

Chapter4 - 제어문과 반복문

Condition

조건문이란?

- 조건에 따라 특정한 동작을 하게 하는 명령어
- 조건문은 **조건을 나타내는 기준**과 **실행해야 할 명령**으로 구성됨
- if, else, elif

If-else문

- 가장 기본적인 조건문으로 조건에 따른 명령을 실행

조건 판단 방법

- if 다음에 조건을 표기하여 참 또는 거짓을 판단함
- `==`: 할당

비교연산자	비교상태	설명
<code>x < y</code>	~보다 작음	x과 y보다 작은지 검사
<code>x > y</code>	~보다 큼	x과 y보다 큰지 검사
<code>x == y</code>	같음	x와 y과 같은지 검사 (값과 메모리 주소)
<code>x is y</code>		
<code>x != y</code>	같지 않음	x와 y과 다른지 검사 (값과 메모리 주소)
<code>x is not y</code>		
<code>x >= y</code>	크거나 같음	x과 y보다 이상인지 검사
<code>x <= y</code>	작거나 같음	x과 y보다 이하인지 검사

논리 키워드 : and, or, not

LOOP

반복문이란?

- 정해진 동작을 반복적으로 수행하게 하는 명령문
- 반복 시작 조건, 종료 조건, 수행 명령
- ex) for, while

For문

- 기본적인 반복문, 반복 범위를 지정하여 반복문 수행

```
for loop in [1,2,3,4,5]:  
    print ("hello")
```

- ① loop 변수에 1 할당
- ② "Hello" 출력
- ③ 리스트(대괄호속 숫자들) 있는 값 차례로 loop 할당
- ④ 5까지 할당한 후 반복 block 수행 후 종료

range () 사용하기

```
for loop in [1,2,3,4,5]:
    print ("hello")
for loop in range(0,5):
    print "hello"
```

왜 range(1,5) 과 아닌 range (0,5) 인가?

: range()는 마지막 숫자 바로 앞까지 리스트를 만들어줌

즉, range(1,5) = [1,2,3,4] 까지 같은 의미

※ range(0,5) = [0,1,2,3,4] = range(5)는 같은 의미

For문의 다양한 반복문 조건 표현

1. 문자열을 한자씩 리스트로 처리
2. 각각의 문자열 리스트로 처리
3. 간격을 두고 세기
4. 역순으로 반복문 수행

While문

-조건이 만족하는 동안 반복 명령문을 수행

조건이 만족하는 동안 반복 명령문을 수행

```
i = 1
while i < 10:
    print (i)
    i += 1
```

- ① i 변수에 1 할당
- ② i가 10 미만인지 판단
- ③ 조건에 만족할 때 i 출력, i에 1을 더함
- ④ i가 10이 되면 반복 종료

for문은 while문으로 변환 가능

반복 실행횟수를 명확히 알 때

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

반복 실행횟수가 명확하지 않을 때

```
i = 0
while i < 5:
    print (i)
    i = i + 1
```

반복의 제어 – break, continue

break 특정 조건에서 반복 종료

```
for i in range(10):
    if i == 5: break
    print (i)
print ("EOP")
```

i가 5가 되면 반복 종료

반복 종료 후 "EOP" 출력

continue 특정 조건에서 남은 반복 명령 skip

```
for i in range(10):
    if i == 5: continue
    print (i)
print ("EOP")
```

i가 5가 되면 i를 출력하지 않음

반복 종료 후 "EOP" 출력

반복의 제어 – else

반복 조건이 만족하지 않을 경우 반복 종료 시 1회 수행

```
for i in range(10):  
    print (i),  
else:  
    print ("EOP")
```

Editor

```
i = 0  
while i < 10:  
    print (i),  
    i += 1  
else:  
    print ("EOP")
```

Editor

※ break로 종료된 반복문은 else block이 수행되지 않음

가변적인 중첩 반복문(variable nested loops)

- 반복문은 사용자의 입력에 따라 가변적으로 반복되고 하나의 반복이 아닌 중복되어 반복이 일어남

How to debug code

Debugging(디버깅)

- 코드의 오류를 발견하여 수정하는 과정
- 문법적 에러를 찾기 위한 에러 메시지 분석
- 논리적 에러를 찾기 위한 테스트

문법적 에러

오타자, 대소문자 구분, 들여쓰기, 잘못된 괄호 사용

논리적 에러

뜻대로 실행이 안되는 코드, 중간 중간 프린터 문을 찍어서 확인

If__name__ == "__main__" - 파이썬 셸에서 실행이 되지 않는다. (import문에서 실행되지 않는다.)