getting-started-with-numpy

August 24, 2023

[1]:
```
! pip install numpy
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(1.23.5)
```

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

[3]:
```
np
```

[3]:
```
<module 'numpy' from '/usr/local/lib/python3.10/dist-
packages/numpy/__init__.py'>
```

[4]:
```
lst =[1,5,6]
```

[5]:
```
type(lst)
```

[5]:
```
list
```

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

[7]:
```
arr
```

[7]:
```
array([1, 5, 6])
```

[8]:
```
type(arr)
```

[8]:
```
numpy.ndarray
```

[12]:
```
import numpy as np

# Create a NumPy array for demonstration
arr = np.array([[1, 2, 3], [4, 5, 6]])

# Get the number of dimensions
num_dimensions = arr.ndim

# Print the number of dimensions
print("Number of dimensions:", num_dimensions)
```

```
Number of dimensions: 2
```

[13]:
```
arr.ndim
```

[13]:
```
2
```

[14]:
```
arr.shape
```

[14]:
```
(2, 3)
```

[15]:
```
arr.size
```

[15]:
```
6
```

[16]:
```
arr.dtype
```

[16]:
```
dtype('int64')
```

[20]:
```
np.zeros = np.zeros((3,5))
```

[22]:
```
import numpy as np

zeros = np.zeros((3, 5))

print(zeros)
```

```
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
```

[23]:
```
ones = np.ones((6,8))
```

[24]:
```
ones
```

[24]:
```
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.]])
```

Reshape and Random Number Generator

[25]:
```
import numpy as np
```

[26]:
```
np.random.random((3,3))
```

[26]:
```
array([[0.11243489, 0.18211982, 0.0357582 ],
       [0.99550492, 0.52014732, 0.82495231],
       [0.80978037, 0.76606401, 0.77608561]])
```

[27]:
```
np.random.random((3,3))
```

[27]:
```
array([[0.93145476, 0.29595108, 0.85351513],
       [0.28003738, 0.92473162, 0.22079262],
       [0.10283949, 0.79139398, 0.66288511]])
```

[28]:
```
if np.random.random() > 0.5:
  print("Run the function")
else:
  print("Don't run the function ")
```

```
Don't run the function
```

[29]:
```
np.arange(1,10,1)
```

[29]:
```
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

[30]:
```
np.linspace(1,10,10)
```

[30]:
```
array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

[31]:
```
np.linspace(1,7,10)
```

[31]:
```
array([1.        , 1.66666667, 2.33333333, 3.        , 3.66666667,
       4.33333333, 5.        , 5.66666667, 6.33333333, 7.        ])
```

[34]:
```
arr = np.random.random((4,3))
arr
```

[34]:
```
array([[0.98502697, 0.36895527, 0.51141099],
       [0.74081599, 0.24685389, 0.6615342 ],
       [0.24043535, 0.29418725, 0.15746477],
       [0.97531813, 0.61469407, 0.40573485]])
```

[35]:
```
np.reshape(arr, (3,4))
```

[35]:
```
array([[0.98502697, 0.36895527, 0.51141099, 0.74081599],
       [0.24685389, 0.6615342 , 0.24043535, 0.29418725],
       [0.15746477, 0.97531813, 0.61469407, 0.40573485]])
```

[36]:
```
arr
```

[36]:
```
array([[0.98502697, 0.36895527, 0.51141099],
       [0.74081599, 0.24685389, 0.6615342 ],
       [0.24043535, 0.29418725, 0.15746477],
       [0.97531813, 0.61469407, 0.40573485]])
```

[38]:
```
arr = np.reshape(arr , (6,2))
arr
```

[38]:
```
array([[0.98502697, 0.36895527],
       [0.51141099, 0.74081599],
       [0.24685389, 0.6615342 ],
       [0.24043535, 0.29418725],
       [0.15746477, 0.97531813],
       [0.61469407, 0.40573485]])
```

[39]:
```
arr.flatten()
```

[39]:
```
array([0.98502697, 0.36895527, 0.51141099, 0.74081599, 0.24685389,
       0.6615342 , 0.24043535, 0.29418725, 0.15746477, 0.97531813,
       0.61469407, 0.40573485])
```

