



p y t h o n





## 12. 반복해서 돌기-while



이번시간에 학습할 내용은..

for 문처럼 특정한 프로그램영역을  
반복해서 수행하는 while문이 있습니다.  
while문에 대해 알아보고 break,continue와  
같은 명령어도 공부해보아요.

## ▶ 오늘의 학습 – 반복문 while



### 조건을 만족하는 동안 while



난 5시 10분에 공부가 끝나!!!  
그 때 까지 쭈욱 공부할래^^

5시 10분 01초 입니다!!!  
주인님 얼른 나가세요^^



**While - 조건을 만족할 때 까지 문장을 반복수행**

➤ 형식

While 조건식 :  
반복 실행문장

while 시간 <=5시 10분  
    공부!!

## ▶ 오늘의 학습 – 반복문 while



for문과 다른점은?



for 도 특정영역을 반복수행하고, while 도 반복수행하잖아. 어떤 경우 while을 써야 하는거지?



while은 마지막 종료값이 확실하지 않은 경우 이용해요.

### for를 사용해요 – 명확한 목표

키보드로부터 3번  
값 입력받기



1부터 100까지  
합계 계산하기



### while문을 사용 – 끝이 어딘지 몰라요

키보드로부터 99가  
입력될 때 까지 숫자 입력받기



사용자가 언제쯤  
99를 입력할까?



숫자의 합계가 5000을  
넘을 때 까지 덧셈작업하기



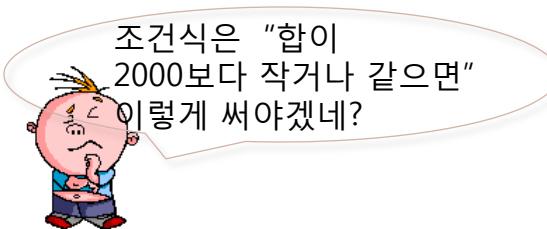
얼마까지 더해야  
5000을 넘어서지?



## while 이용하여 임의의 숫자 입력받기



▶ 숫자 1부터 100까지를 더하는 과정에서 합이 2000보다 크면 while 루프를 종료하고, 얼마까지 더할 때 2000보다 큰 값이 되는지를 출력하는 프로그램



```
sum=0  
i=0  
while sum <=2000:  
    i=i+1  
    if i<=100:  
        sum+=i  
  
print ("2000초과 i=", i, "합계=", sum)
```

python18-1.py

```
>>>  
2000초과 i= 63 합계= 2016  
>>>
```

## ▶ 오늘의 학습 – 반복문 while



while은 무한하게 반복될 수 있어요



끝도없이 특정영역을 반복한단 말이야?



예 ^^ 무한loop라고 해요.  
형식은 아래와 같아요.



while 1:  
반복문장

그럼 어떻게 실행을 중지시키지?



실행을 중지하려면 ctrl 키와 c를 눌러주세요^^



```
while 1:  
    print("안녕하세요")
```



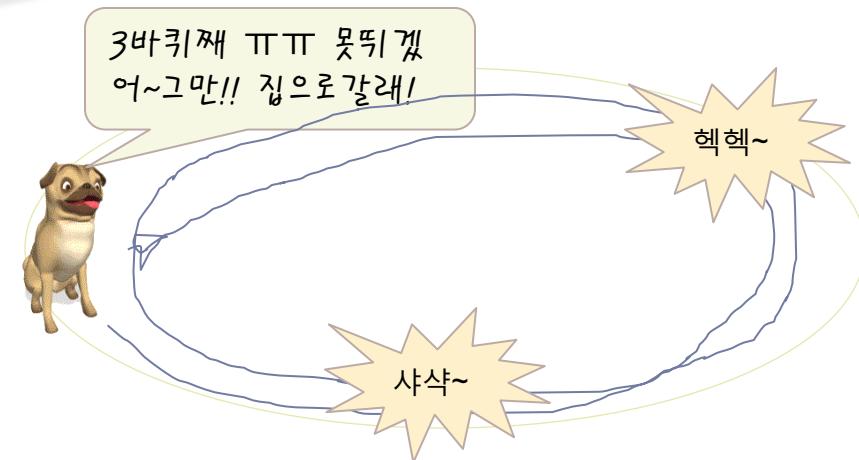
```
안녕하세요  
안녕하세요  
안녕하세요  
안녕하세요  
Traceback (most recent call last):  
  File "C:/Python34/aaaaaa.py", line 3, in <module>  
    print("안녕하세요")  
  File "C:/Python34/lib/idlelib/PyShell.py", line 1  
    return self.shell.write(s, self.tags)  
KeyboardInterrupt  
>>> |
```

## ▶ 오늘의 학습 – 반복문 while



### 루프를 빠져나가는 break

똘똘아!! 운동장 10바퀴  
돌아^^



#### ▶ 반복작업에서 도중에 빠져나올 때 사용

- break 예제 – 앞서 봤던 1부터 100까지 숫자의 합계를 구하되 합이 2000넘어서면 합계작업 중단하고 해당 값을 출력하는 프로그램

```
sum=0
for i in range(1,101):
    if(sum>=2000):
        break → sum이 2000이상이면 for문을 더 이상 수행하지 않아요
    else:
        sum+=i
print("합계 2000이 넘는 지점", i, "합계=", sum)
```

