

Ask Django

흐름 제어 (조건문, 반복문)

조건문, if문

if 조건:

위 조건에 부합될 때 실행할 코드 블록

elif 두번째조건:

두번째조건에 부합될 때 실행할 코드 블록

elif 세번째조건:

세번째조건에 부합될 때 실행할 코드 블록

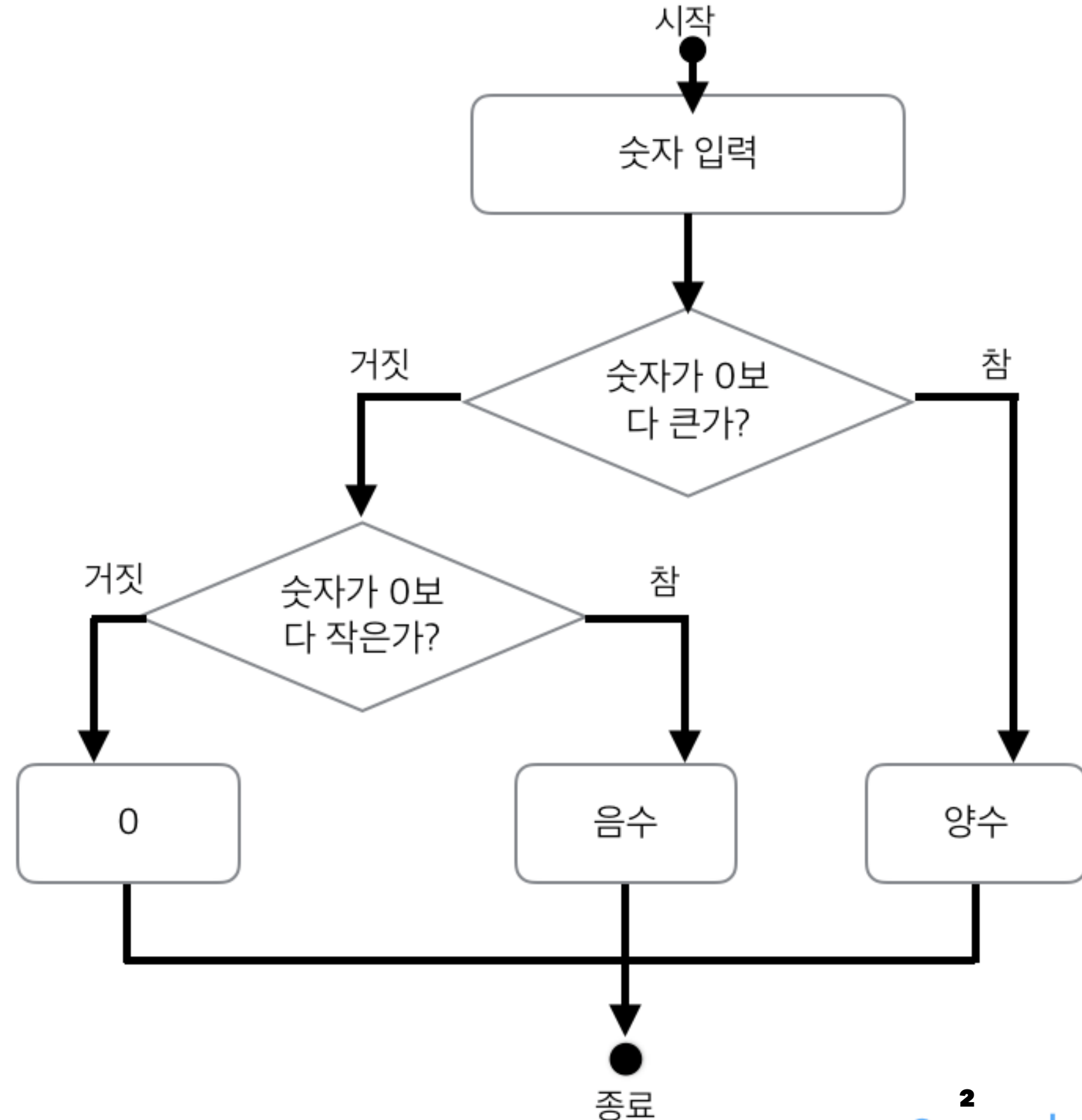
elif 네번째조건:

네번째조건에 부합될 때 실행할 코드 블록

else:

위 조건에 모두 맞지 않을 때 실행할 코드 블록

한 if statement 에서 if / else 는 1회만 쓸 수 있으나, elif 는 원하는 조건의 수만큼 쓸 수 있습니다.



어떤 차이가 있을까요?