Arithmetic Operations on Array

[40]:
```
import numpy as np
```

[41]:
```
arr = np.array([1,3, 4, 5, 7, 8])
```

[42]:
```
arr
```

[42]:
```
array([1, 3, 4, 5, 7, 8])
```

[43]:
```
arr + 1
```

[43]:
```
array([2, 4, 5, 6, 8, 9])
```

[44]:
```
arr * 5
```

[44]:
```
array([ 5, 15, 20, 25, 35, 40])
```

[45]:
```
print(arr+1)
print(arr-1)
print(arr*1)
print(arr/1)
print(arr % 1)
```

```
[2 4 5 6 8 9]
[0 2 3 4 6 7]
[1 3 4 5 7 8]
```

```
[1. 3. 4. 5. 7. 8.]
[0 0 0 0 0 0]
```

[46]:
```
arr
```

[46]:
```
array([1, 3, 4, 5, 7, 8])
```

[47]:
```
arr = arr + 1
```

[48]:
```
arr
```

[48]:
```
array([2, 4, 5, 6, 8, 9])
```

[49]:
```
arr += 1
```

[50]:
```
arr
```

[50]:
```
array([ 3,  5,  6,  7,  9, 10])
```

[52]:
```
arr =+ 1
arr
```

[52]:
```
1
```

[53]:
```
arr*=2
```

[54]:
```
arr
```

[54]:
```
2
```

Arithmetic Operations on Multiple Arrays

[55]:
```
import numpy as np
```

[56]:
```
arr1 = np.array([[1,2,3],
                 [4,5,6],
                 [7,8,9]])

arr2 = np.array([[9,8,7],
                 [6,5,4],
                 [3,2,1]])

arr3 = np.array([[9,8,7,5],
                 [6,5,4,2],
                 [3,2,1,5]])
```

[57]:
```
arr1
```

[57]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

[58]:
```
arr2
```

[58]:
```
array([[9, 8, 7],
       [6, 5, 4],
       [3, 2, 1]])
```

[59]:
```
arr1 + arr2
```

[59]:
```
array([[10, 10, 10],
       [10, 10, 10],
       [10, 10, 10]])
```

[60]:
```
arr2*arr1
```

[60]:
```
array([[ 9, 16, 21],
       [24, 25, 24],
       [21, 16,  9]])
```

[61]:
```
arr1
```

[61]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

[62]:
```
arr2
```

[62]:
```
array([[9, 8, 7],
       [6, 5, 4],
       [3, 2, 1]])
```

[63]:
```
(1*9) + (2*6) + (3*3)
```

[63]:
```
30
```

[64]:
```
arr1.dot(arr2)
```

[64]:
```
array([[ 30,  24,  18],
       [ 84,  69,  54],
       [138, 114,  90]])
```

[65]:
```
arr2.dot(arr1)
```

[65]:
```
array([[ 90, 114, 138],
       [ 54,  69,  84],
       [ 18,  24,  30]])
```

[66]:
```
arr1
```

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

[68]:
```
arr = np.array([1,2,3,4,5])
```

[69]:
```
arr[:-2]
```

[69]:
```
array([1, 2, 3])
```

[70]:
```
arr1
```

[70]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

[71]:
```
arr1[ 1: , 1 : ]
```

[71]:
```
array([[5, 6],
       [8, 9]])
```

Array Sorting

[72]:
```
arr = np.array([[7,3,8,6,4], [7,2,9,8,6],[5,4,2,3,1]])
arr
```

[72]:
```
array([[7, 3, 8, 6, 4],
       [7, 2, 9, 8, 6],
       [5, 4, 2, 3, 1]])
```

[73]:
```
np.sort(arr , axis = 1, kind = 'mergesort')
```

[73]:
```
array([[3, 4, 6, 7, 8],
       [2, 6, 7, 8, 9],
       [1, 2, 3, 4, 5]])
```

[74]:
```
np.sort(arr , axis = 0, kind = 'mergesort')
```

[74]:
```
array([[5, 2, 2, 3, 1],
       [7, 3, 8, 6, 4],
       [7, 4, 9, 8, 6]])
```

