

Ch8. break, continue문

- **while에서 break로 반복문 끝내기**
 - while 무한 루프에서 숫자를 증가시키다가 변수 i가 100일 때 반복문을 끝내도록 만들어보자
 - IDLE의 소스 코드 편집 창에 입력한 뒤 실행해보자

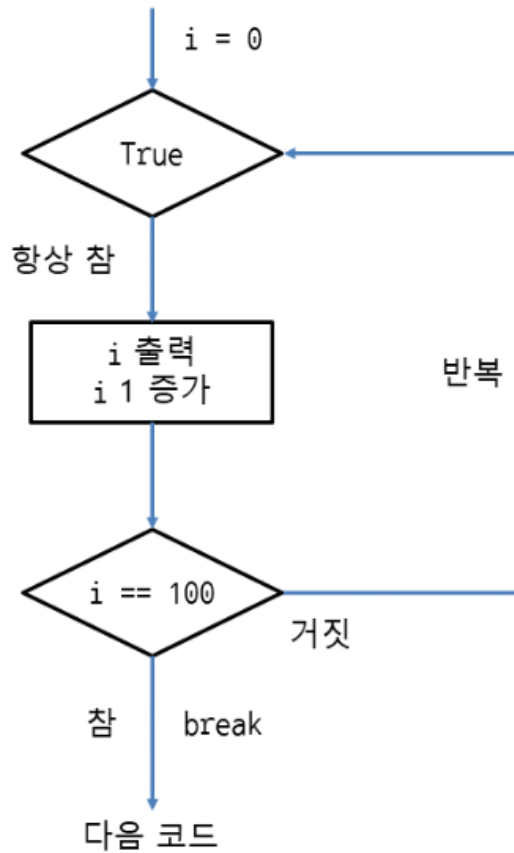
break_while.py

```
i = 0
while True:    # 무한 루프
    print(i)
    i += 1      # i를 1씩 증가시킴
    if i == 100: # i가 100일 때
        break   # 반복문을 끝냄. while의 제어흐름을 벗어남
```

실행 결과

```
0
1
2
... (생략)
97
98
99
```

- while에서 break로 반복문 끝내기



- for에서 break로 반복문 끝내기
 - while뿐만 아니라 for에서도 break의 동작은 같음

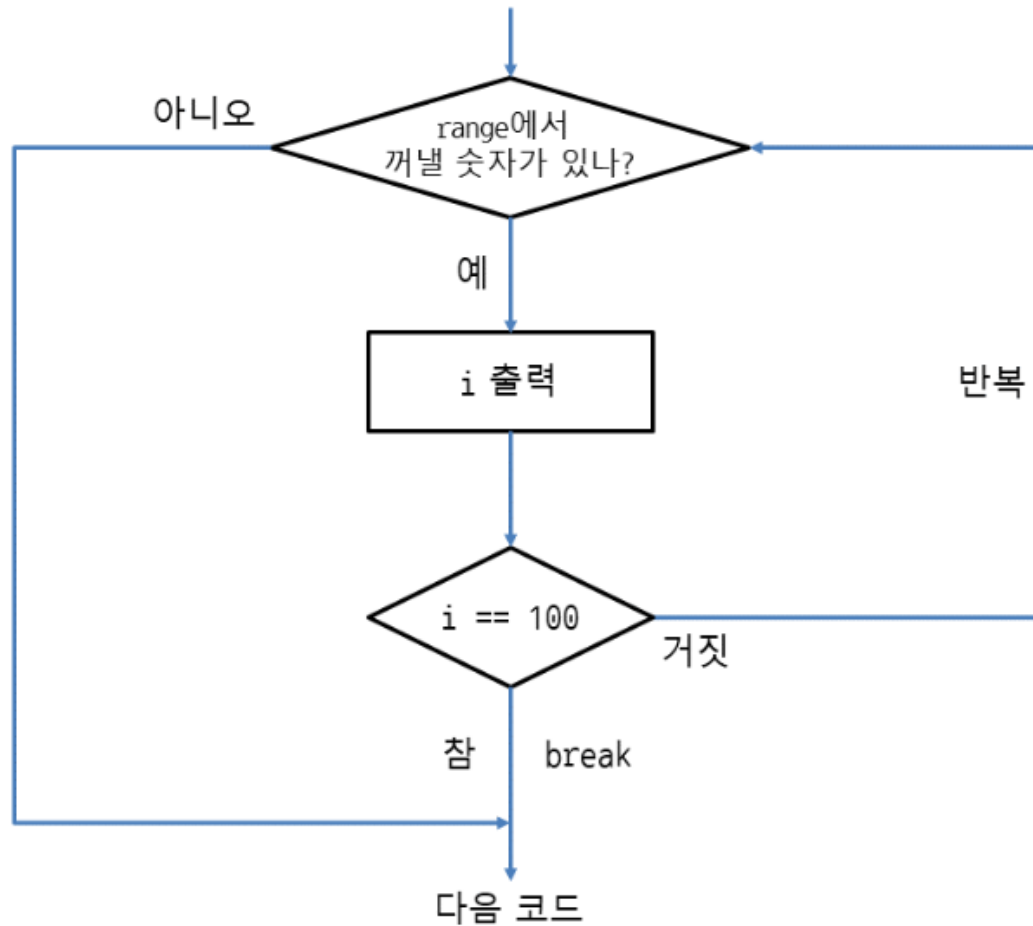
break_for.py

```
for i in range(10000):    # 0부터 9999까지 반복
    print(i)
    if i == 100:         # i가 100일 때
        break           # 반복문을 끝냄. for의 제어흐름을 벗어남
```

