NumPy

- ## How to create a NumPy Array ?

- ## import numpy as np

In [31]:
```
import numpy as np

a = np.array(['a', 'He', 2,3])
print(a) # ['a' 'He' '2' '3']
print(type(a)) # <class 'numpy.ndarray'>
```

```
['a' 'He' '2' '3']
<class 'numpy.ndarray'>
```

- ## Combined types are not allowed.

In [5]:
```
import numpy as np

a = np.array(25)
print(a) # 25
```

```
25
```

In [7]:
```
import numpy as np
a = np.array([10,20,30,40,50])
print(a)
# [10 20 30 40 50]
```

```
[10 20 30 40 50]
```

In [12]:
```
import numpy as np
a = np.array([10,20,30,40,50])
print("Dimension is :", a.ndim) # 1
print(a) # [10 20 30 40 50]
print(a.shape) # (5,)
```

```
Dimension is : 1
[10 20 30 40 50]
(5,)
```

- ## Dimenstions :- 0-D || 1-D || 2-D || 3-D

In [16]:
```python
import numpy as np
a = np.array([[10,20,30],[40,50]])
print("Dimension is :", a.ndim) # 1
print(a) # [list([10, 20, 30]) list([40, 50])]
print(a.shape) # (2,)
```

```
Dimension is : 1
[list([10, 20, 30]) list([40, 50])]
(2,)

<ipython-input-16-6fd59125c966>:2: VisibleDeprecationWarning: Creating an
ndarray from ragged nested sequences (which is a list-or-tuple of lists-or
-tuples-or ndarrays with different lengths or shapes) is deprecated. If yo
u meant to do this, you must specify 'dtype=object' when creating the ndar
ray
  a = np.array([[10,20,30],[40,50]])
```

- ## Each row must have same column

In [17]:
```python
import numpy as np
a = np.array([[10,20,30],[40,50,60]])
print("Dimension is :", a.ndim) # 2
print(a)
# [[10 20 30]
# [40 50 60]]
print(a.shape) # (2, 3)
```

```
Dimension is : 2
[[10 20 30]
 [40 50 60]]
(2, 3)
```

In [18]:
```python
import numpy as np
a = np.array([[[10,20,30],[40,50,60]],[[1,2,3],[4,5,6]]])
print("Dimension is :", a.ndim) # 3
print(a)
# [[[10 20 30]
#   [40 50 60]]

#  [[ 1  2  3]
#   [ 4  5  6]]]
print(a.shape) # (2, 2, 3)
```

```
Dimension is : 3
[[[10 20 30]
  [40 50 60]]

 [[ 1  2  3]
  [ 4  5  6]]]
(2, 2, 3)
```

- **In 3-D array 1-D && 0-D is Equal.**

In [19]:
```
import numpy as np
a = np.array([[[[1,2,3]]]])
print("Dimension is :", a.ndim) # 4
print(a) # [[[[1 2 3]]]]
print(a.shape) # (1, 1, 1, 3)
```

```
Dimension is : 4
[[[[1 2 3]]]]
(1, 1, 1, 3)
```

In [22]:
```
import numpy as np
a = np.array([1,2,3,4,5])

print(a[-4:]) # [2 3 4 5]
print(a[1:]) # [2 3 4 5]
```

```
[2 3 4 5]
[2 3 4 5]
```

In [24]:
```
import numpy as np
a = np.array([[1,2,3,4,5],[6,7,8,9,10]])
print(a[1:,1:4]) # [[7 8 9]]
print(a[:,1:4])
# [[2 3 4]
#  [7 8 9]]
```

```
[[7 8 9]]
[[2 3 4]
 [7 8 9]]
```

In [29]:
```
import numpy as np
a = np.array([[[1,2,3],[4,5,6,2,3],[7,8,9,2,3]],
             [[2,3,4,8,5],[4,2,3,5,6],[8,7,2,3,9]]])
print(a[:,1:,1:])

# [[[5 6]
#   [8 9]]

#  [[5 6]
#   [7 9]]]
```

```
[[[5 6]
  [8 9]]

 [[5 6]
  [7 9]]]
```

In [30]:
```
import numpy as np
a = np.array([[[1,2,9,8,3],[4,5,6,2,3],[7,8,9,2,3]],
             [[2,3,4,8,5],[4,2,3,5,6],[8,7,2,3,9]]])

print(a[:,::2,1::2])
# [[[2 8]
#   [8 2]]

#  [[3 8]
#   [7 3]]]
```

```
[[[2 8]
  [8 2]]

 [[3 8]
  [7 3]]]
```