[75]:
```
np.sort(arr , kind = 'mergesort')
```

[75]:
```
array([[3, 4, 6, 7, 8],
       [2, 6, 7, 8, 9],
       [1, 2, 3, 4, 5]])
```

Array Merging

[76]:
```
arr1 = np.array([[1,2,3,4], [5,6,7,8]])
arr2 = np.array([[8,7,6,5], [4,3,2,1]])
```

[77]:
```
arr1
```

[77]:
```
array([[1, 2, 3, 4],
       [5, 6, 7, 8]])
```

[78]:
```
arr2
```

[78]:
```
array([[8, 7, 6, 5],
       [4, 3, 2, 1]])
```

[79]:
```
np.vstack((arr1,arr2))
```

[79]:
```
array([[1, 2, 3, 4],
       [5, 6, 7, 8],
       [8, 7, 6, 5],
       [4, 3, 2, 1]])
```

[80]:
```
np.hstack((arr1,arr2))
```

[80]:
```
array([[1, 2, 3, 4, 8, 7, 6, 5],
       [5, 6, 7, 8, 4, 3, 2, 1]])
```

[81]:
```
np.hstack((arr2,arr1))
```

[81]:
```
array([[8, 7, 6, 5, 1, 2, 3, 4],
       [4, 3, 2, 1, 5, 6, 7, 8]])
```

[82]:
```
arr = np.concatenate((arr1,arr2), axis = 0)
```

[83]:
```
np.concatenate((arr1,arr2), axis = 1)
```

[83]:
```
array([[1, 2, 3, 4, 8, 7, 6, 5],
       [5, 6, 7, 8, 4, 3, 2, 1]])
```

[84]:
```
arr
```

[84]:
```
array([[1, 2, 3, 4],
       [5, 6, 7, 8],
       [8, 7, 6, 5],
       [4, 3, 2, 1]])
```

[85]:
```
np.hsplit(arr, 2)
```

[85]:
```
[array([[1, 2],
        [5, 6],
        [8, 7],
        [4, 3]]),
 array([[3, 4],
        [7, 8],
        [6, 5],
        [2, 1]])]
```

[86]:
```
np.vsplit(arr, 2)
```

[86]:
```
[array([[1, 2, 3, 4],
        [5, 6, 7, 8]]),
 array([[8, 7, 6, 5],
        [4, 3, 2, 1]])]
```

Array Slicing - DAP

[87]:
```
arr = np.array([1,2,3,4,5,6,7,8])
```

[88]:
```
arr
```

[88]:
```
array([1, 2, 3, 4, 5, 6, 7, 8])
```

[89]:
```
arr[3: ]
```

[89]:
```
array([4, 5, 6, 7, 8])
```

[90]:
```
arr[ : -3]
```

[90]:
```
array([1, 2, 3, 4, 5])
```

[91]:
```
arr[3  :  -3]
```

[91]:
```
array([4, 5])
```

[92]:
```
arr = np.array([[1,2,3,4],[5,6,7,8],[8,7,6,5],[4,3,2,1]])
```

[93]:
```
arr
```

[93]:
```
array([[1, 2, 3, 4],
       [5, 6, 7, 8],
       [8, 7, 6, 5],
       [4, 3, 2, 1]])
```

[94]:
```
arr[ 3 , 1]
```

[94]:
```
3
```

[95]:
```
arr[ 1: , 1:]
```

[95]:
```
array([[6, 7, 8],
       [7, 6, 5],
       [3, 2, 1]])
```

Automating using Numpy

[96]:
```
lst = [1,2,3,4,5,6,7,8,9,10]
```

[97]:
```
q = 0

for i in lst:
  if (i > 5):
    print(i)
    q += 1

print(q)
```

```
6
7
8
9
10
5
```

[99]:
```
lst = np.array([1,2,3,4,5,6,7,8,9,10])
```

[100]:
```
print(len(lst[lst > 5]))
print(lst[lst>5])
```

```
5
[ 6  7  8  9 10]
```

[101]:
```
print(len(lst[lst % 3 == 0]))
print(lst[lst % 3 == 0])
```

```
3
[3 6 9]
```