코드 A

```
number = int(input('Enter number : '))

if number > 0:
    print('%d is positive number.' % number)
elif number < 0:
    print('%d is negative number.' % number)
else:
    print('%d is zero.' % number)
```

코드 B

```
number = int(input('Enter number : '))

if number > 0:
    print('%d is positive number.' % number)
if number < 0:
    print('%d is negative number.' % number)
else:
    print('%d is zero.' % number)
```

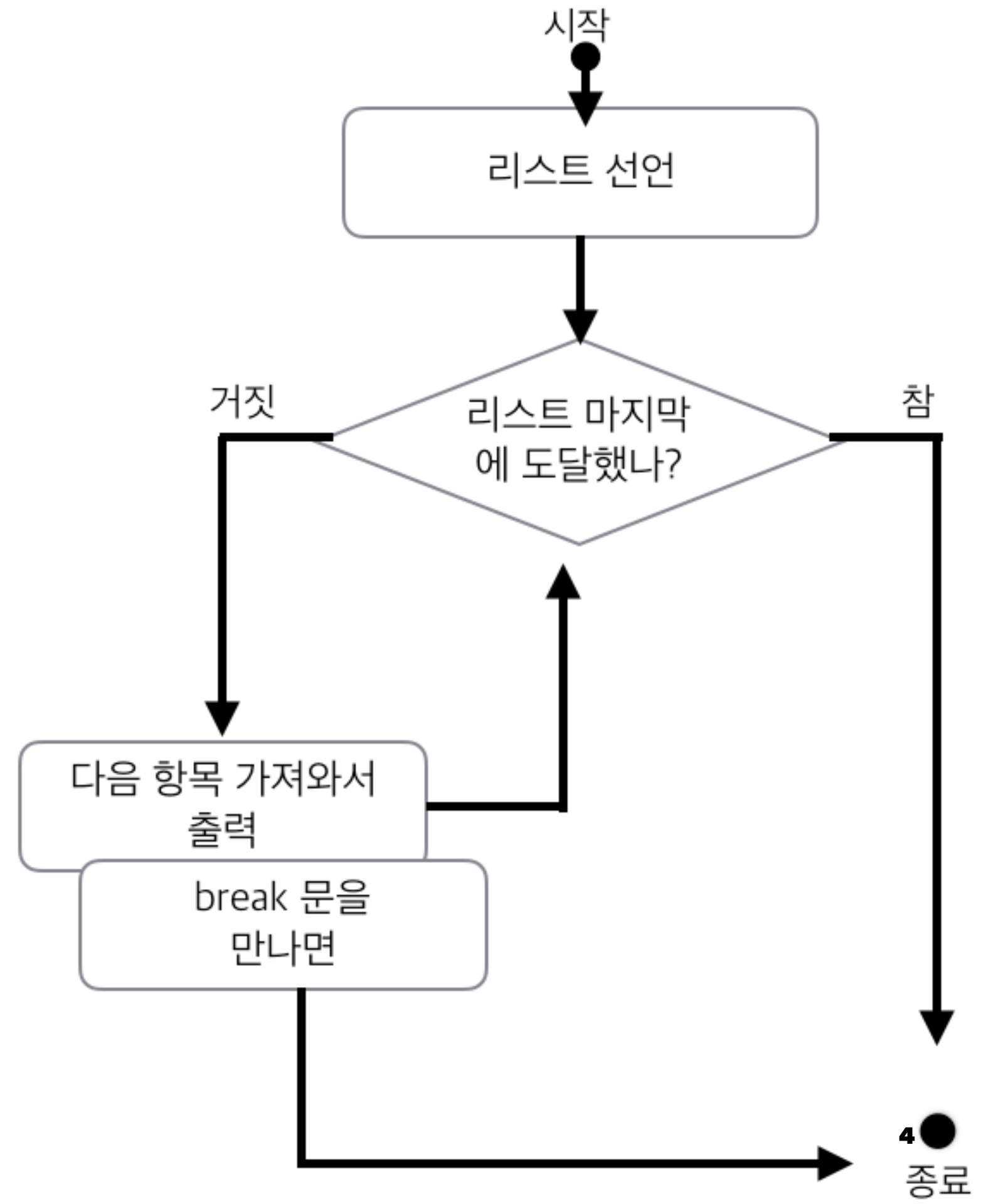
반복문 : **for**

Iterable Object로부터 가져올 값들을 모두
가져올 때까지 반복

for 변수 **in** 리스트:
 매 항목마다 수행할 코드 블록

for 내에서 **break** 문을 만나면 해당 반복문 종
료

```
for i in range(20):  
    print(i)  
    if i > 10:  
        break
```



중첩 반복문

반복문 내에서의 break는 근접한 반복문만 종료시킵니다.

