

Numpy Crash Course

```
In [41]: import numpy as np
```

```
In [42]: np.__version__
```

```
Out[42]: '1.26.4'
```

```
In [43]: import sys
sys.version
```

```
Out[43]: '3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2024, 13:17:27) [MSC v.1929 6
4 bit (AMD64)]'
```

Creating List

```
In [44]: my_list = [0,1,2,3,4,5]
my_list
```

```
Out[44]: [0, 1, 2, 3, 4, 5]
```

```
In [45]: type(my_list)
```

```
Out[45]: list
```

below code we are converting list to array

```
In [46]: !pip install numpy
```

```
Requirement already satisfied: numpy in d:\anaconda python in windows 11\lib\site-packages (1.26.4)
```

```
In [48]: arr = np.array(my_list)
```

```
In [50]: arr
```

```
Out[50]: array([0, 1, 2, 3, 4, 5])
```

```
In [51]: type(arr)
```

```
Out[51]: numpy.ndarray
```

```
In [52]: my_list = [0,1,2,3,4,5]
my_list
```

```
Out[52]: [0, 1, 2, 3, 4, 5]
```

```
In [53]: type(my_list)
```

```
Out[53]: list
```

```
In [32]: np. # we Learn important function
```

```
Cell In[32], line 1
  np. # we learn important function
  ^
SyntaxError: invalid syntax
```

```
In [55]: np.arange(10)
```

```
Out[55]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [57]: np.arange(5.0)
```

```
Out[57]: array([0., 1., 2., 3., 4.])
```

```
In [58]: np.arange(2.0)
```

```
Out[58]: array([0., 1.])
```

```
In [59]: np.arange(9)
```

```
Out[59]: array([0, 1, 2, 3, 4, 5, 6, 7, 8])
```

```
In [60]: np.arange(0, 5)
```

```
Out[60]: array([0, 1, 2, 3, 4])
```

```
In [61]: np.arange(10,20)
```

```
Out[61]: array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

```
In [62]: np.arange(20,10) # 1st arg < 2nd arg
```

```
Out[62]: array([], dtype=int32)
```

```
In [63]: np.arange(-20,10)
```

```
Out[63]: 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 [64]: np.arange(-16,10)
```

```
Out[64]: array([-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 [65]: np.arange(-20,-10) # 1st arg always be < then 2nd arg
```

```
Out[65]: array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11])
```

```
In [66]: ar = np.arange(-30,20)
ar
```

```
Out[66]: array([-30, -29, -28, -27, -26, -25, -24, -23, -22, -21, -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, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

```
In [67]: np.arange()
```

```
-----
```

```
TypeError
Cell In[67], line 1
----> 1 np.arange()
```

```
Traceback (most recent call last)
```

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

```
In [68]: np.arange(10,30,5) # 10 - starting from 30- end point 5 - step count
```

```
Out[68]: array([10, 15, 20, 25])
```

```
In [69]: np.arange(0,10,3)
```

```
Out[69]: array([0, 3, 6, 9])
```

```
In [70]: np.arange(10,30,5,8)
```

```
-----
```

```
TypeError
Cell In[70], line 1
----> 1 np.arange(10,30,5,8)
```

```
Traceback (most recent call last)
```

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

```
In [71]: np.zeros(10) # parameter tuning
```

```
Out[71]: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

```
In [72]: np.zeros(10,dtype=int)
```

```
Out[72]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
In [74]: np.zeros(5, dtype=int) # hyperparameter tuning
```

```
Out[74]: array([0, 0, 0, 0, 0])
```

```
In [75]: np.zeros((2,2), dtype=int)
```

```
Out[75]: array([[0, 0],
[0, 0]])
```

```
In [77]: zero = np.zeros([2,2])
print(zero)
print(type(zero))
```

```
[[0. 0.]
 [0. 0.]]
<class 'numpy.ndarray'>
```

```
In [78]: np.zeros((2,10))
```

```
Out[78]: array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],  
                 [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]])
```

