

## 제6장 반복문

### ■ 학습목표

- 기본 for 문, 중첩 for 문의 형식과 사용법을 익힌다
- while 문의 형식과 사용법을 익힌다
- 반복문을 다양하게 활용하는 방법을 익힌다

## 기본 for문

### ■ 기본 형식

```
for 변수 in range(시작값, 끝값+1, 증가값):  
    반복할문장
```

- 예 : range() 함수 사용과 내부적 변경

```
for i in range(0, 3, 1):  
    print("for 문 공부중~")
```

```
for i in [0,1,2]:  
    print("for 문 공부중~")
```

## 기본 for문

### • 예

```
for i in range(0, 3, 1):  
    print("안녕? for 문 공부 중. ^^")
```

```
안녕? for 문 공부 중. ^^  
안녕? for 문 공부 중. ^^  
안녕? for 문 공부 중. ^^
```

```
for _ in range(0, 3, 1):  
    print("안녕? for 문 공부 중. ^^")
```

```
안녕? for 문 공부 중. ^^  
안녕? for 문 공부 중. ^^  
안녕? for 문 공부 중. ^^
```

```
for i in range(2, -1, -1):  
    print("%d : 안녕? for 문 공부 중. ^^" %i)
```

```
2 : 안녕? for 문 공부 중. ^^  
1 : 안녕? for 문 공부 중. ^^  
0 : 안녕? for 문 공부 중. ^^
```

## 기본 for문

### • 예

```
for i in range(1, 6, 1):  
    print("%d " %i, end = " ")
```

```
1 2 3 4 5
```

```
for i in range(1, 6, 1):  
    print("%d " %i)
```

```
1  
2  
3  
4  
5
```

## 기본 for문

- 예: 1~10 의 합계구하기

```
i = 0
hap = hap + i
NameError: name 'hap' is not defined. Did you mean: 'map'?

for i in range(1, 11, 1):
    hap = hap + i

print("1에서 10까지의 합계 : %d" % hap)

i, hap = 0, 0

for i in range(1, 11, 1):
    hap = hap + i

print("1에서 10까지의 합계 : %d" % hap)
```

1에서 10까지의 합계 : 55

## 기본 for문

- 예: 500~1000 사이의 홀수 합계

```
i, hap = 0, 0
500~1000 홀수 합계 : 187500

for i in range(501, 1001, 2):
    hap = hap + i

print("500~1000 홀수 합계 : %d" % hap)
```

## 기본 for문

- 예: 키보드로 입력한 값까지 합계 구하기

```
i, hap = 0, 0
num = 0

num = int(input("값 : "))

for i in range(1, num+1, 1):
    hap = hap + i

print("1 ~ %d 합계 : %d" % (num, hap))
```

값 : |

값 : 100  
1 ~ 100 합계 : 5050

## 기본 for문

- 예: 시작값, 끝값, 종료값 입력받아 합계 구하기

```
i, hap = 0, 0
num1, num2, num3 = 0, 0, 0

num1 = int(input("시작값 : "))
num2 = int(input("끝값 : "))
num3 = int(input("증가값 : "))

for i in range(num1, num2+1, num3):
    hap = hap + i

print("%d ~ %d 까지 %d씩 증가시킨 값의 합계 : %d" % (num1, num2, num3, hap))
```

시작값 : 2  
끝값 : 300  
증가값 : 3  
2 ~ 300 까지 3씩 증가시킨 값의 합계 : 15050

## 기본 for문

- 예: 단 수 입력받아 해당 구구단 출력

```
i, dan = 0, 0
```

```
dan = int(input("단 : "))
```

```
for i in range(1, 10, 1):
    print("%d X %d = %2d" % (dan, i, dan * i))
```

```
단 : 7
7 X 1 = 7
7 X 2 = 14
7 X 3 = 21
7 X 4 = 28
7 X 5 = 35
7 X 6 = 42
7 X 7 = 49
7 X 8 = 56
7 X 9 = 63
```

## 중첩 for문

- 중첩 for문

- 실행횟수 = 바깥 for문 반복횟수 X 안쪽 for문 반복횟수

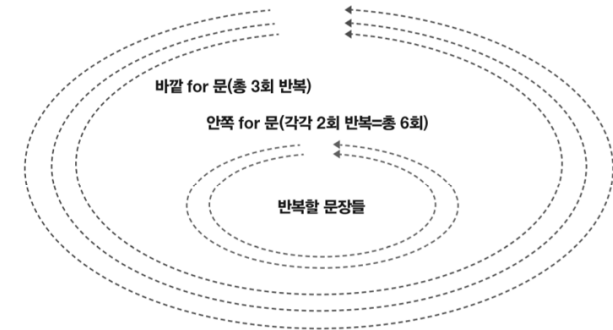


그림 6-2 중첩 for 문의 작동 개념

## 중첩 for문

- 예: 

```
for i in range(0, 3, 1):
    for k in range(0, 2, 1):
        print("파이썬은 꿀잼 (i값 : %d, k값 : %d)" % (i,k))
```



