

# NumPy

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

```
In [2]: np._version_
```

```
-----  
AttributeError                                Traceback (most recent call last)  
Cell In[2], line 1  
----> 1 np._version_  
  
File ~\anaconda3\Lib\site-packages\numpy\__init__.py:414, in __getattr__(attr)  
    411     import numpy.char as char  
    412     return char.chararray  
--> 414 raise AttributeError("module {!r} has no attribute "  
    415                        "{!r}".format(__name__, attr))  
  
AttributeError: module 'numpy' has no attribute '_version_'
```

## Assistant

The error occurs because you're trying to access an attribute `_version_` that doesn't exist in the NumPy module. The correct attribute to check NumPy's version is `__version__` (with double underscores on both sides).

```
In [4]: np.__version__ #with double underscores on both sides
```

```
Out[4]: '2.1.3'
```

```
In [5]: # Method 1: Update NumPy using pip  
!pip install --upgrade numpy
```

Requirement already satisfied: numpy in c:\users\acer\anaconda3\lib\site-packages (2.1.3)

Collecting numpy

Downloading numpy-2.3.5-cp313-cp313-win\_amd64.whl.metadata (60 kB)

Downloading numpy-2.3.5-cp313-cp313-win\_amd64.whl (12.8 MB)

```
----- 0.0/12.8 MB ? eta -:--:--
--- 1.0/12.8 MB 5.1 MB/s eta 0:00:03
----- 1.8/12.8 MB 4.4 MB/s eta 0:00:03
----- 2.6/12.8 MB 4.2 MB/s eta 0:00:03
----- 3.4/12.8 MB 4.1 MB/s eta 0:00:03
----- 4.2/12.8 MB 4.1 MB/s eta 0:00:03
----- 5.2/12.8 MB 4.1 MB/s eta 0:00:02
----- 5.8/12.8 MB 4.0 MB/s eta 0:00:02
----- 6.8/12.8 MB 4.0 MB/s eta 0:00:02
----- 7.9/12.8 MB 4.0 MB/s eta 0:00:02
----- 8.7/12.8 MB 4.0 MB/s eta 0:00:02
----- 9.4/12.8 MB 4.0 MB/s eta 0:00:01
----- 10.2/12.8 MB 4.0 MB/s eta 0:00:01
----- 11.0/12.8 MB 4.0 MB/s eta 0:00:01
----- 11.8/12.8 MB 4.0 MB/s eta 0:00:01
----- 12.6/12.8 MB 3.9 MB/s eta 0:00:01
----- 12.8/12.8 MB 3.9 MB/s eta 0:00:00
```

Installing collected packages: numpy

Attempting uninstall: numpy

Found existing installation: numpy 2.1.3

Uninstalling numpy-2.1.3:

Successfully uninstalled numpy-2.1.3

Successfully installed numpy-2.3.5

ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.

numba 0.61.0 requires numpy<2.2,>=1.24, but you have numpy 2.3.5 which is incompatible.

```
In [3]: import numpy as np
        np.__version__
```

Out[3]: '2.3.5'

```
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

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

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

Out[7]: numpy.ndarray

```
In [8]: arr
```

Out[8]: array([0, 1, 2, 3, 4, 5])

```
In [9]: np.arange(10)
```

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

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

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

```
In [11]: np.arange(20)
```

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

```
In [12]: np.arange(5.0)
```

```
Out[12]: array([0., 1., 2., 3., 4.])
```

```
In [13]: np.arange(5,10)
```

```
Out[13]: array([5, 6, 7, 8, 9])
```

```
In [14]: np.arange(10, 20)
```

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

```
In [15]: np.arange(10,30,5)
```

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

```
In [20]: np.zeros(3) #parameter, default displays value in float
```

```
Out[20]: array([0., 0., 0.])
```

```
In [19]: np.zeros(3, dtype=int) #hyperparameter
```

```
Out[19]: array([0, 0, 0])
```

```
In [24]: np.zeros([2,2])
```

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

```
In [25]: np.zeros([2,2], dtype=int)
```

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

```
In [27]: np.ones([5,4],dtype=int) #5/4 matrix or 5x4 matrix
```

```
Out[27]: array([[1, 1, 1, 1],  
               [1, 1, 1, 1],  
               [1, 1, 1, 1],  
               [1, 1, 1, 1],  
               [1, 1, 1, 1]])
```

