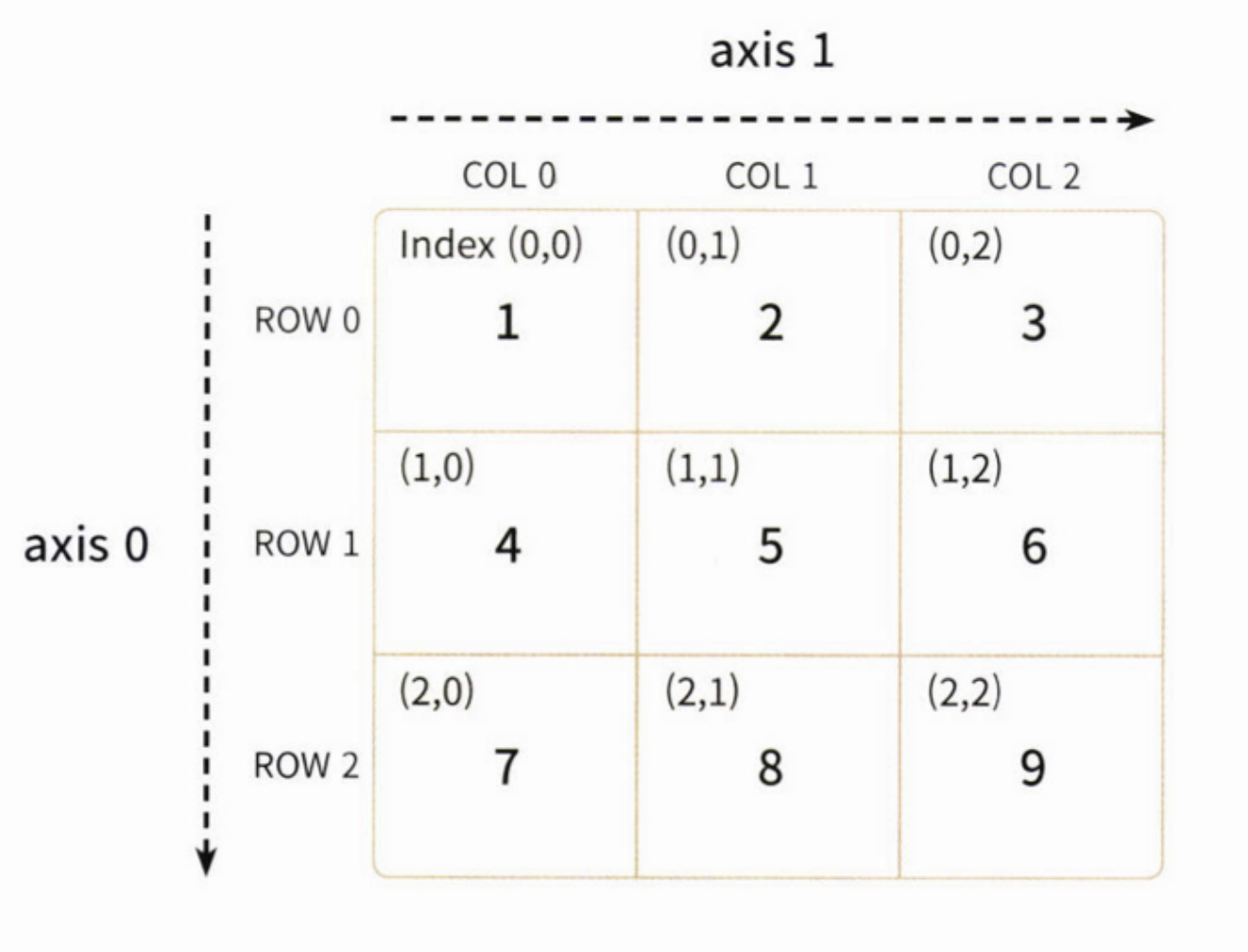

axis 1
COL 0
COL 1
COL 2
axis 0
ROW 0
ROW 1
ROW 2
Index (0,0)
1
(0,1)
2
(0,2)
3
(1,0)
4
(1,1)
5
(1,2)
6
(2,0)
7
(2,1)
8
(2,2)
9

넘파이의 sum 함수를 예로 axis의 값에 따라 어떻게 연산이 처리되는지를 시각화해 본다.

먼저 x는 다음과 같다.

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

위의 x를 행렬로 시각화 하면 다음과 같다.

이 x에 대한 axis=0으로 한 sum 함수에 대한 코드는 다음과 같으며 그 결과는 바로 다음의 그림과 같다.

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

← 행 벡터 방향(수직 방향)

이 x에 대한 axis=1으로 한 sum 함수에 대한 코드는 다음과 같으며 그 결과는 바로 다음의 그림과 같다.

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

← 열 벡터 방향(수평 방향)

2차원의 행렬이 있다고 해본다.

In [7]:
```
import numpy as np

array = np.random.randint(10, size=(5,3))
array
```

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

여기서 axis = 0은 무엇일까? 5,3 행렬에서 처음 5라고 생각하면되고, 다섯개의 원소를 죄다 더한다고 보면 된다.

In [8]:
```
np.sum(array, axis = 0)
```

Out[8]:
```
array([25, 21, 25])
```

더했기 때문에, 다섯개는 모두 더해지고 세개만 남는다.

만약 axis = 1로 놓는다면, 이제는 (5,3) 중 3이 합쳐지고 다섯개의 원소가 남는다.

In [11]:
```
np.sum(array, axis = 1)
```

Out[11]:
```
array([23,  6, 22, 13,  7])
```

이차원의 데이터는 이해하기 쉬운데, 삼차원의 데이터는 어떨까?

In [13]:
```
array3 = np.random.randint(10, size=(5,3,2))
array3
```

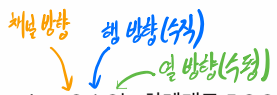
Out[13]:
```
array([[[9, 4],
        [9, 7],
        [5, 2]],

       [[8, 9],
        [6, 4],
        [0, 9]],

       [[9, 4],
        [3, 9],
        [0, 8]],

       [[4, 6],
        [0, 2],
        [3, 6]],

       [[7, 7],
        [4, 1],
        [8, 5]]])
```



이제, 세가지의 데이터 방향이 있다. np.sum에서 axis = 0,1,2는 차례대로 5,3,2 데이터를 하나로 합쳐준다. 합계를 구하는 간단한 np.sum은 0은 5를 없애고 3,2를 남기고, 1은 3을, 2는 2를 없애고 나머지 차원을 살린다.

In [23]:
```
print(np.sum(array3, axis = 0))
print("")
print(np.sum(array3, axis = 0).shape)
```

```
[[37 30]
 [22 23]
 [16 30]]

(3, 2)
```

In [24]:
```
print(np.sum(array3, axis = 1))
print("")
print(np.sum(array3, axis = 1).shape)
```

```
[[23 13]
 [14 22]
 [12 21]
 [ 7 14]
 [19 13]]

(5, 2)
```

In [25]:
```
print(np.sum(array3, axis = 2))
print("")
print(np.sum(array3, axis = 2).shape)
```

```
[[13 16  7]
 [17 10  9]
 [13 12  8]
 [10  2  9]
 [14  5 13]]

(5, 3)
```