

01_Numpy 기초

• 배열 인덱싱(Boolean Indexing) 논리 인덱싱

02

> Bool 타입을 사용한 인덱싱

```
# True, False
```

```
print(arr[[True, False, True, True, False]])
```

 요소의 갯수, shape은 꼭 같아야 함

```
# 기본 데이터
```

true 인 요소만 인덱싱 가능

```
arr = np.array([1,2,3,4,5])
```

Out : [1 3 4]

```
# 조건
```

비교연산자가 있으면 불리언 인덱싱 가능

```
print(arr[arr >= 3])
```

Out : [3 4 5]

• 배열 인덱싱(Fancy Indexing)

03

> 인덱스의 값을 가지는 배열을 통해 인덱싱

```
a = np.arange(1, 11)
b = np.array([3, 4, 5, 6])
print(a)
print(b)
```

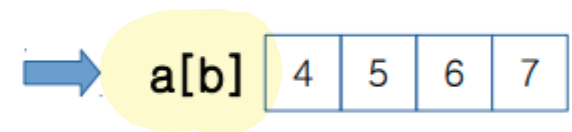
```
Out: [ 1  2  3  4  5  6  7  8  9 10]
      [3  4  5  6]
```

```
print(a[b])
```

```
Out: [4 5 6 7]
```

a	0	1	2	3	4	5	6	7	8	9
	1	2	3	4	5	6	7	8	9	10

b	3	4	5	6
---	---	---	---	---



배열 요소가 인덱싱을 위한 자리번호처럼 됨

Numpy 기초 연습문제

• 배열 인덱싱 불리언 인덱싱

> [문제2]

```
score = np.array([[78, 91, 84, 89, 93, 65],  
                  [82, 87, 96, 79, 91, 73]])
```

위와 같은 2x6 형태의 2차원 배열을 생성하고
각각 0행과 1행에서 90이상인 숫자를 추출해봅시다.

<전 체>

```
yes=(score>90)
```

```
score[score > 90]
```

0행에서

```
score[0][score[0] > 90]
```

1행에서

```
score[1][score[1] > 90]
```