

13차시

반복을 제어하는 for문과 while문



♪ 학습개요

- … 반복 개요
- ··· for in
 - in 뒤의 시퀀스
 - 내장 함수 range()
- ··· while
 - 조건 블록

⚠ 학습목표

- … 반복이 필요한 상황을 설명할 수 있다.
- ··· 반복을 위한 while 과 for 문장을 사용할 수 있다.
- … 함수 range()를 사용할 수 있다.
- … 구구단을 출력하는 프로그램을 작성할 수 있다.

Chapter 1.

반복 개요

PYTHON PROGRAMMING



⚠ 일상생활과 같은 반복의 실행

★ 콜론과 반복 몸체인 블록 구성이 반드시 필요

<반복 조건>에 따른 반복

while <반복 조건>: 반복 몸체인_문장들 <시퀀스>항목마다 반복

for 변수 in <시퀀스>: 반복 몸체인_문장들

[그림13-1] 파이썬의 while문 과 for문

일상생활과 같은 반복의 실행

```
print ('로그인')
print ('캐릭터 및 아이템 등 여러 옵션 선택')
print ('게임 실행')
print ('게임 종료')
```

반복 결과 로그인 캐릭터 및 아이템 등 여러 옵션 선택 게임 종료 로그인 캐릭터 및 아이템 등 여러 옵션 선택 게임 실행 게임 종료 로그인 캐릭터 및 아이템 등 여러 옵션 선택 게임 실행 게임 종료

[그림13-2] 게임과 반복

Chapter 2.

for in

PYTHON PROGRAMMING





+ 정해져 있는 시퀀스의 항목 값으로 반복을 실행

- 여러 개의 값을 갖는 **시퀀스**에서 변수에 하나의 값을 순서대로 할당한다.
- 할당된 변숫값을 갖고 **블록의 문장들(문장 1, 문장 2)을 순차적으로 실행한다.**
- 반복 몸체인 문장 1, 문장 2에서 변수를 사용할 수 있다.
- 시퀀스의 그 다음 값을 변수에 할당해 다시 **반복 블록**을 실행한다.
- 이러한 과정을 시퀀스의 마지막 항목까지 수행한다.
- 시뭔스의 마지막 항목까지 실행한 후 선택 사항인 **else:** 블록을 실행하고 반복을 종료한다.

→ III이선 프로그래밍 반복을 제어하는 for문과 while문





문자열과 같은 시퀀스(레인지, 리스트, 튜플, 딕셔너리, 집합 등)가 위치하고 반드시 콜론이 필요하다.

for 변수 in 시퀀스:



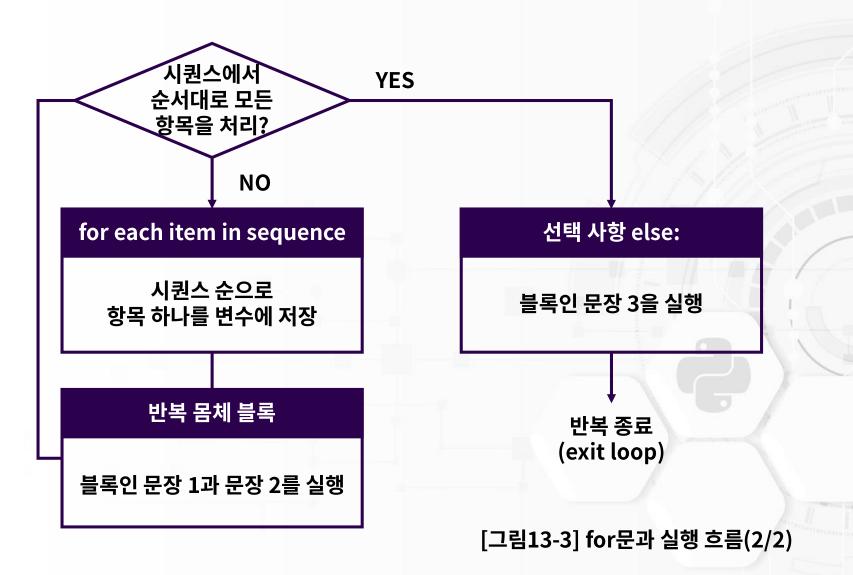
문장 3

else: 이후의 블록인 문장 3은 for 시퀀스의 모든 항목으로, 반복이 종료된 마지막에 실행된다. 블록인 문장 1과 문장 2를 반복 몸체라고 하며, 변수에 할당된 하나의 값으로 실행되는 반복 단위다.

