

2. PYTHON 기초, 자료형

KOREA DIGITAL MEDIA HIGH SCHOOL 2020 PYTHON LECTURE



2. PYTHON 기초, 자료형

- 학습주제
 - 자료형이란 무엇인가?
 - PYTHON의 자료형의 종류를 알아보자

2-4. 튜플 자료형

- 튜플(tuple)은 리스트(list)와 거의 비슷하다.
 - 리스트는 []로 둘러싸지만 튜플은 ()로 둘러싼다.
 - 리스트는 그 값의 생성, 삭제, 수정이 가능하지만 튜플은 그 값을 바꿀 수 없다.

- 튜플의 모습

T1=()

T2=(1,)

T3=(1,2,3)

T4=1,2,3

T5=('a','b',('ab','cd'))

2-4. 튜플 자료형

- 튜플(tuple)의 인덱싱과 슬라이싱
 - 문자열, 리스트와 마찬가지로 `t1[0]`, `t2[1]`과 같이 인덱싱이 가능하다.
 - 문자열, 리스트와 마찬가지로 `t1[1:]`, `t2[1:3]`과 같이 슬라이싱이 가능하다.
 - 튜플의 더하기(+), 곱하기(*) 가능
 - 튜플의 길이 구하기 `len()`

2-5. 딕셔너리 자료형

- 딕셔너리란?
 - 대응관계를 나타낼 수 있는 자료형. (name='kim' , birth='20-05-01')
 - 연관배열(Associative array), 해시(Hash)
 - Key를 통해 value를 얻음.
 - 순차검색이 아니라 key값만 확인
- 딕셔너리의 모습
{key1:value1, key2:value2, key3:value3,...}

2-5. 딕셔너리 자료형

- 딕셔너리의 예

`dic={'name':'kim', 'phone':'01012345678', 'birth':'200426'}`

`A={1:'hi'}` key가 정수, value가 문자열

`B={'a':[1,2,3]}` value에 리스트 사용가능

2-5. 딕셔너리 자료형

- 딕셔너리 추가 삭제하기

- 추가하기

```
>>> a={1:'a'}  
>>> a[2]='b'  
>>> a['name']='kim'  
>>> a[3]=[1,2,3]
```

- 삭제하기

```
>>> del a[1]
```

2-5. 딕셔너리 자료형

- 딕셔너리 사용 유의사항
 - Key는 고유한 값이므로 중복되는 Key값을 설정하면 하나를 제외한 나머지 것들이 모두 무시된다.

```
>>> a={1:'a', 1:'b'}  
>>> a
```

- Key에 리스트를 사용할 수 없다.

2-5. 딕셔너리 자료형

- 딕셔너리 관련함수
 - keys() : 딕셔너리 key만을 모아서 객체를 반환함.
※ 2.7까지는 리스트로 반환했으나 3.0 이상에는 객체로 필요한 경우는 list()형변환 사용가능
 - values() : 딕셔너리 value만을 모어소 객체를 반환함.
 - items() : 딕셔너리 key, value 쌍을 튜플로 반환
 - clear() : 딕셔너리 요소를 모두 삭제
 - get() : 딕셔너리에서 key에 대응하는 value를 반환
※print(dic['key']) vs print(dic.get('key'))
 - in() : 해당키가 딕셔너리 안에 있는지 조사, key in dic

Thank You

KOREA DIGITAL MEDIA HIGH SCHOOL 2020 PYTHON LECTURE

