

넌넌한 교수의 기초 파이썬

19_3. NumPy: 배열 인덱싱

창원대학교 정보통신공학과 박동규

동영상 강의자료와 코드

- 동영상 강의(넬넬한 교수의 기초 파이썬)
 - <https://www.youtube.com/watch?v=TK6RLGaM3Ao&list=PL2P1Vm9k53HOEn-QYXAfbIQWkR0OW9mKX>

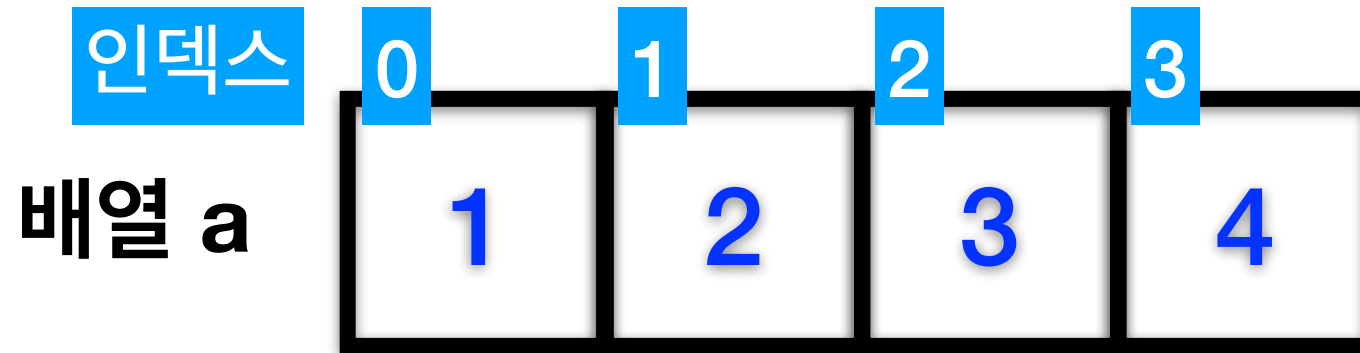


- 소스코드(GitHub)
 - <https://github.com/dongupak/Basic-Python-Programming>

배열의 인덱스

- numpy 배열에서 원소를 참조하기 위해서는 원소의 위치를 알아야 한다
- 이를 위해서는 인덱스를 사용한다
- 1차원 배열의 경우 인덱스 값이 파이썬 리스트의 인덱스 값과 동일하다(이전 강의 참조)
 - 인덱스를 사용하기 위해서는 배열이름[x]와 같이 대괄호를 사용
- 2차원 배열과 3차원 배열의 경우 인덱스를 각각 2개, 3개 사용하여 개별 원소를 참조할 수 있다

1차원 배열의 인덱스



```
[>>> a = np.array([1, 2, 3, 4])
>>> print(a[0])
1
>>> print(a[1])
2
>>> print(a[2])
3
>>> print(a[3])
4
>>> print(a[-1])
4
>>> print(a[-2])
3
>>>
```

