

# 2018학년도 2학기 언어와 컴퓨터

## 제5강 조건문

박수지

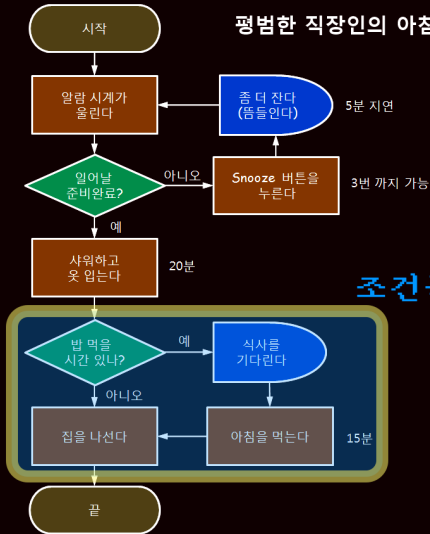
서울대학교 인문대학 언어학과

2018년 9월 17일 월요일

## 오늘의 목표

- 1 프로그램의 구성 요소 중 조건 실행을 수행할 수 있다.
- 2 `if, else, elif` 문을 사용할 수 있다.

## 평범한 직장인의 아침 시간 흐름도



조건문

# 복합문

[https://docs.python.org/ko/3/reference/compound\\_stmts.html](https://docs.python.org/ko/3/reference/compound_stmts.html)

## 절 (clause)의 구성 요소와 형식

헤더 (header) 키워드로 시작해서 콜론 (:)으로 끝나는 행  
스위트 (suite) 같은 수준의 들여쓰기로 이루어진 코드들

## if 문

헤더의 키워드가 if인 복합문

# if 문

조건이 참일 때 실행할 스위트

## 형식

```
1  if <조건>:  
2      <실행할 스위트>
```

## cold\_or\_not.py

```
1  temp = 26  
2  print('Hello!')  
3  if temp > 20:  
4      print("It isn't cold.")  
5  
6  print('Bye!')
```

temp의 값을 바꾸면서 실행해 보자.

## 질문

여기서 큰따옴표를 작은따옴표로 바꾸어도 괜찮은가?

# else 문

if 문의 조건이 거짓일 때 실행할 스위트

## 형식

```
1  if <조건>:  
2      <실행할 스위트>  
3  else:  
4      <실행할 스위트>
```

## cold\_or\_hot.py

```
1  temp = 17  
2  print('Hello!')  
3  if temp > 20:  
4      print("It isn't cold.")  
5  else:  
6      print("It isn't hot.")  
7  
8  print('Bye!')
```

# elif 문

if 문의 조건이 거짓이고 elif 문의 조건이 참일 때 실행할 스위트

## 형식

```
1  if <조건>:  
2      <실행할 스위트>  
3  elif <조건>:  
4      <실행할 스위트>  
5  elif <조건>:  
6      <실행할 스위트>  
7  else:  
8      <실행할 스위트>
```

## cold\_or\_mild\_or\_hot.py

```
1  temp = 17  
2  print('Hello!')  
3  if temp < 15:  
4      print("It's cold.")  
5  elif temp < 25:  
6      print("It's mild.")  
7  else:  
8      print("It's hot.")  
9  
10 print('Bye!')
```

## 조건식의 조건

- 불 값으로 평가되는 표현식
- ...뿐만 아니라 임의의 표현식

### empty.py

```
1 basket = []  
2 print("Q: Is the  
   basket empty?")  
3 if basket:  
4     print("A: No.")  
5 else:  
6     print("A: Yes.")
```

### 주의

basket의 자료형은  
bool이 아니라 list이다.



# “파이썬다운” 조건문

## 3행의 조건 비교하기

### less\_pythonic.py

```
1 basket = ['egg']
2 print("Q: Is the basket
   empty?")
3 if len(basket) > 0:
4     print("A: No.")
5 else:
6     print("A: Yes.")
```

### more\_pythonic.py

```
1 basket = ['egg', 'ham']
2 print("Q: Is the basket
   empty?")
3 if basket:
4     print("A: No.")
5 else:
6     print("A: Yes.")
```

### 관찰

- 리스트뿐만 아니라 모든 자료형으로 일반화할 수 있다.
- `bool()`에 다양한 자료형의 값을 넣어 보자.

## 3의 배수 판정

### 1. 정수를 입력받아 3의 배수인지 아닌지 판정하기

3의 배수이다.  $\iff$  3으로 나눈 나머지가 0이다.

multi3.py

```
1 # -*- coding: utf-8 -*-  
2 n = int(input('정수를 입력하세요: '))  
3 if n % 3:  
4     print('3의 배수가 아닙니다.')  
5 else:  
6     print('3의 배수입니다.')
```

### 질문

이 코드에서 어떤 예외가 발생할 수 있는가?

