



# 파이썬과 친해지기

개념 복습, 집합, 교집합, 합집합

15주차

# 복습

## int(A)

A를 정수로 변환하여 뱉는다.

```
>>> int("10")  문자를 정수로 변환  
10
```

```
>>> int(3.5)   실수를 정수로 변환  
3
```

# 복습

```
a = input("숫자를 입력해주세요.")  
b = int(a)  
c = b * 2  
print("2를 곱하면", c, "가 됩니다.")
```



```
>>> %Run test.py  
숫자를 입력해주세요. 5  
2를 곱하면 10 가 됩니다.
```

# 같이 해보기

1. 사용자에게 정수  $a$ 와  $b$ 를 입력 받으면, 두 수를 더한 값, 뺀 값, 곱한 값을 출력하자.

```
>>> %Run test.py
```

```
정수 a를 입력하세요. 5
```

```
정수 b를 입력하세요. 3
```

```
더하면 8
```

```
빼면 2
```

```
곱하면 15
```

# 복습

for A in B:

B 안의 모든 것을 한번 씩 A에 넣고 블록을 실행한다.

```
>>> for a in [1, 2, 3, 4, 5]:  
        print("양", a, "마리...")
```

```
양 1 마리...
```

```
양 2 마리...
```

```
양 3 마리...
```

# 복습

```
chickens = ["간장", "양념", "뿌링클", "갈릭"]  
for a in chickens:  
    print(a + "치킨이 최고!")
```



```
>>> %Run test.py
```

```
간장치킨이 최고!  
양념치킨이 최고!  
뿌링클치킨이 최고!  
갈릭치킨이 최고!
```

# 복습

## range(a, b)

“a 이상 b 미만인 모든 정수”라는 표현을 뱉는다.

```
>>> for a in range(1, 6):  
        print("양", a, "마리...")
```

# 같이 해보기

2. 좋아하는 음식을 담은 리스트를 만들자.  
그 리스트에 있는 모든 음식 a에 대해 for 반복문을 통해 “a는 맛있다!”를 출력해보자.

```
>>> %Run test.py
```

```
치킨은 맛있다!  
과자는 맛있다!  
고기는 맛있다!  
치즈는 맛있다!
```



# 복습

## A and B

A와 B가 모두 참일 때만 True를 받고 아니면 False이다.

```
>>> 1+1==2 and 2+2==4  
True
```

```
>>> 1+1==2 and 2*2==3  
False
```

# 복습

## A or B

A와 B가 하나라도 참이면 True를 뱉고, 아니면 False이다.

```
>>> 1+1==2 or 1+1==3
```

```
True
```

```
>>> 1+3==5 or 6+6==5
```

```
False
```

## 같이 해보기

3. 어떤 정수를 입력 받자. 그 정수가 2로 나누어지고 5으로 나누어지면, “10으로도 나누어집니다.”를, 아니면 “10으로 나누어지지 않습니다.”를 출력하는 프로그램을 만들자.

```
>>> %Run set.py
```

```
정수를 입력해주세요. 31
```

```
10으로 나누어지지 않습니다.
```

# import A as B

## import A as B

A라는 모듈을 가져와 쓰지만 이름만 B로 바꾼다.

```
>>> import turtle as t
```

```
>>> t.forward(100)
```

# 거북이 움직이기

▲ t.forward(100)

100만큼 앞으로 가기

▼ t.backward(50)

50만큼 뒤로 가기

◀ t.left(90)

왼쪽으로 90도 돌기

▶ t.right(90)

오른쪽으로 90도 돌기

♡ t.write("안녕!")

"안녕!"이라고 쓰기

○ t.circle(50)

크기 50의 원 그리기

☆ t.reset()

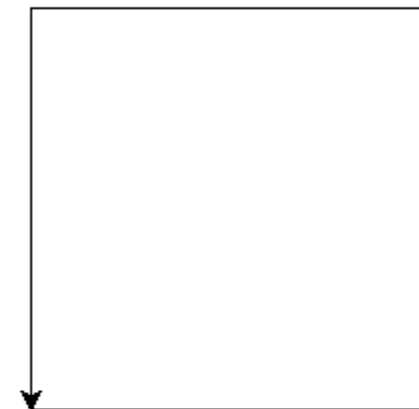
모두 지우고 처음으로 돌아간다.

# 같이 해보기

4. 사용자에게 숫자  $n$ 을 입력 받자. turtle 모듈을 이용해서 정 $n$ 각형을 그려주는 프로그램을 만들자.

```
>>> %Run set.py
```

```
숫자 n을 입력해주세요. 4  
정 4 각형을 그립니다.
```



# 직접 해보기

## 1. 거북이로 정삼각형을 그려보자.

힌트: 아래 명령들 중 알맞는 것을 사용하면 된다.

