

# Bool, 불 타입

## bool 타입이란?

bool type은 참 True와 거짓 False를 나타내는 자료형입니다.

bool 자료형은 참과 거짓 2가지 값만을 가질 수 있습니다.

```
print(2 > 5) # False
```

2는 5보다 큰 값이 아니기에 거짓(False)의 값이 나옵니다.

## Python False

파이썬에서 거짓의 값을 가지는 경우를 살펴보겠습니다.

- ''
- ""
- []
- list()
- ()
- tuple()
- {}
- set()
- dict()
- 0
- None
- False

이러한 경우에는 Python에서는 False의 값을 가집니다. (cf. 더 있지만, 주로 쓰는 부분만 다루었습니다.)

```
print(bool('')) # False
print(bool("")) # False
print(bool([])) # False
print(bool(list())) # False
print(bool(())) # False
print(bool(tuple())) # False
print(bool({})) # False
print(bool(set())) # False
print(bool(dict())) # False
print(bool(0)) # False
print(bool(None)) # False
print(bool(False)) # False
```

[], list(), (), tuple(), {}, set(), dict()는 추후 다른 데이터 타입입니다.

맞보기

[ ] 와 list() 는 리스트 자료형을 나타냅니다.

() 와 tuple() 은 튜플 자료형을 나타냅니다.

{ } 와 set() 은 집합 자료형을 나타냅니다.

{ } 와 dict() 은 딕셔너리(사전형) 자료형을 나타냅니다.

여기서, {}는 집합 자료형과 딕셔너리 자료형에 쓰인다고 나와있는데, 이도 추후 집합과 딕셔너리 자료형을 다루다보면 차이점을 알 수 있으니 넘어가도록 하겠습니다.

## type()

위의 내용을 확인해보면 타입으로 값을 감싸면 해당 값으로 변하는 모습을 확인할 수 있습니다.

type(값) 을 쓰게 된다면 해당 값의 자료형을 파악할 수 있습니다.

```
print(type(5)) # <class 'int'>
print(type('5')) # <class 'str'>
```

프로그래밍을 하다보면 웹이나 엑셀과 같은 다른 곳에서 자료를 가져와서 사용해야할 때가 있습니다.

내가 필요한 데이터 타입은 숫자형인데 문자형을 가져와서 곤란한 경우가 있습니다.

아래의 예시를 들어보겠습니다.

```
crawling_data = '20200525'
print(type(crawling_data)) # <class 'str'>
print(type(int(crawling_data))) # <class 'int'>
```

문자열이 숫자형태로 변환이 가능할 때, 정수형으로 바꾸고 싶다면 int() 를 사용하며, 실수형으로 변경하고 싶다면 float() 를 사용하여 주시면 됩니다.

숫자를 문자열로 바꾸고 싶을 때는 str() 을 이용하시면 됩니다.

```
crawling_data = 20200525
conversion_str_data = str(crawling_data)

print(conversion_str_data) # 20200525
print(type(conversion_str_data)) # <class 'str'>
```