## 3의 배수 판정

### 2. 입력받은 값이 정수인지 아닌지 먼저 판정하기

multi3.py

```
1 # -*- coding: utf-8 -*-
2 n = input('정수를 입력하세요: ')
3 if n.isnumeric():
4     if int(n) % 3:
5         print('3의 배수가 아닙니다.')
6     else:
7         print('3의 배수입니다.')
8 else:
9     print('정수를 입력하지 않았습니다.')
```

#### 관찰

if 문의 스위트 안에 if 문을 중첩해서 넣을 수 있다.

## 3의 배수 판정

### 3. 조건문 중첩을 피하기

multi3.py

```
1  # -*- coding: utf-8 -*-
2  n = input('정수를 입력하세요: ')
3  if not n.isnumeric():
4      print('정수를 입력하지 않았습니다.')
5  elif int(n) % 3:
6      print('3의 배수가 아닙니다.')
7  else:
8      print('3의 배수입니다.')
```

#### 관찰

왜 `n.isnumeric()`이 아닌 `not n.isnumeric()`으로 조건을 시작했을까?

## 3의 배수 판정

덜 효율적인 조건문

```

1  # -*- coding: utf-8 -*-
2  n = input('정수를 입력하세요: ')
3  if n.isnumeric() and (int(n) % 3):
4      print('3의 배수가 아닙니다.')
5  elif n.isnumeric() and not (int(n) % 3):
6      print('3의 배수입니다.')
7  else:
8      print('정수를 입력하지 않았습니다.')
```

### 관찰

3의 배수인 경우나 정수가 아닌 경우, `n.isnumeric()`와 `int(n)`를 `if` 문에서 이미 연산했는데도 `elif` 문에서 또 연산하게 된다.

## 3의 배수 판정

더욱 비효율적인 조건문

```
1 # -*- coding: utf-8 -*-
2 n = input('정수를 입력하세요: ')
3 if n.isnumeric() and (int(n) % 3):
4     print('3의 배수가 아닙니다.')
5 if n.isnumeric() and not (int(n) % 3):
6     print('3의 배수입니다.')
7 if not n.isnumeric():
8     print('정수를 입력하지 않았습니다.')
```

### 관찰

3의 배수가 아닌 정수인 경우에도 두 번째와 세 번째 if 문의 조건을 검사하게 된다.

# 패스워드 검사

길이가 8 이상이고 숫자나 특수문자를 포함했는지 판정하기

password.py

```
1 # -*- coding: utf-8 -*-
2 password = input('패스워드를 입력하세요: ')
3 if len(password) < 8:
4     print('너무 짧습니다.')
5 elif password.isalpha():
6     print('숫자나 특수문자를 포함하세요.')
7 else:
8     print('좋습니다.')
```

# 냉장고에서 물건 꺼내기

물건이 냉장고에 있다면 꺼내고 남은 물건들을 알려주기

fridge.py

```
1 # -*- coding: utf-8 -*-
2 fridge = ['주스', '두부', '요거트', '요거트']
3 food = input('찾는 것을 입력하세요: ')
4 if food in fridge:
5     fridge.remove(food)
6     print('냉장고에 남은 것: {}'.format(fridge))
7 else:
8     print('냉장고에 없습니다.')
```



# 산수 문제 내기

사람이 입력한 답이 맞는지 평가하기

fridge.py

```
1 # -*- coding: utf-8 -*-  
2 n = float(input('0.1+0.2='))  
3 if n == 0.1 + 0.2:  
4     print('맞았습니다.')  
5 else:  
6     print('틀렸습니다.')
```

## 질문

이 코드의 문제점은 무엇인가?

# 산수 문제 내기

컴퓨터 부동소수점 표현의 오차를 고려해서 판정하기

arith\_test.py

```
1 # -*- coding: utf-8 -*-
2 n = float(input('0.1+0.2='))
3 if abs(n - (0.1 + 0.2)) < 10 ** -7:
4     print('맞았습니다.')
5 else:
6     print('틀렸습니다.')
```

# 조건문 잘 쓰기

## 체크해 보기

- `else`를 썼는가?
- 조건문이 중첩되지 않았는가?
- `if` 문과 `elif` 문의 조건에서 중복된 내용이 없는가?
- 조건이 참이 되는 경우를 정확하게 판단했는가?

# 요약

## 복합문의 구성과 문법

- 헤더에 콜론을 썼는가?
- 스위트를 들어 썼는가?

## 다음 시간 예고

## 반복문

## 조건문 키워드

- if/else: 2가지 경우
- if/elif/else: 3가지 경우
- if/elif/elif/else: 4가지 경우