

## **02 최소한의 도구로 딥러닝을 시작합니다**

## 02-2 딥러닝을 위한 도구들을 알아봅니다

### 파이썬 리스트 복습

```
my_list = [10, 'hello list', 20]  
print(my_list[1])
```

```
hello list
```

실행 결과입니다.

왼쪽에 있는 코드를 코랩에  
입력하여 실행해 보세요!

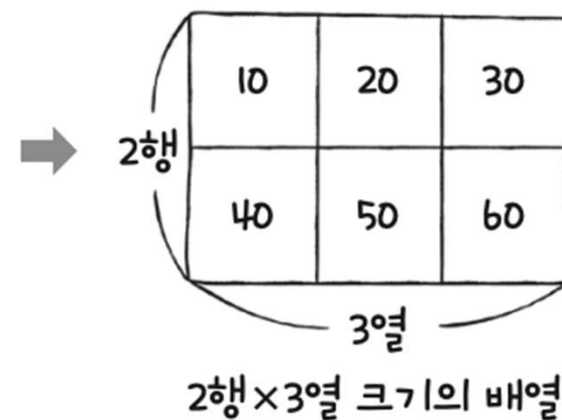
# 2차원 배열

2 x 3 크기, 2개의 행과 3개의 열

```
my_list_2 = [[10, 20, 30], [40, 50, 60]]
```

```
print(my_list_2[1][1])  
50
```

[[10, 20, 30],  
[40, 50, 60]]



# 코랩에서 넘파이 임포트하고 버전 확인하기

코랩에 이미 넘파이가 설치되어 있습니다

```
import numpy as np  
print(np.__version__)  
1.16.3
```

현재는 1.16.5

# 넘파이로 배열 만들기

array( ) 함수로 2차원 배열 만들기

```
my_arr = np.array([[10, 20, 30], [40, 50, 60]])  
print(my_arr)  
[[10 20 30]  
 [40 50 60]]
```

# type( ) 함수로 넘파이 배열인지 확인하기

코드 셀의 마지막 줄은 print( ) 함수를 사용하지 않아도 자동으로 그 결과가 출력됩니다

```
type(my_arr)  
numpy.ndarray
```

# 넘파이 배열에서 요소 선택하기

리스트와 마찬가지로 인덱스는 0부터 시작합니다

```
my_arr[0][2]  
30
```

10	20	30
40	50	60

# 넘파이 내장 함수 사용하기

```
np.sum(my_arr)  
210
```

10	20	30
40	50	60

▶ 1 np.

🔧 abs  
🔧 absolute  
🔧 absolute\_import  
🔧 add  
🔧 add\_docstring  
📦 add\_newdoc  
🔧 add\_newdoc\_ufunc  
📦 alen  
📦 all  
📦 allclose  
🔧 ALLOW\_THREADS  
📦 alltrue

시작하기

지금 읽고 계신

예를 들어 다음:

[ ] 1 sec  
2 sec

86400

np.abs(\*args, \*\*kwargs) ⓘ

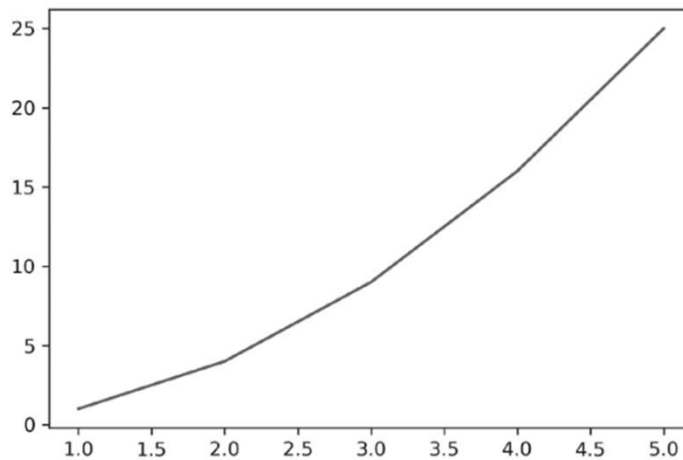


# 맷플롯립으로 그래프 그리기

`plot(x, y)`

```
import matplotlib.pyplot as plt
```

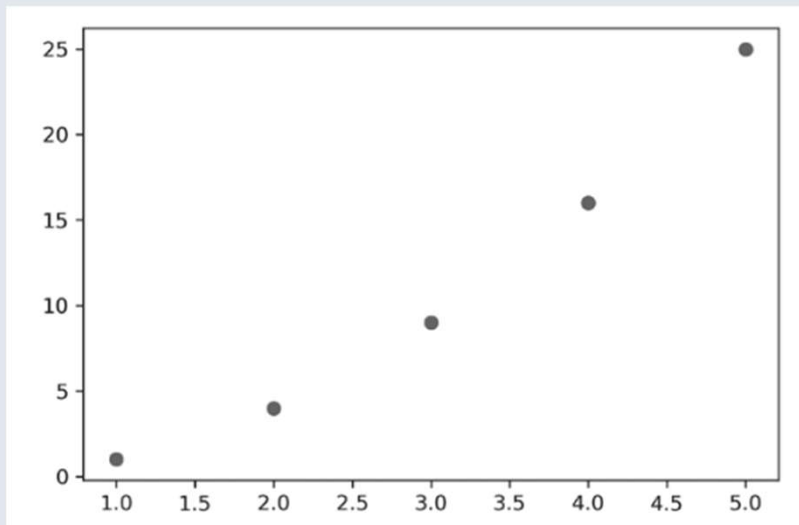
```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 25]) # x축, y축의 값을 파이썬 리스트로 전달합니다.  
plt.show( )
```



# 산점도 그리기

`scatter(x, y)`

```
plt.scatter([1, 2, 3, 4, 5], [1, 4, 9, 16, 25])  
plt.show( )
```



# 넘파이 배열로 산점도 그리기

```
x = np.random.randn(1000) # 표준 정규 분포를 따르는 난수 1,000개를 만듭니다.  
y = np.random.randn(1000) # 표준 정규 분포를 따르는 난수 1,000개를 만듭니다.  
plt.scatter(x, y)  
plt.show( )
```

