
array([100, 100, 100, 100, 100,   6,   7,   8,   9,  10])
```

In [10]:

```
# Reset array, we'll see why i had to reset in  a moment
arr = np.arange(0,11)

#Show
arr
```

Out[10]:

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

In [11]:

```
#Important notes on Slices
slice_of_arr = arr[0:6]

#Show slice
slice_of_arr
```

Out[11]:

```
array([0, 1, 2, 3, 4, 5])
```

In [12]:

```
#Change Slice
slice_of_arr[:]=99

#Show Slice again
slice_of_arr
```

Out[12]:

```
array([99, 99, 99, 99, 99, 99])
```

In [13]:

```
# Now note the changes also occur in our original array!
arr
# Data is not copied, it's a view of the original array! This avoids memory problems!
```

Out[13]:

```
array([99, 99, 99, 99, 99, 99,  6,  7,  8,  9, 10])
```

In [14]:

```
#To get a copy, need to be explicit
arr_copy = arr.copy()

arr_copy
```

Out[14]:

```
array([99, 99, 99, 99, 99, 99,  6,  7,  8,  9, 10])
```

In [18]:

```
arr_copy[:2] = 1
arr_copy
```

Out[18]:

```
array([ 1,  1, 99, 99, 99, 99,  6,  7,  8,  9, 10])
```

In [16]:

```
arr
```

Out[16]:

```
array([99, 99, 99, 99, 99, 99,  6,  7,  8,  9, 10])
```

In [19]:

```
# Indexing a 2D array
arr_2d = np.array(([5,10,15],[20,25,30],[35,40,45]))

#Show
arr_2d
```

Out[19]:

```
array([[ 5, 10, 15],
       [20, 25, 30],
       [35, 40, 45]])
```

In [24]:

```
#Indexing row
arr_2d[1]
```

Out[24]:

```
array([20, 25, 30])
```

In [25]:

```
# Format is arr_2d[row][col] or arr_2d[row,col]

# Getting individual element value
arr_2d[1][0]
```

Out[25]:

```
20
```

In [27]:

```
# Getting individual element value
# trả về phần tử ở hàng 0, cột 2
arr_2d[0,2]
```

Out[27]:

```
15
```

In [28]:

```
# 2D array slicing

#Shape (2,2) from top right corner
# hang :2 lay tu dau den vi tri 2
#cot 1: lay tu vitri 1 den het
arr_2d[:2,  1:]
```

Out[28]:

```
array([[10, 15],
       [25, 30]])
```

In [29]:

```
#Shape bottom row
arr_2d[2]
```

Out[29]:

```
array([35, 40, 45])
```

In [32]:

```
#Shape bottom row
arr_2d[2,:2]
```

Out[32]:

```
array([35, 40])
```

In [42]:

```
# Fancy Indexing
#Set up matrix
arr2d = np.zeros((10,10))
arr2d
```

Out[42]:

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

In [43]:

```
#Length of array
arr_length = arr2d.shape[1]
arr_length
```

Out[43]:

```
10
```

In [44]:

```
#Set up array
for i in range(arr_length):
    arr2d[i] = i

arr2d
```

Out[44]:

```
array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [1., 1., 1., 1., 1., 1., 1., 1., 1., 1.],
       [2., 2., 2., 2., 2., 2., 2., 2., 2., 2.],
       [3., 3., 3., 3., 3., 3., 3., 3., 3., 3.],
       [4., 4., 4., 4., 4., 4., 4., 4., 4., 4.],
       [5., 5., 5., 5., 5., 5., 5., 5., 5., 5.],
       [6., 6., 6., 6., 6., 6., 6., 6., 6., 6.],
       [7., 7., 7., 7., 7., 7., 7., 7., 7., 7.],
       [8., 8., 8., 8., 8., 8., 8., 8., 8., 8.],
       [9., 9., 9., 9., 9., 9., 9., 9., 9., 9.]])
```

In [53]:

```
#Fancy indexing allows the following
arr2d[[2,4,6,8]]
```

Out[53]:

```
array([[2., 2., 2., 2.],
       [4., 4., 4., 4.],
       [6., 6., 6., 6.],
       [8., 8., 8., 8.]])
```

In [52]:

```
#Allows in any order
arr2d[[6,4,2,7]]
```

Out[52]:

```
array([[6., 6., 6., 6., 6., 6., 6., 6., 6., 6.],
       [4., 4., 4., 4., 4., 4., 4., 4., 4., 4.],
       [2., 2., 2., 2., 2., 2., 2., 2., 2., 2.],
       [7., 7., 7., 7., 7., 7., 7., 7., 7., 7.]])
```

In [ ]:

```

```