NumPy

Visual Studio code installation

steps to install vs code -->

1- google - vscode - download the software & install the software

2- start - open vscode application

3- systm cmd - check your path location

4- manually -- c:drive and reach to path location -- create folder (vscode project)

5- extension -->

- github copilot
- python debugger
- arepl for python
- autodocstring
- coderunner
- intellicode
- jupyter
- llm-vscode
- markdownlint
- office viewer
- pdfviwer
- python
- pylance
- python image preview
- python extension pack
- Notepad ++ keymap
- python extension pack

NUMPY

- NumPy is a powerful library in Python used for numerical computing. It's especially useful for working with arrays and matrices, and it provides a wide range of mathematical functions to operate on these data structures efficiently.

- NumPy stands for Numerical Python
- The core object in NumPy is the ndarray, which stands for "N-dimensional array."

usage of numpy :

- arithmetic operation
- statistical operation
- bitwise operation
- copy & view array
- stacking
- matrix operation
- linear algebra
- broadcasting
- math operation
- sorting, searching

In [3]:
```
'welcome'
```

Out[3]:
```
'welcome'
```

In [4]:
```
import numpy as np
```

In [5]:
```
np.__version__
```

Out[5]:
```
'1.26.4'
```

In [6]:
```
import sys
sys.version
```

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

creating a array

In [7]:
```
my_list = [0,1,2,3,4,5,6,7]
my_list
```

Out[7]:
```
[0, 1, 2, 3, 4, 5, 6, 7]
```

In [8]:
```
len(my_list)
```

Out[8]:
```
8
```

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

Out[9]:
```
list
```

converting list to array

```
In [10]: import numpy as np
```

```
In [11]: arr = np.array(my_list)
         arr
```

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

```
In [12]: type(arr)
```

```
Out[12]: numpy.ndarray
```

12th march

```
In [13]: import numpy as np
```

```
In [14]: np.arange(20)
```

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

```
In [15]: np.arange(1)
```

```
Out[15]: array([0])
```

```
In [16]: np.arange(3)
```

```
Out[16]: array([0, 1, 2])
```

```
In [17]: np.arange(3.0) # float values
```

```
Out[17]: array([0., 1., 2.])
```

```
In [18]: np.arange(0,5)
```

```
Out[18]: array([0, 1, 2, 3, 4])
```

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

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

```
In [20]: np.arange(20,10) # always 1st arg < 2nd arg
```

```
Out[20]: array([], dtype=int32)
```

```
In [21]: np.arange(-20,10)
```

```
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,   8,   9])
```

```
In [22]: np.arange(10,30,5) # 1st index,end index, step index
```

```
Out[22]: array([10, 15, 20, 25])
```

```
In [23]: np.arange(0,10,3)
```

```
Out[23]: array([0, 3, 6, 9])
```

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

```
Out[24]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [25]: np.zeros(3) # parameter tunning (by default prints in float)
```

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

```
In [26]: np.zeros(10)
```

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

```
In [27]: np.zeros(6,dtype=int) # Hyperameter tunning
```

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

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

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

```
In [29]: np.zeros((2,2),dtype=int) # (rows,col)
```

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

```
In [30]: np.zeros((2,10)) # 2D
```

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

```
In [31]: np.zeros((5,3))
```

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

```
In [32]: np.zeros((5,5),dtype=int)
```

```
Out[32]: 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 [33]: np.zeros((5,10),dtype=int)
```

```
Out[33]: 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]])
```

```
In [34]: np.zeros((2,10))
```

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

```
In [35]: np.ones(3)
```

```
Out[35]: array([1., 1., 1.])
```

```
In [36]: np.ones(3,dtype=int)
```

```
Out[36]: array([1, 1, 1])
```

```
In [37]: np.ones((2,2,2)) #
```

```
Out[37]: array([[[1., 1.],
                 [1., 1.]],

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

```
In [38]: np.ones((3,3,3))
```

```
Out[38]: 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.]]])
```

```
In [39]: np.ones((5,4),dtype=int)
```

```
Out[39]: array([[1, 1, 1, 1],
                [1, 1, 1, 1],
                [1, 1, 1, 1],
                [1, 1, 1, 1],
                [1, 1, 1, 1]])
```

```
In [40]: np.ones(5)
```

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

