

K G 아 이 티 뱅 크

C 언 어

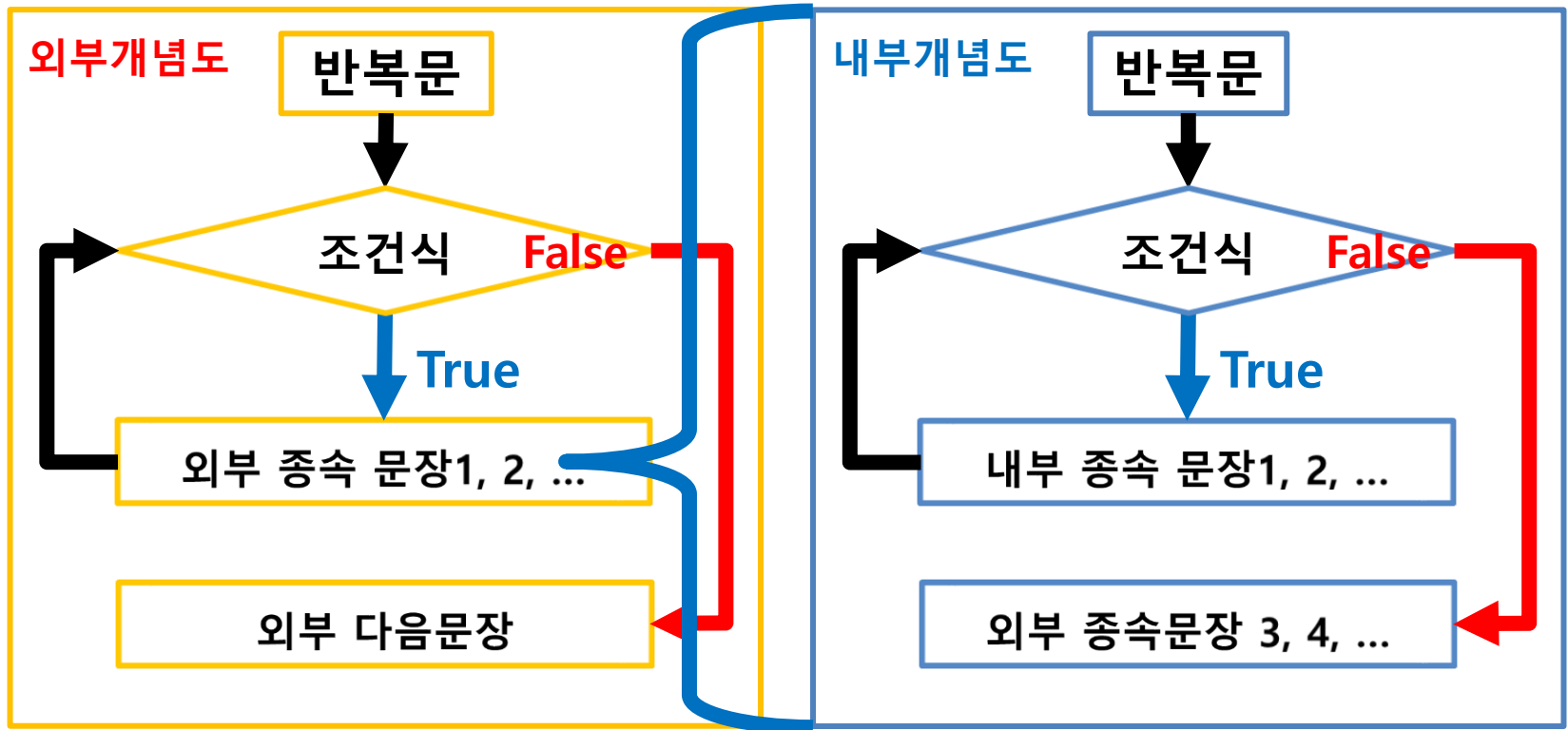
C L A N G U A G E

반복문 Part. 3

반복문 Part. 3

❖ 반복문의 종속문장으로 반복문이 들어간 코드

- while과 for 둘 다 가능하며, 자유롭게 구성할 수 있음
- 내부의 반복문의 반복이 다 끝나야 외부 1회가 끝남
- 서로 동시에 병렬로 반복이 진행되지 않음



❖ 다중 반복문은 종속문의 반복되는 내용을 반복시키는 것

1) 코드의 간략화가 가능하며, 좀 더 쉽게 코드작성이 가능

➤ 단일 반복으로 감당이 안되는 복잡한 경우에 사용

2) 서로 연동되어 작동하며, 변수가 중첩되지 않도록 주의

➤ **while의 경우, 변수에 저장되는 값을 조심해야 함**

단일반복문 : 계단모양으로 출력

```
int count = 0;
int limit = 1;
for (int i=1; i<=12; i+=1) {
    printf("%3d");
    count += 1;
    if (count == limit) {
        printf("\n");
        count = 0;
        limit += 1;
    }
}
```

이중반복문 : 계단모양으로 출력

```
int num = 1;
for (int y=1; y<=4; y+-1) {
    for (int x=1; x<=y; x+=1) {
        printf("%3d", num);
        num += 1
    }
    printf("\n");
}
```

반복문 Part. 3

< 파일이름 : 07. 반복문3_EX1.c >

실습문제1. 아래의 조건을 만족하는 코드를 작성하세요.

✓ 조건

1. 구구단을 2번 출합니다.
2. 첫번째 출력은 for을 이용한 이중반복으로 구성합니다.
3. 두번째 출력은 while을 이용한 이중반복으로 구성합니다.

✓ 결과

- 출력(for) -

```
1 x 1 = 1
1 x 2 = 2
...
3 x 1 = 3
...
9 x 9 = 81
```

- 출력(while) -

```
1 x 1 = 1
1 x 2 = 2
...
3 x 1 = 3
...
9 x 9 = 81
```

반복문 Part. 3

< 파일이름 : 07. 반복문3_EX2.c >

실습문제2. 아래의 조건을 만족하는 코드를 작성하세요.

✓ 조건

1. 1부터 9까지의 정수를 아래와 같이 2번 출력합니다.
2. 첫번째 출력은 for만 이용합니다.
3. 두번째 출력은 while만 이용합니다.

✓ 결과

- 출력(for) -

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

- 출력(while) -

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