python18-2.py

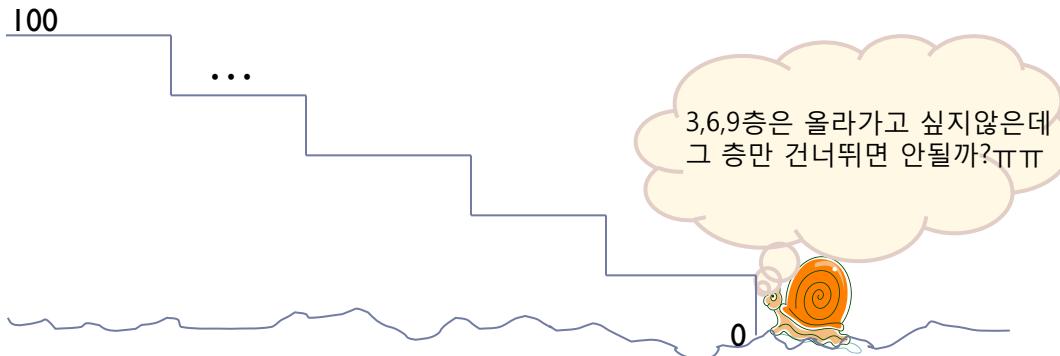


```
>>> 합계 2000이 넘는 지점 64 합계= 2016
>>> |
```



## 다음으로 넘어가기 continue

➤ Loop에서 현재 작업은 반복생략하고 다음 반복작업으로 넘어가고자 할 때 사용



```
for i in range(1,11):
    if i==5:
        continue
    else:
        print(i, "번째 메시지 즐거운 하루")
print("여러분!! 행복하세요")
```

python18-3.py

Continue 문을 만나면 아래 내용이 실행되지 않고 Loop 시작부분으로 돌아가게됩니다.

```
>>>
1 번째 메시지 즐거운 하루
2 번째 메시지 즐거운 하루
3 번째 메시지 즐거운 하루
4 번째 메시지 즐거운 하루
6 번째 메시지 즐거운 하루
7 번째 메시지 즐거운 하루
8 번째 메시지 즐거운 하루
9 번째 메시지 즐거운 하루
10 번째 메시지 즐거운 하루
여러분!! 행복하세요
```

## ▶ 오늘의 학습 – 반복문 while

### 도전!! 절대 틀리지 않는 3,6,9게임 – 1탄



▶ 우리는 1부터 10까지 숫자 중에서 3,6,9가 나오면 해당숫자를 보여주지 않고 “삑” 소리를 낸 다음 2초 동안 프로그램이 정지한 후 다시 다음 숫자를 보여주는 프로그램을 작성해봅니다.

잠깐!! 나는 사운드음을  
어떻게 표시하는지 몰라요ㅠ<sup>ㅠ</sup>  
알려주세요^^



모듈중에 winsound 모듈이  
있어요. 얘는 윈도우의 기본  
사운드를 제공해요^^



실습해보니  
진짜 소리가 나네?



```
import winsound  
winsound.Beep(600, 250)
```

Beep – pc의 기본 사운드를 들려줌  
600- 사운드의 헤르츠  
250 – 사운드의 시간(100만분의 1) 단위

## ▶ 오늘의 학습 – 반복문 while

1. 기본사운드와 시간을 설정하기 위해 winsound,time 모듈을 가져오는 작업을 코딩 합니다.

```
import winsound  
import time
```

3. 해당 숫자가 3,6,9에 해당하면 경고음을 들려 준 다음 2초 동안 정지한 후 for로 이동하여 다음 숫자를 반복하도록 합니다.

```
import winsound  
import time  
for i in range(1,11):  
    if i==3 or i==6 or i==9 :  
        winsound.Beep(600,250)  
        time.sleep(2)  
    continue
```

2. for를 이용하여 1부터 10까지 숫자를 반복하도록 설정합니다. 그 다음 현재 숫자가 3,6,9에 해당하는지 조건문을 코딩해요.

```
import winsound  
import time  
for i in range(1,11):  
    if i==3 or i==6 or i==9 :
```

4. 3,6,9에 해당하지 않는 일반숫자들은 당연히 print해야 되겠죠? print 함수를 코딩하고 저장합니다. 파일명-python18-4.py로 저장하세요.

```
import winsound  
import time  
for i in range(1,11):  
    if i==3 or i==6 or i==9 :  
        winsound.Beep(600,250)  
        time.sleep(2)  
    continue  
    print(i)
```

```
>>>  
1  
2  
4  
5  
7  
8  
10  
>>>
```



## 오늘의 핵심내용

1. 어디까지 반복해야 하는지 명확하지 않는 경우 while문을 이용합니다.
2. while, for 문 수행 중 특정조건등으로 반복문을 빠져나갈경우 break문을 이용합니다.
3. 반복 실행에서 현재 반복을 중단하고 다음 수치에 대한 반복을 실행할 경우 continue문을 이용합니다.

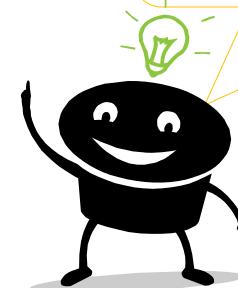


## 잠깐!!! 연습문제

I. 아래는 1~50 사이의 숫자 중에서 짝수만을 추출하여 합계를 계산하는 프로그램을 `continue` 문을 이용하여 작성한 내용입니다. 빈 칸에 어떤 부분이 필요한지 깊이 생각하여 프로그램을 완성하고 실행해 주세요^^

```
sum=0
for i in range ( (여기) ) :
    (여기)
    continue
    sum+=i
print("1----50까지 짝수합계=", sum)
```

힌트) 짝수 인 경우에만 합계가 출력되는 거예요^^



```
>>>
>>> 1----50까지 짝수합계= 650
>>> |
```

프로그램 전체소스는 뒤에...

## ▶ 오늘의 학습 – 반복문 while

프로그램 소스 python18-test.py

```
sum=0
for i in range(1, 51):
    if i%2!=0:
        continue
    sum+=i
print("1----50까지 짝수합계=", sum)
```