

2021학년도 2학기 언어와 컴퓨터

제5강 조건문

박수지

서울대학교 인문대학 언어학과

2021년 9월 15일 수요일

오늘의 목표

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

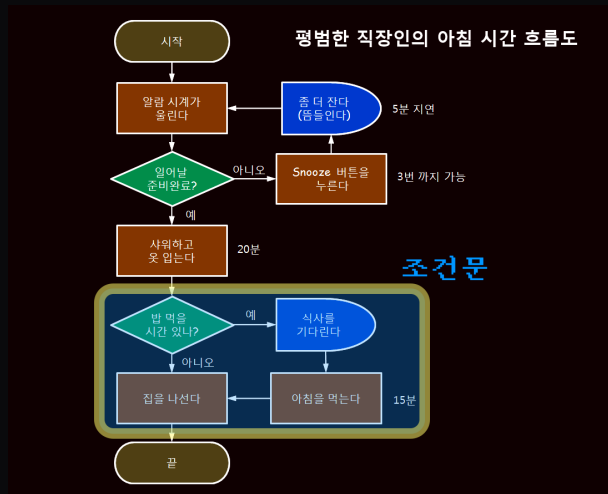
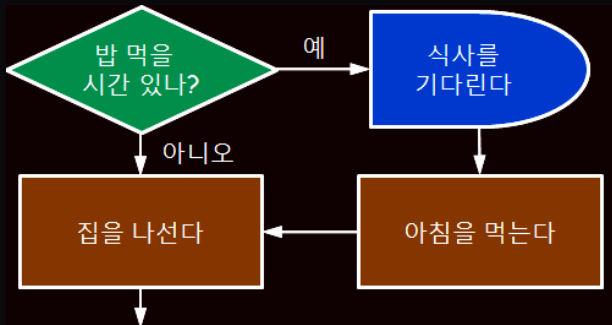


Figure: <https://www.flickr.com/photos/phploveme/4154450017>



파이썬 문법으로 표현하면...

```
1 if <밥 먹을 시간 있다>:  
2     <식사를 기다린다>  
3     <아침을 먹는다>  
4     <집을 나선다>  
5 else:  
6     <집을 나선다>
```

들여쓰기에 주의할 것!

복합문

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 들여쓰기(일정한 개수의 공백 문자)로 이루어진 코드들

if 문

헤더의 키워드가 if인 복합문

if 문

조건이 참일 때 실행할 스위트

형식

```
1 if <조건>:  
2     <실행할 스위트>
```

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

1행에서 temp의 값을 바꾸면서 실행해 보자.

질문

4행에서 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>
```

```
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>
```

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


조건문의 조건

- 불 값으로 평가되는 표현식
- ...뿐만 아니라 임의의 표현식

```
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

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

More pythonic

```
1 basket = ['egg', 'tofu']
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.

while 문

“답정너”

사용자가 정답을 입력할 때까지
반복해서 입력을 요구하는 프로그램

무한반복의 필수 요소

정지 조건 설정

- if 문과 결합
- break로 정지

```
1 while True:
2     n = input('2 + 3 = ')
3     if n == '5':
4         print('정답입니다.')
5         break
6     else:
7         print('정답을 입력하세요.')
```

for 문

100 이상이 될 때까지 합 구하기

문제

$S = \sum_{i=1}^n i \geq 100$ 을 만족하는
S의 최솟값과 i를 구하라.

정지 조건

$S \geq 100$

반복 범위

$i = 1, 2, \dots, ???$

코드

```
>>> S = 0 # 초기화
>>> for i in range(1, 100):
...     S = S + i
...     if S >= 100:
...         break
...     else:
...         pass
...
>>> S, i
(105, 14)
```

3의 배수 판정

1. 정수를 입력받아 3의 배수인지 아닌지 판정하기

3의 배수이다. \iff 3으로 나눈 나머지가 0이다.

```
1 n = input('정수를 입력하세요: ')
2 if int(n) % 3:
3     print('3의 배수가 아닙니다.')
4 else:
5     print('3의 배수입니다.')
```

질문

이 코드에서 어떤 예외가 발생할 수 있는가?

3의 배수 판정

2. 입력받은 값이 정수인지 아닌지 먼저 판정하기

```
1 n = input('정수를 입력하세요: ')
2 # 문자열이 수치로 이루어져 있는지 판단하는 메소드
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

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

관찰

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

3의 배수 판정

덜 효율적인 조건문

```
1 n = input('정수를 입력하세요: ')
2 if n.isnumeric() and (int(n) % 3):
3     print('3의 배수가 아닙니다.')
4 elif n.isnumeric() and not (int(n) % 3):
5     print('3의 배수입니다.')
6 else:
7     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 n = input('정수를 입력하세요: ')
2 if n.isnumeric() and (int(n) % 3):
3     print('3의 배수가 아닙니다.')
4 if n.isnumeric() and not (int(n) % 3):
5     print('3의 배수입니다.')
6 if not n.isnumeric():
7     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

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

냉장고에서 물건 꺼내기

물건이 냉장고에 있다면 꺼내고 남은 물건들을 알려주기

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

산수 문제 내기

사람이 입력한 답이 맞는지 평가하기

```
1 n = float(input('0.1 + 0.2 = '))
2 if n == 0.1 + 0.2:
3     print('맞았습니다.')
4 else:
5     print('틀렸습니다.')
```

질문

이 코드의 문제점은 무엇인가?

산수 문제 내기

컴퓨터 부동소수점 표현의 오차를 고려해서 판정하기

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

`abs()` 수치의 절댓값을 반환하는 함수

문자열에서 문장 부호 제거하기

입출력 예시

'(배고파...)' \mapsto '배고파'

코드

....

반복 범위

문자열의 모든 문자

조건부 연산

글자나 숫자일 때만 남기기

영어 명사에 부정관사 붙이기

입출력 예시

- 'apple' \mapsto 'an apple'
- 'banana' \mapsto 'a banana'

조건부 연산

모음으로 시작하면 'an'을, 아니면 'a'를 붙이기

코드

....

영어 명사의 복수형 만들기

입출력 예시

- 'apple' \mapsto 'apples'
- 'peach' \mapsto 'peaches'
- 'strawberry' \mapsto 'strawberries'

조건부 연산

....

코드

....

빈도 계산

입력 예시

리스트

```
['bacon', 'egg', 'bacon', 'ham', 'cheese', 'egg', 'bacon']
```

출력 목표

딕셔너리 {'bacon': 3, 'egg': 2, 'ham': 1, 'cheese': 1}

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 들여 썼는가?

다음 시간 예고

- 예외 처리 (try 문)
- 함수 정의 (def 문)

조건문 키워드

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