실행 결과

```
0
1
2
... (생략)
98
99
100
```

- for에서 break로 반복문 끝내기



- **continue로 코드 실행 건너뛰기**
 - for에서 continue로 코드 실행 건너뛰기
 - for로 0부터 99까지 반복하면서 홀수만 출력함

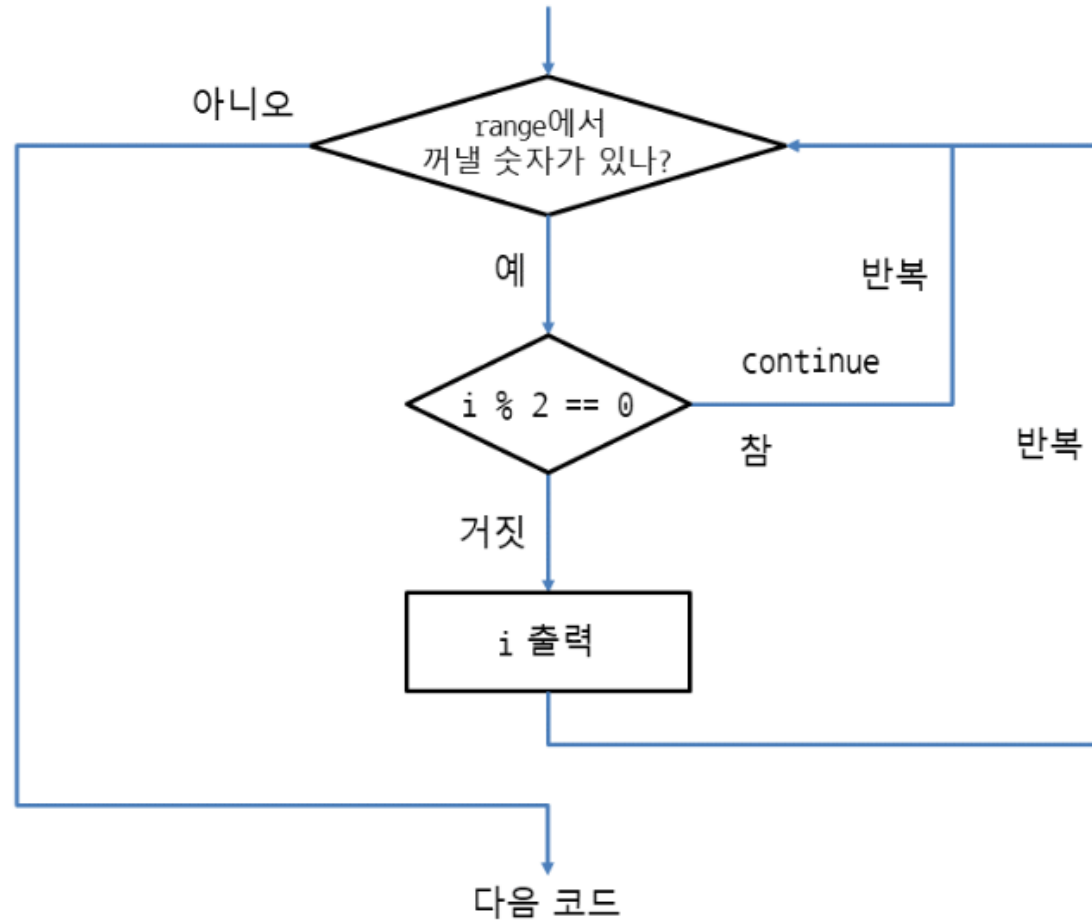
continue_for.py

```
for i in range(100):      # 0부터 99까지 증가하면서 100번 반복
    if i % 2 == 0:        # i를 2로 나누었을 때 나머지가 0면 짝수
        continue         # 아래 코드를 실행하지 않고 건너뛴다
    print(i)
```

