

```

1 import numpy as np
2 arr4=np.random.randint(0,100,(2,5,5))
3 arr4

```

*Handwritten notes: "x 5 number of array size" and "stop" with an arrow pointing to line 2.*

```

array([[[34, 87, 1, 38, 53],
        [68, 65, 20, 89, 90],
        [ 9, 29, 78, 15, 72],
        [51, 44, 36, 83, 1],
        [12, 20, 18, 54, 40]],
       [[ 7, 94, 23, 53, 25],
        [58, 68, 79, 68, 56],
        [50, 47, 88, 77, 16],
        [42, 18, 28, 9, 99],
        [ 9, 94, 32, 93, 49]]])

```

*Handwritten circled numbers 1 and 2 next to the first and second sub-arrays respectively.*

```

1 arr5=arr4
2 arr5[0,0,1]=1997
3 arr5

```

*Handwritten annotations: a red circle around the value 1997 in the output, and a red circle with an equals sign next to line 2.*

```

array([[[ 34, 1997, 1, 38, 53],
        [ 68, 65, 20, 89, 90],
        [ 9, 29, 78, 15, 72],
        [ 51, 44, 36, 83, 1],
        [ 12, 20, 18, 54, 40]],
       [[ 7, 94, 23, 53, 25],
        [ 58, 68, 79, 68, 56],
        [ 50, 47, 88, 77, 16],
        [ 42, 18, 28, 9, 99],
        [ 9, 94, 32, 93, 49]]])

```

Saved successfully!



```
2 arr5
```

```

array([[[ 34, 1997, 1, 38, 53],
        [ 68, 65, 1976, 89, 90],
        [ 9, 29, 78, 15, 72],
        [ 51, 44, 36, 83, 1],
        [ 12, 20, 18, 54, 40]],
       [[ 7, 94, 23, 53, 25],
        [ 58, 68, 79, 68, 56],
        [ 50, 47, 88, 77, 16],
        [ 42, 18, 28, 9, 99],
        [ 9, 94, 32, 93, 49]]])

```

```
1 arr4 # note = updates original array also update
```

*Handwritten annotations: red circles around 'arr4' and '1997' in the output.*

```

array([[[ 34, 1997, 1, 38, 53],
        [ 68, 65, 1976, 89, 90],
        [ 9, 29, 78, 15, 72],
        [ 51, 44, 36, 83, 1],
        [ 12, 20, 18, 54, 40]],
       [[ 7, 94, 23, 53, 25],
        [ 58, 68, 79, 68, 56],
        [ 50, 47, 88, 77, 16],
        [ 42, 18, 28, 9, 99],
        [ 9, 94, 32, 93, 49]]])

```

```
[[ 7, 94, 23, 53, 25],
 [ 58, 68, 79, 68, 56],
 [ 50, 47, 88, 77, 16],
 [ 42, 18, 28, 9, 99],
 [ 9, 94, 32, 93, 49]]])
```

```
1 arr6=np.copy(arr4)
2 arr6
```

```
array([[[ 34, 1997, 1, 38, 53],
 [ 68, 65, 1976, 89, 90],
 [ 9, 29, 78, 15, 72],
 [ 51, 44, 36, 83, 1],
 [ 12, 20, 18, 54, 40]],
 [[ 7, 94, 23, 53, 25],
 [ 58, 68, 79, 68, 56],
 [ 50, 47, 88, 77, 16],
 [ 42, 18, 28, 9, 99],
 [ 9, 94, 32, 93, 49]]])
```

*changes both*

*copy = only is one*

```
1 arr6[0,0,0] = 1961
2 arr6
```

```
array([[[1961, 1997, 1, 38, 53],
 [ 68, 65, 1976, 89, 90],
 [ 9, 29, 78, 15, 72],
 [ 51, 44, 36, 83, 1],
 [ 12, 20, 18, 54, 40]],
 [[ 7, 94, 23, 53, 25],
 [ 58, 68, 79, 68, 56],
 [ 50, 47, 88, 77, 16],
 [ 42, 18, 28, 9, 99],
 [ 9, 94, 32, 93, 49]]])
```

Saved successfully!

```
1 arr4 #note arr4 not changes
```

```
array([[[ 34, 1997, 1, 38, 53],
 [ 68, 65, 1976, 89, 90],
 [ 9, 29, 78, 15, 72],
 [ 51, 44, 36, 83, 1],
 [ 12, 20, 18, 54, 40]],
 [[ 7, 94, 23, 53, 25],
 [ 58, 68, 79, 68, 56],
 [ 50, 47, 88, 77, 16],
 [ 42, 18, 28, 9, 99],
 [ 9, 94, 32, 93, 49]]])
```

```
1 t1=(0,1,4)
2 print(t1,type(t1))
3 arr6[t1]
```

*arr6* (0, 1, 4) <class 'tuple'>

90

```
1 l1=[0,1,0]
2 l2=[[1],[2],[3]]
3 print(arr6[l1])
```

Ask  
doubt

```
[[[1961 1997 1 38 53]
 [ 68 65 1976 89 90]
 [ 9 29 78 15 72]
 [ 51 44 36 83 1]
 [ 12 20 18 54 40]]]
```

```
[[ 7 94 23 53 25]
 [ 58 68 79 68 56]
 [ 50 47 88 77 16]
 [ 42 18 28 9 99]
 [ 9 94 32 93 49]]]
```

```
[[1961 1997 1 38 53]
 [ 68 65 1976 89 90]
 [ 9 29 78 15 72]
 [ 51 44 36 83 1]
 [ 12 20 18 54 40]]]
```

[[1], [2], [3]]

Duplicates  
①

```
1 print(arr6[l2])
```

```
[77]
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Using
""Entry point for launching an IPython kernel.
```

```
1 ones = np.ones((3,3))
```

Saved successfully!

