30th April 2025

Numpy

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

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

Out[3]:
```
'1.26.1'
```

In [4]:
```
import sys
sys.version
```

Out[4]:
```
'3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:03:56) [MSC v.1929 6
4 bit (AMD64)]'
```

Creating a List

In [5]:
```
my_list = [0,1,2,3,4,5]
print(my_list)
```

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

In [6]:
```
print(type(my_list))
```

```
<class 'list'>
```

Creating an Array using List

In [9]:
```
np.array([my_list])
```

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

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

Out[10]:
```
array(10)
```

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

Out[11]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [13]:
```
np.arange(5)
```

Out[13]:
```
array([0, 1, 2, 3, 4])
```

In [14]:
```
np.arange(5.0)
```

```
Out[14]:  array([0., 1., 2., 3., 4.])
```

```
In [15]:  np.arange(10,20)
```

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

```
In [18]:  np.arange(20,10)  # first num < sec num
```

```
Out[18]:  array([], dtype=int32)
```

```
In [19]:  np.arange(-20,10)
```

```
Out[19]:  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,   8,   9])
```

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

```
Out[20]:  array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11])
```

```
In [23]:  ar =np.arange(30,20)   # 1st argg always < 2nd arg
          ar
```

```
Out[23]:  array([], dtype=int32)
```

```
In [24]:  np.arange(10,10)
```

```
Out[24]:  array([], dtype=int32)
```

```
In [25]:  np.arange()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[25], line 1
----> 1 np.arange()

TypeError: arange() requires stop to be specified.
```

```
In [26]:  np.arange(1,10,5)
```

```
Out[26]:  array([1, 6])
```

```
In [27]:  np.arange(10,30,5,8)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[27], line 1
----> 1 np.arange(10,30,5,8)

TypeError: Cannot interpret '8' as a data type
```

```
In [30]:  np.zeros(10)   # Parameter tuning
```

```
Out[30]:  array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

```
In [31]:  np.zeros(10, dtype=int)    # Hyperparameter tuning
```

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

```
In [32]:  np.zeros((2,2), dtype=int)
```

```
Out[32]:  array([[0, 0],
                 [0, 0]])
```

```
In [33]:  np.zeros((3,3), dtype=int)
```

```
Out[33]:  array([[0, 0, 0],
                 [0, 0, 0],
                 [0, 0, 0]])
```

```
In [34]:  np.array([my_list])
```

```
Out[34]:  array([[0, 1, 2, 3, 4, 5]])
```

```
In [35]:  np.zeros((4,4),dtype=int)
```

```
Out[35]:  array([[0, 0, 0, 0],
                 [0, 0, 0, 0],
                 [0, 0, 0, 0],
                 [0, 0, 0, 0]])
```

```
In [36]:  np.zeros((5,5),dtype=int)
```

```
Out[36]:  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]])
```

```
In [37]:  np.zeros((2,10))
```

```
Out[37]:  array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
                 [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]])
```

```
In [38]:  np.zeros((10,30),dtype=int)
```

```
Out[38]: 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, 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],
                [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, 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],
                [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, 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],
                [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 [39]: np.zeros((10,10), dtype=int)
```

```
Out[39]: 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, 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],
                [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]])
```

ONE Functions

```
In [40]: np.ones(3)
```

```
Out[40]: array([1., 1., 1.])
```

```
In [41]: np.ones(3, dtype=int)
```

```
Out[41]: array([1, 1, 1])
```

```
In [42]: np.ones((4,4))
```

```
Out[42]: array([[1., 1., 1., 1.],
                [1., 1., 1., 1.],
                [1., 1., 1., 1.],
                [1., 1., 1., 1.]])
```

```
In [43]: np.ones((3,3))
```

```
Out[43]: array([[1., 1., 1.],
                [1., 1., 1.],
                [1., 1., 1.]])
```

3 RAND Functions

```
In [44]: np.ones(4)

Out[44]: array([1., 1., 1., 1.])