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

```
2 1  
3 1  
4 1  
5 1  
6 1  
7 1  
8 1  
9 1  
>>>
```

중첩 반복문을 한 번에 종료시키기

예외발생이나 함수 내에서 return문을 통해 중첩 반복문을 한 번에 종료 가능

```
>>> def gugu():  
    for i in range(2, 10):  
        for j in range(1, 10):  
            print(i, j)  
        return None
```

```
>>> gugu()
```

```
2 1
```

```
>>>
```

내장함수 **range** #doc

- `range(stop)` : 0부터 stop 미만의 범위에서 1 씩 증가시킨 값으로 리스트를 구성
 - 문법적으로는 리스트가 아니라, 순회가능한 (Iterable) 객체
- `range(start, stop[, step])` : start 값 이상, stop 값 미만의 범위에서 step 씩 증가시킨 값으로 리스트를 구성

```
>>> range(10)
range(0, 10)
```

```
>>> list(range(10))                                     # 순회가능한 객체로부터 list 생성
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
>>> tuple(range(10))                                    # 순회가능한 객체로부터 tuple 생성
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
```

반복문 : **while**

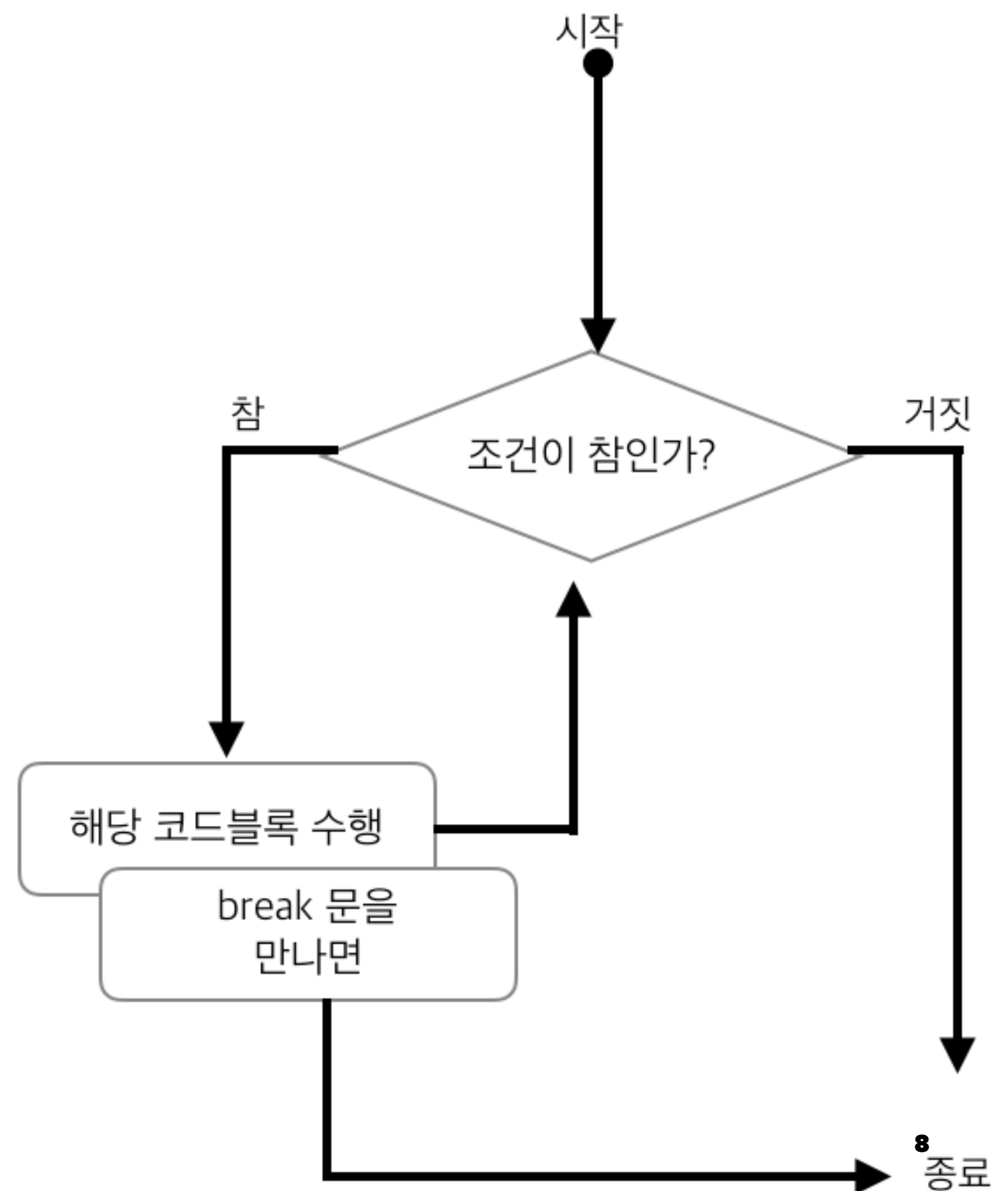
조건이 만족하는 동안에 반복문 수행

while 조건:

매 항목마다 수행할 코드 블록

while 내에서 break 문을 만나면 해당 반복문 종료

```
i = 0
while i < 20:
    print(i)
    if i > 10:
        break
    i += 1
```



무한루프

<반복문 조건> 이 항상 True 일 때

```
i = 10
while i < 13:
    print(i)
    i -= 1
```


for 에서는 itertools.count 함수를 통해 가능

```
from itertools import count
```

```
for i in count(1):
    print(i)
```

연습문제

2단, 4단, 6단, 8단 구구단을 출력하는 코드를 작성하세요.



*Life is short,
use Python3/Django.*