Colab: https://colab.research.google.com/drive/1-xBKiV-ZdLJaEUSHWlsvvGDAM6HERWYk?usp=sharing

```
import numpy as np
```

```
x = np.arange(9)
x
```

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

```
np.split(x, 3)
```

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

```
np.split(x, [3, 5, 6])
```

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

```
x = np.arange(16).reshape(4, 4)
x
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15]])
```

```
np.split(x, 2, axis=1)
```

```
[array([[ 0,  1],
        [ 4,  5],
        [ 8,  9],
        [12, 13]]), array([[ 2,  3],
        [ 6,  7],
        [10, 11],
        [14, 15]])]
```

```
np.split(x, 2, axis=0)
```

```
[array([[0, 1, 2, 3],
        [4, 5, 6, 7]]), array([[ 8,  9, 10, 11],
        [12, 13, 14, 15]])]
```

```
np.hsplit(x, 2)
```

```
[array([[ 0,  1],
        [ 4,  5],
        [ 8,  9],
        [12, 13]]), array([[ 2,  3],
        [ 6,  7],
        [10, 11],
        [14, 15]])]
```

```
np.vsplit(x, 2)
```

```
[array([[0, 1, 2, 3],
        [4, 5, 6, 7]]), array([[ 8,  9, 10, 11],
        [12, 13, 14, 15]])]
```

```
data = np.arange(5)
data
```

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

```
np.vstack((data, data, data))
```

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

```
data = np.arange(5).reshape(5, 1)
data
```

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

```
np.hstack((data, data, data))

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

```
a = np.array([[1], [2], [3]])
b = np.array([[4], [5], [6]])
np.hstack((a, b))

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

```
z = np.array([[2, 4]])
z

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

```
np.concatenate((z,z), axis=0)

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

```
np.concatenate((z,z), axis=1)

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

```
# why not np.stack()?
# another function in numpy np.stack, but it something different?
# np.stack is different np.concatenate
```

Broadcasting

```
 [[0,   0,  0],      [[0, 1, 2],
  [10, 10, 10], and   [0, 1, 2],
  [20, 20, 20],       [0, 1, 2],
  [30, 30, 30]]       [0, 1, 2]]
```

```
data = np.array([0, 10, 20, 30]).reshape(4,1)
np.hstack((data, data, data))

    array([[ 0,  0,  0],
           [10, 10, 10],
           [20, 20, 20],
           [30, 30, 30]])
```

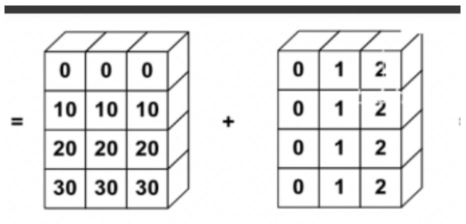
```
np.tile(np.array([0, 10, 20, 30]), (3,2))

    array([[ 0, 10, 20, 30,  0, 10, 20, 30],
           [ 0, 10, 20, 30,  0, 10, 20, 30],
           [ 0, 10, 20, 30,  0, 10, 20, 30]])
```

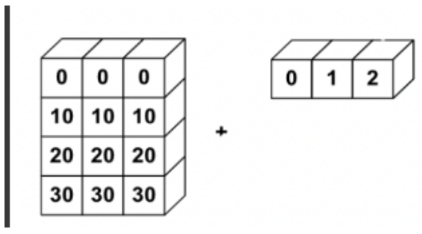
```
a = np.tile(np.arange(0,40,10), (3,1)).T
a

    array([[ 0,  0,  0],
           [10, 10, 10],
           [20, 20, 20],
           [30, 30, 30]])
```

```
b = np.tile(np.arange(3), (4, 1))
b

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

```
a + b
```

```
    array([[ 0,  1,  2],
           [10, 11, 12],
           [20, 21, 22],
           [30, 31, 32]])
```





```
a
```

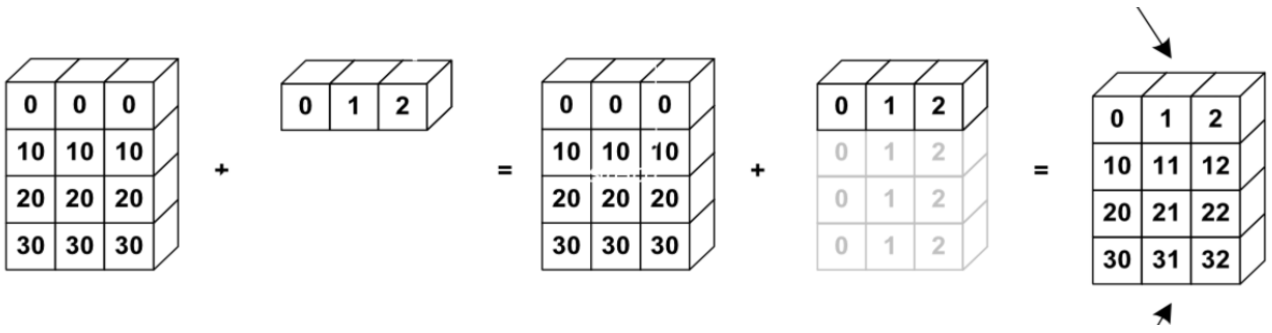
```
    array([[ 0,  0,  0],
           [10, 10, 10],
           [20, 20, 20],
           [30, 30, 30]])
```

