



제어문

최석재

lingua@naver.com

프로그램의 흐름을 제어하기

- 지금까지는 프로그램이 기록된 대로만 진행하는 것을 보았다
- 그러나 어떤 부분에 조건문과 반복문을 사용하면 그 부분은 특정 상황에서만 진행되도록 할 수 있다
- 특정 조건에서만 진행되도록 하는 것은 조건문 if
- 일정 상황까지 반복하게 하는 것은 반복문 for, while 이다

조건문 if

- if는 특정 조건이 참인 경우만 그 내용을 실행시킨다

```
gender = input("당신의 성별은?(남자/여자): ")
```

```
if gender=="남자":  
    print("안녕하십니까?")  
    print("당신은 남자이군요!")
```

콜론
tab

```
if gender=="여자":  
    print("안녕하세요?")  
    print("당신은 여자이군요~")
```

if와 비교 연산자

```
age = int(input("당신의 나이는?: "))
```

```
if age == 19:  
    print("[1] 올해 성년이 되셨군요!")
```

```
if age != 19:  
    print("[2] 당신은 19세가 아닙니다.")
```

```
if age >= 19:  
    print("[3] 당신은 19세 이상입니다.")
```

```
if age > 19:  
    print("[4] 당신은 20세 이상입니다.")
```

```
if 19 < age < 30:  
    print("[5] 당신은 20대입니다.")
```

if와 논리 연산자

```
animals = ['닭', '소', '돼지']
```

```
legs = int(input("다리가 몇 개입니까?: "))
```

```
childs = bool(input("새끼가 2마리 이상입니까?(True/False): "))
```

```
foods = input("초식입니까, 잡식입니까?(초식/잡식): ")
```

```
if legs == 2:  
    print("닭입니다.")
```

```
if legs == 4 and childs == True:  
    print("소 또는 돼지이나, 돼지일 것입니다.")
```

```
if legs == 4 and childs == True and foods == "잡식":  
    print("돼지가 분명합니다.")
```

연습문제 1

- 동물이 소인 경우를 처리하세요

in과 not in

- 어떤 항목이 있는지를 살펴볼 때 쉽게 사용할 수 있다

```
animals = ['닭', '소', '돼지']
```

```
if '소' in animals:  
    print("소, 있습니다!")
```

```
if '오리' not in animals:  
    print("오리.. 없습니다...")
```

연습문제 2

- 사용자로부터 입력을 받아서 해당 가축이 리스트에 있는지 여부를 알려주는 프로그램을 만드세요
 - ① 있는 경우, in 을 사용하여 해당 가축이 있음을 알려주세요
 - ② 없는 경우, not in 을 사용하여 해당 가축이 없음을 알려주세요

if else 구문

- 조건에 일치하는 경우와 그렇지 않은 경우를 묶을 수가 있다

```
animals = ['닭', '소', '돼지']
```

```
if '소' in animals:  
    print("소, 있습니다!")  
else:  
    print("소.. 없습니다...")
```

연습문제 3

- 앞에서 작성한 연습문제 2를 if else 구문으로 변환하세요

if elif else 구문

- 조건에 일치하지 않는 경우, 추가 조건을 줄 수 있다

```
animals = ['닭', '소', '돼지']  
animals2 = ['오리', '펭', '노루']
```

```
if '오리' in animals:  
    print("오리, 있습니다!")  
elif '오리' in animals2:  
    print("오리, 앞으로 살 것입니다.")  
else:  
    print("오리.. 없습니다...")
```

연습문제 4

- 앞의 내용을 사용자 입력을 받아 처리하는 프로그램으로 변환하세요

반복문 while

- while 구문은 일정 조건에 이를 때까지 반복된다

```
animals = ['닭', '소', '돼지']
```

```
animals_length = len(animals)    # 길이를 세어준다  
print("가축은 {}마리 입니다.\n".format(animals_length))
```

```
count = 0  
while count < animals_length:  
    print(animals[count])  
    count = count + 1
```

*# 보통 밖에 통제 변수를 두게 된다
이 조건이 참일 때까지 반복한다
현재 count의 리스트 원소 출력
count를 올려준다*

반복문 for

- for 반복문의 사용 빈도가 더 높다
- 해당 원소의 수만큼, 해당 원소를 하나씩 가져오며 반복한다
- 통제 변수를 미리 두지 않아도 된다

```
animals = ['닭', '소', '돼지']
```

※ animals의 원소가 하나씩 animal로 들어간다

```
for animal in animals:  
    print(animal)
```

연습문제 5-1

- 사용자로부터 입력을 받아서 원하는 가축이 있으면 알려주는 프로그램을 작성하세요. 단, for 문을 사용하세요

연습문제 5-2

- 연습문제 5-1에서 다음의 기능이 추가되도록 하세요
 - 못 찾은 경우, 해당 가축을 “못 찾았습니다” 라는 메시지 출력
- hint:
 - 프로그램 초반부에 `found = False` 라는 변수를 사용
 - 프로그램 후반부에 `if not found:` 로 시작하는 부분을 작성

range() 함수

- range() 함수는 for문 안에서 수를 자동 생성한다

```
for i in [0, 1, 2, 3, 4]:  
    print(i)
```

```
print("#"*10)
```

```
for i in range(0, 5):  
    print(i)
```

0~4까지 자동생성

```
print("#"*10)
```

```
for i in range(5):  
    print(i)
```

시작이 0일 때는 생략 가능

중첩 for 문

- for 문 안에는 다시 for 문이 들어갈 수 있다

```
farms = ['제일농장', '우리농장', '한국농장']  
animals = ['닭', '소', '돼지']
```

```
for farm in farms:  
    print('*'*5, farm, '*'*5)  
  
    for animal in animals:  
        print(animal)
```

```
***** 제일농장 *****  
닭  
소  
돼지  
***** 우리농장 *****  
닭  
소  
돼지  
***** 한국농장 *****  
닭  
소  
돼지
```

연습문제 6

- 각 농장의 가축 수가 다음과 같이 수집되었습니다

제일농장 = [100, 50, 80]

우리농장 = [50, 10, 20]

한국농장 = [200, 100, 150]

- 이를 이용하여 아래와 같이 결과가 나오도록 하세요

***** 제일농장 *****

닭, 100마리입니다.

소, 50마리입니다.

돼지, 80마리입니다.

***** 우리농장 *****

닭, 50마리입니다.

소, 10마리입니다.

돼지, 20마리입니다.

***** 한국농장 *****

닭, 200마리입니다.

소, 100마리입니다.

돼지, 150마리입니다.

break

- 조건이 만족되면 빠르게 반복문을 중지시킬 수 있다

```
animals = ['닭', '소', '돼지']
```

```
for animal in animals:  
    if animal == "소":  
        print(animal)  
        break
```

'돼지'를 검사하지 않고 바로 종료한다

연습문제 7

- 사용자로부터 입력을 받아 원하는 가축이 찾아지면 바로 종료하도록 프로그램을 작성하세요