In [45]: np.ones(4,dtype=int)

Out[45]: array([1, 1, 1, 1])

In [48]: np.random.rand(3,2)

Out[48]: array([[0.53215214, 0.55113967],
                [0.01919176, 0.44031336],
                [0.56235325, 0.55182636]])

In [49]: np.random.rand(5)

Out[49]: array([0.24715308, 0.64198416, 0.02278619, 0.6869118 , 0.86693751])

In [52]: np.random.rand(3,5)

Out[52]: array([[0.84762806, 0.44238312, 0.52780289, 0.58573994, 0.93903461],
                [0.81864088, 0.62356401, 0.77740526, 0.45181751, 0.30339386],
                [0.53287994, 0.32679012, 0.58395963, 0.19686035, 0.61915998]])

In [55]: np.random.randint(5)

Out[55]: 3

In [58]: np.random.randint(2,5)

Out[58]: 2

In [59]: np.random.randint(5,2)    # 1st arg < 2nd arg
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[59], line 1
----> 1 np.random.randint(5,2)

File numpy\\random\\mtrand.pyx:779, in numpy.random.mtrand.RandomState.randint()

File numpy\\random\\_bounded_integers.pyx:2885, in numpy.random._bounded_integers._r
and_int32()

ValueError: low >= high
```

```
In [60]: np.random.randint(5,14)
```

```
Out[60]: 11
```

```
In [61]: np.random.randint(5,20,4,5)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[61], line 1
----> 1 np.random.randint(5,20,4,5)

File numpy\\random\\mtrand.pyx:760, in numpy.random.mtrand.RandomState.randint()

TypeError: Cannot interpret '5' as a data type
```

```
In [62]: np.random.randint(5,15,3)
```

```
Out[62]: array([ 7, 14,  9])
```

```
In [64]: np.random.randint(5,10,(10,10))
```

```
Out[64]: array([[6, 6, 5, 8, 8, 6, 9, 6, 7, 9],
                [8, 7, 6, 6, 6, 5, 6, 9, 8, 7],
                [9, 8, 6, 9, 7, 7, 5, 7, 5, 7],
                [9, 5, 5, 5, 6, 9, 8, 5, 5, 5],
                [6, 7, 5, 8, 5, 8, 9, 8, 9, 8],
                [7, 9, 8, 5, 9, 9, 7, 7, 7, 8],
                [6, 9, 8, 7, 6, 5, 5, 7, 6, 8],
                [9, 5, 8, 9, 9, 9, 8, 8, 8, 8],
                [9, 5, 5, 6, 9, 8, 9, 7, 7, 8],
                [9, 8, 8, 7, 9, 8, 5, 5, 7, 8]])
```

```
In [65]: np.random.randint(1,100,(12,12))
```

```
Out[65]: array([[18, 50, 53, 17, 84,  7, 79, 56, 56, 75, 79, 47],
                [47, 34, 13, 44, 81, 27, 35, 60, 50, 37, 80, 78],
                [77, 73, 66, 96, 22, 82, 93, 10, 31, 45, 38, 56],
                [45, 88, 35, 23, 25, 16, 72, 97, 97, 69,  5, 37],
                [35, 90, 47, 67, 91, 11, 61, 75, 50, 57, 64, 79],
                [89, 30, 91, 61, 12, 18, 44, 69, 96, 22, 54, 34],
                [37,  5, 87, 57, 14, 88, 50, 49, 78, 96, 76, 91],
                [24, 13, 51, 69, 75, 52, 87,  4, 43, 50, 92, 35],
                [65, 88, 96, 79, 61, 81, 86, 92, 50, 91, 48, 91],
                [ 9, 72, 13, 99, 51, 57, 29, 65, 53, 85,  4, 33],
                [19, 24, 46, 18, 49, 88, 28, 84, 76, 47,  5, 39],
                [31, 99, 93, 20, 42, 42, 11, 37, 90,  6, 65, 63]])
```

```
In [66]: np.arange(1,13).reshape(4,3)
```

```
Out[66]: array([[ 1,  2,  3],
                [ 4,  5,  6],
                [ 7,  8,  9],
                [10, 11, 12]])
```

In [68]:

```
b = np.random.randint(10,20,(5,4))
b
```

Out[68]:

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

In [ ]: