



## 반복문

### while문

조건식을 검사한 후, 참인 동안 실행문을 반복

ex) `int a = 1;`

```
while (a < 10)
{
    a = a * 2;
}
```

⇒ a가 16이 되었을 때 반복문을 종료한다.

\* 퀴즈일래는 반복문을 한 문장으로 인식한다.

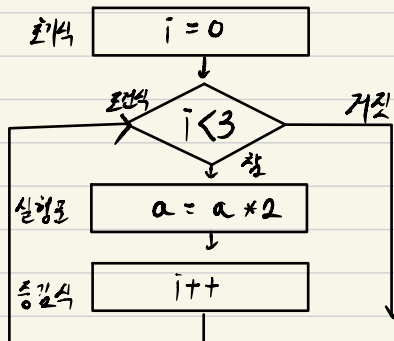
∴ if문의 실행문이 반복문 하나라면 동결로 생각 가능

### for문

초기식      조건식      증감식

↑            ↑            ↑

```
for (i = 0; i < 3; i++)
{
    a = a * 2;
}
```



## do while 문

while, for 문은 조건식을 먼저 확인하는 반면, do~while 문은 일단 실행한 뒤 조건을 검사한다.

⇒ 반드시 한번은 실행된다.

ex) int a = 1;

```
do
{
    a = a * 2;
} while (a < 10); // 서미콜론 주의!
```

만약 a = 10으로 초기화되었다 하면, do~while 문 이후에 a는 20이 된다.

## break

반복문 안에서 음간에 임의로 반복을 끝내고 싶을 때 사용한다.

ex) int sum = 0;

```
for (int i = 1; i <= 10; i++)
{
    sum += i;
    if (sum > 30) break; // sum > 30이면 반복문을 빠져나간다.
}
```

※ break는 자신이 속한 반복문 하나만 벗어난다.

+ 반복문 이외의 블록에서 사용하면 그 블록을 포함한 반복문을 빠져나간다.

but, if 문과 달리 switch~case 문이 블록 안에서 사용시, switch~case 간 빠져나간다.

## Continue

continue 사용시 다음 실행 위치가 백문 블록의 끝이 된다.

```
ex) for(i=1; i <= 100; i++)  
    {  
        if((i%3) == 0)  
        {  
            continue;  
        }  
        sum += i;  
    }
```

백문 블록으로 이동.

즉, 다음 실행은 증강식이 된다.