`t.forward(200)`

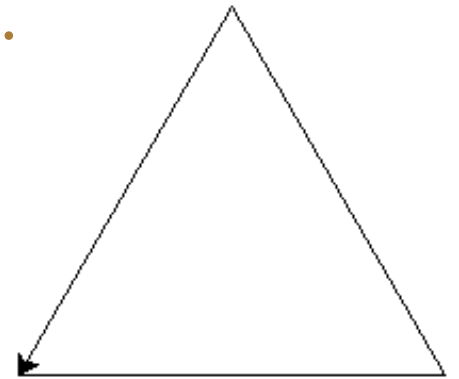
200만큼 앞으로 가기

`t.left(120)`

왼쪽으로 120도 돌기

`t.right(120)`

오른쪽으로 120도 돌기



뭔가 잘못 했을 때에는 `t.reset()` 으로 처음부터!

# 직접 해보기

2. 1부터 100까지의 숫자 중 2로 나누어지고 그리고 5로 나누어지는 숫자만을 출력하자.

힌트: for문, range(a, b), and 와 % 연산자를 이용해보자.

```
>>> %Run set.py
```

```
10
```

```
20
```

```
30
```

```
40
```



# 직접 해보기

3. 1부터 100까지의 숫자 중 2로 나누어지거나 3으로 나누어지는 숫자를 모두 출력하자.

힌트: for문, range(a, b), or 와 % 연산자를 이용해보자.

```
>>> %Run set.py
```

```
2
```

```
3
```

```
4
```

```
6
```

# 복습

## myList[i]

myList의 i번째 항목을 뱉는다.

```
>>> food = ["딸기주스", "오렌지주스"]
```

```
>>> food[0]
```

```
'딸기주스'
```

```
>>> food[1]
```

```
'오렌지주스'
```

# 복습

```
menus = ["햄버거", "치즈버거", "콜라", "얼음컵"]
```

```
>>> menus  
['햄버거', '치즈버거', '콜라', '얼음컵']
```

```
>>> menus[0]   menus 목록의 0번째 항목  
'햄버거'
```

```
>>> menus[1]   menus 목록의 1번째 항목  
'치즈버거'
```

# 복습

```
menus = ["햄버거", "치즈버거", "콜라", "얼음컵"]
```

          ↑                  ↑                  ↑                  ↑  
          0번째              1번째              2번째              3번째

컴퓨터는 숫자를 셀 때 0부터 센다!

# 복습

apple



사과. 사과나무의 열매.

bear



곰. 갈색의 털북숭이 포유류.

car



자동차. 바퀴가 달린 이동수단.

key

열쇠

value

값

# 복습

## hello[A]

hello 사전의 A 에 해당하는 값을 뱉는다.

```
>>> like = {"선생님": "치킨",  
            "동현": "과자"}
```

```
>>> like["동현"]  
'과자'
```

```
>>> like["선생님"]  
'치킨'
```

# 복습

hello[A] = B

hello 사전의 A 의 값을 B 로 한다.

```
>>> a = {}  
>>> a["치킨"] = "맛있음"  
>>> a["샐러드"] = "상쾌함"  
>>> a  
{ '치킨': '맛있음', '샐러드': '상쾌함' }
```

# 복습

```
eng = {}  
eng["apple"] = "사과"  
eng["bike"] = "자전거"  
eng["carrot"] = "당근"  
eng["diet"] = "다이어트"  
eng["elephant"] = "코끼리"
```

