

# while 명령문

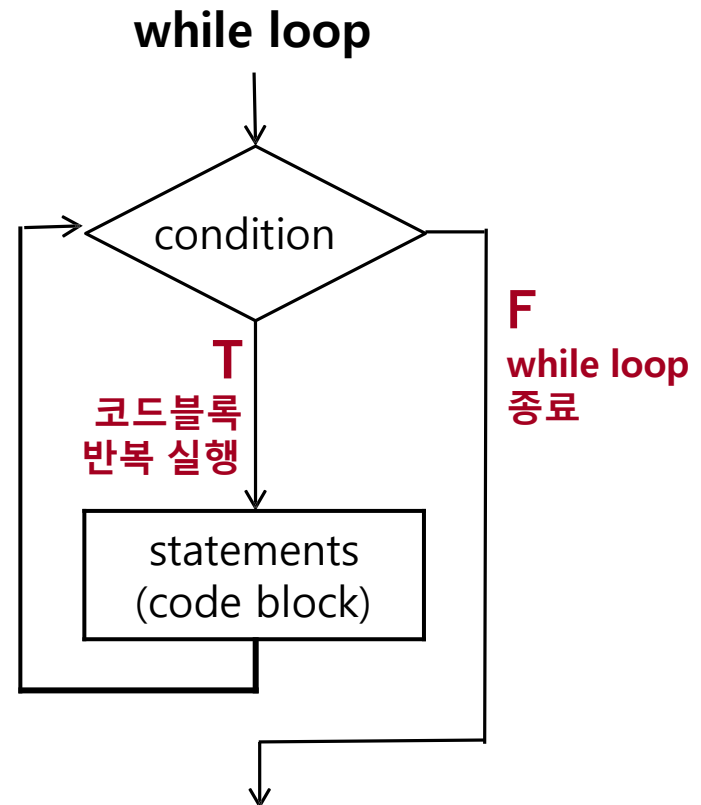
- while 문은 조건을 정해놓고 반복을 하는 구조
- while statement syntax

**while condition :** -----> 반드시 있어야 함

들여  
쓰기

any statements

**while 문:** condition의 결과가 True이면  
코드 블록 내의 명령어들을 반복 실행



## while 명령문

- while을 사용한  $1 + 2 + \dots + N$  계산 프로그램

```
N = 10
sum = 0                # sum을 0으로 초기화
i = 1                  # range() 대신, counting 위한 변수 i를 사용
while i <= N :
    sum = sum + i       # sum을 누적(accumulation)
    i = i+1             # counting 변수 1 증가
print("The sum is", sum) # 출력 : The sum is 55
```

- 1부터 100사이의 수에서 3의 배수인 수들 합을 계산하는 프로그램

```
sum = 0
number = 1
while number <= 100 :
    if number %3 == 0 :
        sum += number
    number = number + 1
print("1부터 100 사이의 모든 3의 배수의 합은 %d입니다." % sum)
# 출력 : 1부터 100 사이의 모든 3의 배수의 합은 1683입니다.
```

# while 명령문

- 정수의 각 자리수의 합을 계산하는 프로그램(1234 경우,  $1+2+3+4$ 를 계산하는 것)

```
number = 1234
sum = 0
while number > 0 : dig
    it = number % 10
    um = sum + digit
    number = number // 10
    print("자리수의 합은 %d입니다." % su
m) # 출력 : 자리수의 합은 10입니다.
```

while 반복문 코드 블록 안에서

1. digit = 4, sum = 0 + 4, number = 123
2. digit = 3, sum = 4 + 3, number = 12
3. digit = 2, sum = 7 + 2, number = 1
4. digit = 1, sum = 9 + 1, number = 0
5. number > 0 이 False이기 때문에 while 문 종료
6. print(sum) 실행

# while 명령문

- 숫자 맞추기 게임

random() : 0.0 이상 1.0 미만 실수를 반환  
random() + 1.0 : 1.0 이상 2.0 미만 실수를 반환  
(난수의 범위를 설정할 수 있음)

```
import random

tries = 0
number = random.randint(1, 100)
print("1부터 100 사이의 숫자 맞추기")
while tries < 10:
    guess = int(input("숫자 입력: "))
    tries = tries + 1
    if guess < number:
        print("더 큰 수를 입력하세요!")
    elif guess > number:
        print("더 작은 수를 입력하세요!")
    else:
        break
if guess == number:
    print("축하합니다. 시도횟수=", tries)
else:
    print("정답은 ", number)
```

randint(n, m) : n 이상 m 이하 정수를 반환

출력

1부터 100 사이의 숫자 맞추기  
숫자 입력: 25  
더 큰 수를 입력하세요!  
숫자 입력: 50  
더 작은 수를 입력하세요!  
숫자 입력: 33  
더 큰 수를 입력하세요!  
숫자 입력: 39  
더 큰 수를 입력하세요!  
숫자 입력: 45  
더 큰 수를 입력하세요!  
숫자 입력: 48  
더 작은 수를 입력하세요!  
숫자 입력: 47  
더 작은 수를 입력하세요!  
숫자 입력: 46  
축하합니다. 시도횟수= 8

# while 명령문

- 여러 개의 정수를 입력 받아서 양수와 음수 개수를 출력하는 프로그램

```
countP= 0                #양수 개수
countN= 0                #음수 개수
sumNums = 0              #전체 합

data = list(int(x) for x in input("정수들을 입력 : \n").split())
dataNum = len(data)      #리스트 내포형 for를 사용한 입력
                        #입력한 숫자가 없을 경우를 위한 변수

If dataNum == 0:          #입력한 숫자가 없는 경우
    print( " 입력한 숫자가 없습니다 " )
Else :
    while (dataNum > 0):  #dataNum을 인덱스로 반복, dataNum 개수만큼
        if data[dataNum-1] > 0: #양수일 경우
            countP += 1
        elif data[dataNum-1] < 0: #음수일 경우
            countN += 1

        sumNums += data[dataNum-1] #양수든 음수든 합계 계산
        dataNum -= 1 #인덱스 카운트

Print( " 양수: " , countP, " 개, " , " 음수: " , countN, " 개, " , " 합계: " , sumNums)
```

