In [1]:
```
import sys
sys.version
```

Out[1]:
```
'3.12.7 | packaged by Anaconda, Inc. | (main, Oct  4 2024, 13:17:27) [MSC v.1929 6
4 bit (AMD64)]'
```

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

In [3]:
```
np.__version__
```

Out[3]:
```
'1.26.4'
```

In [5]:
```
# creat list
my_list = [0,1,2,3,4,5]
my_list
```

Out[5]:
```
[0, 1, 2, 3, 4, 5]
```

In [7]:
```
type(my_list)
```

Out[7]:
```
list
```

In [8]:
```
arr = np.array(my_list)
arr
```

Out[8]:
```
array([0, 1, 2, 3, 4, 5])
```

In [9]:
```
print(type(arr))
print(type(my_list))
```

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

In [10]:
```
np.arange(10)
```

Out[10]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [11]:
```
np.arange(10,20)
```

Out[11]:
```
array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

In [12]:
```
np.arange(10,50,5)
```

Out[12]:
```
array([10, 15, 20, 25, 30, 35, 40, 45])
```

In [13]:
```
np.arange(10,30,3)
```

Out[13]:
```
array([10, 13, 16, 19, 22, 25, 28])
```

In [14]:
```
np.arange(10,30,30, 3)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[14], line 1
----> 1 np.arange(10,30,30, 3)

TypeError: Cannot interpret '3' as a data type
```

In [15]:
```
np.arange(8,20)
```

Out[15]:
```
array([ 8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

In [16]:
```
np.arange(20,8)
```

Out[16]:
```
array([], dtype=int32)
```

In [17]:
```
np.arange(-20,8) #1st arg < 2nd arg
```

Out[17]:
```
array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11, -10,  -9,  -8,
        -7,  -6,  -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5,
         6,   7])
```

In [18]:
```
n = np.arange(-20,8)
n
```

Out[18]:
```
array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11, -10,  -9,  -8,
        -7,  -6,  -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5,
         6,   7])
```

In [19]:
```
np.zeros(3)
```

Out[19]:
```
array([0., 0., 0.])
```

In [20]:
```
np.zeros(3, dtype=int)
```

Out[20]:
```
array([0, 0, 0])
```

In [25]:
```
z = np.zeros(5)
z
```

Out[25]:
```
array([0., 0., 0., 0., 0.])
```

In [26]:
```
z = np.zeros(5, dtype = complex)
z
```

Out[26]:
```
array([0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j, 0.+0.j])
```

In [27]:
```
np.zeros((2,2)) # 2d array
```

Out[27]:
```
array([[0., 0.],
       [0., 0.]])
```

In [28]:
```
np.zeros((3,3), dtype = int)
```

Out[28]:
```
array([[0, 0, 0],
       [0, 0, 0],
       [0, 0, 0]])
```

In [29]:
```
nd = np.zeros((5,9), dtype = int)
nd
```

Out[29]:
```
array([[0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0]])
```

In [30]:
```
np.ones(3)
```

Out[30]:
```
array([1., 1., 1.])
```

In [31]:
```
np.ones(3, dtype=int)
```

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

In [32]:
```
nd1 = np.ones((10,10), dtype = int)
nd1
```

Out[32]:
```
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, 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, 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 [34]:
```
nd1 = np.zeros((5,9), dtype = int)
nd1
```

Out[34]:
```
array([[0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0]])
```

In [ ]: