

무한 루프

무한 루프(infinite loop)란?

- 순환 작업을 끝없이 반복하는 것을 말한다.

일반적인 무한 루프란

컴퓨터에서 프로그램이 끝없이 동작하는 것으로,
주로 순환문의 논리적 오류로 종료 조건을 충족할 수 없거나
종료 조건이 없는 경우에 발생한다.

while문 무한 루프를 실행해보자.

무한 루프를 빠져나오려면 **Ctrl + C** 를 눌러야 한다.

In []:

```
# --- 주피터에서는 실행하지 말 것
while True:
    print('안녕 파이썬')
```

따라해보기

다음 **while**문은

처음에는 일반적인 무한 루프처럼 계속 실행되지만

10회 후에 멈추고 **while**문을 빠져나온다.

In []:

```
status = True
stamina = 10
while status:
    print('운동을 시킵시다. 체력이 1 감소합니다.')
    stamina -= 1
    if stamina == 0:
        status = False
else:
    print('체력이 고갈되었습니다. 더 이상 운동을 할 수 없습니다.')
```

순환문의 흐름을 제어하는 **break** 명령어를 사용하면

앞의 코드를 좀 더 간단하게 작성할 수 있다.

In []:

```
stamina = 10
while True:
    print('운동을 시킵시다. 체력이 1 감소합니다.')
    stamina -= 1
    if not stamina:
        print('체력이 고갈되었습니다. 더 이상 운동을 할 수 없습니다.')
        break
```

break와 continue 명령어

break와 **continue**는 순환문의 흐름을 제어하는데 사용한다.

break

break문을 만나면

이후의 모든 코드를 건너뛰고

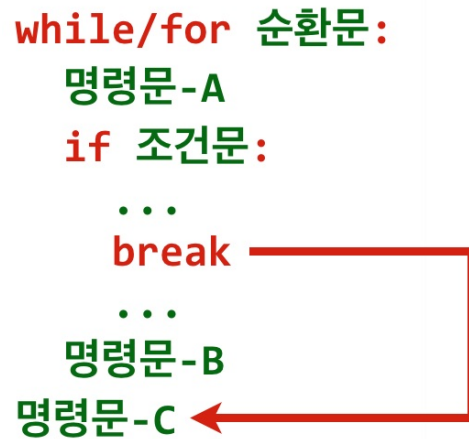
현재 **break**문이 속한 순환문을 종료하고 빠져나간다.

break문이

중첩 순환문(순환문 안의 또 다른 순환문) 안에 있으면

break문이 속한 **가장 안쪽 순환문**을 종료하고 빠져나와서

통제권을 상위 순환문으로 넘긴다.



따라해보기

break문을 이용해서 **while**문을 빠져나가보자.

In []:

```
L = list(range(10))
print(L)

i = 0
while True:
    print(L[i])
    i += 1
    if i == 5:    # i가 5면 while 순환문을 빠져나간다
        break
else:
    print('종료') # 절대로 실행되지 않는다.
```

In []:

```
# --- 특정 값을 입력하면 무한 루프를 빠져나간다.
while True:
    answer = input("아무거나 입력하시오 (마치려면 -1을 입력): ")
    if answer == '-1':    # answer 가 '-1'이면 while문을 빠져나간다.
        print('종료합니다...')
        break
    print(answer)
```

In []:

```
# --- 그냥 <enter> 키를 누르면 종료한다.
while True:
    answer = input('무언가를 입력하세요 (마치려면 <ENTER> 키를 누르세요): ')
    if not answer:
        print('종료합니다...')
        break
    print(answer)
```

break 명령어를 사용하여 무한 루프에 빠지지 않고

정상적으로 작동할 수 있도록 **while**문 흐름 제어해보자.

다음 예는 **0~10** 사이의 정수를 무작위로 생성한 후 정수가 **9** 이하면 **while**문을 계속해서 순환하고, **9**보다 크면 **while**문을 빠져나간다.

In []:

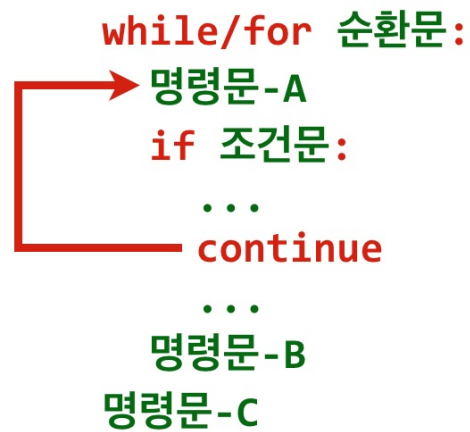
```
# --- while문 흐름 제어(while_break_continue.py)
import random
i = 0
while True:
    i = random.randint(0, 10)
    if i > 9:
        print('{}: 무한 루프를 빠져나갑니다.'.format(i))
        break      # 여기 break 명령어가 없으면 무한루프가 된다.
    else:
        print('{}: 무한 루프에 빠져있습니다.'.format(i))
```

continue

continue문을 만나면

continue 이후의 모든 코드를 건너뛰고

순환문의 첫 명령문으로 되돌아가 다음 반복 처리를 실행한다.



따라해보기

continue문을 이용해서 **1~100** 사이 정수 중 홀수만 더해보자.

In []:

```
total = 0
for i in range(1, 101):
    if i % 2 == 0: # 짝수면 건너뛰다.
        continue
    total += i
else:
    print(total)
```

In []:

```
# --- 홀수만 더해서 합하고, 짝수면 건너뛰고, 음수면 즉시 for문을 종료한다.
data = [1, 4, 7, 3, 8, 6, 9, 5, 0, 2, 7, -1, 22, 31]
total = 0
for i in data:
    if i < 0:
        print('음수(i = {})라 break문을 실행합니다.'.format(i))
        break
    elif i % 2 == 0:
        print('짝수(i = {})라 continue문을 실행합니다.'.format(i))
        continue
    else:
        print(i)
        total += i
print('-' * 35)
print('홀수의 합:', total)
```