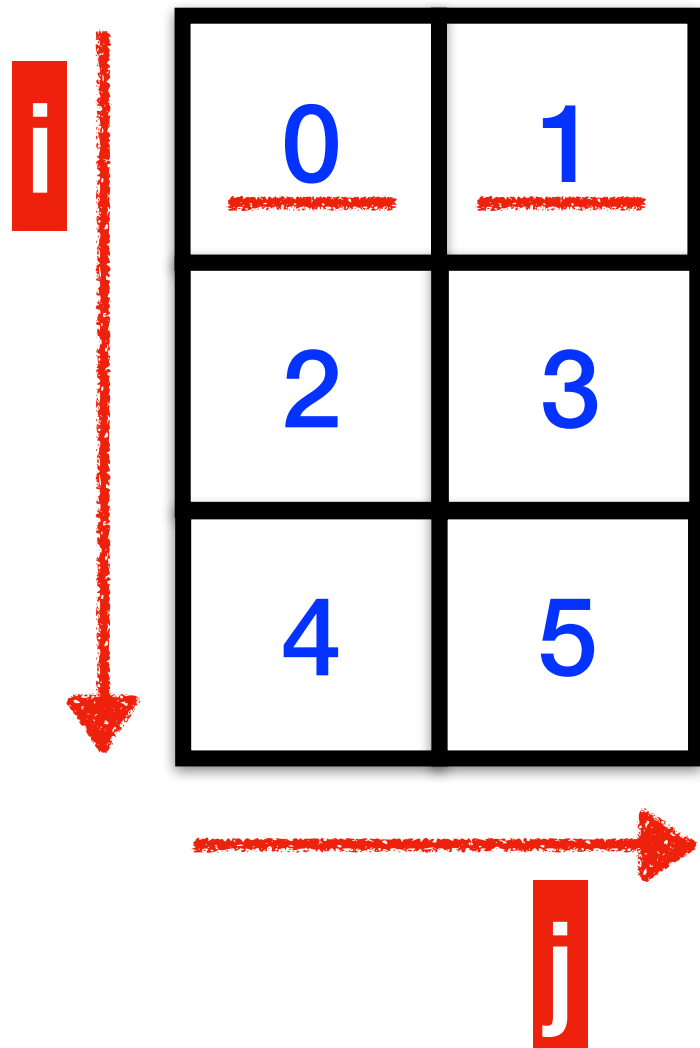
1차원 배열의 인덱스는
리스트의 인덱스와 동일

```
[>>> print(a[np.array([0,1])])
[1 2]
>>> print(a[np.array([0,2])])
[1 3]
>>> print(a[np.array([0,3])])
[1 4]
>>> print(a[np.array([1,3])])
[2 4]
>>> print(a[np.array([0,1,2])])
[1 2 3]
>>> print(a[np.array([0,1,3])])
[1 2 4]
>>>
```

여러개의 원소를 인덱싱

Lab

2차원 배열의 인덱스



```
>>> b = np.arange(0, 6).reshape(3, 2)
```

```
>>> b
```

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

```
>>> print(b[0, 0])
```

```
0
```

```
>>> print(b[0, 1])
```

```
1
```

```
>>> print(b[0, 2])
```

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

```
  File "<stdin>", line 1, in <module>
```

```
IndexError: index 2 is out of bounds for axis 1 with size 2
```

```
>>> print(b[1, 0])
```

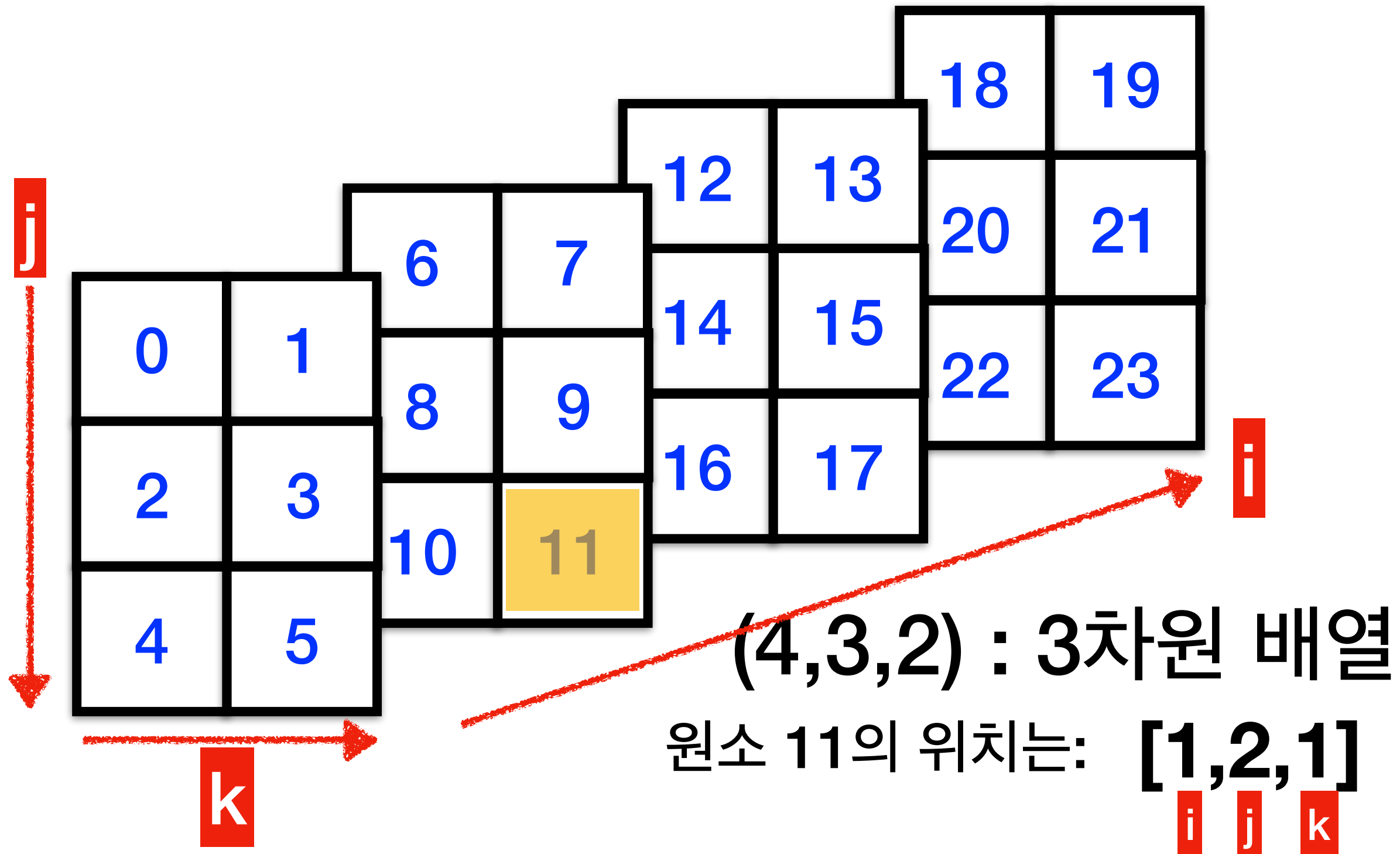
```
2
```

```
>>> print(b[1, 1])
```

```
3
```

```
>>>
```