실행 결과

```
1
3
5
... (생략)
95
97
99
```

- continue로 코드 실행 건너뛰기



- while 반복문에서 continue로 코드 실행 건너뛰기
 - for뿐만 아니라 while에서도 continue의 동작은 같음

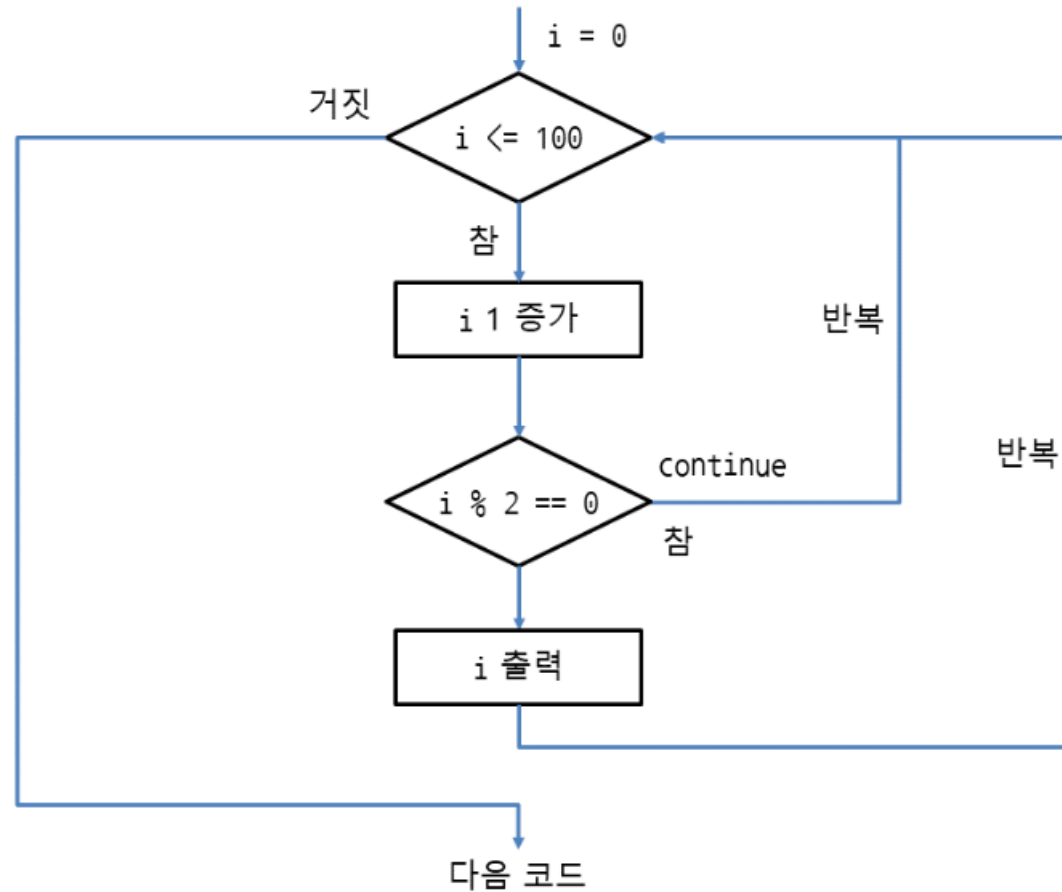
continue_while.py

```
i = 0
while i < 100:      # i가 100보다 작을 때 반복. 0부터 99까지 증가하면서 100번 반복
    i += 1          # i를 1씩 증가시킴
    if i % 2 == 0:  # i를 2로 나누었을 때 나머지가 0이면 짝수
        continue   # 아래 코드를 실행하지 않고 건너뛴
    print(i)
```

실행 결과

```
1
3
5
... (생략)
95
97
99
```


- continue로 코드 실행 건너뛰기



- 입력한 횟수대로 반복하기
 - 입력한 횟수대로 반복을 해보자
 - IDLE의 소스 코드 편집 창에 입력하자

break_input.py

```
count = int(input('반복할 횟수를 입력하세요: '))

i = 0
while True:    # 무한 루프
    print(i)
    i += 1
    if i == count:    # i가 입력받은 값과 같을 때
        break        # 반복문을 끝냄
```

