Day 12 - Numpy 1

Numpy Introduction

What is NumPy?

NumPy (Numerical Python) is a core library for numerical and matrix operations in Python. It introduces a powerful object called the ndarray (N-dimensional array) that enables fast, memory-efficient computations.

- NumPy is faster and more memory-efficient than regular Python lists.
- It allows you to perform operations on entire arrays without writing loops.
- It is used in data analysis, machine learning, deep learning, simulations, image processing, and more.

Python List vs NumPy Array

Feature	Python List	NumPy Array
Speed	Slower	Faster
Memory	More memory usage	Less memory usage
Operations	Manual loops	Vectorized
Math Support	Not built-in	Built-in support

In [1]:
```
import numpy as np
```

In [2]:
```
import sys
sys.version
```

Out[2]:
```
'3.13.5 | packaged by Anaconda, Inc. | (main, Jun 12 2025, 16:37:03) [MSC v.1929 64 bit (AMD64)]'
```

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

Out[3]:
```
'2.1.3'
```

Create List

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

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

In [5]:
```
type(my_list)
```

Out[5]:
```
list
```

List to Array Conversion

In [6]:
```
arr = np.array(my_list)
arr
```

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

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

Out[7]:
```
numpy.ndarray
```

In [8]:
```
print(type(arr))
print(type(my_list))
```

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

Numpy Functions

In [32]:
```
np.arange(10)
```

Out[32]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [33]:
```
np.arange(3.0)
```

Out[33]:
```
array([0., 1., 2.])
```

In [13]:
```
np.arange(9)
```

Out[13]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8])
```

In [14]:
```
np.arange(0,5)
```

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 [16]:
```
np.arange(10,50,5)
```

Out[16]:
```
array([10, 15, 20, 25, 30, 35, 40, 45])
```

In [17]:
```
np.arange(10,30,3)
```

Out[17]:
```
array([10, 13, 16, 19, 22, 25, 28])
```

In [18]:
```
np.arange(10,30,30,3)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[18], line 1
----> 1 np.arange(10,30,30,3)

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

In [19]:
```
np.arange(20,8)
```

Out[19]:
```
array([], dtype=int64)
```

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

Out[20]:
```
array([ 8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

In [21]:
```
np.arange(-20,8) #1st arg < 2n arg
```

Out[21]:
```
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 [22]:
```
np.zeros(10, dtype=int) #parameter tunning (hyper parameter tunning)
```

Out[22]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

In [23]:
```
np.zeros(10) #parameter tunning
```

Out[23]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

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

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

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

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

In [26]:
```
np.zeros((2,2)) #2d array
```

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

In [27]:
```
np.zeros((3,3), dtype = int)
```

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

In [28]:
```
nd = np.zeros((5,9), dtype = int)
nd
```

Out[28]:
```
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 [29]:
```
np.ones(3)
```

Out[29]:
```
array([1., 1., 1.])
```

In [30]:
```
np.ones(3, dtype=int)
```

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

In [31]:
```
nd1 = np.ones((10,10), dtype = int)
nd1
```

Out[31]:
```
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]])
```