```
In [41]: range(3)
```

```
Out[41]: range(0, 3)
```

```
In [42]: r = range(3)
         r
```

```
Out[42]: range(0, 3)
```

```
In [43]: for i in r:
             print(i)
```

```
0
1
2
```

```
In [44]: range(10)
```

```
Out[44]: range(0, 10)
```

```
In [45]: list(range(10))
```

```
Out[45]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [46]: random.rand(3,2)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[46], line 1
----> 1 random.rand(3,2)

NameError: name 'random' is not defined
```

```
In [47]: np.random.rand(3,2) # OTP # package, module, function
```

```
Out[47]: array([[0.97029527, 0.59763627],
                [0.0775351 , 0.79066731],
                [0.89840852, 0.74667884]])
```

```
In [48]: np.random.rand(3,6) # randomly print the values
```

```
Out[48]: array([[0.91866375, 0.00835754, 0.53264361, 0.9468736 , 0.63774721,
                 0.72626917],
                [0.78394124, 0.91946879, 0.87894591, 0.17403569, 0.11065635,
                 0.48273015],
                [0.53236809, 0.92712125, 0.16705466, 0.96231846, 0.53796566,
                 0.52768601]])
```

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

```
Out[49]: array([0.17680571, 0.93394426, 0.47048345, 0.37918993, 0.26012573])
```

```
In [50]: np.random.rand(4)
```

```
Out[50]: array([0.60030609, 0.6497575 , 0.59544817, 0.75177505])
```

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

```
Out[52]: array([[0.46308493, 0.42835455, 0.07013622, 0.07741275, 0.64578159],
                [0.75808021, 0.39703333, 0.21229871, 0.87739427, 0.41581956],
                [0.51942009, 0.40972139, 0.85040184, 0.87118917, 0.55142477]])
```

```
In [53]: np.random.rand(3,5)
```

```
Out[53]: array([[0.46707892, 0.52454769, 0.68339274, 0.64783038, 0.02631103],
                [0.99766624, 0.71118337, 0.39962008, 0.73854209, 0.53798319],
                [0.00175637, 0.81462303, 0.55110953, 0.36267551, 0.39381878]])
```

```
In [54]: np.random.rand(5,10)
```

```
Out[54]: array([[0.44350196, 0.01375308, 0.95656587, 0.17831232, 0.58837554,
                 0.21819238, 0.45264073, 0.78743648, 0.54480822, 0.28920284],
                [0.30744041, 0.48985106, 0.61741693, 0.63035012, 0.87523543,
                 0.3691376 , 0.62925208, 0.25285067, 0.59541113, 0.77711777],
                [0.80211805, 0.57437385, 0.88505682, 0.81517662, 0.40194904,
                 0.10987263, 0.56891229, 0.62726847, 0.11667466, 0.91984805],
                [0.87460331, 0.14481012, 0.83055951, 0.38338141, 0.81130101,
                 0.33244107, 0.23067444, 0.96796374, 0.64907872, 0.55707154],
                [0.72911488, 0.62015796, 0.54830338, 0.36118339, 0.53790443,
                 0.30168188, 0.92394231, 0.19750654, 0.42940052, 0.98384627]])
```

```
In [55]: np.random.rand(2,3)
```

```
Out[55]: array([[0.4883288 , 0.59100333, 0.78667778],
                [0.70267496, 0.19730628, 0.89341683]])
```

```
In [56]: np.random.randint(0,1)
```

```
Out[56]: 0
```

```
In [57]: np.random.randint(10,23)
```

```
Out[57]: 12
```

```
In [58]: np.random.randint(0,10,2)
```

Out[58]:

```
array([6, 4])
```

In [61]:

```
np.random.randint(3)
```

Out[61]:

```
2
```

In [62]:

```
np.random.randint(20,10,2)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[62], line 1
----> 1 np.random.randint(20,10,2)

File numpy\\random\\mtrand.pyx:780, in numpy.random.mtrand.RandomState.randint()

File numpy\\random\\_bounded_integers.pyx:1425, in numpy.random._bounded_integers._r
and_int32()

ValueError: low >= high
```

In [63]:

```
np.random.randint(5,9)
```

Out[63]:

```
5
```

In [64]:

```
np.random.randint(0,10,(2,4)) # (1st index,end indexing,(row,col))
```

Out[64]:

