Slicing

In [51]:

```
import numpy as np
```

In [4]:

```
#1D
arr1d = np.array([44,55,66,77])
print(arr1d)
print(arr1d[0])
print(arr1d[3])
print(arr1d[-1])
print(arr1d[-4])
print(arr1d[2:4])

```

```
[44 55 66 77]
44
77
77
44
[66 77]
```

In [5]:

```
#2D
arr2d = np.array([[22,33,44],[55,66,77]])
print(arr2d)
print(arr2d[0])
print(arr2d[1])
print(arr2d[0][1])
print(arr2d[1][1])
print(arr2d[1][-2])
print(arr2d[0][1:3])
print(arr2d[1][-1:-1])
print(arr2d[0:,2:])  # here after commma spacing
print(arr2d[1:,2:])
print(arr2d[0:,1:2]) #(rowstart:end, colomn start:end, spacing)
```

```
[[22 33 44]
 [55 66 77]]
[22 33 44]
[55 66 77]
33
66
66
[33 44]
[]
[[44]
 [77]]
[[77]]
[[33]
 [66]]
```

In [23]:

```
print(arr2d[0:,1:2])
print(arr2d[0:3:2,0:3:2])
```

```
[[33]
 [66]]
[[22 44]]
```

In [24]:

```
##another method :this is not slicing
arr2d[1,2]  # 2d has 2 rows one row is 22 33 44 and seond row is 55 66 77.1,2 means 1
#remember indexing starts from 0

```

Out[24]:

```
77
```

In [36]:

```
print(arr2d)
print(arr2d[:2])
print(arr2d[0:2,1:3]) # :2,1:
print(arr2d[:2,1:2])  # showing 2 D
print(arr2d[:2,1])    #showing 1 D
print(arr2d[1:,1:])
```

```
[[22 33 44]
 [55 66 77]]
[[22 33 44]
 [55 66 77]]
[[33 44]
 [66 77]]
[[33]
 [66]]
[33 66]
[[66 77]]
```

In [33]:

```
# 3D

arr3d = np.array([[[10,20,30],[40,50,60],[70,80,90]]])  # 3d having 3 1d arrays ..pora
print(arr3d)
print(arr3d[0])
print(arr3d[0][0])
print(arr3d[0][1])
print(arr3d[0][2])
print(arr3d[0][0][0])
print(arr3d[0][0][1])
print(arr3d[0][0][2])
print(arr3d[0][0][0:2])
```

```
[[[10 20 30]
  [40 50 60]
  [70 80 90]]]
[[10 20 30]
 [40 50 60]
 [70 80 90]]
[10 20 30]
[40 50 60]
[70 80 90]
10
20
30
[10 20]
```

In [68]:

```
print(arr3d[0][0:2])
print(arr3d[0][0:3])  #becomes 2d
print(arr3d[:2])
print(arr3d[0,2:])  # not slicing  1st 2 d then 2:
print(arr3d[0][2])  #ans 1 d but same
print(arr3d[0,:2])
print(arr3d[0,:2,1:2])
```

```
[[10 20 30]
 [40 50 60]]
[[10 20 30]
 [40 50 60]
 [70 80 90]]
[[[10 20 30]
  [40 50 60]
  [70 80 90]]]
[[70 80 90]]
[70 80 90]
[[10 20 30]
 [40 50 60]]
[[20]
 [50]]
```

In [61]:

```
print(arr3d[0,2:])
```

```
[[70 80 90]]
```

In [62]:

```
print(arr3d[0,2:,1:2])  # 0 means 1st 2D then 2: means 2nd means 3rd acc to index 1 d
```

```
[[80]]
```

In [63]:

```
print(arr3d[0,:2,1:2])
```

```
[[20]
 [50]]
```

In [64]:

```
print(arr3d)
```

```
[[[10 20 30]
  [40 50 60]
  [70 80 90]]]
```

In [77]:

```
print(arr3d.ndim)
print(arr3d.shape) # 1 2D array having 3 rows n 3 columns mmeans 3 axis
```

```
3
(1, 3, 3)
```

In [72]:

```
tdarray = np.arange(36).reshape(6,6)
tdarray
```

Out[72]:

```
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],
       [24, 25, 26, 27, 28, 29],
       [30, 31, 32, 33, 34, 35]])
```