[그림13-3] for문과 실행 흐름(1/2)



⚠ 정해져 있는 시퀀스의 항목 값으로 반복을 실행





① 실수의 나열인 시퀀스에서 for 구문으로 합과 평균 구하기

[코딩실습] 수의 나열에서 합과 평균 구하기 난이도 기본 1. sum = 02. for i in 1.1, 2.5, 3.6, 4.2, 5.4: sum += i 3. print(i, sum) 5. else: print('합: %.2f, 평균: %2f' % (sum, sum / 5)) 6. 2, 5번 줄에는 콜론을 잊지 말에야 하고, 3, 4, 6번 줄은 들여쓰기로 입력해야 한다. 1.1 1.1 2.5 3.6 3.6 7.2 결과 4.2 11.4 5.4 16.8 합: 16.80, 평균: 3.36



⚠ 내장 함수 range()를 사용한 for문

- **+** range(5)
 - 정수 0에서 4까지 5개의 항목인 정수로 구성되는 시퀀스
 - 01234

```
+ range(1, 10, 2)
```

```
for i in range(1, 10, 2)
... print(i, end = ' ')
...
1 3 5 7 9
```

- start: 시퀀스의 시작
- stop: 시퀀스의 끝(미만), 실질적으로 stop-1까지
- step: 증가 값

파이썬 프로그래밍 반복을 제어하는 for문과 while문

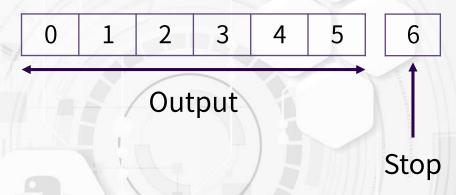




+ 범위라는 단어를 쓰는 함수 range()

■ 내장 함수 range()는 일정 간격의 정숫값의 나열인 시퀀스를 생성해주는 함수다. 함수의 인자는 총 3개로, 적어도 1개의 인자를 지정해 줘야 한다. 만약 인자 중 하나만 지정하면 자동으로 stop 값이 지정되는데, stop의 이전 정숫값까지 포함된다. 시퀀스의 시작인 start와 인접한 숫자의 간격인 step은 자동으로 각각 0, 1로 지정된다. 인자를 2개 지정하면 자동으로 start와 stop 값을 설정하게 되며, 증가값은 자동으로 1로 지정된다.

Python range(6)



- 모든 인자는 문자열 또는 부동소수점 숫자나 다른 유형이 올 수 없으며, 반드시 정수여야 한다





+ 범위라는 단어를 쓰는 함수 range()

■ 다음 코드에서 볼 수 있듯이 **list() 함수 내부에 range() 함수를 사용**하면 바로 **시퀀스 결과**를 알 수 있다.

```
>>> list(range(4))
[0, 1, 2, 3]
>>> list(range(6, 10))
[6, 7, 8, 9]
>>> list(range(10, 20, 2))
[10, 12, 14, 16, 18]
```

함수호출	인자	결과
range(4)	stop	0, 1, 2, 3
range(6,10)	start, stop	6,7,8,9
range(10, 20, 2)	start, stop, step	10, 12, 14, 16, 18



① 표준 입력값을 최소 한 번 포함하는 두 자리 정수 찾기

[코딩실습] 지정된 최소 한 자릿수가 포함된 두 자리 정수 찾기 난이도 응용 1. n = input('10진수의 한 자릿수 입력 >> ') 2. print('두 자릿수 정수에서 최소 한 자릿수가 %s인 정수 찾기' % n) 3. print('결과').center(50, '= ')) 5. for i in range(10, 100): snum = str(i)6. if n in snum: print(i, end=' ') 8. 6, 7번 줄은 들여쓰기로 입력한다. 10진수의 한 자릿수 입력 >> 7 두 자릿수 정수에서 최소 한 자릿수가 7인 정수 찾기: 결과 ========= 결과 ======= 17 27 37 47 57 67 70 71 72 73 74 75 76 77 78 79 87 97

Chapter 3.

while

PYTHON PROGRAMMING



⚠ 반복 구조가 간단한 while 반복

논리 표현식이 위치하며 반드시 콜론이 필요하다.

while 논리 표현식:

문장 1

문장 2

블록인 문장 1과 문장 2를 반복 몸체라고 하며, 논리 표현식이 True이면 실행되는 반복 단위다.

else:

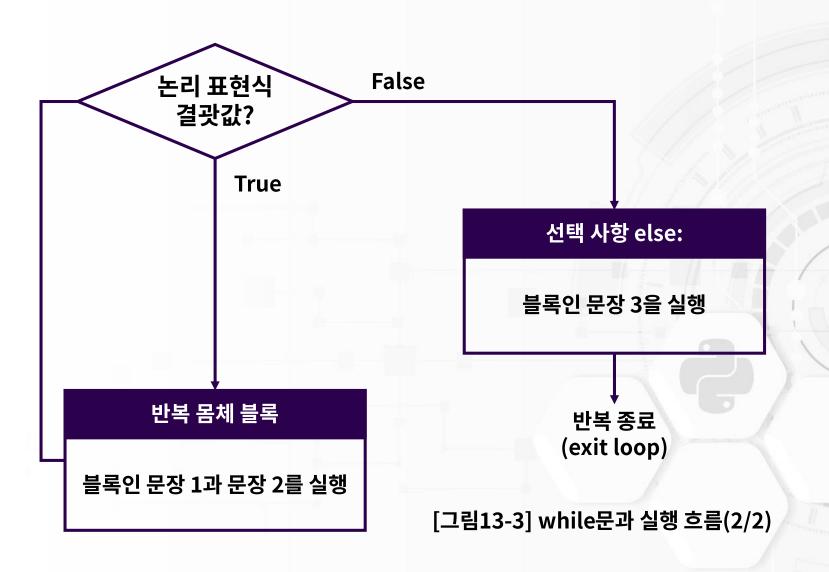
문장 3

else: 이후의 블록인 문장 3은 논리 표현식이 False가 돼 반복이 종료된 마지막에 실행된다.