```
In [79]: np.zeros((2,2))
```

```
Out[79]: array([[0., 0.],  
                 [0., 0.]])
```

```
In [80]: np.zeros((3,3))
```

```
Out[80]: array([[0., 0., 0.],  
                 [0., 0., 0.],  
                 [0., 0., 0.]])
```

```
In [64]: np.zeros((10,30))
```

```
In [81]: np.zeros((5,10)) # by default large -- will give row & 2nd arg - columns
```

```
Out[81]: 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.]])
```

```
In [82]: np.zeros((10,10),dtype = int)
```

```
In [83]: n = (6,7)
         n1 = (6,8)
         print(np.zeros(n1)) # parameter tuning g
```

```
In [84]: print(np.zeros(n,dtype=int)) ## hpyerparameter tunning
```

```
[[0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0]
 [0 0 0 0 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 [85]: n

```
Out[85]: (6, 7)
```

In [86]: n1

```
Out[86]: (6, 8)
```

```
In [87]: print(nn.zeros(n1))
```

```
In [88]: np.ones(3)
```

```
Out[88]: array([1., 1., 1.])
```

```
In [89]: np.ones((3,3))
```

```
Out[89]: array([[1., 1., 1.],
   [1., 1., 1.],
   [1., 1., 1.]])
```

```
In [91]: np.ones(4, dtype=int)
```

```
Out[91]: array([1, 1, 1, 1])
```

```
In [92]: np.ones(4)
```

```
Out[92]: array([1., 1., 1., 1.])
```

```
In [93]: n
```

```
Out[93]: (6, 7)
```

```
In [94]: np.ones(n)
```

```
Out[94]: 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.]])
```

```
In [96]: np.ones((5,4), dtype=int) # by default 5- rows & 4 - columns
```

```
Out[96]: array([[1, 1, 1, 1],
   [1, 1, 1, 1],
   [1, 1, 1, 1],
   [1, 1, 1, 1],
   [1, 1, 1, 1]])
```

```
In [97]: np.t
```

```
Cell In[97], line 1
```

```
np.
```

```
^
```

```
SyntaxError: invalid syntax
```

```
In [98]: np.two((2,3))
```

```
-----
```

```
AttributeError
Cell In[98], line 1
----> 1 np.two((2,3))
```

```
Traceback (most recent call last)
```

```
File D:\Anaconda Python in Windows 11\Lib\site-packages\numpy\__init__.py:333, in __
getattr__(attr)
 330     "Removed in NumPy 1.25.0"
 331     raise RuntimeError("Tester was removed in NumPy 1.25.")
--> 333 raise AttributeError("module {!r} has no attribute "
 334                     "{!r}".format(__name__, attr))
```

```
AttributeError: module 'numpy' has no attribute 'two'
```

```
In [99]: np.three((2,3))
```

```
-----  
AttributeError                                     Traceback (most recent call last)  
Cell In[99], line 1  
----> 1 np.three((2,3))
```

```
File D:\Anaconda Python in Windows 11\Lib\site-packages\numpy\__init__.py:333, in __  
getattr__(attr)  
 330     "Removed in NumPy 1.25.0"  
 331     raise RuntimeError("Tester was removed in NumPy 1.25.")  
--> 333 raise AttributeError("module {!r} has no attribute "  
 334                     "{!r}".format(__name__, attr))
```

```
AttributeError: module 'numpy' has no attribute 'three'
```

```
In [100... np.ones(2)
```

```
Out[100... array([1., 1.])
```

```
In [101... np.ones((2,4))
```

```
Out[101... array([[1., 1., 1., 1.],  
                  [1., 1., 1., 1.]])
```

```
In [102... np.ones((6,10),dtype = int)
```

```
Out[102... 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]])
```

```
In [103... np.ones(2,4))
```

```
Cell In[103], line 1  
np.ones(2,4))  
^  
SyntaxError: unmatched ''
```

```
In [104... np.
```

```
Cell In[104], line 1  
np.  
^  
SyntaxError: invalid syntax
```

```
In [105... np.three((2,4))
```

```
-----  
AttributeError Traceback (most recent call last)  
Cell In[105], line 1  
----> 1 np.three((2,4))  
  
File D:\Anaconda Python in Windows 11\Lib\site-packages\numpy\__init__.py:333, in __  
getattr__(attr)  
 330     "Removed in NumPy 1.25.0"  
 331     raise RuntimeError("Tester was removed in NumPy 1.25.")  
--> 333 raise AttributeError("module {!r} has no attribute "  
 334                 "{}!".format(__name__, attr))  
  
AttributeError: module 'numpy' has no attribute 'three'
```