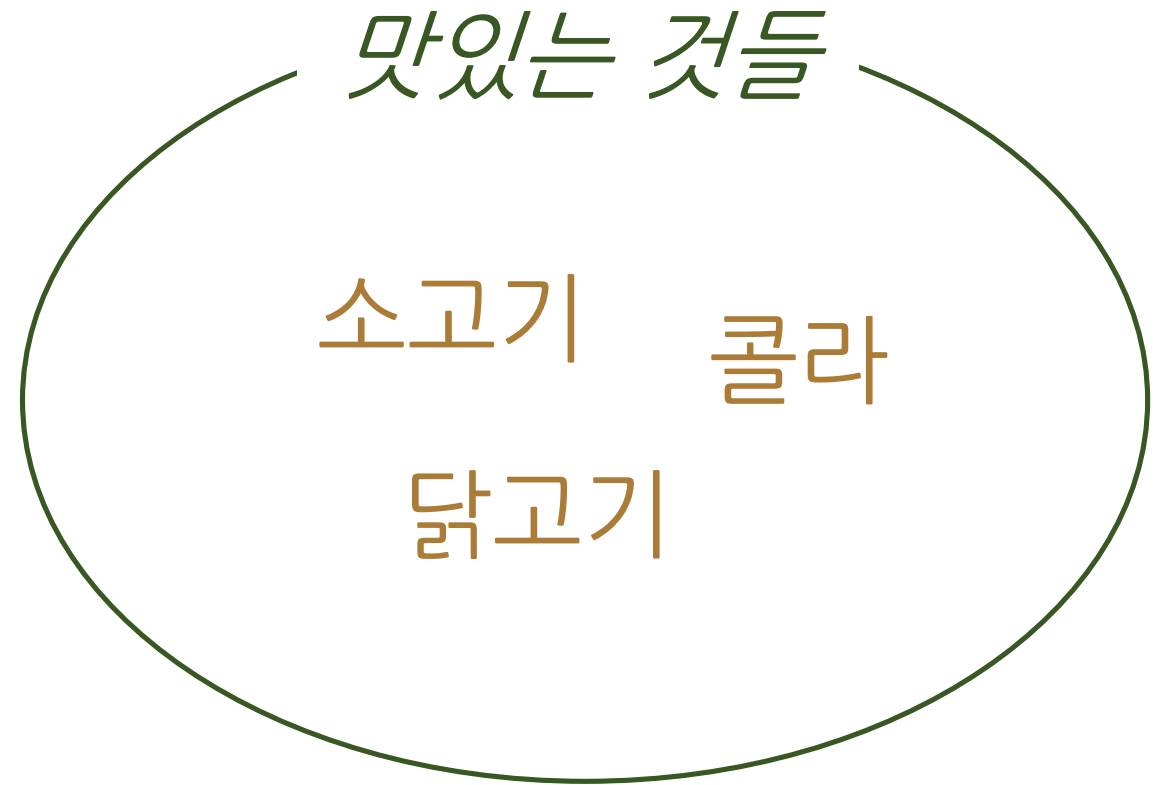
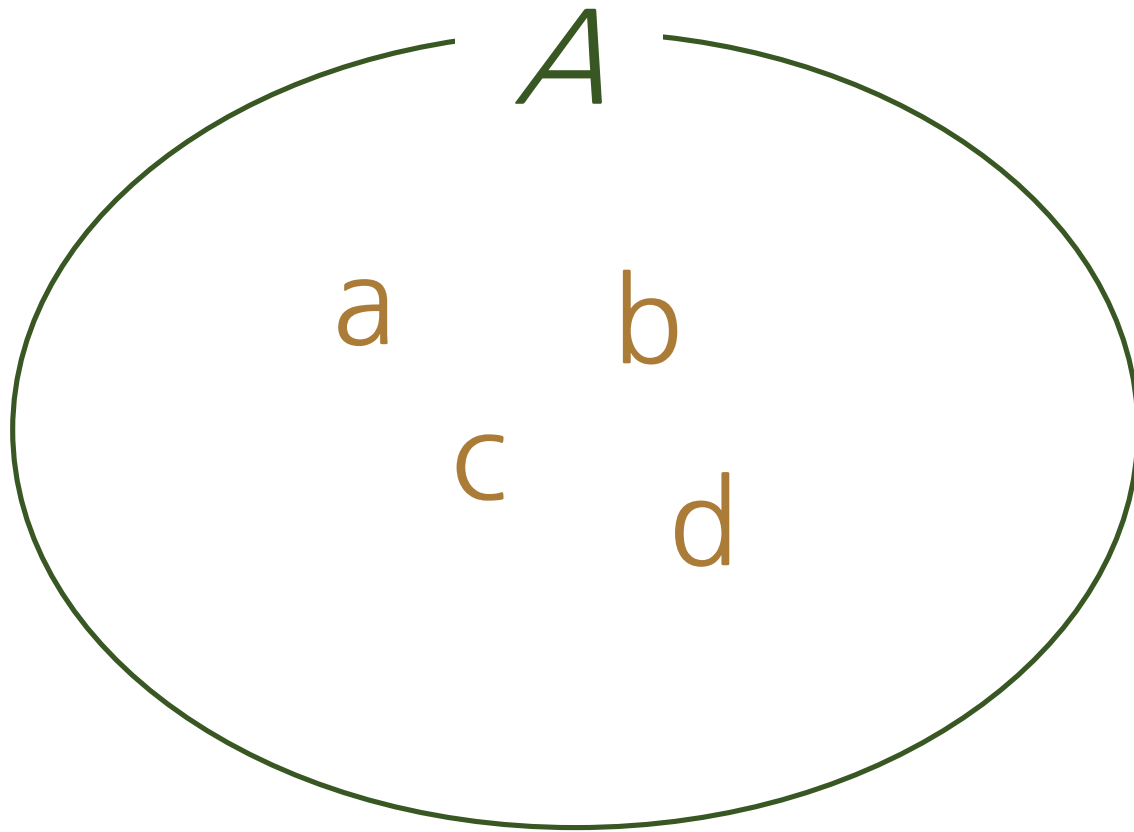


# 집합



반별로  
집합!

# 집합



모여있다.

# 집합의 예



털북숭이

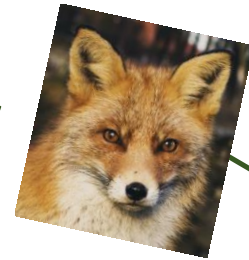
카펫

먼지털이

곰

여우

네발동물



곰

여우

악어

도마뱀

# 집합

집합은 중괄호를 사용한다.

```
>>> a = {5, 2, "흐압!", 70}
```

```
>>> print(a)
```

```
{'흐압!', 2, 5, 70}
```

```
>>> a = set() # 비어있는 집합
```

# 사전과 다른 점

```
>>> 집합 = {1, 2, 3}
```

```
>>> 사전 = {1: "하나", 2: "둘"}
```

# 사전과 다른 점

```
>>> 집합 = {1, 2, 3}
```

```
>>> 사전 = {1: "하나", 2: "둘"}
```

집합은 심표로 구분된 값들이 있다.

사전은 열쇠와 값이 1:1로 맞춰져 있다.

# 사전과 다른 점

```
>>> 집합 = {1, 2, 3}
>>> 사전 = {1: "하나", 2: "둘"}
>>> type(집합)
<class 'set'>
>>> type(사전)
<class 'dict'>
```

# 리스트와 다른 점 - 1

```
>>> a = {4, "치킨", 2, 100}
>>> b = [4, "치킨", 2, 100]
>>> print(a)
{'치킨', 100, 2, 4}
>>> print(b)
[4, '치킨', 2, 100]
```



# 리스트와 다른 점 - 1

```
>>> a = {4, "치킨", 2, 100}
```

```
>>> b = [4, "치킨", 2, 100]
```

```
>>> print(a)
```

```
{'치킨', 100, 2, 4}
```

집합은

```
>>> print(b)
```

```
[4, '치킨', 2, 100]
```

순서가 없다!

# 리스트와 다른 점 - 2

```
>>> a = {1, 1, 2, 3, 3, 3}
```

```
>>> b = [1, 1, 2, 3, 3, 3]
```

```
>>> print(a)
```

```
{1, 2, 3}
```

```
>>> print(b)
```

```
[1, 1, 2, 3, 3, 3]
```

# 리스트와 다른 점 - 2

```
>>> a = {1, 1, 2, 3, 3, 3}
```

```
>>> b = [1, 1, 2, 3, 3, 3]
```

```
>>> print(a)
```

```
{1, 2, 3}
```

```
>>> print(b)
```

```
[1, 1, 2, 3, 3, 3]
```

중복은

중복이 없다!

# A.add(b)

## A.add(b)

집합 A에 b를 추가한다.

```
>>> a = {1, 2, 3}
```

```
>>> a.add(100)
```

```
>>> a.add("치킨엔 콜라지")
```

```
>>> print(a)
```

```
{1, 2, 3, 100, '치킨엔 콜라지'}
```

# A.add(b) 예제

```
>>> a = {1, 2}
>>> a.add(3)
>>> a.add("치킨")
>>> a.add(2)
>>> a.add(2)
>>> print(a)
{1, 2, 3, '치킨'}
```

중복된 것은 무시된다.

# in 연산자를 집합에서!

```
>>> a = {1, 2, 3, 4, 5, 6}
```

```
>>> 1 in a
```

```
True
```

```
>>> 8 in a
```

```
False
```

```
>>> "1" in a
```

```
False
```

# in 연산자를 집합에서!

```
네발동물 = {"도마뱀", "호랑이",  
            "곰", "치타", "강아지"}  
a = input("동물 이름을 입력하세요: ")  
  
if a in 네발동물:  
    print(a, "는 네발동물이죠!")  
else:  
    print("잘 모르겠습니다.")
```

# in 연산자를 집합에서!

```
print(a, "는 네발동물이죠!")  
else:  
    print("잘 모르겠습니다.")
```



```
>>> %Run set.py
```

```
동물 이름을 입력하세요: 치타  
치타 는 네발동물이죠!
```



# in 연산자를 집합에서!

```
print(a, "는 네발동물이요: ")  
else:  
    print("잘 모르겠습니다.")
```



```
>>> %Run set.py
```

```
동물 이름을 입력하세요: 치타  
치타 는 네발동물이죠!
```

```
>>> %Run set.py
```

```
동물 이름을 입력하세요: 두부과자  
잘 모르겠습니다.
```

# for문으로 집합 반복하기

```
a = {1, 2, 3, 4, 5}
for b in a:
    print(b)
```

# for문으로 집합 반복하기

```
a = {1, 2, 3, 4, 5}
for b in a:
    print(b)
```



```
>>> %Run set.py
```

```
1
2
3
4
```

# 직접 해보기

1. 귀여운 동물 이름들을 가진 집합 **a**을 만들자. 사용자에게 동물 이름을 입력 받아 그 동물이 귀여운 지 아닌지 출력하자.

힌트: if문과 in 문을 이용해보자.

```
>>> %Run set.py
```

```
동물 이름을 입력하세요. 고양이  
귀엽습니다!
```

# 직접 해보기

2.  $a = [1, 2, 2, 3, 3, 3]$  인 리스트가 있다. 빈 집합을 만들자. 이 리스트의 모든 숫자들을 반복하여 집합에 모두 집어넣어보자. 이렇게 하면 중복인 숫자들이 사라진다. 이렇게 한 후 집합을 출력해보자.

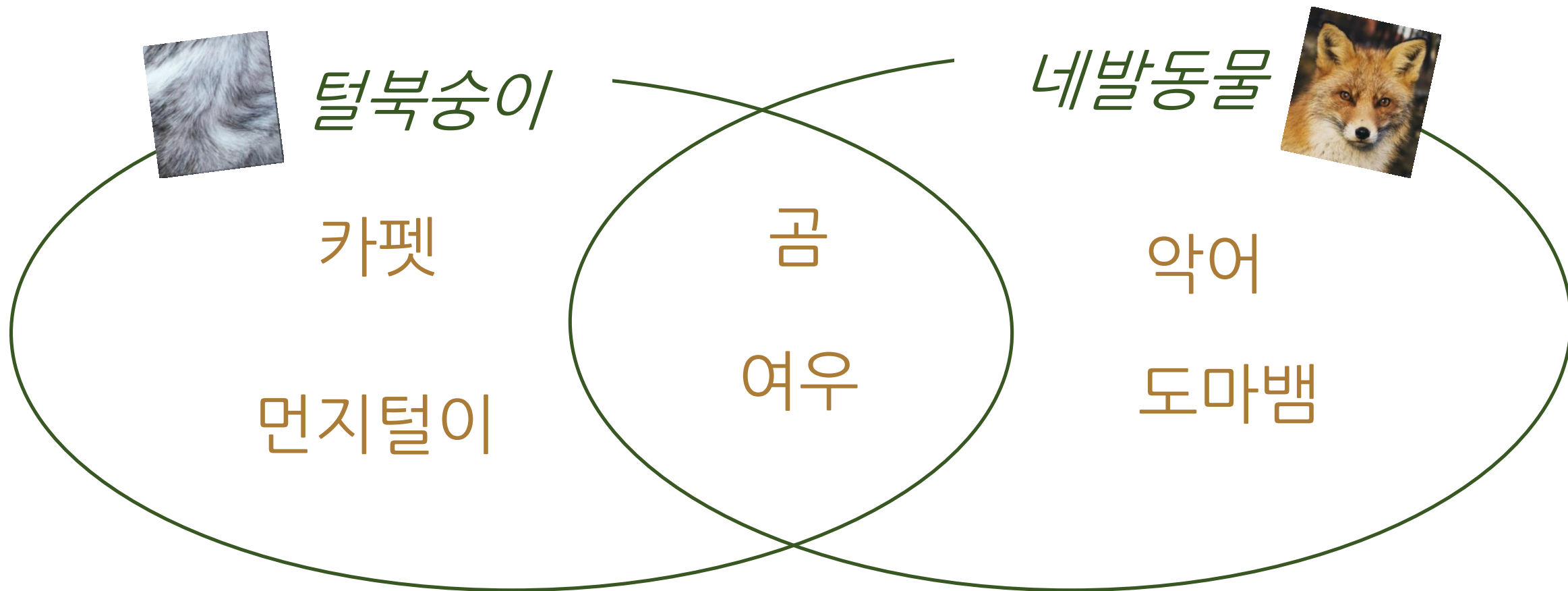
힌트: for문, set(), 집합.add(b) 를 사용해보자.

# 직접 해보기

3. 빈 집합 **a**를 만들자. 무한 반복을 이용하여 계속 입력을 받고, 입력 받은 숫자를 집합 **a**안에 넣는다. 단 “0”이 입력되면 반복을 빠져나온다. 이후 집합을 출력해보자.

힌트: `set()`, `while True`, `break`, `A.add(b)`, `int()` 를 이용하자.

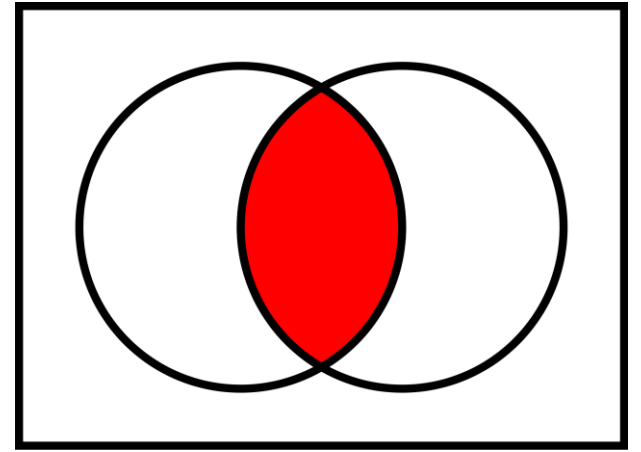
# 집합의 다른 표현



# 교집합

## 교집합 ( $\cap$ )

집합과 집합의 공통점, 겹치는 부분!



$$\begin{aligned} \text{곤충} \cap \text{한국} &= \{ \text{애벌레, 잠자리, \dots} \} \\ \text{곤충} \cap \text{날아다님} &= \{ \text{나방, 나비, \dots} \} \end{aligned}$$



