

데이터 자료형 (복합자료형)

오 동 인

경희대학교 의과대학 의공학교실

리스트

- 리스트 = 목록 : 값을 일렬로 늘어놓은 형태
- 변수에 값을 저장할 때 `[]`(대괄호)로 묶어 줌
- 각 값은 `,`(кома)로 구분
- **리스트 = `[값, 값, 값]`**
- 리스트에 저장된 각각의 값 = 요소(element)
- 문자열, 정수, 실수, 불 등 모든 자료형 저장 가능
- 자료형을 섞어서 저장 가능
- **리스트 = `[]`**
- **리스트 = `list()`**
- **리스트 = `list(range(횟수))`**
- **리스트 = `list(range(시작, 끝))`**
- **리스트 = `list(range(시작, 끝, 증가폭))`**

```
a = [38, 21, 53, 62, 19]
```

```
person = ['james', 17, 175.3, True]
```

```
a = []
```

```
b = list()
```

```
a = list(range(10))
```

```
b = list(range(5, 12))
```

```
c = list(range(-4, 10, 2))
```

```
d = list(range(10, 0, -1))
```

튜플

- 리스트처럼 요소를 일렬로 저장
- 저장된 **요소를 변경, 추가, 삭제를 할 수 없음**
- 읽기 전용 리스트
- **()**(괄호)로 묶어 줌
- **,**(кома)로 구분
- 괄호로 묶지 않고 값만 콤마로 구분해도 튜플이 됨
- **튜플 = (값, 값, 값)**
- **튜플 = 값, 값, 값**
- 요소가 절대 변경되지 않고 유지되어야 할 때 사용
- **튜플 = (값,)**
- **튜플 = 값,**
- 튜플 ↔ 리스트 변환 가능

```
a = (38, 21, 53, 62, 19)
a = 38, 21, 53, 62, 19
person = ('james', 17, 175.3, True)
```

```
(38,)
38,
```

```
a = [1, 2, 3]
tuple(a)
```

```
b = (4, 5, 6)
list(b)
```

```
list('Hello')
tuple('Hello')
```

딕셔너리

- 연관된 값을 묶어서 저장하는 용도
- 값마다 이름을 붙여서 저장하는 방식
- 사전(dictionary)에서 단어를 찾듯 값을 참조
- **{ }**(중괄호) 안에 **키: 값** 형식으로 저장
- 키를 먼저 지정하고 **:**(콜론)을 붙여서 값 표현
- 각 키와 값은 **,**(콤마)로 구분
- **딕셔너리 = {키1: 값1, 키2: 값2}**
- 키-값 쌍(key-value pair) 1:1 대응
- 키가 중복되면 가장 뒤에 있는 값만 사용
- 키는 문자열뿐만 아니라 정수, 실수, 불 자료형도 사용 가능
- 값에는 리스트, 딕셔너리 등을 포함 모든 자료형 사용
- 키에는 리스트와 딕셔너리를 사용할 수 없음
- **딕셔너리 = { }**
- **딕셔너리 = dict()**

Price = [1500, 3500, 800, 1100]

Price = {'apple': 1500, 'banana': 3500, 'lemon': 800, 'grape': 1100}

Price = {'apple': 1500, 'apple': 2500, 'banana': 3500, 'lemon': 800, 'grape': 1100}

X = {100: 'hundred', False: 0, 3.5: [3.5, 3.6]}

x = {[10, 20]: 100}

x = {'a': 10}: 100}

딕셔너리 만들기

- **딕셔너리 = dict(키1=값1, 키2=값2)**
- dict에서 **키=값** 형식으로 딕셔너리 생성. 키에 ' '(작은따옴표)나 " "(큰따옴표)를 사용하지 않음. 키는 딕셔너리를 만들고 나면 문자열로 변경됨.
- **딕셔너리 = dict(zip([키1, 키2], [값1, 값2]))**
- dict에서 zip 함수를 이용. 키가 들어있는 리스트와 값이 들어있는 리스트를 차례대로 zip에 넣은 뒤 다시 dict에 입력함.
- **딕셔너리 = dict([(키1, 값1), (키2, 값2)])**
- 리스트 안에 (키, 값) 형식의 튜플을 나열하는 방법
- **딕셔너리 = dict({키1: 값1, 키2: 값2})**
- dict 안에서 중괄호로 딕셔너리를 생성하는 방법

```
Price1 = dict(apple=1500, banana=3500, lemon=800, grape=1100)
```

```
Price2 = dict(zip(['apple', 'banana', 'lemon', 'grape'], [1500, 3500, 800, 1100]))
```

```
Price3 = dict([('apple', 1500), ('banana', 3500), ('lemon', 800), ('grape', 1100)])
```

```
Price4 = dict({'apple': 1500, 'banana': 3500, 'lemon': 800, 'grape': 1100})
```

딕셔너리 참조

- 딕셔너리의 키에 접근할 때는 딕셔너리 뒤에 [](대괄호)를 사용하며 [] 안에 키를 지정
 - **새로운 값 할당: 딕셔너리[키] = 값**
 - 딕셔너리는 없는 키에 값을 할당하면 해당 키가 추가되고 값이 할당
 - 딕셔너리는 없는 키에서 값을 가져오려고 하면 에러 발생
 - 딕셔너리 키 여부 확인
 - **키 in 딕셔너리**
 - **키 not in 딕셔너리**
 - 딕셔너리의 키 개수
 - **len(딕셔너리)**
- ```
Price = {'apple': 1500, 'banana': 3500, 'lemon': 800, 'grape': 1100}
print(Price['apple'])

Price['apple'] = 2500
Price['grape'] = 900
print(Price)

Price['berry'] = 3800

'apple' in Price
'pineapple' in Price

len(Price)
```

# EOD