그림 6-3 중첩 for 문에서 i와 k값 변화

```
파이썬은 꿀잼 (i값 : 0, k값 : 0)
파이썬은 꿀잼 (i값 : 0, k값 : 1)
파이썬은 꿀잼 (i값 : 1, k값 : 0)
파이썬은 꿀잼 (i값 : 1, k값 : 1)
파이썬은 꿀잼 (i값 : 2, k값 : 0)
파이썬은 꿀잼 (i값 : 2, k값 : 1)
```

## 중첩 for문

- 예:



1에서 9까지 계속 반복해서 증가(안쪽 for 문 : k 변수)

그림 6-4 구구단에서 i와 k 변수 추출

## 중첩 for문

### 예:

i, k = 0, 0

```
for i in range(2, 10, 1):
    for k in range(1, 10, 1):
        print("%d X %d = %2d" %(i, k, i * k))
    print("")
```

2 X 1 = 2	8 X 1 = 8
2 X 2 = 4	8 X 2 = 16
2 X 3 = 6	8 X 3 = 24
2 X 4 = 8	8 X 4 = 32
2 X 5 = 10	8 X 5 = 40
2 X 6 = 12	8 X 6 = 48
2 X 7 = 14	8 X 7 = 56
2 X 8 = 16	8 X 8 = 64
2 X 9 = 18	8 X 9 = 72
3 X 1 = 3	9 X 1 = 9
3 X 2 = 6	9 X 2 = 18
3 X 3 = 9	9 X 3 = 27
3 X 4 = 12	9 X 4 = 36
3 X 5 = 15	9 X 5 = 45
3 X 6 = 18	9 X 6 = 54
3 X 7 = 21	9 X 7 = 63
3 X 8 = 24	9 X 8 = 72
3 X 9 = 27	9 X 9 = 81

## while문

### while 문

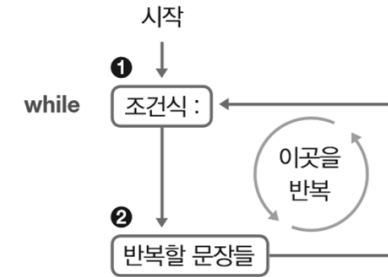


그림 6-6 while 문의 형식과 순서도

- for 문: range() 함수에서 결정된 반복횟수만큼 반복
- while 문 : 반복 횟수를 결정하기보다는 조건식이 참일 때 반복

## while문

### 예

0 : 안녕? for 문 공부 중. ^^	0 : while 문 공부 중. ^^
1 : 안녕? for 문 공부 중. ^^	1 : while 문 공부 중. ^^
2 : 안녕? for 문 공부 중. ^^	2 : while 문 공부 중. ^^

```
for i in range(0, 3, 1):
    print("%d : 안녕? for 문 공부 중. ^^" %i)
```

```
i=0
while i<3:
    print("%d : while 문 공부 중. ^^" %i)
    i = i+1
```

## while문

### 예: 1~10까지의 합계 구하기

i, hap = 0, 0

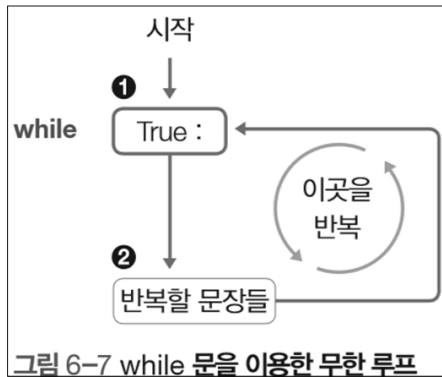
```
i = 1
while i < 11:
    hap = hap + i
    i = i + 1
```

```
print("1 ~10까지의 합계 : %d" % hap)
```

