In [2]:
```
import numpy as np

arr1 = np.array([1,2,3])
print(arr1)
```

```
[1 2 3]
```

In [3]:
```
print(type(arr1))
```

```
<class 'numpy.ndarray'>
```

In [4]:
```
print(arr1.shape)
```

```
(3,)
```

In [5]:
```
arr1[2]
```

Out[5]:
```
3
```

In [12]:
```
arr2 = np.array([[1,2,3],[3,4,5]])
arr2
```

Out[12]:
```
array([[1, 2, 3],
       [3, 4, 5]])
```

In [13]:
```
arr2[1,2]
```

Out[13]:
```
5
```

In [14]:
```
arr2.shape
```

Out[14]:
```
(2, 3)
```

In [15]:
```
arr2[1,2]
```

Out[15]:
```
5
```

In [16]:
```
arrR = np.arange(0,20,2)
arrR
```

Out[16]:
```
array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18])
```

In [17]:
```
arrL = np.linspace(0,10,20)
arrL
```

Out[17]:
```
array([ 0.        ,  0.52631579,  1.05263158,  1.57894737,  2.10526316,
        2.63157895,  3.15789474,  3.68421053,  4.21052632,  4.73684211,
        5.26315789,  5.78947368,  6.31578947,  6.84210526,  7.36842105,
        7.89473684,  8.42105263,  8.94736842,  9.47368421, 10.        ])
```

In [18]:
```
np.random.rand(10)
```

Out[18]:
```
array([0.32293416, 0.27645373, 0.478658  , 0.02845125, 0.46153056,
       0.23836405, 0.74327623, 0.48198542, 0.34010441, 0.42037102])
```

In [24]:
```
np.zeros(10)
np.zeros((2,3))
```

Out[24]:
```
array([[0., 0., 0.],
       [0., 0., 0.]])
```

In [26]:
```
np.ones((3,4,3))
```

Out[26]:
```
array([[[1., 1., 1.],
        [1., 1., 1.],
        [1., 1., 1.],
        [1., 1., 1.]],

       [[1., 1., 1.],
        [1., 1., 1.],
        [1., 1., 1.],
        [1., 1., 1.]],

       [[1., 1., 1.],
        [1., 1., 1.],
        [1., 1., 1.],
        [1., 1., 1.]]])
```

In [27]:
```
arr = [0,1,2]
print(np.repeat(arr, 3))

#repeats each element
```

```
[0 0 0 1 1 1 2 2 2]
```

In [29]:
```
print(np.tile(arr, 3))
```

```
[0 1 2 0 1 2 0 1 2]
```

In [31]:
```
np.eye(5)
```

Out[31]:
```
array([[1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1.]])
```

In [35]:
```
np.diag([1,2,3,4,5])
```

Out[35]:
```
array([[1, 0, 0, 0, 0],
       [0, 2, 0, 0, 0],
       [0, 0, 3, 0, 0],
       [0, 0, 0, 4, 0],
       [0, 0, 0, 0, 5]])
```

In [36]:
```
np.random.rand(5,5)
```

Out[36]:
```
array([[0.57328107, 0.88232737, 0.43546297, 0.40178088, 0.72342087],
       [0.22823096, 0.54875533, 0.90767305, 0.34650668, 0.2940847 ],
       [0.47594824, 0.67783127, 0.92279552, 0.0834686 , 0.47231927],
       [0.26159029, 0.98973384, 0.41148031, 0.42980154, 0.43071394],
       [0.31181657, 0.31301312, 0.3942165 , 0.92925098, 0.03214381]])
```

In [73]:
```
np.diag(arr)
```

Out[73]:
```
array([[0, 0, 0],
       [0, 1, 0],
       [0, 0, 2]])
```

In [76]:
```
arrr = np.array([[1,2,3],[3,4,5],[5,6,7]])
arrr
```

Out[76]:
```
array([[1, 2, 3],
       [3, 4, 5],
       [5, 6, 7]])
```

In [77]:
```
arrr.size
```

Out[77]:
```
9
```

In [78]:
```
arrr.ndim
```

Out[78]:
```
2
```

In [79]:
```
arrr.shape
```

Out[79]:
```
(3, 3)
```

In [80]:
```
np.random.randint(-10,10,4)
```

Out[80]:
```
array([ -8,  -5,   2, -10])
```

In [81]:
```
arrr+10
```

Out[81]:
```
array([[11, 12, 13],
       [13, 14, 15],
       [15, 16, 17]])
```

In [82]:
```
np.log10(arrr)
```

Out[82]:
```
array([[0.        , 0.30103   , 0.47712125],
       [0.47712125, 0.60205999, 0.69897   ],
       [0.69897   , 0.77815125, 0.84509804]])
```