```
array([[2, 0, 9, 3],
       [1, 5, 8, 9]])
```

In [65]:

```
np.random.randint(10,99,(5,6))
```

Out[65]:

```
array([[39, 85, 20, 12, 83, 35],
       [19, 80, 23, 12, 68, 22],
       [94, 19, 57, 81, 41, 17],
       [54, 81, 95, 38, 31, 11],
       [72, 42, 79, 68, 71, 85]])
```

In [66]:

```
np.arange(10,30).reshape(5,4)
```

Out[66]:

```
array([[10, 11, 12, 13],
       [14, 15, 16, 17],
       [18, 19, 20, 21],
       [22, 23, 24, 25],
       [26, 27, 28, 29]])
```

In [67]:

```
np.arange(0,20).reshape(5,4)
```

Out[67]:

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

In [68]:

```
a = np.arange(0,20).reshape(5,4)
a
```

```
Out[68]:  array([[ 0,  1,  2,  3],
                 [ 4,  5,  6,  7],
                 [ 8,  9, 10, 11],
                 [12, 13, 14, 15],
                 [16, 17, 18, 19]])
```

```
In [69]:  type(a)
```

```
Out[69]:  numpy.ndarray
```

```
In [70]:  a
```

```
Out[70]:  array([[ 0,  1,  2,  3],
                 [ 4,  5,  6,  7],
                 [ 8,  9, 10, 11],
                 [12, 13, 14, 15],
                 [16, 17, 18, 19]])
```

```
In [71]:  a[2,3]
```

```
Out[71]:  11
```

```
In [72]:  a[4,2]
```

```
Out[72]:  18
```

```
In [73]:  a[0:2] # slicing
```

```
Out[73]:  array([[0, 1, 2, 3],
                 [4, 5, 6, 7]])
```

```
In [74]:  a
```

```
Out[74]:  array([[ 0,  1,  2,  3],
                 [ 4,  5,  6,  7],
                 [ 8,  9, 10, 11],
                 [12, 13, 14, 15],
                 [16, 17, 18, 19]])
```

```
In [75]:  a[3,2]
```

```
Out[75]:  14
```

```
In [76]:  a[-2,-4]
```

```
Out[76]:  12
```

```
In [77]:  l = [1,2,3,4,5]
          l
```

```
Out[77]:  [1, 2, 3, 4, 5]
```

```
In [78]:  l[:-4]
```

```
Out[78]: [1]
```

```
In [79]: a
```

```
Out[79]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11],
                [12, 13, 14, 15],
                [16, 17, 18, 19]])
```

```
In [80]: a[0:2]
```

```
Out[80]: array([[0, 1, 2, 3],
                [4, 5, 6, 7]])
```

```
In [81]: a[1:2]
```

```
Out[81]: array([[4, 5, 6, 7]])
```

```
In [82]: a[:]
```

```
Out[82]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11],
                [12, 13, 14, 15],
                [16, 17, 18, 19]])
```

```
In [83]: a[-1:-3] # always 1st argument should be < 2nd argument
```

```
Out[83]: array([], shape=(0, 4), dtype=int32)
```

```
In [84]: a[-3:-1] # slicing with negative indices
```

```
Out[84]: array([[ 8,  9, 10, 11],
                [12, 13, 14, 15]])
```

```
In [85]: a[:]
```

```
Out[85]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11],
                [12, 13, 14, 15],
                [16, 17, 18, 19]])
```

```
In [86]: a[0:2:]
```

```
Out[86]: array([[0, 1, 2, 3],
                [4, 5, 6, 7]])
```

```
In [87]: a[-3:-1]
```

```
Out[87]: array([[ 8,  9, 10, 11],
                [12, 13, 14, 15]])
```

```
In [88]: a[-4:-1]
```

```
Out[88]: array([[ 4,  5,  6,  7],
               [ 8,  9, 10, 11],
               [12, 13, 14, 15]])
```

```
In [89]: a[-4:-2]
```

```
Out[89]: array([[ 4,  5,  6,  7],
               [ 8,  9, 10, 11]])
```

```
In [90]: a[-3:-2]
```

```
Out[90]: array([[ 8,  9, 10, 11]])
```

```
In [91]: a[1:3]
```

```
Out[91]: array([[ 4,  5,  6,  7],
               [ 8,  9, 10, 11]])
```

```
In [ ]:
```