In [106...]: `range(5)`

Out[106...]: `range(0, 5)`

In [107...]: `r = range(5)`
r

Out[107...]: `range(0, 5)`

In [108...]: `for i in r:`
 `print(i)`

0
1
2
3
4

In [109...]: `list(range(5))`

Out[109...]: `[0, 1, 2, 3, 4]`

In [110...]: `range(1,10)`

Out[110...]: `range(1, 10)`

In [111...]: `list(range(1,10))`

Out[111...]: `[1, 2, 3, 4, 5, 6, 7, 8, 9]`

In [112...]: `list(range(1,10,3))`

Out[112...]: `[1, 4, 7]`

In [113...]: `y = list(range(12))`
y

Out[113...]: `[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]`

In [114...]: `rand(3,2)`

```
NameError Traceback (most recent call last)
Cell In[114], line 1
----> 1 rand(3,2)

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

In [115... random.rand(3,2)

```
NameError Traceback (most recent call last)
Cell In[115], line 1
----> 1 random.rand(3,2)

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

In [117... np.random.rand(5)

Out[117... array([0.23833784, 0.22953063, 0.00520161, 0.37617671, 0.65044562])

In [119... np.rand(4)

```
AttributeError Traceback (most recent call last)
Cell In[119], line 1
----> 1 np.rand(4)

File D:\Anaconda Python in Windows 11\Lib\site-packages\numpy\__init__.py:333, in __
getattr__(attr)
    330     "Removed in NumPy 1.25.0"
    331     raise RuntimeError("Tester was removed in NumPy 1.25.")
--> 333 raise AttributeError("module {!r} has no attribute "
    334                 "{!r}".format(__name__, attr))

AttributeError: module 'numpy' has no attribute 'rand'
```

In [120... np.random.rand(3,5)

Out[120... array([[0.6274539 , 0.28364037, 0.69072444, 0.41114175, 0.81934126],
 [0.17863562, 0.09939568, 0.15026301, 0.68103177, 0.01805326],
 [0.92545381, 0.64740874, 0.78585733, 0.80478655, 0.46262653]])

In [123... np.random.randint(4,6)

Out[123... 4

In [124... np.random.randint(2,20) # 2nd argument is exclusive

Out[124... 10

In [127... np.random.randint(2,20,4)

Out[127... array([4, 4, 9, 7])

In [128... np.random.randint(0,1)

Out[128... 0

In [129... np.random.randint(10,20,5)

Out[129... array([19, 11, 10, 12, 19])

In [130... np.random.randint(1,6,4)

Out[130... array([4, 5, 1, 3])

In [131... np.random.rand(3)

Out[131... array([0.95213076, 0.88134768, 0.41479083])

In [132... np.random.randint(1)

Out[132... 0

In [133... np.random.randint(30,20,10)

ValueError

Traceback (most recent call last)

Cell In[133], line 1

----> 1 np.random.randint(30,20,10)

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

File numpy\\random_bounded_integers.pyx:1425, in numpy.random._bounded_integers._rand_int32()

ValueError: low >= high

In [134... np.random.randint(-30,20,10)

Out[134... array([-10, -7, 9, 8, -18, 18, 5, 15, -8, -2])

In [135... np.random.randint(20,30,10)

Out[135... array([25, 20, 20, 22, 28, 28, 29, 24, 28, 27])

In [136... np.random.randint(5,9) # Get the value <=1 & >=5

Out[136... 5

In [137... np.random.randint(10,21,3)

Out[137... array([14, 17, 19])