```
In [39]: np.random.rand(2,3)
```

```
Out[39]: array([[0.88142974, 0.4682977 , 0.50540358],  
               [0.96593308, 0.01730841, 0.11247497]])
```

```
In [72]: np.random.randint(5) #Generates random numbers from 0 to 5(exclusive)
```

```
Out[72]: 1
```

```
In [69]: np.random.randint(1,8) #Generates random numbers from 1(inclusive) to 8(exclusiv
```

```
Out[69]: 6
```

```
In [71]: np.random.randint(1,8,4) #Generates random numbers from 1(inclusive) to 8(exclus
```

```
Out[71]: array([2, 2, 5, 3], dtype=int32)
```

```
In [73]: np.random.randint(10,40,(10,10))
```

```
Out[73]: array([[35, 22, 33, 35, 20, 15, 28, 23, 20, 13],  
               [14, 20, 12, 23, 39, 23, 38, 30, 32, 38],  
               [18, 19, 13, 33, 25, 21, 19, 33, 38, 22],  
               [24, 27, 36, 24, 34, 20, 31, 36, 14, 36],  
               [37, 14, 37, 14, 13, 18, 31, 14, 36, 33],  
               [26, 11, 12, 38, 29, 25, 19, 10, 38, 38],  
               [21, 27, 30, 17, 26, 21, 26, 33, 20, 17],  
               [29, 35, 30, 31, 32, 27, 32, 34, 26, 24],  
               [17, 16, 38, 33, 22, 33, 25, 28, 30, 15],  
               [10, 13, 17, 32, 24, 14, 21, 10, 17, 15]], dtype=int32)
```

```
In [74]: np.arange(1,13).reshape(3,4)
```

```
Out[74]: array([[ 1,  2,  3,  4],  
               [ 5,  6,  7,  8],  
               [ 9, 10, 11, 12]])
```

## Slicing in Matrix

```
In [76]: s = np.random.randint(10,20,(3,4))  
s
```

```
Out[76]: array([[17, 19, 18, 17],  
               [15, 10, 15, 16],  
               [13, 18, 14, 13]], dtype=int32)
```

```
In [77]: s[:]
```

```
Out[77]: array([[17, 19, 18, 17],  
               [15, 10, 15, 16],  
               [13, 18, 14, 13]], dtype=int32)
```

```
In [78]: s[0]
```

```
Out[78]: array([17, 19, 18, 17], dtype=int32)
```

```
In [79]: s[0:3]
```

```
Out[79]: array([[17, 19, 18, 17],
               [15, 10, 15, 16],
               [13, 18, 14, 13]], dtype=int32)
```

```
In [80]: s[0,1]
```

```
Out[80]: np.int32(19)
```

```
In [82]: s[::-1] #to reverse an array
```

```
Out[82]: array([[13, 18, 14, 13],
               [15, 10, 15, 16],
               [17, 19, 18, 17]], dtype=int32)
```

```
In [83]: print(s)
```

```
[[17 19 18 17]
 [15 10 15 16]
 [13 18 14 13]]
```

```
In [84]: print(s[::-1])
```

```
[[13 18 14 13]
 [15 10 15 16]
 [17 19 18 17]]
```

```
In [94]: s[-3,-2] #comma used to call an element from an array
```

```
Out[94]: np.int32(18)
```

```
In [86]: s.max()
```

```
Out[86]: np.int32(19)
```

```
In [87]: s.min()
```

```
Out[87]: np.int32(10)
```

```
In [88]: s.mean()
```

```
Out[88]: np.float64(15.416666666666666)
```

```
In [89]: s
```

```
Out[89]: array([[17, 19, 18, 17],
               [15, 10, 15, 16],
               [13, 18, 14, 13]], dtype=int32)
```

```
In [91]: s[0] #Displays the 1st row
```

```
Out[91]: array([17, 19, 18, 17], dtype=int32)
```

```
In [93]: s[:,0] #Displays the 1st column
```

```
Out[93]: array([17, 15, 13], dtype=int32)
```

```
In [95]: s[:]
```

```
Out[95]: array([[17, 19, 18, 17],  
               [15, 10, 15, 16],  
               [13, 18, 14, 13]], dtype=int32)
```

```
In [96]: s[1:2,2:3]
```

```
Out[96]: array([[15]], dtype=int32)
```

```
In [ ]:
```