3차원 배열의 인덱스



Lab

```
[>>> c = np.arange(0, 24).reshape(4, 3, 2)
[>>> c
array([[[ 0,  1],
        [ 2,  3],
        [ 4,  5]],

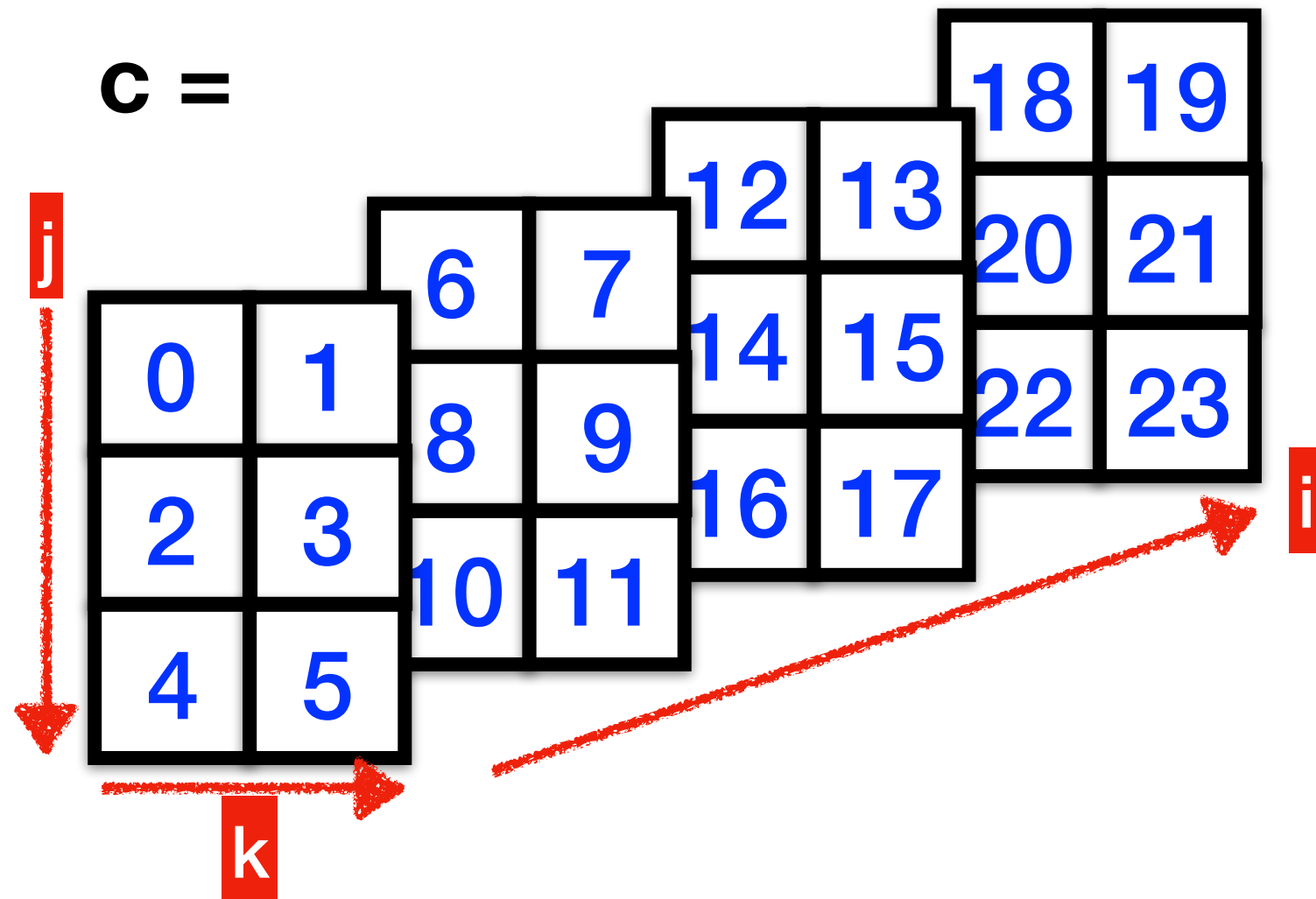
       [[ 6,  7],
        [ 8,  9],
        [10, 11]],

       [[12, 13],
        [14, 15],
        [16, 17]],

       [[18, 19],
        [20, 21],
        [22, 23]])])
```

```
[>>> print(c[0, 0, 0])
0
[>>> print(c[0, 0, 1])
1
[>>> print(c[0, 1, 0])
2
[>>> print(c[1, 1, 0])
8
[>>> print(c[1, 1, 1])
9
[>>> print(c[1, 2, 1])
11
[>>> █
```