1 ~10까지의 합계 : 55

## While문 : 무한 루프 적용

- 무한 루프 적용
  - 조건식을 True로 지정 → 중지하려면 Ctrl+C 누름



## While문 : 무한 루프 적용

- 예: 숫자 2개 입력받아 더하기

```
hap = 0
a, b = 0, 0

while True :
    a = int(input("첫 번째 수 : "))
    b = int(input("두 번째 수 : "))
    hap = a + b
    print("%d + %d = %d" % (a, b, hap))
```

```
첫 번째 수 : 55
두 번째 수 : 22
55 + 22 = 77
첫 번째 수 : 77
두 번째 수 : 128
77 + 128 = 205
첫 번째 수 :
Traceback (most recent call last):
  File "C:\atopos\_LECTURE\_2022_2\파이썬
\code\Code06-10.py", line 5, in <module>
    a = int(input("첫 번째 수 : "))
KeyboardInterrupt
```

## While문 : 무한 루프 적용

- 예: 두 수와 연산자 입력받아 연산 수행하는 프로그램

```
ch = ""
a, b = 0, 0

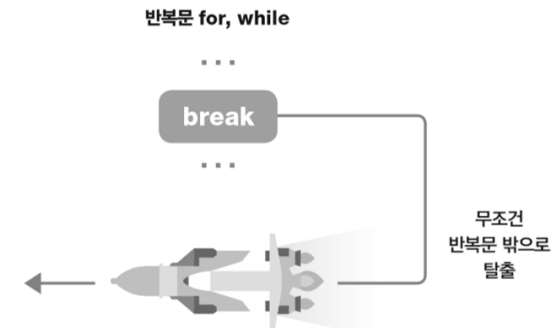
while True :
    a = int(input("첫 번째 수 : "))
    b = int(input("두 번째 수 : "))
    ch = input("연산자(+, -, *, /, %, //, **) : ")

    if (ch == "+") : print("%d + %d = %d" % (a, b, a + b))
    elif (ch == "-") : print("%d - %d = %d" % (a, b, a - b))
    elif (ch == "*") : print("%d * %d = %d" % (a, b, a * b))
    elif (ch == "/") : print("%d / %d = %5.2f" % (a, b, a / b))
    elif (ch == "%") : print("%d %% %d = %d" % (a, b, a % b))
    elif (ch == "//") : print("%d // %d = %d" % (a, b, a // b))
    elif (ch == "**") : print("%d ** %d = %d" % (a, b, a ** b))
    else : print("연산자를 잘못 입력했습니다")
```

```
첫 번째 수 : 22
두 번째 수 : 33
연산자(+, -, *, /, %, //, **) : *
22 * 33 = 726
첫 번째 수 : 10
두 번째 수 : 4
연산자(+, -, *, /, %, //, **) : %
10 % 4 = 2
첫 번째 수 :
```

## break 문과 continue문

- break문
  - 계속되는 반복을 논리적으로 빠져나가는 방법



## break 문과 continue문

```

for i in range(1, 100) :
    print("for문을 %d 번 실행" %i)
    break

```

for문을 1 번 실행

```

첫 번째 수 : 55
두 번째 수 : 22
55 + 22 = 77
첫 번째 수 : 77
두 번째 수 : 128
77 + 128 = 205
첫 번째 수 : 0
0을 입력해 반복문 탈출

```

```

hap = 0
a, b = 0, 0

while True :
    a = int(input("첫 번째 수 : "))
    if a == 0 :
        break
    b = int(input("두 번째 수 : "))
    hap = a + b
    print("%d + %d = %d" % (a, b, hap))

print("0을 입력해 반복문 탈출")

```

## break 문과 continue문

```

hap, i = 0,0

for i in range(1,101) :
    hap += i

    if hap >= 1000 :
        break

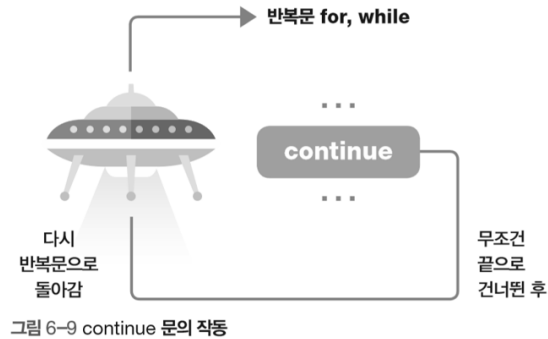
print("1~100의 합계를 최초로 1000이 넘게 하는 숫자 : %d" % i)

```

1~100의 합계를 최초로 1000이 넘게 하는 숫자 : 45

## break 문과 continue문

- continue문 : 반복문으로 다시 돌아가게 하는 문장



## break 문과 continue문

```

hap, i = 0,0

for i in range(1, 101) :
    if i % 3 == 0 :
        continue

    hap += i

print("1~100의 합계(3의 배수 제외) : %d" % hap)

```

1~100의 합계(3의 배수 제외) : 3367