In [73]:

```
#to identify 20 thsi is indexing 3rd row n 2nd column satrts from 0
tdarray[3,2]
```

Out[73]:

```
20
```

In [74]:

```
#slicing 13 14 15 19 20 21 26 27 28 (3x3 matrix)
#tdarray[row,colomn]
#tdarray[start:end, start:end]
tdarray[2:5, 1:4] # end at 5 means 4  tk 5 not included
```

Out[74]:

```
array([[13, 14, 15],
       [19, 20, 21],
       [25, 26, 27]])
```

In [75]:

```
#33 usind slicing
tdarray[5:,3:4] # 33 aega but value nhi he 2 dimentional array aega
```

Out[75]:

```
array([[33]])
```

In [76]:

```
tdarray[1:2,2:5]
```

Out[76]:

```
array([[ 8,  9, 10]])
```

In [78]:

```
#2D
```

In [80]:

```
arr2d = np.array([[2,4,6,8,10],[1,2,3,4,5],[4,0,6,2,5],[7,5,1,6,5],[8,2,4,9,7]])
arr2d
```

Out[80]:

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

In [81]:

```
arr2d[2:4,1:4]
```

Out[81]:

```
array([[0, 6, 2],
       [5, 1, 6]])
```

In [93]:

```
print(arr2d[::2,2])
print(arr2d[:5:5])
print(arr2d[:5,:5:2])
```

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

In [95]:

```
print(arr2d[:4,:3:2])  # spacing always give colomns
```

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

*** 2 D = (row start: end, col start: end :spacing) ***

In [8]:

```
arr = np.arange(1,37).reshape(6,6)
arr

```

Out[8]:

```
array([[ 1,  2,  3,  4,  5,  6],
       [ 7,  8,  9, 10, 11, 12],
       [13, 14, 15, 16, 17, 18],
       [19, 20, 21, 22, 23, 24],
       [25, 26, 27, 28, 29, 30],
       [31, 32, 33, 34, 35, 36]])
```

In [99]:

```
arr[2:3,0:] #row,colom count with index that starts with zero
```

Out[99]:

```
array([[13, 14, 15, 16, 17, 18]])
```

In [100]:

```
arr[2:4,0:]
```

Out[100]:

```
array([[13, 14, 15, 16, 17, 18],
       [19, 20, 21, 22, 23, 24]])
```

In [101]:

```
arr[2:4,0:6]
```

Out[101]:

```
array([[13, 14, 15, 16, 17, 18],
       [19, 20, 21, 22, 23, 24]])
```

In [102]:

```
arr[4:5,:] # khali colon in colum means all colomn
```

Out[102]:

```
array([[25, 26, 27, 28, 29, 30]])
```

In [104]:

```
arr[4:5,]
```

Out[104]:

```
array([[25, 26, 27, 28, 29, 30]])
```

In [109]:

```
print(arr[4:5])
print(arr[4])
print(arr[:2])
print(arr[1:5,])
```

```
[[25 26 27 28 29 30]]
[25 26 27 28 29 30]
[[ 1  2  3  4  5  6]
 [ 7  8  9 10 11 12]]
[[ 7  8  9 10 11 12]
 [13 14 15 16 17 18]
 [19 20 21 22 23 24]
 [25 26 27 28 29 30]]
```

In [9]:

```
print(arr)
```

```
[[ 1  2  3  4  5  6]
 [ 7  8  9 10 11 12]
 [13 14 15 16 17 18]
 [19 20 21 22 23 24]
 [25 26 27 28 29 30]
 [31 32 33 34 35 36]]
```

In [112]:

```
print(arr[2:4,5])
print(arr[2:4,5:])
```

```
[18 24]
[[18]
 [24]]
```

In [113]:

```
print(arr[::-2])
```

```
[[31 32 33 34 35 36]
 [19 20 21 22 23 24]
 [ 7  8  9 10 11 12]]
```

In [114]:

```
print(arr[:,:-2]) #not showing last 2 col
```

```
[[ 1  2  3  4]
 [ 7  8  9 10]
 [13 14 15 16]
 [19 20 21 22]
 [25 26 27 28]
 [31 32 33 34]]
```

