
문화 A0019

# 파이썬프로그래밍

김 태 완

kimtwan21@dongduk.ac.kr

# 오늘 배울 내용

- 반복문
  - 주로 조건문 (if문)과 함께 많이 사용

  - for
    - 반복하는 횟수를 알 수 있을 때
    - 이중 for문

  - while
    - 반복하는 횟수를 알 수 없을 때
    - 무한 반복 및 break

# 반복문

- 반복문을 사용해야 하는 이유
  - 수백 줄의 코드를 단 몇 줄로 줄이는 효율을 발휘함
  - 예시

```
안녕하세요. 어서 오세요. QR 체크 부탁드립니다.
안녕하세요. 어서 오세요. QR 체크 부탁드립니다.
안녕하세요. 어서 오세요. QR 체크 부탁드립니다.
```

  - 방법 1 : print 함수 세 번 작성

```
print("안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
print("안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
print("안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
```

  - 방법 2 : 반복문 (for문) 사용

```
for i in range(3) :
    print("안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
```

# for문

- for문 형식

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

  - range(시작값, 끝값+1, 증가값) : 지정된 범위의 값을 반환함
  - range(0, 3, 1) : 0에서 시작해서 2까지 1씩 증가하는 값들을 반환함
    - 증가 값을 생략할 경우 1로 인식함
    - 따라서 range(0, 3, 1)은 range(0, 3)과 동일함
    - 시작 값이 0이기 때문에 시작 값도 생략 가능함
    - 따라서 range(3)만 써도 range(0, 3, 1)과 동일하게 인식함

| 일상생활에서의 범위 | 컴퓨터에서의 범위 |
|---|---|
| 1~10 | 0~10 |
| 11~20 | 10~20 |
| 21~30 | 20~30 |
| 31~40 | 30~40 |
| 41~50 | 40~50 |

# for문

- for문 + range 함수

```
print("안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
print("안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
print("안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
```

=

```
for i in range(3) :
    print("안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
```

=

```
for i in range(0, 3, 1) :
    print("안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
```

## for문

- i 값에 접근
  - 각 행의 맨 앞에 i 값 출력

```
for i in range(0, 3, 1) :
    print("안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
```

```
for i in range(0, 3, 1) :
    print(i, ": 안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
```

```
print("0: 안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
print("1: 안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
print("2: 안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
```

  - 예시 : 1~10까지 숫자들을 차례대로 출력하기

```
for i in range(1, 11, 1) :
    print(i)
```

12345678910

## 예시

for문을 이용하여 실행 결과와 같은 형태로 출력해보자.

```
* * * * * * * * * * *
* * * * * * * * * * *
* * * * * * * * * * *
* * * * * * * * * * *
* * * * * * * * * * *
```

## 예시

1부터 30까지의 숫자 중 3의 배수만 출력하세요.

```
3
6
9
12
...
```

## for문

- 예시 : 팩토리얼 (!) 값 구하기

2 이상의 숫자를 입력해 주세요 : 7

5040

2 이상의 숫자를 입력해 주세요 : 3

6

# for문

- for문 활용한 반복적인 덧셈

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

print("1에서 10까지의 합 = \n", res)
```

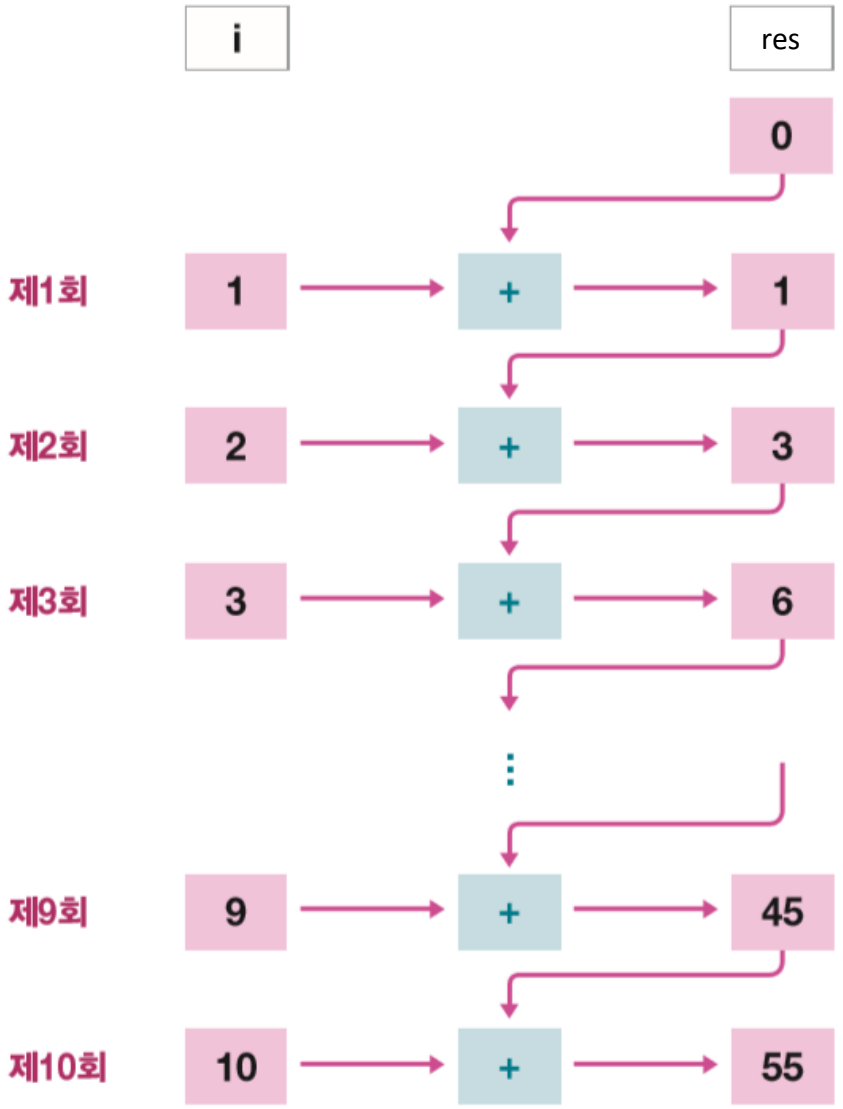
NameError : name ‘res’ is not defined

- res 변수를 선언하지 않아서 오류
- res 변수에 어떤 값이 있어야만 다시 res 변수에 자기 자신에 누적해서 계산 가능

```
res = 0

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

print("1에서 10까지의 합 = \n", res)
```

# for문

```
res = 0

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

print("1에서 10까지의 합 = \n", res)
```



# for문

- for문 활용한 반복적인 덧셈 : 1000~2000 사이에서 홀수의 합을 구하는 프로그램