# 교집합



털북숭이

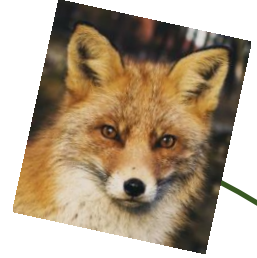
카펫

먼지털이

곰

여우

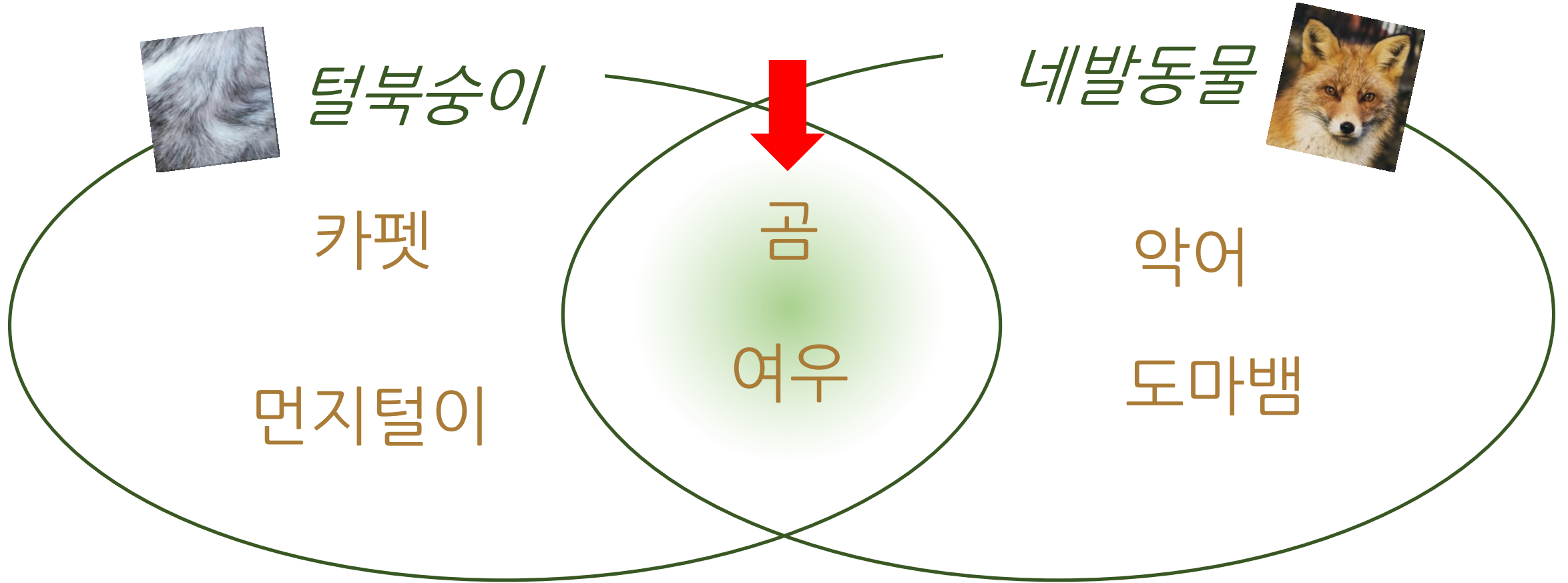
네발동물



악어

도마뱀

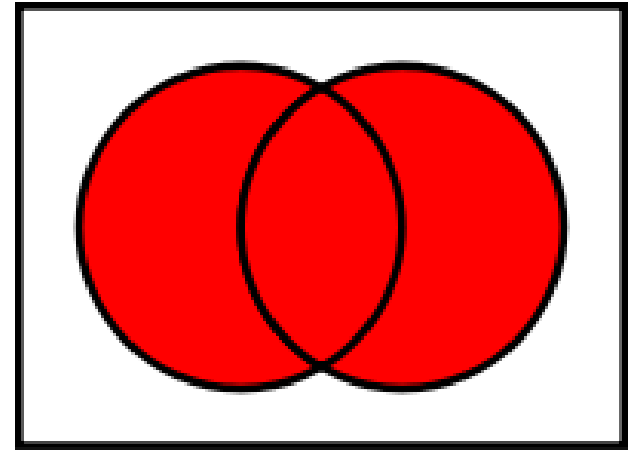
# 교집합



# 합집합

## 합집합 ( $\cup$ )

집합과 집합을 더한 것, 집합의 합!



과자  $\cup$  라면

매움  $\cup$  달콤함

= { 신라면, 오레오, ... }

= { 설탕, 불닭볶음면, ... }

# 합집합



털북숭이

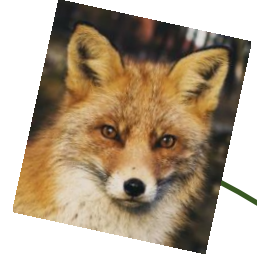
카펫

먼지털이

곰

여우

네발동물



악어

도마뱀

# 합집합



털북숭이

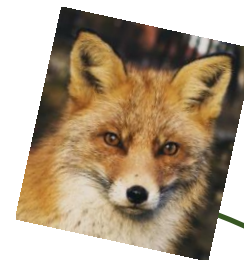
카펫

먼지털이

곰

여우

네발동물



악어

도마뱀

# A.union(B)

## A.union(B)

집합 A와 집합 B의 합집합을 뱉는다. (합친 것)

```
>>> a = {1, 2, 3}
>>> b = {2, 3, 4}
>>> c = a.union(b)
>>> c
{1, 2, 3, 4}
```

# A.union(B) 예제

```
오전수업 = {"국어", "체육"}
```

```
오후수업 = {"수학", "가정"}
```

```
오늘수업 = 오전수업.union(오후수업)
```

```
print("오늘 수업들: ", 오늘수업)
```

# A.union(B) 예제

```
오전수업 = {"국어", "체육"}  
오후수업 = {"수학", "가정"}
```

```
오늘수업 = 오전수업.union(오후수업)  
print("오늘 수업들: ", 오늘수업)
```



```
>>> %Run set.py
```

```
오늘 수업들: {'가정', '체육', '수학', '국어'}
```



# A.intersection(B)

## A.intersection(B)

집합 A와 집합 B의 교집합을 뱃는다. (겹치는 부분)

```
>>> a = {1, 2, 3}
>>> b = {2, 3, 4}
>>> c = a.intersection(a, b)
>>> c
{2, 3}
```

# A.intersection(B) 예제

```
엄마 = {"비빔밥", "청국장", "나물"}
```

```
아들 = {"피자", "치킨", "비빔밥"}
```

```
저녁 = 엄마.intersection(아들)
```

```
print("오늘 저녁:", 저녁)
```

# A.intersection(B) 예제

```
엄마 = {"비빔밥", "청국장", "나물"}  
아들 = {"피자", "치킨", "비빔밥"}
```

```
저녁 = 엄마.intersection(아들)  
print("오늘 저녁:", 저녁)
```