In [115]:

```
print(arr[:,:2])
```

```
[[ 1  2]
 [ 7  8]
 [13 14]
 [19 20]
 [25 26]
 [31 32]]
```

In [117]:

```
arr[:,::2] col spacing 2
```

Out[117]:

```
array([[ 1,  3,  5],
       [ 7,  9, 11],
       [13, 15, 17],
       [19, 21, 23],
       [25, 27, 29],
       [31, 33, 35]])
```

In [118]:

```
arr[:,::-2] col spacing reverse
```

Out[118]:

```
array([[ 6,  4,  2],
       [12, 10,  8],
       [18, 16, 14],
       [24, 22, 20],
       [30, 28, 26],
       [36, 34, 32]])
```

In [121]:

```
# 1 n 3
arr[::2,:3:2]
```

Out[121]:

```
array([[ 1,  3],
       [13, 15],
       [25, 27]])
```

In [10]:

```
arr[::2,0:]
```

Out[10]:

```
array([[ 1,  2,  3,  4,  5,  6],
       [13, 14, 15, 16, 17, 18],
       [25, 26, 27, 28, 29, 30]])
```

In [11]:

```
arr[0:4:4,0:]
```

Out[11]:

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

In [18]:

```
arr[0:1,0:3:2]
```

Out[18]:

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

In [19]:

```
arr[0:4:5,0:3:2]
```

Out[19]:

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

In [13]:

```
arr
```

Out[13]:

```
array([[ 1,  2,  3,  4,  5,  6],
       [ 7,  8,  9, 10, 11, 12],
       [13, 14, 15, 16, 17, 18],
       [19, 20, 21, 22, 23, 24],
       [25, 26, 27, 28, 29, 30],
       [31, 32, 33, 34, 35, 36]])
```

In [29]:

```
# 3D having 2 2Ds
arr3d2d = np.array([[[1,2,3,4],[5,6,7,8]],
                    [[20,30,40,50],[60,70,80,90]]])
arr3d2d
```

Out[29]:

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

       [[20, 30, 40, 50],
        [60, 70, 80, 90]]])
```

In [30]:

```
arr3d2d.ndim
```

Out[30]:

```
3
```

In [31]:

```
arr3d2d.shape # having 2 2ds havind 2 rows n 4 columns
```

Out[31]:

```
(2, 2, 4)
```

In [34]:

```
arr3d # having
```

Out[34]:

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

In [35]:

```
arr3d.shape
```

Out[35]:

```
(1, 3, 3)
```

In [36]:

```
arr3d2d
```

Out[36]:

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

       [[20, 30, 40, 50],
        [60, 70, 80, 90]]])
```

In [37]:

```
arr3d2d[0]
```

Out[37]:

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

In [38]:

```
arr3d2d[1]
```

Out[38]:

```
array([[20, 30, 40, 50],
       [60, 70, 80, 90]])
```

In [39]:

```
arr3d2d[0][0]
```

Out[39]:

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

In [40]:

```
arr3d2d[0][0][1]
```

Out[40]:

```
2
```

In [48]:

```
arr3d2d[0,0:2,:2]
```

Out[48]:

```
array([[1, 2],
       [5, 6]])
```

In [49]:

```
arr3d2d[0,0:2,2]
```

Out[49]:

```
array([3, 7])
```

In [50]:

```
arr3d2d[1,0:2,::2]
```

Out[50]:

```
array([[20, 40],
       [60, 80]])
```

In [54]:

```
d3 = np.arange(40).reshape(2,4,5) # 5*4 = 20 * 2 = 40
d3
```

Out[54]:

```
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, 24],
        [25, 26, 27, 28, 29],
        [30, 31, 32, 33, 34],
        [35, 36, 37, 38, 39]]])
```

In [55]:

```
d3.shape
```

Out[55]:

```
(2, 4, 5)
```

In [56]:

```
#d3[0,3,2]
d3[0][3][1:4]

```

Out[56]:

```
array([16, 17, 18])
```

In [57]:

```
d3[1,:,1:2]
```

Out[57]:

```
array([[21],
       [26],
       [31],
       [36]])
```

In [ ]:

```

```