In [32]:
```
import numpy as np
a = np.array([[[1,2,9,8,3],[4,5,6,2,3],[7,8,9,2,3]],
             [[2,3,4,8,5],[4,2,3,5,6],[8,7,2,3,9]]])
print(a[:,0,1]) # [2 3]
```

```
[2 3]
```

In [39]:
```
import numpy as np
a = np.array((1,2,3,4,5,6))
a = a.reshape(2,3)
print(a)
# [[1 2 3]
#  [4 5 6]]
```

```
[[1 2 3]
 [4 5 6]]
```

In [40]:
```
import numpy as np
a = np.array((1,2,3,4,5,6))
a = a.reshape(2,4)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call las
t)
<ipython-input-40-ad54128a8811> in <module>
      1 import numpy as np
      2 a = np.array((1,2,3,4,5,6))
----> 3 a = a.reshape(2,4)

ValueError: cannot reshape array of size 6 into shape (2,4)
```

In [44]:
```python
import numpy as np
a = np.array((1,2,3,4,5,6))
a = a.reshape(1,3,2)
print(a)
# [[[1 2]
#   [3 4]
#   [5 6]]]
```

```
[[[1 2]
  [3 4]
  [5 6]]]
```

In [45]:
```python
import numpy as np
a = np.array((1,2,3,4,5,6))
a = a.reshape(1,3,2,-1)
print(a)
# [[[[1]
#     [2]]

#   [[3]
#     [4]]

#   [[5]
#     [6]]]]
```

```
[[[[1]
   [2]]

  [[3]
   [4]]

  [[5]
   [6]]]]
```

In [48]:
```
import numpy as np
a = np.array(range(1,51)).reshape(2,5,5)
print(a)

# [[[ 1  2  3  4  5]
#   [ 6  7  8  9 10]
#   [11 12 13 14 15]
#   [16 17 18 19 20]
#   [21 22 23 24 25]]

#  [[26 27 28 29 30]
#   [31 32 33 34 35]
#   [36 37 38 39 40]
#   [41 42 43 44 45]
#   [46 47 48 49 50]]]
```

```
[[[ 1  2  3  4  5]
  [ 6  7  8  9 10]
  [11 12 13 14 15]
  [16 17 18 19 20]
  [21 22 23 24 25]]

 [[26 27 28 29 30]
  [31 32 33 34 35]
  [36 37 38 39 40]
  [41 42 43 44 45]
  [46 47 48 49 50]]]
```

In [49]:
```
import numpy as np
a = np.array(([[1,2],[5,6],[8,9]]))
a = a.reshape(-1)
print(a) # [1 2 5 6 8 9]
```

```
[1 2 5 6 8 9]
```

In [50]:
```
import numpy as np
a = np.array(([[1,2],[5,6],[8,9]]))
a = a.reshape(1,-1)
print(a) # [[1 2 5 6 8 9]]
```

```
[[1 2 5 6 8 9]]
```

In [67]:
```python
import numpy as np
a = np.array(range(1,19)).reshape(2,3,3)
print(a)
sum = 0
x = a[:,:,1].reshape(-1)
print(x)

for i in x:
    sum += i
print(sum) # 57
```

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

 [[10 11 12]
  [13 14 15]
  [16 17 18]]]
[ 2  5  8 11 14 17]
57
```

In [68]:
```python
import numpy as np
a = np.array([1,2,3,4,5])
b = np.array([6,7,8,10])
print(a.shape) # (5,)

n = np.concatenate((a,b))
print(n) # [ 1  2  3  4  5  6  7  8 10
print(n.shape) # (9,)
```

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

In [69]:
```python
import numpy as np
a = np.array([[1,2,3],[4,5,6],[7,8,9]])
b = np.array([[1,2,3],[4,5,6],[7,8,9]])
n = np.concatenate((a,b))
print(n)
# [[1 2 3]
#  [4 5 6]
#  [7 8 9]
#  [1 2 3]
#  [4 5 6]
#  [7 8 9]]
print(n.shape) # (6, 3)
```

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

In [72]:
```
import numpy as np
a = np.array([[1,2,3],[4,5,6],[7,8,9]])
b = np.array([[1,2,3],[4,5,6],[7,8,9]])
n = np.concatenate((a,b),axis = 1)
print(n)
# [[1 2 3 1 2 3]
#  [4 5 6 4 5 6]
#  [7 8 9 7 8 9]]
print(n.shape) # (3, 6)
```

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

In [ ]:
```
 
```

In [ ]:
```
 
```

In [ ]:
```
 
```