

# 2018학년도 2학기 언어와 컴퓨터

## 제6강 반복문

박수지

서울대학교 인문대학 언어학과

2018년 9월 19일 수요일

## 오늘의 목표

- 1 프로그램의 구성 요소 중 반복을 수행할 수 있다.
- 2 while, for 문을 사용할 수 있다.

## 연산자를 부정하는 방법

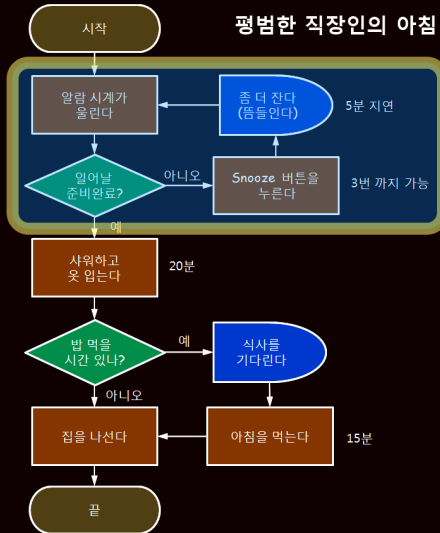
`==` `!=`

`is` `is not`

`in` `not in`

## 평범한 직장인의 아침 시간 흐름도

반복문



# 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

# while 문

조건이 참인 동안 반복해서 실행할 스위트

## 형식

```
1 while <조건>:  
2     <반복 실행할 스위트>
```

## 주의

if 문과 어떻게 다른가?

# while 문

## 무한반복

### 조건부 1회 실행

```
>>> if True:  
...     print('집에 가고 싶어')  
집에 가고 싶어
```

### 관찰

언제 >>> 대신 ...가 나오는가?

### 무한반복

```
>>> while True:  
...     print('집에 가고 싶어')  
집에 가고 싶어  
집에 가고 싶어  
집에 가고 싶어  
집에 가고 싶어  
집에 가고 싶어  
집에 가고 싶어  
집에 가고 싶어  
집에 가고 싶어  
집에 가고 싶어  
집에 가고 싶어  
.....
```

# while 문

“답정너”

## 무한반복의 필수 요소

### 정지 조건 설정

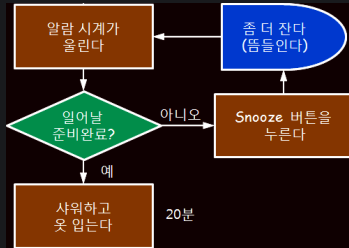
- if 문과 결합
- break로 정지

사용자가 정답을 입력할 때까지  
반복해서 입력을 요구하는 프로그램

```
1 while True:
2     n = input('2_+_3_=_')
3     if n == '5':
4         break
5     else:
6         print('정답을 입력하세요.')
```

# while 문

평범한 직장인의 아침 시간 흐름도



snooze.py

```

1 # -*- coding: utf-8 -*-
2 print('알람 시계가 울린다')
3 input('일어날 준비 완료?(Y/N)')
4 print('Snooze 버튼을 누른다')
5 print('좀 더 잔다')
6 print('샤워하고 옷 입는다')
  
```

## 할 일

위의 코드에 반복 요소를 추가해 보자.



# while 문

100 이상이 될 때까지 합 구하기

## 문제

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

## 반복 조건

$S \geq 100$ 이 달성되지 않았을 때  
 $\iff S < 100$ 일 때

## 코드

```
>>> S, i = 0, 0
>>> while S < 100:
...     i = i + 1
...     S = S + i
...
>>> S, i
(105, 14)
```

# while 문

## 초기화의 중요성

### 코드

```
>>> S, i = 0, 0
>>> while S < 100:
...     i = i + 1
...     S = S + i
...
>>> S, i
(105, 14)
```

### 주의

왼쪽 코드에서

- S, i = 0, 0을 빠뜨리면 어떻게 되는가?
- i = i + 1과 S = S + i의 순서를 바꾸면 어떻게 되는가?

# while 문

1부터 10까지의 합 구하기

## 문제

$S = \sum_{i=1}^{10} i$ 의 값을 구하라.

## 반복 조건

$i > 10$ 이 달성되지 않았을 때  
 $\iff i \leq 10$ 일 때

## 코드

```
>>> S, i = 0, 0 # 초기화
>>> while i <= 10:
...     S = S + i
...     i = i + 1
...
>>> S, i
(55, 11)
```

# while 문

업데이트 순서에 유의하기

## $S$ 를 먼저 갱신하는 경우

```
>>> S, i = 0, 0
>>> while i <= 10:
...     S = S + i
...     i = i + 1
...
>>> S, i
(55, 11)
```

## $S$ 를 나중에 갱신하는 경우

```
>>> S, i = 0, 0
>>> while i < 10:
...     i = i + 1
...     S = S + i
...
>>> S, i
(55, 10)
```

# for 문

반복 가능한 객체의 반복자를 따라 반복되는 스위트

## 형식

```
1  for <반복자> in <반복 가능한 객체>:  
2      <반복 실행할 스위트>
```

## 자료형의 반복 가능성

```
>>> '.__iter__' in dir(str) # 반복 가능  
True  
>>> '.__iter__' in dir(int) # 반복 불가능  
False
```

# for 문

1부터 10까지의 합 구하기

## 문제

$S = \sum_{i=1}^{10} i$ 의 값을 구하라.