- 소스 코드를 실행하면 '반복할 횟수를 입력하세요:'가 출력
- 3을 입력하고 엔터 키를 누름

실행 결과

```
반복할 횟수를 입력하세요: 3 (입력)
0
1
2
```

- 입력한 숫자까지 홀수 출력하기
 - 입력한 숫자까지 해당하는 홀수를 출력해보자

continue_input.py

```
count = int(input('반복할 횟수를 입력하세요: '))

for i in range(count + 1):    # 0부터 증가하면서 count까지 반복(count + 1)
    if i % 2 == 0:           # i를 2로 나누었을 때 나머지가 0이면 짝수
        continue            # 아래 코드를 실행하지 않고 건너뛴다
    print(i)
```

- 소스 코드를 실행하면 '반복할 횟수를 입력하세요:'가 출력
- 9를 입력하고 엔터 키를 누르자

반복할 횟수를 입력하세요: 9 (입력)

1
3
5
7
9

- 계단식으로 별 출력하기

- for 반복문과 if 조건문을 사용하여 계단식으로 별(*)을 출력해보자

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

- 계단식으로 별 출력하기

```
for i in range(횟수):      # 바깥쪽 루프
```

```
    for j in range(횟수):  # 안쪽 루프
```

```
        가로 처리 코드
```

```
    세로 처리 코드
```

↑
가로 처리 코드

↑
세로 처리 코드

- 중첩 루프 사용하기

- i를 사용하는 바깥쪽 루프는 세로 방향을 처리하고, j를 사용하는 안쪽 루프는 가로 방향을 처리

multiple_loop.py

```
for i in range(5):          # 5번 반복. 바깥쪽 루프는 세로 방향
    for j in range(5):      # 5번 반복. 안쪽 루프는 가로 방향
        print('j:', j, sep='', end=' ')  # j값 출력. end에 ' '를 지정하여 줄바꿈 대신 한 칸 띄움
    print('i:', i, '\\n', sep='')        # i값 출력, 개행 문자 모양도 출력
                                          # 가로 방향으로 숫자를 모두 출력한 뒤 다음 줄로 넘어감
                                          # (print는 기본적으로 출력 후 다음 줄로 넘어감)
```

안쪽 루프

j:0 j:1 j:2 j:3 j:4 i:0\\n
j:0 j:1 j:2 j:3 j:4 i:1\\n
j:0 j:1 j:2 j:3 j:4 i:2\\n
j:0 j:1 j:2 j:3 j:4 i:3\\n
j:0 j:1 j:2 j:3 j:4 i:4\\n

← 바깥쪽 루프

- 사각형으로 별 출력하기

- 중첩된 반복문을 사용하여 5x5 사각형 형태로 별을 그려보자

star_square.py

```
for i in range(5):          # 5번 반복. 바깥쪽 루프는 세로 방향
    for j in range(5):      # 5번 반복. 안쪽 루프는 가로 방향
        print('*', end='')  # 별 출력. end에 ''를 지정하여 줄바꿈을 하지 않음
    print()                # 가로 방향으로 별을 다 그린 뒤 다음 줄로 넘어감
                            # (print는 출력 후 기본적으로 다음 줄로 넘어감)
```

실행 결과

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

- 사각형으로 별 출력하기

- 안쪽의 `for j in range(5):` 반복문에서 `print`로 별을 그리면 다음과 같이 가로 방향으로 별이 나열됨

```
for j in range(5):  
    print('*', end='')
```

j 0 1 2 3 4

for i in range(5):

i	
0	***** \n
1	***** \n
2	***** \n
3	***** \n
4	***** \n

↑
print()



- 사각형 모양 바꾸기

- for 반복문의 조건식을 수정하여 사각형의 모양을 바꿔보자

star_rectangle.py

```
for i in range(3):          # 3번 반복. 세로 방향
    for j in range(7):      # 7번 반복. 가로 방향
        print('*', end='')  # 별 출력. end에 ''를 지정하여 줄바꿈을 하지 않음
    print()                 # 가로 방향으로 별을 다 그린 뒤 다음 줄로 넘어감
                            # (print는 출력 후 기본적으로 다음 줄로 넘어감)
```

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

- 안쪽 반복문은 7번 반복하고, 바깥쪽 반복문은 3번 반복하도록 수정함
- 7×3 크기의 사각형을 그림
- 반복문 조건식의 숫자를 바꿔서 다양한 크기로 출력해보자

