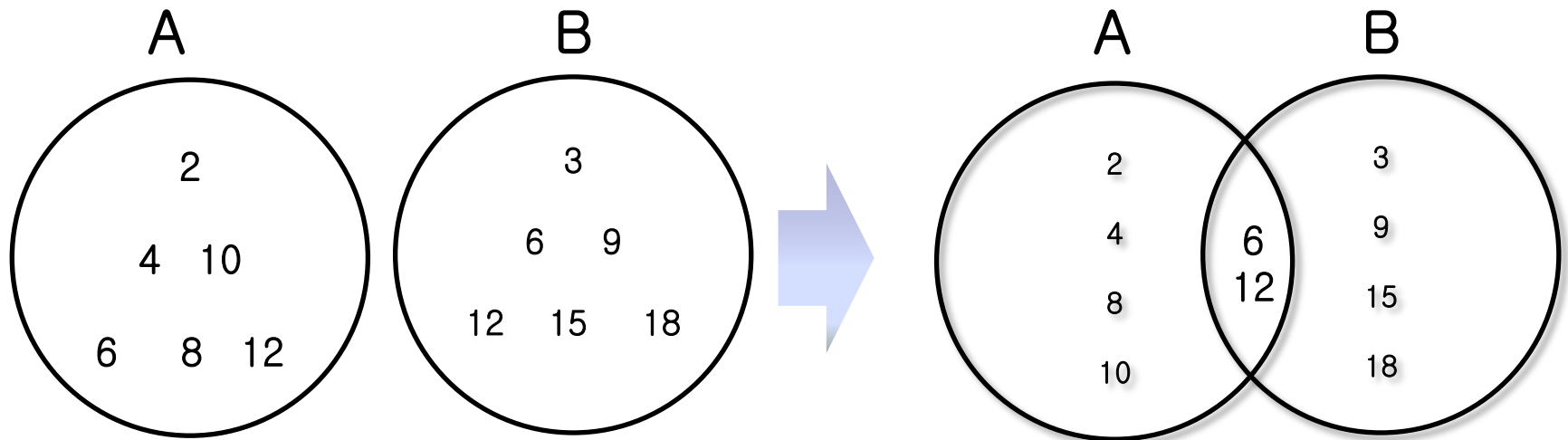




PYTHON_PROGRAMMING SET



Set(집합)



set

❖ set

- 집합은 여러 값들의 모임이며 저장 순서가 보장되지 않고, 중복값을 허용하지 않음.
 - 집합은 사전과 마찬가지로 {}로 표현하지만 key, value쌍이 아닌 데이터가 각각 존재한다는 점이 사전과 다름
 - set 함수는 공집합을 만들기도 하지만 다른 자료형을 집합으로 변환 가능함
- ❖ 저장속도는 빠르나 검색은 반복하여 찾아야 함.

```
names = {'허준','신사임담','권율','홍길동','허준'}
```

```
print(type(names))
```

```
print(len(names))
```

```
print(names)
```

내장 함수 set()

```
s = {}
```

```
print(type(s))
```

```
print(set('programming'))
```

```
print(set([12,15,17,11,15]))
```

```
dic = {'a':1, 'b':2, 'c':3}
```

```
print(set(dic))
```

예제

```
for x in {'가','나','다','라'}:  
    print(x)
```

※ 결과를 여러 번 반복 실행해 보세요!!!

Set 조작함수

Set 조작함수

함 수	설 명	사용법
add()	집합의 요소를 추가함	변수.add()
update()	집합을 결함함	변수.update()
remove()	집합의 요소를 삭제	변수.remove()

예제

```
asia = {'korea','china','japan'}  
print(asia)
```

```
asia.add('thailand')  
print(asia)
```

```
asia.add('china')  
print(asia)
```

```
asia.remove('japan')
```

```
print(asia)
```


예제

```
asia = {'korea', 'china', 'japan'}  
print(asia)  
asia2 = {'iraq', 'singapore', 'korea'}  
asia.update(asia2)  
print(asia)
```

```
print("-"*40)
```

```
asia = {'korea', 'china', 'japan'}  
asia2 = {'iraq', 'singapore', 'korea'}  
asia3=asia+asia2  
print(asia3)
```

set

❖ 집합의 연산

- 1] 합집합 (\cup) : 두 집합의 전체 요소들의 모음.
- 2] 교집합 (\cap) : 두 집합의 공통 요소들의 모음.
- 3] 차집합 ($-$) : 왼쪽 집합의 요소에서 오른쪽 집합의 요소를 제거
- 4] 배타적 차집합 (Δ) : 합집합 - 교집합
- 5] 부분집합 (\subseteq) : 왼쪽 집합이 오른쪽 집합의 부분집합인지의 여부를 확인
- 6] 진성 부분집합 (\subset) : 부분집합이면서 추가로 요소가 더 존재하는지를 확인.

부분집합과 진성부분집합의 차이는

부분집합(\subseteq)일 경우는 좌,우 집합이 같아도 부분집합이다.

진성부분집합(\subset)인 경우는 좌,우 집합이 모두 같을 경우 성립되지 않는다.

예제

```
two= {2, 4, 6, 8,10,12}  
three = {3, 6, 9, 12, 15}  
print('교집합', two & three)  
print('합집합', two | three)  
print('차집합', two - three)  
print('배타적 차집합', two ^ three)
```

예제

```
animal = {'호랑이', '사자', '강아지', '치타', '햄스터', '고양이'}
```

```
pet= {'강아지', '고양이', '햄스터'}
```

```
print(pet <= animal)
```

```
print(pet <= pet)
```

```
print(pet < animal)
```

```
print(pet < pet)
```