## 반복 범위

$i = 1, 2, \dots, 10$

## 코드

```
>>> S = 0 # 초기화
>>> for i in range(1, 11):
...     S = S + i
...
>>> S, i
(55, 10)
```

# for 문

100 이상이 될 때까지 합 구하기

## 문제

$S = \sum_{i=1}^n i \geq 100$ 을 만족하는  
 $S$ 의 최솟값과  $i$ 를 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)
```

# while 문 vs. for 문

100 이상이 될 때까지 합 구하기

## while

```
>>> S, i = 0, 0
>>> while S < 100:
...     i += 1
...     S += i
...
>>> S, i
(105, 14)
```

## for

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



# while 문 vs. for 문

1부터 10까지 합 구하기

## while

```
>>> S, i = 0, 0
>>> while i < 10:
...     i += 1
...     S += i
...
>>> S, i
(55, 10)
```

## for

```
>>> S = 0 # 초기화
>>> for i in range(1, 11):
...     S += i
...
>>> S, i
(55, 10)
```

# for 문

임의의 "반복가능한" 자료형에 적용하기

## 리스트

```
>>> basket = ['bacon', 'egg', 'ham']
>>> for item in basket:
...     print(item.upper())
...
BACON
EGG
HAM
```

## 문자열

```
>>> word = 'egg'
>>> for char in word:
...     print(2 * char)
...
ee
gg
gg
```

# for 문

enumerate() 함수로 인덱스와 항목 모두 사용하기

## 예시

```
>>> word = 'egg'
>>> for i, char in enumerate(word):
...     print(i, char)
...
0 e
1 g
2 g
```

# for 문

딕셔너리에 대한 반복문

## 딕셔너리의 특징

인덱스가 아닌 키로 값을 찾는다.

## 예시

```
>>> hansol = {'HP': 80, 'MP': 90, 'class': 'mage'}
>>> for k, v in hansol.items():
...     print('속성: {}, 값: {}'.format(k, v))
...
속성: HP, 값: 80
속성: MP, 값: 90
속성: class, 값: mage
```

# for 문

## 반복문 중첩

### 출력 목표

```
월요일 오전
월요일 오후
화요일 오전
...
일요일 오후
```

### 반복 범위

- 월-일
- 오전-후

### 코드

```
>>> for day in '월화수목금토일':
...     for period in '전후':
...         print('{}요일 오{}'.format(day, period))
...
월요일 오전
월요일 오후
화요일 오전
화요일 오후
수요일 오전
....
토요일 오후
일요일 오전
일요일 오후
```

# 구구단 만들기

## 출력 목표

```
1 2 3 4 5 6 7 8 9
2 4 6 8 10 .....
.....
..... 49 56 72 81
```

## 반복 범위

- $i$  단 ( $i = 1, 2 \dots 9$ )
- $j$  번째 ( $i = 1, 2 \dots 9$ )

## 코드

```
....
```

# 문자열에서 문장 부호 제거하기

입출력 예시

'(배고파...)'  $\mapsto$  '배고파'

반복 범위

문자열의 모든 문자

조건부 연산

글자나 숫자일 때만 남기기

코드

....

# 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}
```

# 빈도 계산

## 반복 범위

리스트의 모든 항목

## 조건부 연산 과정

- 1 빈도표를 초기화하고
- 2 항목을 하나씩 확인하면서
- 3 빈도표에 항목이 있으면  
기존 빈도값에 1을 더하고,  
없으면 항목을 추가한다.

## 코드

...

# 통계 실험

## 단순 반복

### 방법

range(n)로 **똑같은** 코드를 n번 반복할 수 있다.

### 코드

```
>>> for i in range(3):  
...     print('배고파!')  
...  
배고파!  
배고파!  
배고파!
```

# 통계 실험

## 주사위 던지기

### 방법

1에서 6 사이의 정수를 임의로 추출한다.

### 예시

```
>>> from random import randrange
>>> randrange(1, 7)
2
>>> randrange(1, 7)
1
>>> for i in range(3):
...     print(randrange(1, 7))
...
3
5
5
```

# 통계 실험

주사위를 던져서 나온 눈의 기댓값 확인하기

## 방법

주사위 던지기를 1000번 시행하여 눈의 값들의 평균을 구한다.

- 평균을 구하기 위해서는 시행 결과를 저장해야 한다.

## 코드

```
>>> spots = []  
>>> for i in range(1000): # 1000회 반복  
...     spots.append(randrange(1, 7)) # 주사위를 던져  
...  
>>> print(sum(spots) / len(spots))  
3.404
```

# 통계 실험

6이 나올 때까지 주사위를 던지는 횟수의 기댓값 구하기

## 방법

6이 나올 때까지 주사위를 던지는 시행을 10000회 반복하여 횟수들의 평균을 구한다.

- 6이 나올 때까지 주사위를 던지는 방법

- 1 주사위를 던져서 6이 안 나오면 계속 던진다.
- 2 6이 나오면 그만 던진다.

반복문과 조건문이 모두 필요하다.

# 통계 실험

6이 나올 때까지 주사위를 던지는 횟수의 기댓값 구하기

## 코드

```
>>> ns = []
>>> for i in range(10000):
...     n, k = 0, 0
...     while k != 6:
...         k = randrange(1, 7)
...         n += 1
...     ns.append(n)
...
>>> print(sum(ns) / len(ns))
5.9328
```



# 요약

## 반복문

`while` 연산 결과에 대한 조건부 반복  
`for` 반복 범위를 지정한 반복

## 반복 중단

`break`  
■ 비교  
`pass`

## 정보

- 문자열 메소드 `str.endswith()`
- 탭 문자 `'\t'`
- 정수 추출 `from random import randrange`

## 더 생각해 볼 것

- `continue`에 대해서 알아보자.
- `str.startswith()` 메소드에 대하여 알아보자.