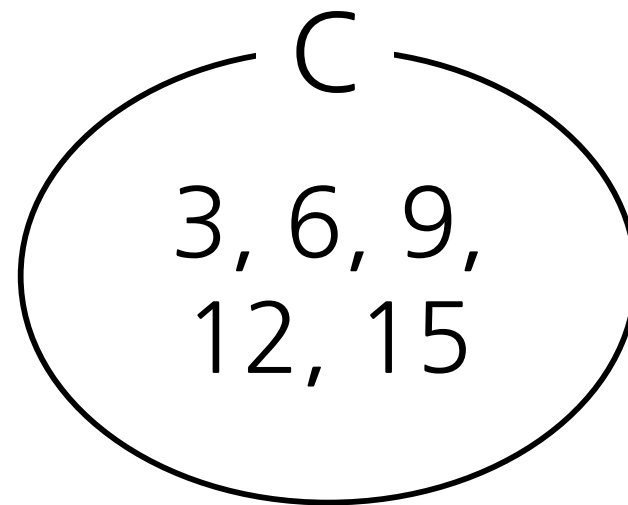
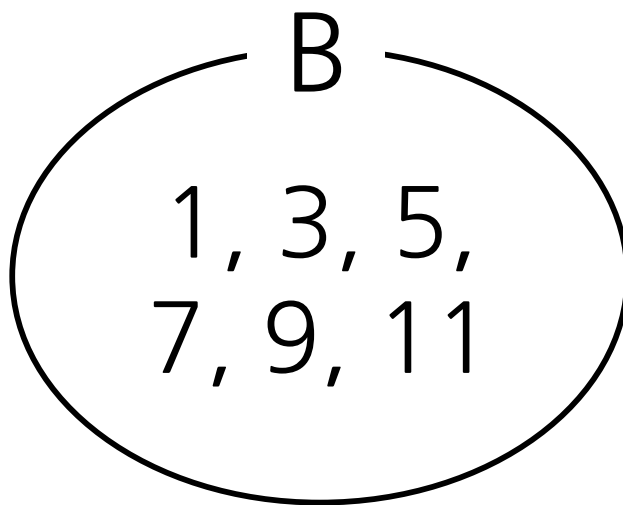
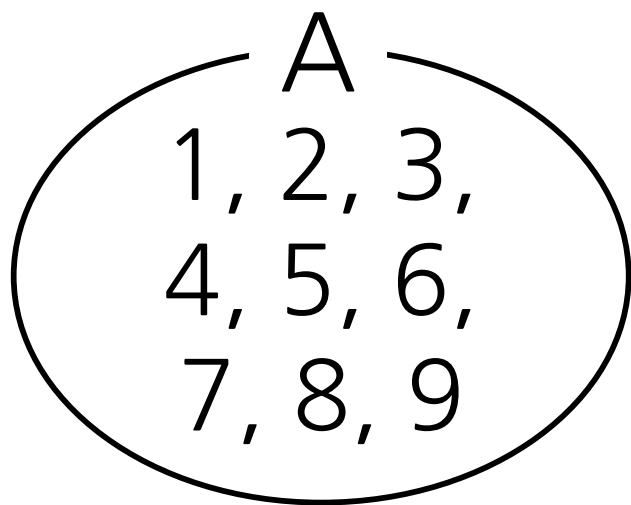


```
>>> %Run set.py  
오늘 저녁: {'비빔밥'}
```

# 직접 해보기

1. 다음과 같이 집합 세 개를 만들어라. 세 집합의 교집합을 구해 출력해보자. (세 집합이 모두 겹치는 부분)

힌트: `A.intersection(B)` 를 여러 번 이용해보자.



## 직접 해보기

2.  $a = \{1, 2, 3\}$  인 집합이 있다. 빈 집합  $b$ 를 만들자. 사용자에게 숫자 입력을 세 번 받아 집합  $b$ 에 집어넣자. 집합  $a$ 와  $b$ 의 합집합인 집합  $c$ 를 만들자. 세 집합을 모두 출력하자.

힌트: `set()`, `A.add(b)`, `A.union(B)` 를 이용해보자.

숫자를 입력해주세요: 4

$a: \{1, 2, 3\}$

$b: \{2, 3, 4\}$

$c: \{1, 2, 3, 4\}$

3. **복슬** = {"개", "고양이", "다람쥐", "곰", "다리털"} 인 집합이 있다. 빈 집합 **a**를 만들자. 사용자에게 입력을 세 번 받아 집합 **a**에 집어넣자. 집합 **a**와 **복슬**의 교집합을 찾아 출력해보자.

힌트: `set()`, `A.add(b)`, `A.intersection(B)` 를 이용해보자.

```
>>> %Run set.py
```

```
입력 1:곰
```

```
입력 2:도마뱀
```

```
입력 3:개
```

```
그 중에 복슬한 것은 {'곰', '개'} 입니다.
```



# 파이썬과 친해지기

개념 복습, 집합, 교집합, 합집합

15주차

## References

왕초보를 위한 Python 2.7  
뱀 인형 이미지  
Stock Photos

<https://wikidocs.net/145>  
<https://bit.ly/2WINL65>  
<https://unsplash.com/>