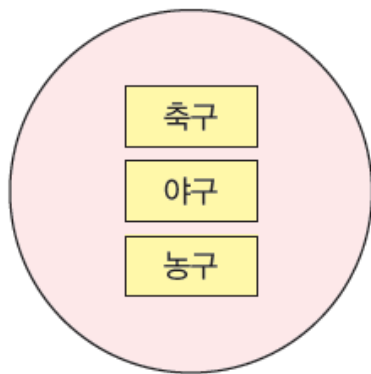
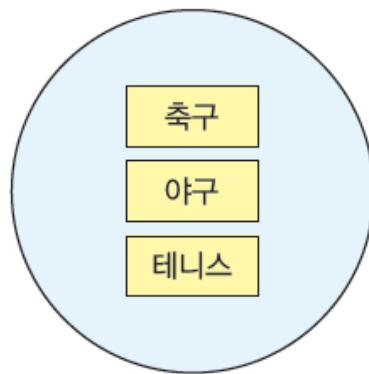


세트

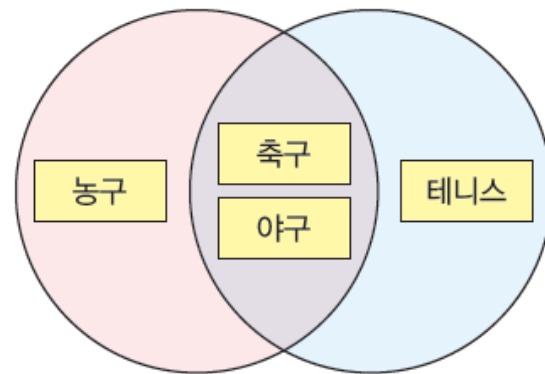
- 세트(set)는 우리가 수학에서 배웠던 집합이다. 세트는 고유한 값들을 저장하는 자료구조라고 할 수 있다.
- 리스트와는 다르게 세트의 요소는 특정 순서로 저장되지 않으며 위치별로 액세스할 수 없다



세트 #1



세트 #2



세트 #1 ∩ 세트 #2

세트 생성하기

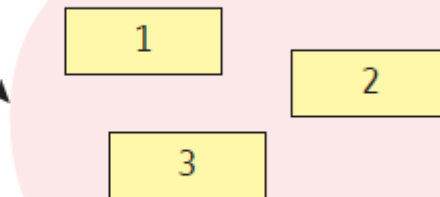
Syntax: 세트

형식 세트_이름 = { 항목1, 항목2, 항목3, ... }

예 numbers = {1, 2, 3} 초기화된 세트를 생성한다.
values = set() 공백 세트를 생성한다.

공백 세트가 {}가 아니다.

numbers

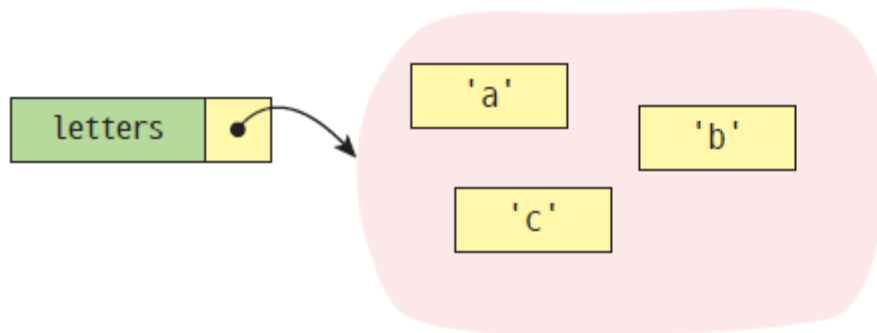


리스트<-> 세트

```
numbers = set([1,2,3,1,2,3])  
print(numbers)
```

{ 1, 2, 3 }

```
letters = set("abc")
```



문자열을 분해하여
세트로 만들 수 있어요.



세트의 연산

- all(), any(), enumerate(), len(), max(), min(), sorted(), sum() 사용 가능

```
fruits = {"apple", "banana", "grape"}  
size = len(fruits)           # size는 3이 된다.
```

```
fruits = { "apple", "banana", "grape" }  
if "apple" in fruits:  
    print("집합 안에 apple이 있습니다.")
```

집합 안에 apple이 있습니다.

세트의 연산

```
fruits={"apple","banana","grape"}  
for x in fruits:  
    print(x, end=" ")
```

grape banana apple

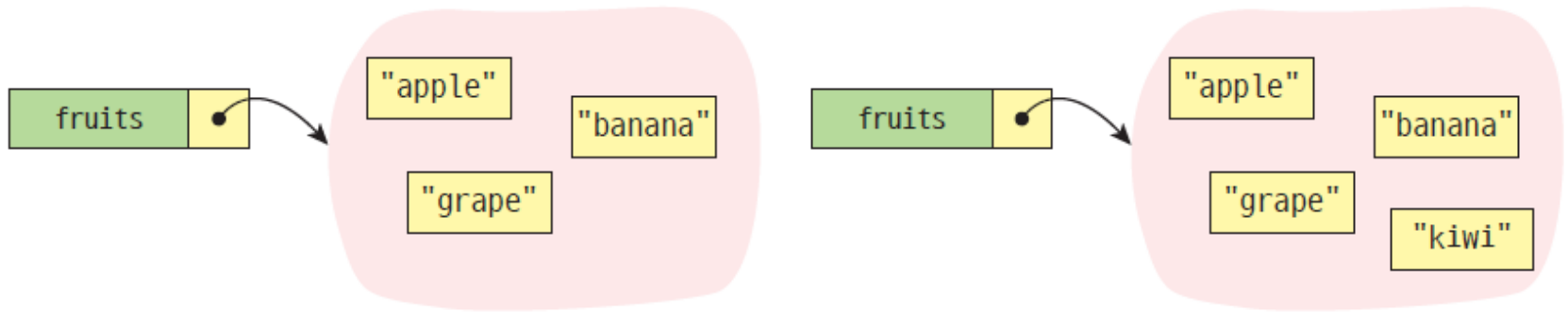
```
fruits={"apple","banana","grape"}  
for x in sorted(fruits):  
    print(x, end=" ")
```

apple banana grape

세트에 요소 추가하기

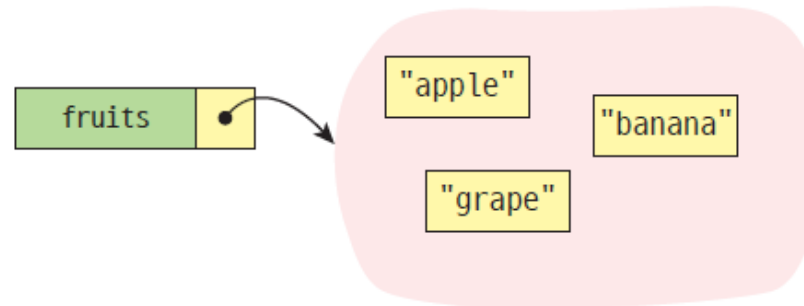
```
fruits = {"apple", "banana", "grape"}
```

```
fruits.add("kiwi")
```



```
fruits.remove("kiwi")
```

삭제하는 요소가 없으면 예외 발생, 예외 발생시키지 않으려면 `discard()` 사용!



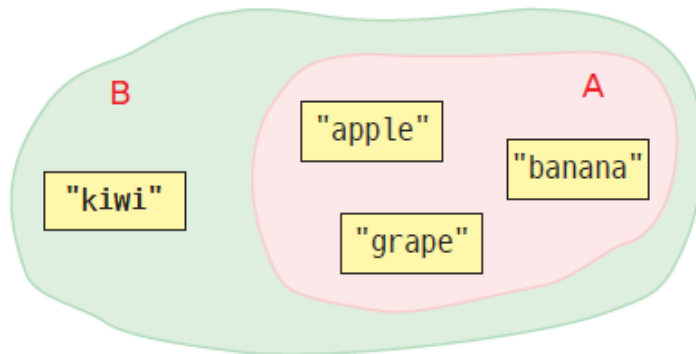
부분 집합 연산

```
A = {"apple", "banana", "grape"}  
B = {"apple", "banana", "grape", "kiwi"}
```

```
if A < B :                               # 또는 A.issubset(B) :  
    print("A는 B의 부분 집합입니다.")
```

🔗 실행결과

A는 B의 부분 집합입니다.



부분 집합은 < 으로
검사할 수 있어요!



==, != 연산

```
A={"apple","banana","grape"}  
B={"apple","banana","grape","kiwi"}
```

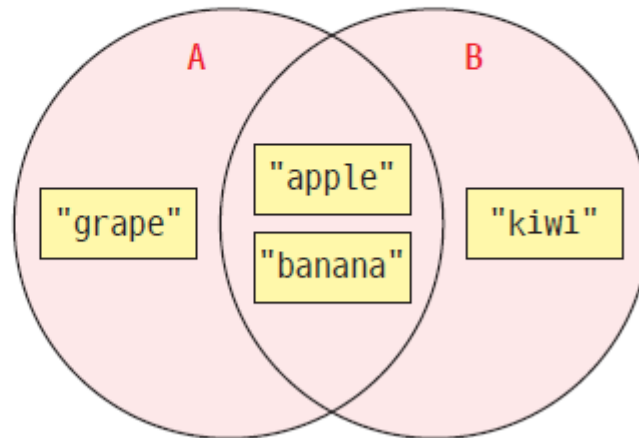
```
if A == B :  
    print("A와 B는 같습니다.")  
else :  
    print("A와 B는 같지 않습니다.")
```

A와 B는 같지 않습니다.