In [83]:
```
np.cos(arrr)
```

Out[83]:
```
array([[ 0.54030231, -0.41614684, -0.9899925 ],
       [-0.9899925 , -0.65364362,  0.28366219],
       [ 0.28366219,  0.96017029,  0.75390225]])
```

In [84]:
```
np.tan(arrr)
```

Out[84]:
```
array([[ 1.55740772, -2.18503986, -0.14254654],
       [-0.14254654,  1.15782128, -3.38051501],
       [-3.38051501, -0.29100619,  0.87144798]])
```

In [85]:
```
np.sum(arrr, axis = 1)
#axis = 0 sums the columns and axis = 1 sums the rows
```

Out[85]:
```
array([ 6, 12, 18])
```

In [86]:
```
np.min(arrr)
# axis works here as well
#same works with np.max(arrr)
```

Out[86]:
```
1
```

In [87]:
```
np.mean(arrr, axis = 0)
```

Out[87]:
```
array([3., 4., 5.])
```

In [88]:
```
np.median(arrr)
```

Out[88]:
```
4.0
```

In [89]:
```
np.std(arrr)
```

Out[89]:
```
1.8257418583505538
```

In [90]:
```
np.var(arrr)
```

Out[90]:
```
3.3333333333333335
```

In [100]:
```
arrr[1:3,0:-1]
#columns 2 to 3, rows 1 to 2
```

Out[100]:
```
array([[3, 4],
       [5, 6]])
```

In [101]:
```
np.sort(arrr)
```

Out[101]:
```
array([[1, 2, 3],
       [3, 4, 5],
       [5, 6, 7]])
```

In [102]:
```
arrr.flatten()
```

Out[102]:
```
array([1, 2, 3, 3, 4, 5, 5, 6, 7])
```

In [104]:
```
arrr.transpose()
#flips the arr
```

Out[104]:
```
array([[1, 3, 5],
       [2, 4, 6],
       [3, 5, 7]])
```

In [109]:
```
arr = np.array([4,5,6,7])
```

In [110]:
```
arr1 = np.append(arr,8)
arr1
```

Out[110]:
```
array([4, 5, 6, 7, 8])
```

In [112]:
```
arr2 = np.insert(arr, 0, [1,2,3])
arr2
```

Out[112]:
```
array([1, 2, 3, 4, 5, 6, 7])
```

In [116]:
```
arr3 = np.delete(arr, [0,1]) # indexes
arr3
```

Out[116]:
```
array([6, 7])
```

In [124]:
```
arrC = arr3.copy()
arrC
```

Out[124]:
```
array([6, 7])
```

In [3]:
```
a = np.array([[1,2,3,4],[1,2,3,4]])
b = np.array([[5,6,7,8],[5,6,7,8]])
arr_cat = np.concatenate((a,b), axis = 1)
print(arr_cat)
```

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

In [6]:
```
catV = np.vstack((a,b))  #no axis mentioned. vstack is verticall stacking, hstack is horizontal
catV
```

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

In [139]:
```
np.unique(a)
```

Out[139]:
```
array([1, 2, 3, 4])
```

In [144]:
```
uniques, counts = np.unique(a, return_counts = True)
print(uniques)  #prints unique values
print(counts)   #prints how many of the values are there
```

```
[1 2 3 4]
[2 2 2 2]
```

In [149]:
```
print(arr1)
print(arr2)
np.intersect1d(arr1, arr2)
```

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

Out[149]:
```
array([4, 5, 6, 7])
```

In [150]:
```
np.union1d(arr1, arr2)
```

Out[150]:
```
array([1, 2, 3, 4, 5, 6, 7, 8])
```

In [151]:
```
np.setdiff1d(arr1, arr2)
```

Out[151]:
```
array([8])
```

In [152]:
```
np.setxor1d(arr1, arr2)
```

Out[152]:
```
array([1, 2, 3, 8])
```