```
res = 0

for i in range(1001, 2001, 2) :
    res = res + i

print("100에서 2000까지의 홀수의 합 = \n", res)
```

```
res = 0

for i in range(1001, 2001, 1) :
    if i % 2 == 1 :
        res = res + i

print("100에서 2000까지의 홀수의 합 = \n", res)
```

## for문

- 예시 : 구구단 구하기

구구단 몇 단을 계산할까요? : 7

7 x 1 = 7

7 x 2 = 14

...

7 x 9 = 63

# 중첩 for문

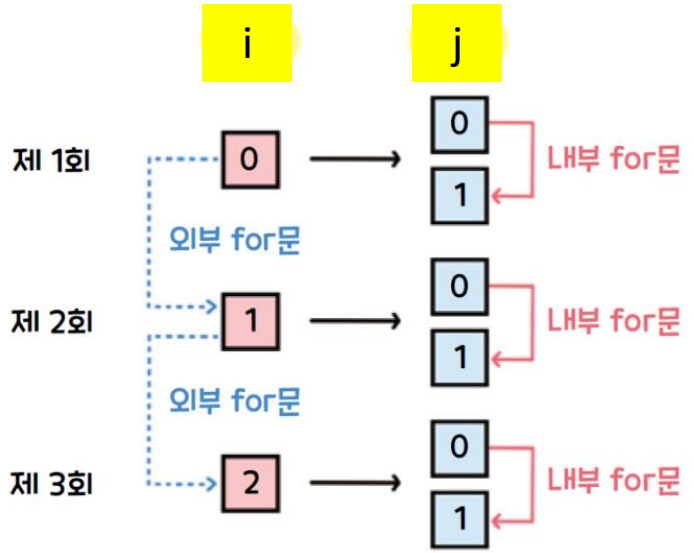
- for문 안에 또 for문을 사용할 수도 있음
- 이를 중첩 for문 또는 중복 for문이라 함
- 중첩 for문의 실행 횟수: 바깥 for문 반복 횟수 × 안쪽 for문 반복 횟수

```
for i in range(3) :
    for j in range(2) :
        print(i, j, “안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
```

```
print(“00안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n”)
print(“01안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n”)
print(“10안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n”)
print(“11안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n”)
print(“20안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n”)
print(“21안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n”)
```

# 중첩 for문

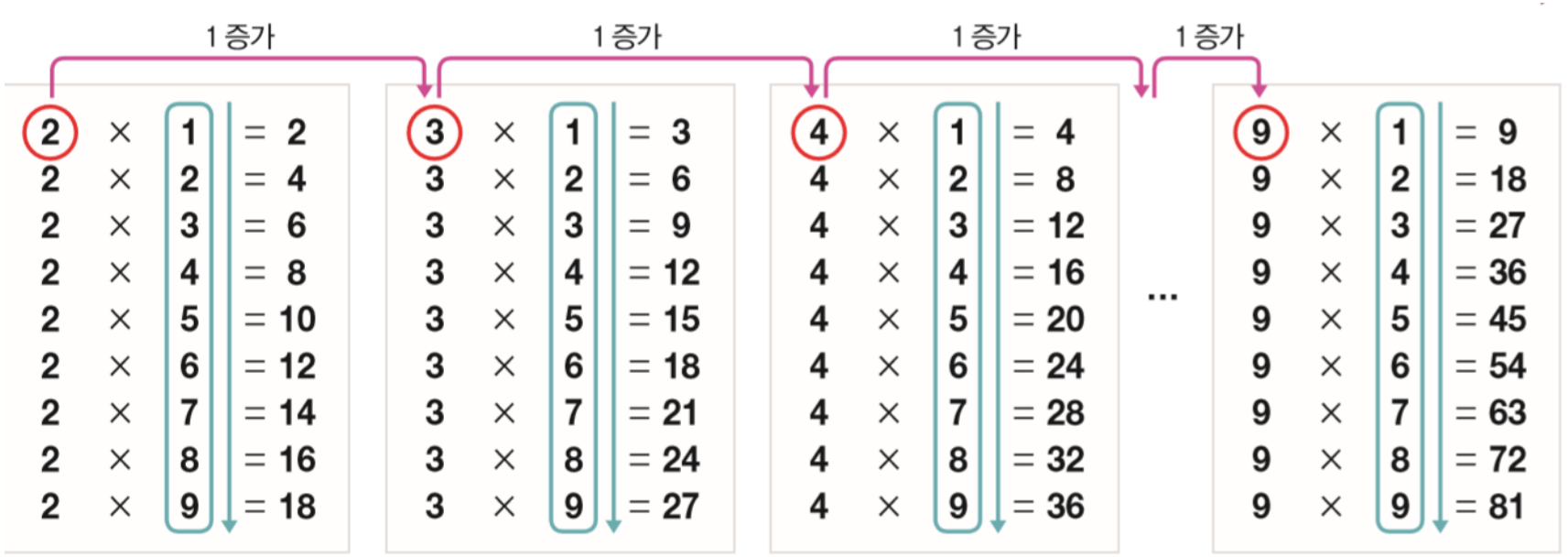
- 중첩 for문의 처리되는 순서
  - 외부 for문 1회: i에 0을 대입
    - 내부 for문 1회: j에 0을 대입한 후에 print() 수행
    - 내부 for문 2회: j에 1을 대입한 후에 print() 수행
  - 외부 for문 2회: i에 1을 대입
    - 내부 for문 1회: j에 0을 대입한 후에 print() 수행
    - 내부 for문 2회: j에 1을 대입한 후에 print() 수행
  - 외부 for문 3회: i에 2를 대입
    - 내부 for문 1회: j에 0을 대입한 후에 print() 수행
    - 내부 for문 2회: j에 1을 대입한 후에 print() 수행



# 중첩 for문

- 이중 for문
  - 구구단