합집합

$C = A \cup B$

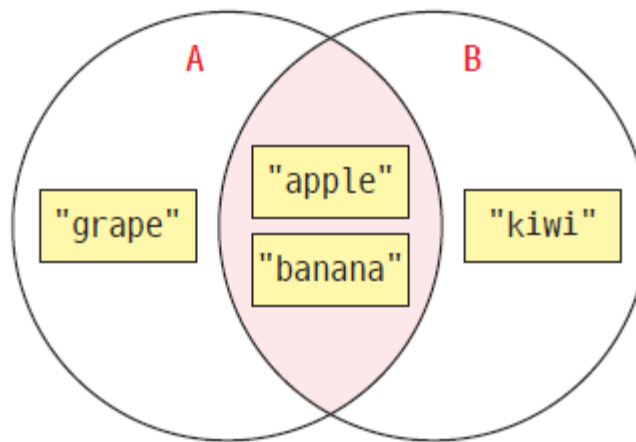
또는 $C = A.union(B)$



교집합

$C = A \cap B$

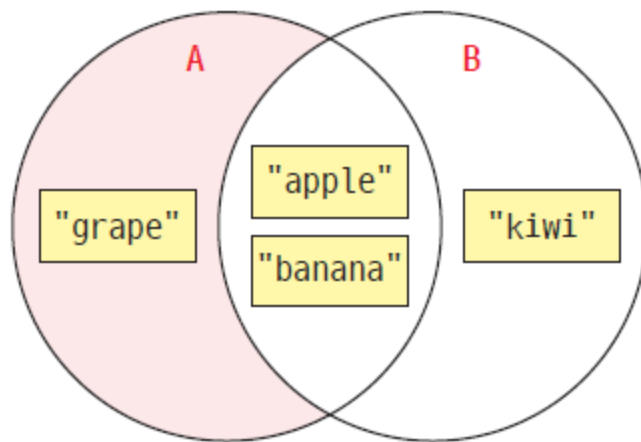
또는 $C = A.intersection(B)$



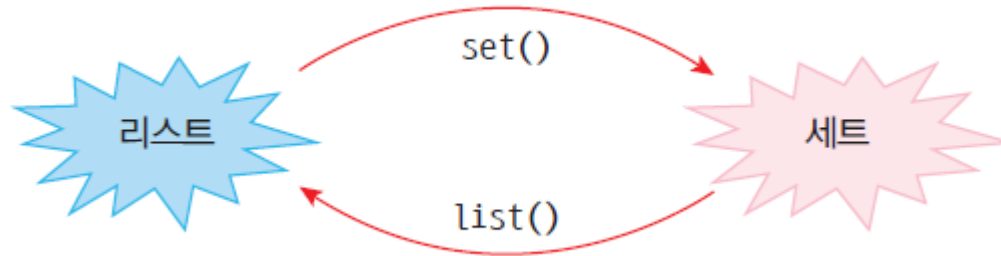
차집합

$$C = A - B$$

또는 $C = A.difference(B)$



리스트 <-> 세트



```
>>> list1 =[1,2,3,4,5,1,2,4 ]  
>>> len(set(list1))  
5
```

서로 다른 정수는 몇 개나 있을까?

```
>>> list1 =[1,2,3,4,5 ]  
>>> list2 =[3,4,5,6,7 ]  
>>> set(list1)&set(list2)  
{3, 4, 5}
```

공통적인 정수는 무엇일까?

세트 연산 정리

연산	설명
<code>set()</code>	공백 세트 생성
<code>set(seq)</code>	시퀀스에서 요소를 꺼내서 세트를 만든다.
<code>s1 = { e1, e2, e3, ... }</code>	초기값이 있는 세트는 중괄호로 만든다.
<code>len(s1)</code>	세트에 있는 요소의 수
<code>e in s1</code>	e가 세트 안에 있는지 여부
<code>add(e)</code>	e를 세트에 추가한다.
<code>remove(e)</code> <code>discard(e)</code>	e를 세트에서 삭제한다.
<code>clear()</code>	세트의 모든 요소를 삭제한다.
<code>s1.issubset(s2)</code>	부분 집합인지를 검사한다.
<code>s1 == s2</code> <code>s1 != s2</code>	동일한 집합인지를 검사한다.
<code>s1.union(s2)</code> <code>s1 s2</code>	합집합
<code>s1.intersection(s2)</code> <code>s1 & s2</code>	교집합
<code>s1.difference(s2)</code> <code>s1 - s2</code>	차집합