3 ones

```
[[1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]]
[[3. 3. 3.]
 [3. 3. 3.]
 [3. 3. 3.]]
```

double  
ones  
double

```
1 sum = ones+double #similarly -
2 sum
```

```
array([[4., 4., 4.],
 [4., 4., 4.],
 [4., 4., 4.]])
```

```
1 sum = ones*double #similarly /
2 sum
```

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

[3., 3., 3.]])

```
1 exp = np.exp(sum)
2 exp
```

*(3)*  
*(3)*  
 array([[20.08553692, 20.08553692, 20.08553692],  
 [20.08553692, 20.08553692, 20.08553692],  
 [20.08553692, 20.08553692, 20.08553692]])

```
1 sinsum = np.sin(sum)
2 sinsum
```

array([[0.14112001, 0.14112001, 0.14112001],  
 [0.14112001, 0.14112001, 0.14112001],  
 [0.14112001, 0.14112001, 0.14112001]])

```
1 sum = sum + 100
2 sum
```

array([[103., 103., 103.],  
 [103., 103., 103.],  
 [103., 103., 103.]])

```
1 #broadcasting
2 a1=np.array([1,2,3,4,5])
3 a2=np.array([9,9])
4 a1+a2
```

*can't do sum*  
*val err*  
*(0,1,5)*  
*(0,1,5)*

-----  
 ValueError Traceback (most recent call last)  
 <ipython-input-36-39432b302aa5> in <module>()  
 4 a1+a2

Saved successfully! X

ValueError: operands could not be broadcast together with shapes (5,) (2,)

SEARCH STACK OVERFLOW

```
1 a1=np.array([[1,2,3,4,5],[1,2,3,4,5],[1,2,3,4,5]])
2 a2=np.array([9,9,9,9,9])
3 a1+a2
```

array([[10, 11, 12, 13, 14],  
 [10, 11, 12, 13, 14],  
 [10, 11, 12, 13, 14]])

*broad cast*

*(0,3,5)*  
*(0,1,5)*

```
1 a1=np.array([[1,2,3,4,5],[1,2,3,4,5],[1,2,3,4,5]])
2 a2=np.array([9,9,9,9,9]+[9,9,9,9,9])
3 a1+a2
```

*X broad cast*  
*(0,3,5)*  
*(0,2,5)*

```

-----
TypeError                                Traceback (most recent call last)
<ipython-input-42-4c1659d5ccc8> in <module>()
      1 a1=np.array([[1,2,3,4,5],[1,2,3,4,5],[1,2,3,4,5]])
----> 2 a2=np.array([9,9,9,9,9]+0)
      3 a1+a2

```

**TypeError:** can only concatenate list (not "int") to list

```

1 a1=np.array([[1,2,3,4,5],[1,2,3,4,5]])
2 a2=np.array([9,9])
3 a1+a2

```

0,1,5  
0,1,2

```

-----
ValueError                                Traceback (most recent call last)
<ipython-input-43-c15d830cc24c> in <module>()
      1 a1=np.array([[1,2,3,4,5],[1,2,3,4,5]])
      2 a2=np.array([9,9])
----> 3 a1+a2

```

**ValueError:** operands could not be broadcast together with shapes (2,5) (2,)

SEARCH STACK OVERFLOW

```

1 a1=np.array([[1,2,3,4,5],[1,2,3,4,5]])
2 a2=np.array([1]) # it treats like[[1,1,1,1,1],[1,1,1,1,1]]
3 a1+a2

```

0,0,1  
0,2,8

```

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

```

a1=np.array([1,2,3,4,5])  
a2=np.array([1,2,3,4,5]) # it  
treats  
like[[1,1,1,1,1],[1,1,1,1,1]]  
a1+a2 # in all possible atleast  
one dimension match

Saved successfully!

×

```

3 a1+a2
4 # in all possible atleast one dimension match , or if each dimension 000

```

```

array([ 2,  4,  6,  8, 10])

```

0,1,5  
0,1,8

```

1 a1=np.arange(4)
2 a1

```

```

array([0, 1, 2, 3])

```

```

1 a2=np.reshape(a2,4)
2 a2

```

```

array([0, 1, 2, 3])

```

```

1 arr1=np.array([[1,2,3],[4,5,6],[7,8,9]])
2 arr2=np.array([1,2,3])
3 #022
4 #002
5 print(arr1,arr2)

```

```
6 arr3=arr1+arr2
7 print(arr3)
```

arr1

1	2	3
4	5	6
7	8	9

arr2

1	2	3
---	---	---

arr1 + 2 =

2	4	6
5	7	9
8	10	12

3, 2, 3, 2

0	3	3
0	1	3

Saved successfully!

Completed at 1:19 PM

