

집합(Set) 자료형

프리젠테이션 사용 불가

집합(Set) 자료형

- 집합(Set) 자료형은 집합 관련 처리를 쉽게 처리하기 위한 자료형이다.
- 집합 자료형은 {}(중괄호)로 감싸서 기술하며 값과 값은 ,(кома)로 구분하여 기술한다.

세트명 = { 값1, 값2, 값n, ... }

- 예)

```
s = { 1, 2, 3 }
print(s)
print(type(s))
```
- 다른 자료형의 객체를 집합 자료형으로 생성하고자 하는 경우 set 생성자 함수를 사용하여 생성한다.

```
numbers = [1, 2, 3]
s = set(numbers)
print(s)
print(type(s))
print(set('HELLO'))
```

{ 1, 2, 3 }
<class 'set'>

list 객체를 전달 인자로 하여 set 클래스의 생성자를 호출
{ 1, 2, 3 }
<class 'set'>
'HELLO' 문자열을 사용하여 집합 객체 {'E', 'O', 'H', 'L'}를 생성

집합(Set) 자료형의 특징

- 집합 자료형은 다음과 같은 특징을 가진다.
 - 중복을 허용하지 않는다.
중복을 허용하지 않는 특징 때문에 중복 제거를 위한 필터(filter)로 사용된다.
 - 내부적으로 정렬이 이뤄지지 않는다.(Unordered)
내부적으로 정렬이 이뤄지지 않는다는 것은 인덱싱이 불가능하다는 것을 의미한다. 그러므로 집합 자료형의 요소에 접근하기 위해서는 인덱싱이 가능한 자료형으로 형변환 후 참조해야 한다.
- 인덱싱이 가능한 자료형의 종류
 - 리스트 : list(집합자료형 변수) 예) l = list([1, 2, 3])
 - 튜플 : tuple(집합자료형 변수) 예) t = tuple([1, 2, 3])

집합 자료형에 대한 연산(교집합, 합집합, 차집합)

- 집합 자료형은 교집합, 합집합, 차집합을 구할 때 유용하다.

$A = \{ 1, 2, 3, 4, 5 \}$
 $B = \{ 3, 4, 5, 6, 7 \}$

- 교집합

함수를 사용한 연산

$A \& B$

`A.intersection(B)`

$\# \{ 3, 4, 5 \}$

- 합집합

$A | B$

`A.union(B)`

$\# \{ 1, 2, 3, 4, 5, 6, 7 \}$

- 차집합

$A - B$

`A.difference(B)`

$\# \{ 1, 2 \}$

$B - A$

`B.difference(A)`

$\# \{ 6, 7 \}$

집합 자료형 관련 함수 #1/2

- add() 함수 : 값 추가하기

```
s = { 1, 2, 3 }  
print(s)           # { 1, 2, 3 }  
s.add(4)  
print(s)           # { 1, 2, 3, 4 }
```

- update() 함수 : 여러 개의 값 추가하기

```
s = { 1, 2, 3 }  
print(s)           # { 1, 2, 3 }  
s.update([4, 5, 6])  
print(s)           # 리스트의 원소를 집합 자료형에 추가 { 1, 2, 3, 4, 5, 6 }  
s.update((7, 8, 9))  
print(s)           # 튜플의 원소를 집합 자료형에 추가 { 1, 2, 3, 4, 5, 6, 7, 8, 9 }  
s.update({7, 8, 9, 10})  
print(s)           # 집합 자료형의 값을 집합 자료형에 추가 { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }
```

집합 자료형 관련 함수 #2/2

- remove() 함수 : 특정 값 제거하기

```
s = { 1, 2, 3 }  
print(s)           # { 1, 2, 3 }  
s.remove(2)  
print(s)           # { 1, 3 }
```

- len() 함수 : 집합 자료형 값의 개수 구하기

```
s = { 1, 2, 3 }  
print(len(s))      # 3
```