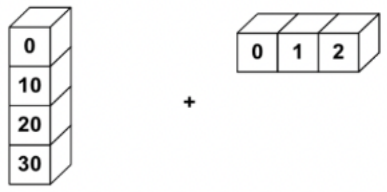
```
b = np.arange(0,3)
b
```

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

```
a + b
```

```
    array([[ 0,  1,  2],
           [10, 11, 12],
           [20, 21, 22],
           [30, 31, 32]])
```





```
a = np.arange(0, 40, 10).reshape(4, 1)
a
```

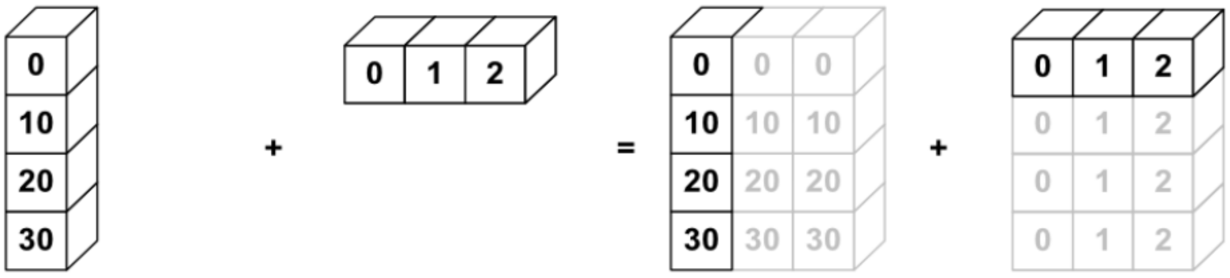
```
    array([[ 0],
           [10],
           [20],
           [30]])
```

```
b = np.arange(3)
b
```

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

```
a + b
```

```
    array([[ 0,  1,  2],
           [10, 11, 12],
           [20, 21, 22],
           [30, 31, 32]])
```



```
a = np.arange(8).reshape(2,4)
b = np.arange(16).reshape(4,4)

print(a*b)
```

```
    ---------------------------------------------------------------------------
    ValueError                                Traceback (most recent call last)
    <ipython-input-43-2e8364a6b7d1> in <module>
          2 b = np.arange(16).reshape(4,4)
          3
    ----> 4 print(a*b)

    ValueError: operands could not be broadcast together with shapes (2,4) (4,4)
```

SEARCH STACK OVERFLOW

```
 For each dimension ( going from right side)
     1. The size of each dimension should be same OR
     2. The size of one dimension should be 1
```

```
import numpy as np
```

```
arr = np.arange(6)
arr
```

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

```
arr.shape
```

```
    (6,)
```

```
arr.reshape(1, -1).shape
```

```
    (1, 6)
```

```
np.expand_dims(arr, axis=0)
```

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

```
np.expand_dims(arr, axis=1)
```

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

```
arr[np.newaxis, :]
```

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

```
arr[:, np.newaxis]
```

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

```
arr = np.arange(9).reshape(1, 1, 9)
arr
```

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

```
arr.shape
```

```
    (1, 1, 9)
```

```
np.squeeze(arr)
```

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

```
np.squeeze(arr, axis=1).shape
```

```
    (1, 9)
```

```
a = np.arange(4)
a
```

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

```
b = a.reshape(2, 2)
b
```

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

```
a[0] = 100
```

```
a
```

```
    array([100,   1,   2,   3])
```

```
b
```

```
    array([[100,   1],
           [  2,   3]])
```

```
a = np.arange(4)
a
```

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

```
c = a + 2
c
```

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

```
np.shares_memory(a, c)
```

```
    False
```

```
# until now, Numpy has been taking decision for me
# whether to create a shallow or a deep copy
```

```
arr = np.arange(10)
arr
```

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

```
view_arr = arr.view()
```

```
np.shares_memory(arr, view_arr)
```

```
True
```

```
deep_copy = arr.copy()
```

```
np.shares_memory(arr, good )
```

```
False
```

✓ 0s completed at 21:43