**numpy.linspace(**start**,**stop**,**num=50)

Return evenly spaced numbers over a specified interval.

N = 8  
y = np.zeros(N)  
x1 = np.linspace(0, 10, N, endpoint=**True**)  
x2 = np.linspace(0, 10, N, endpoint=**False**)  
plt.plot(x1, y, "o")  
plt.plot(x2, y + 0.5, 'o')  
plt.ylim([-0.5, 1])  
  
plt.show()

Ex **>>>** np.linspace(2.0, 3.0, num=5)

array([ 2. , 2.25, 2.5 , 2.75, 3. ])

Return a new array of given shape and type, filled with zeros

np.zeros(5)

array([ 0., 0., 0., 0., 0.])

**>>>** np.zeros((2, 1))

array([[ 0.],

[ 0.]])

>>>

**>>>** s = (2,2)

**>>>** np.zeros(s)

array([[ 0., 0.],

[ 0., 0.]])

**numpy.zeros\_like(***a***,***dtype=None***,***order='K'***,***subok=True***)**[**[source]**](http://github.com/numpy/numpy/blob/v1.12.0/numpy/core/numeric.py#L87-L147)

Return an array of zeros with the same shape and type as a given array.

**>>>** x = np.arange(6)

**>>>** x = x.reshape((2, 3))

**>>>** x

array([[0, 1, 2],

[3, 4, 5]])

**>>>** np.zeros\_like(x)

array([[0, 0, 0],

[0, 0, 0]])