3차원 배열의 인덱스

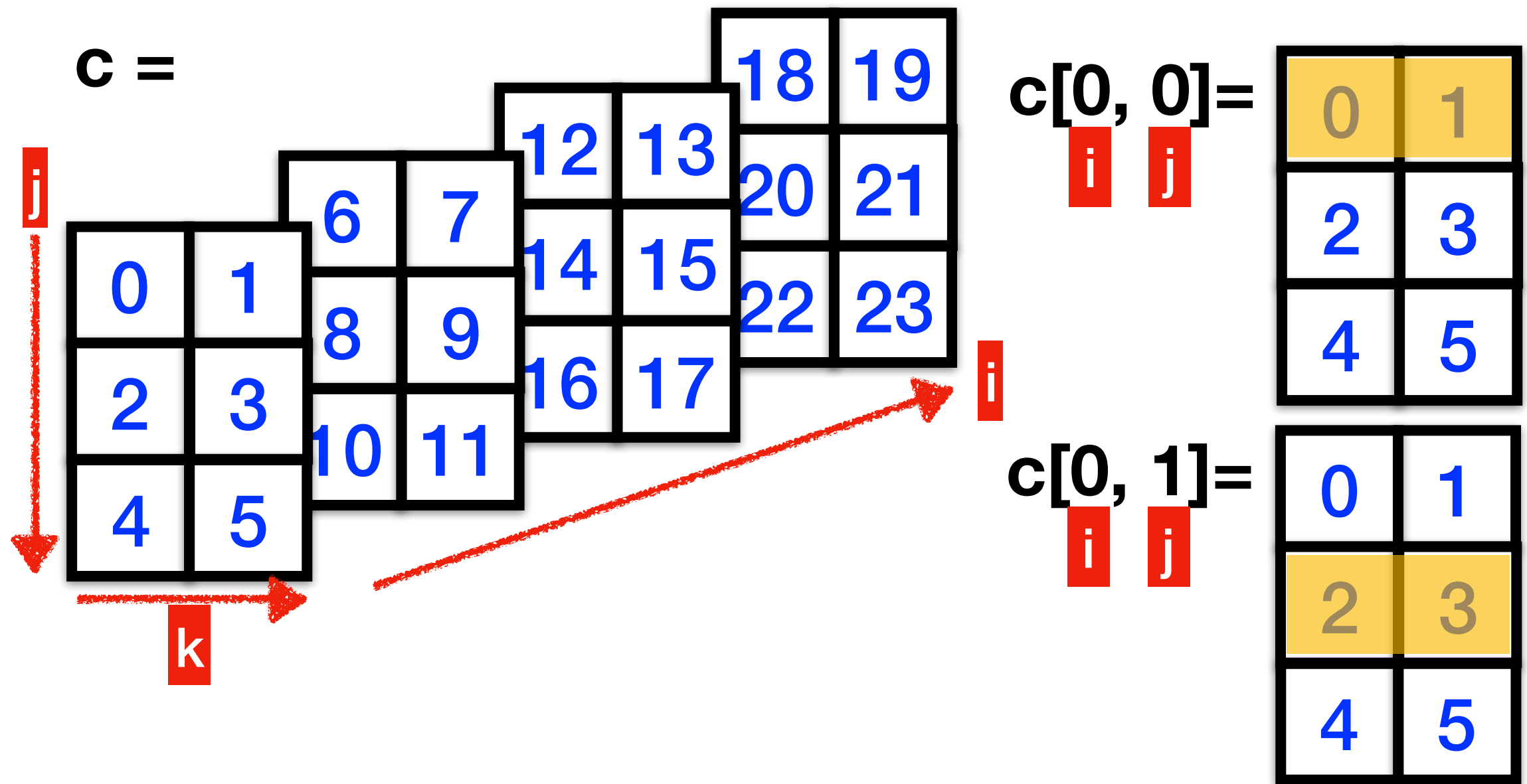


c[0]=
i

0	1
2	3
4	5

[3, 2] 행렬을 가져온다

3차원 배열의 인덱스



```
import numpy as np

c = np.arange(0, 24).reshape(4, 3, 2)

print('c[0]=', c[0])
print('c[0, 0] =', c[0, 0])
print('c[1, 0] =', c[0, 1])
```

수행결과

```
c[0]= [[0 1]
       [2 3]
       [4 5]]
c[0, 0] = [0 1]
c[1, 0] = [2 3]
```

Process finished with exit code 0

Lab

감사합니다