

```
import numpy as np
```

Mathematical Operations In Numpy

```
a = np.arange(0,18).reshape((6,3))
b = np.arange(20,38).reshape((6,3))
```

```
print(a)
print(b)
```

```
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]
 [12 13 14]
 [15 16 17]]
[[20 21 22]
 [23 24 25]
 [26 27 28]
 [29 30 31]
 [32 33 34]
 [35 36 37]]
```

Addition

```
a+b
```

```
array([[20, 22, 24],
       [26, 28, 30],
       [32, 34, 36],
       [38, 40, 42],
       [44, 46, 48],
       [50, 52, 54]])
```

```
np.add(a,b)
```

```
array([[20, 22, 24],
       [26, 28, 30],
       [32, 34, 36],
       [38, 40, 42],
       [44, 46, 48],
       [50, 52, 54]])
```

Subtraction

```
a-b
```

```
array([[ -20,  -20,  -20],
       [ -20,  -20,  -20],
       [ -20,  -20,  -20],
       [ -20,  -20,  -20],
       [ -20,  -20,  -20],
       [ -20,  -20,  -20]])
```

```
np.subtract(a,b)
```

```
array([[ -20,  -20,  -20],
       [ -20,  -20,  -20],
       [ -20,  -20,  -20],
       [ -20,  -20,  -20],
       [ -20,  -20,  -20],
       [ -20,  -20,  -20]])
```

Multiplication

```
a * b
```

```
array([[ 0, 21, 44],
       [69, 96, 125],
```

```
[156, 189, 224],
[261, 300, 341],
[384, 429, 476],
[525, 576, 629]])
```

```
np.multiply(a,b)
```

```
array([[ 0, 21, 44],
       [ 69, 96, 125],
       [156, 189, 224],
       [261, 300, 341],
       [384, 429, 476],
       [525, 576, 629]])
```

Division

```
a/b
```

```
array([[0.          , 0.04761905, 0.09090909],
       [0.13043478, 0.16666667, 0.2          ],
       [0.23076923, 0.25925926, 0.28571429],
       [0.31034483, 0.33333333, 0.35483871],
       [0.375       , 0.39393939, 0.41176471],
       [0.42857143, 0.44444444, 0.45945946]])
```

```
np.divide(a,b)
```

```
array([[0.          , 0.04761905, 0.09090909],
       [0.13043478, 0.16666667, 0.2          ],
       [0.23076923, 0.25925926, 0.28571429],
       [0.31034483, 0.33333333, 0.35483871],
       [0.375       , 0.39393939, 0.41176471],
       [0.42857143, 0.44444444, 0.45945946]])
```

Matrix Multiplication

```
print(a.shape)
```

```
print(b.shape)
```

```
(6, 3)
(6, 3)
```

```
b = b.reshape(3,6)
```

```
print(b.shape)
```

```
(3, 6)
```

```
a @ b
```

```
array([[ 90,  93,  96,  99, 102, 105],
       [ 324, 336, 348, 360, 372, 384],
       [ 558, 579, 600, 621, 642, 663],
       [ 792, 822, 852, 882, 912, 942],
       [1026, 1065, 1104, 1143, 1182, 1221],
       [1260, 1308, 1356, 1404, 1452, 1500]])
```

Min - Max - ArgMax - ArgMin

```
b.max()
```

```
37
```

```
b.min()
```

```
20
```

```
b.argmax()
```

```
17
```

```
b.argmax()
```

```
0
```

Sum of Array - Sum of Rows - Sum of Cols

```
np.sum(b)
```

```
513
```

```
# sum of columns
```

```
np.sum(b,axis = 0)
```

```
array([78, 81, 84, 87, 90, 93])
```

```
# sum of rows
```

```
np.sum(b,axis = 1)
```

```
array([135, 171, 207])
```

Mean - Square Root - Standard Deviation - Log

```
np.mean(b)
```

```
28.5
```

```
np.std(b)
```

```
5.188127472091127
```

```
np.log(b)
```

```
array([[2.99573227, 3.04452244, 3.09104245, 3.13549422, 3.17805383,
        3.21887582],
       [3.25809654, 3.29583687, 3.33220451, 3.36729583, 3.40119738,
        3.4339872 ],
       [3.4657359 , 3.49650756, 3.52636052, 3.55534806, 3.58351894,
        3.61091791]])
```

```
np.sqrt(b)
```

```
array([[4.47213595, 4.58257569, 4.69041576, 4.79583152, 4.89897949,
        5.                ],
       [5.09901951, 5.19615242, 5.29150262, 5.38516481, 5.47722558,
        5.56776436],
       [5.65685425, 5.74456265, 5.83095189, 5.91607978, 6.          ,
        6.08276253]])
```

Start coding or [generate](#) with AI.
