NumPy Array Slicing

May 18, 2022

1 1-D Array Slicing

[2]:
```
import numpy as np

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

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

[5]:
```
b = a[2:5] #arrayName[begin:end-1:step]
```

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

[6]:
```
c = a[2:7:2] #arrayName[begin:end-1:step=2 = every second element]
c
```

[6]:
```
array([3, 5, 7])
```

[8]:
```
d = a[:] #all elements
d
```

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

[9]:
```
e = a[:6] #between very 1st and last-1 elements
e
```

[9]:
```
array([1, 2, 3, 4, 5, 6])
```

[24]:
```
e = a[2:] #between 2nd and very last elements
e
```

[24]:
```
array([3, 4, 5, 6, 7, 8, 9])
```

[11]:
```
f = a[5:2:-1] # minus indexing applicable
f
```

[11]:
```
array([6, 5, 4])
```

[23]:
```
# if we don't put the step then bydefault it will always go left to right␣
 ↪otherwise empty array will show
g = a[-4:-2]
g
```

[23]:
```
array([6, 7])
```

[22]:
```
# if we only put the step in negetive index then it will go right to left␣
 ↪otherwise empty array will show

a = np.array([1,2,3,4,5,6,7,8,9])

h = a[-2:-4:-1]
h
```

[22]:
```
array([8, 7])
```

2 2-D Array Slicing

[25]:
```
ar = np.array([[10,20,30],[40,50,60],[70,80,90]])
ar
```

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

[26]:
```
ar[:,::2] #arrayName[begin:end:step,begin:end:step] -> row, column
```

[26]:
```
array([[10, 30],
       [40, 60],
       [70, 90]])
```

3 3-D Array Slicing

[27]:
```
arr = np.
 ↪array([[[10,20,30],[40,50,60],[70,80,90]],[[100,110,10],[120,130,11],[140,150,12]]])
arr
```

[27]:
```
array([[[ 10,  20,  30],
        [ 40,  50,  60],
        [ 70,  80,  90]],

       [[100, 110,  10],
        [120, 130,  11],
        [140, 150,  12]]])
```

[28]:
```
arr[:,:,2:] #(i,j,k) i = which 2D array, j = which row, k, which column ,,␣
↪arrayName[begin:end:step,begin:end:step,beign:end, step]
```

[28]:
```
array([[[30],
        [60],
        [90]],

       [[10],
        [11],
        [12]]])
```