

Lecture 1 : 증복과 순서가 없는 집합

1. 집합(Set)의 이해

- 원소는 유일하고 순서는 의미 없는 집합
- 집합의 원소는 정수, 실수, 문자열, 튜플 등 수정 불가능한(immutable) 것이어야 함

```
{ element-0, element-1, ... , element-n}
```

집합(set) 생성과 참조 / LAB-1

번호	사용 예	출력(결과)	설 명
1	num_set = {1, 1, 2, 2, 3, 4, 4, 5}		
	print(num_set)	{}	원소는 유일함
2	tpl_set = {(1, 1), (2, 4), (3, 9)}		
	print(tpl_set)	{(1, 1), (3, 9), (2, 4)}	튜플을 원소로 집합 생성
3	s1 = {[1, 1], [2, 4], [3, 9]}		
	print(s1)	unhashable type: 'list'	수정 불가능한(immutable) 객체만 원소로 가능
4	s = set()		
	print(s)	set()	공집합 만들기
5	s = set([1, 2, 3])		
	print(s)	{1, 2, 3}	리스트로 집합 만들기
6	s = set('java')		
	print(s)	{'j', 'v', 'a'}	문자열로 집합 만들기
7	s = set(range(5))		
	print(s)	{0, 1, 2, 3, 4}	문자열로 집합 만들기
8	for c in set('java') :		
	print(c, end=' ')	{'j', 'v', 'a'}	집합 원소 출력

LAB-1 집합 생성과 참조

```
In [ ]: num_set = {1, 1, 2, 2, 3, 4, 4, 5}
         print(num_set)
```

```
In [ ]: tpl_set = {(1, 1), (2, 4), (3, 9)}
        print(tpl_set)
```

```
In [ ]: s1 = {[1, 1], [2, 4], [3, 9]}
        print(s1)
```

```
In [ ]: s = set()
        print(s)
```

```
In [ ]: s = set([1, 2, 3])
        print(s)
```

```
In [ ]: s = set('java')
        print(s)
```

```
In [ ]: s = set(range(5))
        print(s)
```

```
In [ ]: for c in set('java') :
        print(c, end=' ')
```

2. 집합(set) 관련 메서드 / LAB-2

번호	사용 예	출력(결과)	설 명
1	p1 = {'python', 'kotlin'}		
	len(p1)	2	집합 p1의 원소 수
2	p1 = {'python', 'kotlin'}		
	'python' in p1	True	집합 p1에 원소 'python'이 있느냐?
3	odd = {1, 3, 5}		
	odd.add(7)		집합 odd에 원소 7 추가
	print(odd)	{1, 3, 5, 7}	
4	odd = {1, 3, 5}		
	odd.remove(3)		집합 odd에서 원소 3 제거
	print(odd)	{1, 5}	
5	odd = {1, 3, 5}		
	odd.pop()		집합 odd에서 임의의 원소 제거
	print(odd)	{3, 5}	
6	s1 = {1, 2, 3}		

번호	사용 예	출력(결과)	설 명
	s2 = s1		동일 객체 공유
	print(id(s1), id(s2))	2952276645632 2952276645632	객체의 주소 값 출력 (동일)
7	s1 = {1, 2, 3}		
	s2 = s1.copy()		객체 복사본 생성
	print(id(s1), id(s2))	2952276644512 2952276648544	객체의 주소 값 출력 (다름)
8	a = {1, 2, 3}		
	b = {2, 3, 4}		
	print(a b)	{1, 2, 3, 4}	집합 a와 집합 b의 합집합
	print(a.union(b))	{1, 2, 3, 4}	집합 a와 집합 b의 합집합
9	a = {1, 2, 3}		
	b = {2, 3, 4}		
	print(a - b)	{1}	집합 a와 집합 b의 차집합
	print(a.difference(b))	{1}	집합 a와 집합 b의 차집합
10	a = {1, 2, 3}		
	b = {2, 3, 4}		
	print(a & b)	{2, 3}	집합 a와 집합 b의 교집합
	print(a.intersection(b))	{2, 3}	집합 a와 집합 b의 교집합

LAB-2 집합(set) 관련 메서드

```
In [ ]: p1 = {'python', 'kotlin'}
len(p1)
```

```
In [ ]: p1 = {'python', 'kotlin'}
'python' in p1
```

```
In [ ]: odd = {1, 3, 5}
odd.add(7)
print(odd)
```

```
In [ ]: odd = {1, 3, 5}
odd.remove(3)
print(odd)
```

```
In [ ]: odd = {1, 3, 5}
odd.pop()
print(odd)
```

```
In [ ]: s1 = {1, 2, 3}
s2 = s1
print(id(s1), id(s2))
```

```
In [ ]: s1 = {1, 2, 3}
s2 = s1.copy()
print(id(s1), id(s2))
```

```
In [ ]: a = {1, 2, 3}
b = {2, 3, 4}
print(a | b)
print(a.union(b))
```

```
In [ ]: a = {1, 2, 3}
b = {2, 3, 4}
print(a - b)
print(a.difference(b))
```

```
In [ ]: a = {1, 2, 3}
b = {2, 3, 4}
print(a & b)
print(a.intersection(b))
```

LAB-3 집합(set)을 활용한 수행과제

1. 문제 : 고객 이메일 중복 제거

- 다음과 같은 이메일 리스트가 주어졌을 때, 중복을 제거하고 정렬된 리스트를 반환하는 프로그램을 작성하세요.

```
emails = [
    "alice@example.com", "bob@example.com", "alice@example.com",
    "charlie@example.com", "dave@example.com", "bob@example.com",
    "eve@example.com", "frank@example.com"
]
```

2. 요구사항:

- 중복을 제거하세요.
- 이메일을 알파벳 순서로 정렬하여 리스트로 반환하세요.

```
In [ ]: emails = [
    "alice@example.com", "bob@example.com", "alice@example.com",
    "charlie@example.com", "dave@example.com", "bob@example.com",
    "eve@example.com", "frank@example.com"
]
```

```
# 코딩하세요
#
#
print(unique_emails)
```

LAB-4 집합(set)을 활용한 수행과제

1. 문제 : 보안 시스템 - 허용된 IP vs 차단된 IP

- 한 회사의 네트워크 보안 시스템에서 특정 IP 주소 목록을 허용하고, 특정 IP는 차단해야 합니다.
- 다음과 같은 IP 주소 목록이 있습니다.

```
allowed_ips = {"192.168.1.1", "192.168.1.2", "192.168.1.3",
               "192.168.1.4"}
blocked_ips = {"192.168.1.3", "192.168.1.5"}
```

- 현재 접속을 시도하는 IP 리스트

```
current_access_ips = {"192.168.1.1", "192.168.1.3", "192.168.1.5",
                     "192.168.1.6"}
```

2. 요구사항:

- 허용된 IP 중에서 실제 접속 가능한 IP 목록을 출력하세요.
- 차단된 IP 목록을 출력하세요.
- 접속 시도했으나 목록에 없는 IP(미등록 IP)를 출력하세요.

```
In [ ]: allowed_ips = {"192.168.1.1", "192.168.1.2", "192.168.1.3", "192.168.1.4"}
        blocked_ips = {"192.168.1.3", "192.168.1.5"}
        current_access_ips = {"192.168.1.1", "192.168.1.3", "192.168.1.5", "192.168.1.6"}

# 코딩하세요
#
#
#

print("✅ 접속 허용된 IP:")
print(accessible_ips)

print("\n❌ 차단된 IP:")
print(denied_ips)

print("\n⚠️ 미등록 IP:")
print(unregistered_ips)
```