```
for i in range(2,10):
    for j in range(1,10):
        
```



# 오늘 배울 내용

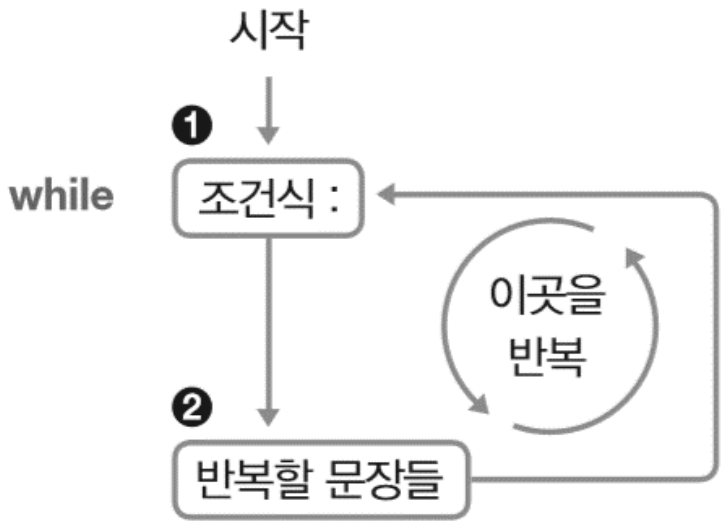
- 반복문
  - 주로 조건문 (if문)과 함께 많이 사용

  - for
    - 반복하는 횟수를 알 수 있을 때
    - 이중 for문

  - while
    - 반복하는 횟수를 알 수 없을 때
    - 무한 반복 및 break

# while문

- while 문은 반복 횟수를 결정하기 보다는 조건식이 참일 때 반복하는 방식




```
for i in range(0, 3, 1) :
    print("%d : 안녕하세요? for 문을 공부 중입니다. ^^" % i)
```

=

```
i = 0
while i < 3 :
    print("%d : 안녕하세요? while 문을 공부 중입니다. ^^" % i)
    i = i + 1
```

출력 결과

```
0 : 안녕하세요? while 문을 공부 중입니다. ^^
1 : 안녕하세요? while 문을 공부 중입니다. ^^
2 : 안녕하세요? while 문을 공부 중입니다. ^^
```

# while문

- for 문 → while 문