- 계단식으로 별 출력하기

- 계단식으로 별이 하나씩 증가하게 출력하려면 어떻게 해야 할까?
- 사각형으로 별을 출력할 때는 사각형에 꽉 채워서 별을 출력했지만, 계단식으로 출력할 때는 별을 출력하지 않는 부분이 있으므로 if 조건문으로 print를 제어

star_triangle.py

```
for i in range(5):      # 0부터 4까지 5번 반복. 세로 방향
    for j in range(5):  # 0부터 4까지 5번 반복. 가로 방향
        if j <= i:      # 세로 방향 변수 i만큼
            print('*', end='') # 별 출력. end에 ''를 지정하여 줄바꿈을 하지 않음
    print()             # 가로 방향으로 별을 다 그린 뒤 다음 줄로 넘어감
                        # (print는 출력 후 기본적으로 다음 줄로 넘어감)
```

실행 결과

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

- 계단식으로 별 출력하기

- 첫 번째 줄에는 별이 한 개, 두 번째 줄에는 별이 두 개, 세 번째 줄에는 별이 세 개, 네 번째 줄에는 별이 네 개, 다섯 번째 줄에는 별이 다섯 개임
- 별의 개수는 세로 방향인 줄의 위치에 비례하므로 조건식도 세로 방향 변수 i 에 맞추어 작성

```
if j <= i:           # 세로 방향 변수 i만큼  
    print('*', end='') # 별 출력. end에 ''를 지정하여 줄바꿈을 하지 않음
```

- if 조건문을 $if\ j\ <= i:$ 와 같이 작성하여 가로 방향(j)으로 별을 출력할 때는 세로 방향(i)의 변수만큼 출력하도록 만들었음
- i 가 증가하면 i 만큼 별이 늘어나게 됨
- i 가 0일 때 j 는 0부터 4까지 반복하지만, $j\ <= i$ 를 만족하는 경우는 j 가 0일 때 뿐이므로 별이 1 번만 출력
- i 가 1일 때는 $j\ <= i$ 를 만족하는 경우는 j 가 0, 1일 때이므로 별이 2번 출력
- 이런 식으로 반복하면서 마지막 5번째에서 별이 5번 출력

- 대각선으로 별 출력하기

- 출력 부분을 수정하여 별을 대각선으로 그려보자

star_diagonal.py

```
for i in range(5):      # 0부터 4까지 5번 반복. 세로 방향
    for j in range(5):  # 0부터 4까지 5번 반복. 가로 방향
        if j == i:      # 세로 방향 변수와 같을 때
            print('*', end='')  # 별 출력. end에 ''를 지정하여 줄바꿈을 하지 않음
    print()             # 가로 방향으로 별을 다 그린 뒤 다음 줄로 넘어감
                        # (print는 출력 후 기본적으로 다음 줄로 넘어감)
```

- if j == i:처럼 가로 방향 변수와 세로 방향 변수가 같을 때만 별을 출력하면
왼쪽 위에서 오른쪽 아래 방향으로 별이 출력될 것임

실행 결과

```
*
*
*
*
*
```

- 대각선으로 별 출력하기

- 별을 출력해보면 그냥 세로로 출력됨
- 문자 출력을 하지 않으면, 다음 문자는 현재 위치에 출력되므로 왼쪽으로 몰림
- 별을 출력하지 않는 부분에서는 print로 공백을 출력해주어야 별이 원하는 위치에 오게 됨

star_diagonal.py

```
for i in range(5):      # 0부터 4까지 5번 반복. 세로 방향
    for j in range(5):  # 0부터 4까지 5번 반복. 가로 방향
        if j == i:      # 세로 방향 변수와 같을 때
            print('*', end='') # 별 출력. end에 ''를 지정하여 줄바꿈을 하지 않음
        else:           # 세로 방향 변수와 다를 때
            print(' ', end='') # 공백 출력. end에 ''를 지정하여 줄바꿈을 하지 않음
    print()             # 가로 방향으로 별을 다 그린 뒤 다음 줄로 넘어감
                        # (print는 출력 후 기본적으로 다음 줄로 넘어감)
```

실행 결과

```
*
 *
  *
   *
    *
```

- 대각선으로 별 출력하기

- 별이 왼쪽 위에서 오른쪽 아래 방향인 대각선 모양으로 그려짐
- 가로 방향과 세로 방향의 변수 값이 같을 때 별을 출력하고,
같지 않을 때는 공백을 출력
- 바깥쪽 루프가 세로 방향, 안쪽 루프가 가로 방향을 처리한다는 점만 기억하면 됨