- 출력 결과

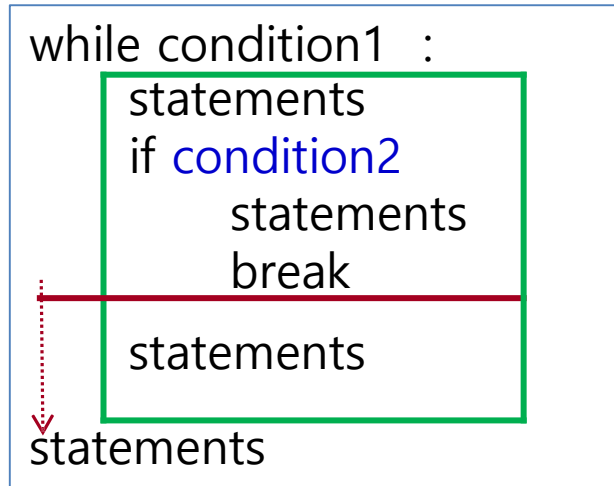
```
정 수 들 을 입 력 :
23 -2 5 89 -43 0 5 0
양 수 : 4 개 , 음 수 : 2 개 , 합 계 : 77
```

```
정 수 들 을 입 력 :

입 력 한 숫 자 가 없 습 니 다
양 수 : 0 개 , 음 수 : 0 개 , 합 계 : 0
```

# break 명령문

- break 명령어의 기능은 for 반복문과 같음
- 자연수 N이 소수(a prime number)인지 판단하는 코드



```
N = int(input("N(> 1)? "))  
k = 2  
primeChk = True  
while k < N :  
    if N % k == 0 :  
        primeChk = False  
        break  
    k = k + 1  
if primeChk == True :  
    print("prime")  
else :  
    print("not prime")
```

# break 명령문

- 무한 루프와 break

**while(True) :**    **#무한 루프**

- 0이 아닌 임의의 자연수를 횡수에 상관없이 입력 받는 프로그램

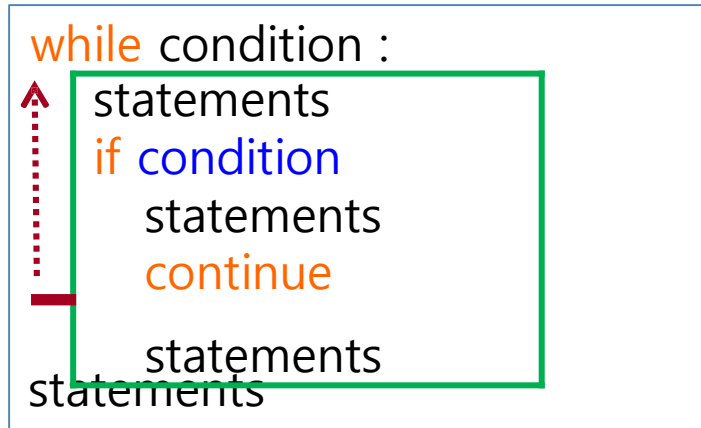
```
while(True) :  
    n = int(input('Enter the number : '))  
    if n==0 :    #0이 입력되면 loop 종료  
        break  
print(n)
```

## **출력**

```
Enter the number : 10  
Enter the number : 20  
Enter the number : -2  
Enter the number : 0  
0
```

# continue 명령문

- continue 명령어의 기능은 for 반복문과 같음





# while 예제(break 사용)

## ● 숫자 맞추기 게임

```
import random

tries = 0
number = random.randint(1, 100)

print("1부터 100 사이의 숫자 맞추기")
while tries < 10:
    guess = int(input( " 숫자 입력: " ))
    tries = tries + 1
    if guess < number:
        print( " 더 큰 수를 입력하세요! " )
    elif guess > number:
        print( " 더 작은 수를 입력하세요! " )
    else:
        break
if guess == number:
    print("축하합니다. 시도횟수=", tries)
else:
    print("정답은 ", number)
```

#random() : 0.0 이상 1.0 미만 실수를 반환  
#random() + 1.0 : 1.0 이상 2.0 미만 실수를 반환 (난수의 범위를 설정할 수 있음)  
#최대 10번까지 시도 가능을 while 사용하여 표현  
#입력된 숫자는 guess에 저장  
#tries 변수(시도횟수 의미) +1  
#입력된 숫자보다 생성된 숫자가 클 경우  
#입력된 숫자가 생성된 숫자 보다 클 경우  
#정답일 경우  
#break 사용하여 while문 종료  
#정답일 경우 시도 횟수 출력  
#실패일 경우 정답 공개

## ● 출력 결과

```
1부 터 100 사 이 의 숫 자 맞 추 기
숫 자 입 력 : 50
더 작 은 수 를 입 력 하 세 요 !
숫 자 입 력 : 25
더 큰 수 를 입 력 하 세 요 !
숫 자 입 력 : 45
더 작 은 수 를 입 력 하 세 요 !
숫 자 입 력 : 40
더 큰 수 를 입 력 하 세 요 !
숫 자 입 력 : 43
더 작 은 수 를 입 력 하 세 요 !
숫 자 입 력 : 42
더 작 은 수 를 입 력 하 세 요 !
숫 자 입 력 : 41
축 하 합 니 다 . 시 도 횟 수 = 7
```

실습