

리스트

파이썬 리스트에 있는 데이터에 하나씩 접근할 때는 인덱싱을 사용하면 됩니다. 그런데 리스트에 있는 여러 개의 데이터에 동시에 접근하려면 어떻게 해야 할까요? 이럴 때 사용하는 것이 바로 파이썬 슬라이싱입니다. 파이썬 리스트의 슬라이싱은 문자열 슬라이싱과 동일합니다.

```
kospi_top10 = ['삼성전자', 'SK 하이닉스', '현대차', '한국전력',  
'아모레퍼시픽', '제일모직', '삼성전자우', '삼성생명', 'NAVER', '현대모비스']
```

```
print("시가총액 5 위: ", kospi_top10[4])
```

```
>>> 시가총액 5 위: 아모레퍼시픽
```

```
kospi_top10[5:10]
```

```
>>> ['제일모직', '삼성전자우', '삼성생명', 'NAVER', '현대모비스']
```

6위부터 9위까지 종목을 가져와야하는 경우

[5:9] 또는 [5:-1] 사용

리스트에 데이터 삽입하기

append

```
kospi_top10.append('SK 텔레콤')
```

```
kospi_top10
```

```
>>> ['삼성전자', 'SK 하이닉스', '현대차', '한국전력', '아모레퍼시픽',  
'제일모직', '삼성전자우', '삼성생명', 'NAVER', '현대모비스', 'SK 텔레콤']
```

insert

```
kospi_top10.insert(3, 'SK 텔레콤')
```

```
kospi_top10
```

```
['삼성전자', 'SK 하이닉스', '현대차', 'SK 텔레콤', '한국전력',  
'아모레퍼시픽', '제일모직', '삼성전자우', '삼성생명', 'NAVER', '현대모비스']
```

리스트에 데이터 삭제

del

```
kospi_top10[-1]
```

```
>>> '현대모비스'
```

```
del kosp_i_top10[-1]
kosp_i_top10
>>> ['삼성전자', 'SK 하이닉스', '현대차', 'SK 텔레콤', '한국전력',
'아모레퍼시픽', '제일모직', '삼성전 자우', '삼성생명', 'NAVER']
```

튜플

- 리스트는 '[' 와 ']'를 사용하는 반면 튜플은 '('와 ')'를 사용한다.
- 리스트는 리스트 내의 원소를 변경할 수 있지만 튜플은 변경할 수 없다.

```
t = ('Samsung', 'LG', 'SK')
t
>>> ('Samsung', 'LG', 'SK')
```

튜플 슬라이싱

```
t[0:2]
t
>>> ('Samsung', 'LG')
```

여기서 한 가지 주의할 점은 튜플을 생성할 때는 '('와 ')' 기호를 사용하지만 데이터에 접근하는 인덱싱이나 슬라이싱에서는 데이터의 범위를 '['와 ']' 기호로 표현한다는 점입니다. 즉, 다음 코드는 파이썬 문법에 어긋나기 때문에 에러가 발생합니다.

딕셔너리

키와 밸류로 이루어진 자료형
딕셔너리는 인덱스 개념이 없음

```
cur_price = {}
type(cur_price)
>>> <class 'dict'>

cur_price['daeshin'] = 30000
```

```
cur_price
>>> {'daeshin': 30000}
```

딕셔너리에 데이터 삽입 및 삭제

삽입

```
cur_price['Daum KAKAO'] = 80000
cur_price
{'Daum KAKAO': 80000, 'daeshin': 30000}

cur_price['naver'] = 800000
cur_price
>>> {'Daum KAKAO': 80000, 'naver': 800000, 'daeshin': 30000}
```

삭제

```
del cur_price['daeshin']
cur_price
>>> {'naver': 800000, 'Daum KAKAO': 80000}
```

딕셔너리로부터 키-값 구하기

키 값만 구하려면

"딕셔너리이름.keys()"라고 호출

```
>>> cur_price = {'Daum KAKAO': 80000, 'naver':800000, 'daeshin':30000}
>>> cur_price.keys()
dict_keys(['naver', 'Daum KAKAO', 'daeshin'])

>>> list(cur_price.keys())
['naver', 'Daum KAKAO', 'daeshin']

>>> 'Samsung' in cur_price.keys()
False

>>> 'naver' in cur_price.keys()
True
```