In [138... np.random.randint(1,12,10)

Out[138... array([10, 6, 9, 10, 1, 1, 11, 3, 3, 11])

```
In [139... np.random.randint(10,40,(10,10)) # generate the element 10 - 30 with 4*4 mtri
```

```
Out[139... array([[13, 26, 10, 19, 17, 38, 30, 26, 14, 14],
 [35, 34, 27, 20, 26, 32, 13, 21, 36, 25],
 [26, 14, 33, 20, 20, 17, 37, 34, 17, 24],
 [33, 32, 30, 28, 20, 17, 10, 39, 30, 28],
 [18, 23, 20, 18, 16, 28, 33, 24, 22, 35],
 [27, 37, 23, 37, 25, 32, 36, 15, 26, 25],
 [18, 23, 27, 25, 29, 39, 28, 15, 36, 26],
 [17, 29, 16, 12, 35, 18, 21, 31, 30, 15],
 [31, 16, 11, 27, 13, 27, 12, 39, 19, 10],
 [11, 37, 26, 21, 18, 39, 24, 11, 38, 35]])
```

```
In [140... np.random.randint(1,100,(12,12)) # generate the element 10 - 30 with 4*4 mtri
```

```
Out[140... array([[22, 13, 21, 42, 56, 12, 62, 74, 73, 56, 85, 56],
 [30, 62, 27, 13, 82, 47, 36, 96, 98, 12, 17, 57],
 [35, 27, 94, 24, 51, 33, 61, 92, 92, 59, 43, 88],
 [7, 87, 49, 20, 10, 14, 5, 55, 28, 69, 22, 82],
 [74, 28, 40, 22, 61, 6, 54, 8, 49, 33, 98, 46],
 [61, 51, 21, 6, 3, 67, 11, 2, 88, 87, 82, 98],
 [88, 74, 5, 17, 32, 5, 95, 80, 43, 54, 34, 1],
 [53, 28, 22, 22, 87, 64, 20, 62, 50, 87, 65, 63],
 [94, 80, 25, 71, 86, 82, 13, 35, 79, 13, 64, 56],
 [25, 5, 9, 20, 59, 71, 13, 82, 17, 12, 16, 96],
 [60, 67, 74, 61, 2, 73, 97, 36, 19, 54, 62, 57],
 [63, 46, 30, 98, 82, 2, 23, 58, 8, 15, 96, 97]])
```

```
In [141... np.arange(1,13)
```

```
Out[141... array([ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
```

```
In [146... np.arange(1,13).reshape(3,4)
```

```
Out[146... array([[ 1, 2, 3, 4],
 [ 5, 6, 7, 8],
 [ 9, 10, 11, 12]])
```

```
In [148... np.arange(1,13).reshape(4,3)
```

```
Out[148... array([[ 1, 2, 3],
 [ 4, 5, 6],
 [ 7, 8, 9],
 [10, 11, 12]])
```

```
In [149... np.arange(1,13).reshape(12,1)
```

```
Out[149... array([[ 1],  
[ 2],  
[ 3],  
[ 4],  
[ 5],  
[ 6],  
[ 7],  
[ 8],  
[ 9],  
[10],  
[11],  
[12]])
```

slicing in matrix

```
In [150... b = np.random.randint(10,20,(5,4))  
b
```

```
Out[150... array([[18, 13, 14, 11],  
[17, 10, 15, 10],  
[13, 19, 15, 19],  
[15, 11, 18, 17],  
[17, 10, 19, 12]])
```

```
In [151... type(b)
```

```
Out[151... numpy.ndarray
```

```
In [153... b
```

```
Out[153... array([[18, 13, 14, 11],  
[17, 10, 15, 10],  
[13, 19, 15, 19],  
[15, 11, 18, 17],  
[17, 10, 19, 12]])
```

```
In [154... b[:]
```

```
Out[154... array([[18, 13, 14, 11],  
[17, 10, 15, 10],  
[13, 19, 15, 19],  
[15, 11, 18, 17],  
[17, 10, 19, 12]])
```

```
In [ ]:
```