[그림13-3] while문과 실행 흐름(1/2)



① 반복 구조가 간단한 while 반복



⚠ 반복 구조가 간단한 while 반복

```
>>> n = 1
>>> while n <= 5:
        print(n, end =' ')
        n += 1
... else:
        print("\n 반복 while 종료: n => %d" % n)
```



⚠ 일상 코딩: 어린이를 위한 놀이 기구 탑승 처리

[코딩실습] 어린이를 위한 놀이 기구 탑승 검사

난이도 응용

```
1. MAXNUM = 4
2. MAXHEIGHT = 130
3.
4. more = True
5. cnt = 0
6. while more:
       height = float(input("키는?"))
       if height < MAXHEIGHT:</pre>
8.
9.
           cnt += 1
           print('들어가세요.', '%명' % cnt)
10.
11.
       else:
12.
           print('커서 못 들어갑니다.')
       if cnt == MAXNUM:
13.
14.
           more = False
15. else:
       print('%d명 모두 찼습니다. 다음 번에 이용하세요.' % cnt)
16.
```



⚠ 일상 코딩: 어린이를 위한 놀이 기구 탑승 처리

주의 변수 cnt를 사용해 한 번에 탑승 인원인 4명을 검사한다.

키는? 120 들어가세요. 1명 키는? 110 들어가세요. 2명 키는? 105 들어가세요. 3명 키는? 107 들어가세요. 4명 4명 모두 찼습니다. 다음 번에 이용하세요. 키는? 125 들어가세요. 1명 키는? 129 들어가세요. 2명 키는? 131 커서 못 들어갑니다. 키는? 110 들어가세요. 3명 키는? 105 들어가세요. 4명 4명 모두 찼습니다. 다음 번에 이용하세요.



↑ 일상 코딩: 반복 for의 중첩 표준 구구단

```
i = 6
for j in range(1, 10):
    print('%d * %d = %2d' % (i, j, i*j), end = ' ')
print()
```

실행 결과

```
6*1=66*2=126*3=186*4=24...중략...6*8=486*9=54
```



⚠ 일상 코딩: 놀이공원에서 키를 제한하는 놀이 기구

```
[코딩실습] 전형적인 구구단 출력
                                                                                          난이도 응용
   1. for i in range(2, 10):
          for j in range(1, 10):
   2.
              print('%d * %d = %2d' % (i, j, i * j), end = ' ')
   3.
          print()
      콜론과 들여쓰기를 잊지 말자.
      2*1 = 2 2*2 = 4 2*3 = 62*4 = 82*5 = 10 2*6 = 12 2*7 = 142*8 = 162*9 = 18
      3 * 1 = 3 3 * 2 = 6 3 * 3 = 9 3 * 4 = 12 3 * 5 = 15 3 * 6 = 18 3 * 7 = 12 3 * 8 = 24 3 * 9 = 27
      4 * 1 = 4 4 * 2 = 8 4 * 3 = 12 4 * 4 = 16 4 * 5 = 20 4 * 6 = 24 4 * 7 = 28 4 * 8 = 32 4 * 9 = 36
      5 * 1 = 5 5 * 2 = 10 5 * 3 = 15 5 * 4 = 20 5 * 5 = 25 5 * 6 = 30 5 * 7 = 35 5 * 8 = 40 5 * 9 = 45
결과
      6 * 1 = 6 6 * 2 = 12 6 * 3 = 18 6 * 4 = 24 6 * 5 = 30 6 * 6 = 36 6 * 7 = 42 6 * 8 = 48 6 * 9 = 54
      7*1=77*2=147*3=217*4=287*5=357*6=427*7=497*8=567*9=63
      8 * 1 = 8 8 * 2 = 16 8 * 3 = 24 8 * 4 = 32 8 * 5 = 40 8 * 6 = 48 8 * 7 = 56 8 * 8 = 64 8 * 9 = 72
      9 * 1 = 9 9 * 2 = 18 9 * 3 = 27 9 * 4 = 36 9 * 5 = 45 9 * 6 = 54 9 * 7 = 63 9 * 8 = 72 9 * 9 = 81
```

⚠ 반복 개요



… 조건 블록



SUMMARY



⚠ for in

- ··· in 뒤의 시퀀스
- ··· 내장 함수 range()