

▼ Hello NumPy !

▼ OBJECTIVE: Familiarize yourself with Numpy

- This page was created for students to learn Python in the AI (717005) class at Hallym University.
- 본 페이지는 한림대학교 인공지능개론(717005) 수업에서 학생들의 Python 학습을 위해 만든 페이지입니다.

```
import numpy as np
```

```
a = np.array([1, 2, 3])
```

```
print(type(a))          # "<type 'numpy.ndarray' >"
print(a.shape)          # "(3,)"
print(a[0], a[1], a[2]) # "1 2 3"
a[0] = 5
print(a)                # "[5, 2, 3]"
```

```
↳ <type 'numpy.ndarray'>
   (3,)
   (1, 2, 3)
   [5 2 3]
```

```
b = np.array([[1,2,3],[4,5,6]]) # Create an array with a rank of 2 (rank가 2인 배열 생성)
print(b)
```

```
↳ [[1 2 3]
    [4 5 6]]
```

```
print(b.shape)          # "(2, 3)"
```

```
↳ (2, 3)
```

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

```
↳ 1
```

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

```
↳ 5
```

```
print(b[1, 2])
```

```
↳ 6
```

▼ Axis / axes

- the nth coordinate to index an array in Numpy.
- multidimensional arrays can have one index per axis.

```
import numpy as np
```

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

```
[[1 2]
 [3 4]]
```

- If not specified, the overall mean will be obtained (지정하지 않으면 전체 평균을 구하게 됨)

```
print np.mean(a) # 2.5
```

```
2.5
```

Axis 0 (↓)

```
print np.mean(a, axis=0) # [ 2.  3.]
```

```
[2. 3.]
```

Axis 1 (→)

```
print np.mean(a, axis=1) # [ 1.5  3.5]
```

```
[1.5 3.5]
```

▼ Broadcast

- Calculate arrays with different shapes 형상이 다른 배열을 계산하기 위해서 지원하는 기능

```
A = np.array([[1, 2], [3, 4]])
B = np.array([10, 20])
print(A)
print('-----')
print(B)
```

```
[[1 2]
 [3 4]]
-----
[10 20]
```

Please observe how it is multiplied. (어떻게 곱해지는지 잘 관찰바랍니다.)

```
print(A*B)
```

```
[[10 40]
 [30 80]]
```

▼ Stack

```
a = np.array([1,2,3,4])  
b = np.array([5,6,7,8])
```

- stack vertically (세로로 쌓기)

```
c = np.vstack((a,b))  
print(c)
```

```
↳   
[[1 2 3 4]  
 [5 6 7 8]]
```

```
print(c.shape)
```

```
↳ (2, 4)
```

- stack horizontally (가로로 쌓기)

```
d = np.hstack((a,b))  
print(d)
```

```
↳ [1 2 3 4 5 6 7 8]
```

```
print(d.shape)
```

```
↳ (8,)
```