```
res = 0

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

print("1에서 10까지의 합 = \n", res)
```

```
i = 1
res = 0

while i < 11 :
    res = res + i
    i = i + 1

print("1에서 10까지의 합 = \n", res)
```

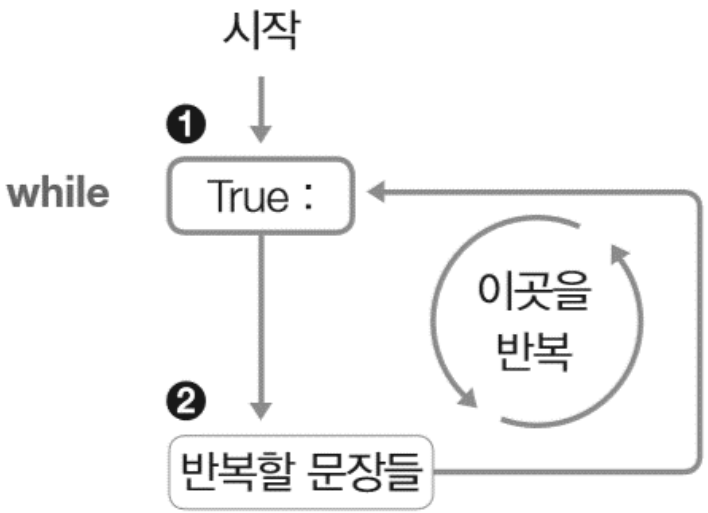
## 예시

while문을 이용하여 실행 결과와 같은 형태로 출력해보자.

```
* * * * * * * * * * *

* * * * * * * * * * *

* * * * * * * * * * *

* * * * * * * * * * *

* * * * * * * * * * *
```

# while문

- 무한 루프를 하는 while문
  - 무한 루프 적용 : ‘while 조건식 : ’에 들어가는 조건식을 True로 지정
  - 사용자가 ctrl + c 누를 때 까지 코드가 동작

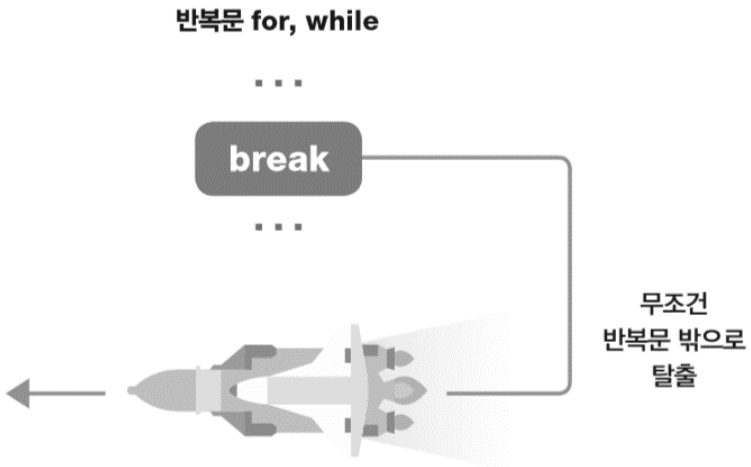



```
i = 1
res = 0

while i < 11 :
    res = res + i
    i = i + 1

    print(i, res)
```

```
while True :
    print(“ㅋㅋ”)
```

# break문

- 반복문을 탈출시키는 break 문
  - 계속되는 반복을 논리 적으로 빠져나가는 방법
  - 반복문 안에서 break문을 만나면 무조건 반복문을 빠져나옴




```
for i in range(3) :
    print("안녕하세요. 어서 오세요. QR 체크 부탁드립니다.\n")
    break
```

# break문

- for문

```
res = 0

for i in range(1, 11, 1) :
    res = res + i
    if res > 40 :
        break

print("1에서 10까지의 합에서 최초로 40을 넘는 숫자는 = %d\n", i)
```

- while문

```
i = 1
res = 0

while i < 11 :
    res = res + I
    if res > 40 : break
    i = i + 1

print("1에서 10까지의 합에서 최초로 40을 넘는 숫자는 = %d\n", i)
```

## continue문

- 반복문으로 다시 돌아가게 하는 continue 문
  - break문은 반복문을 빠져나가지만, 이와 달리 continue문은 남은 부분을 모두 건너뛰고, 반복문의 처음으로 돌아 감


```
res = 0

for i in range(1, 11, 1) :
    if i % 3 == 0 :
        continue
    res = res + i

print("1부터 3의 배수는 제외한 10까지의 합은 %d 입니다.\n" , res)
```

## 예시

1부터 100까지 숫자를 모두 더한 값을 출력하세요.

```
5050
```

## 예시

30과 75의 최대 공약수를 출력해보세요.

```
15
```

# 감사합니다

kimtwan21@dongduk.ac.kr

김 태 완