

K G 아 이 티 뱅 크

C 언 어

C L A N G U A G E

반복문 1. while

## 반복문( while )

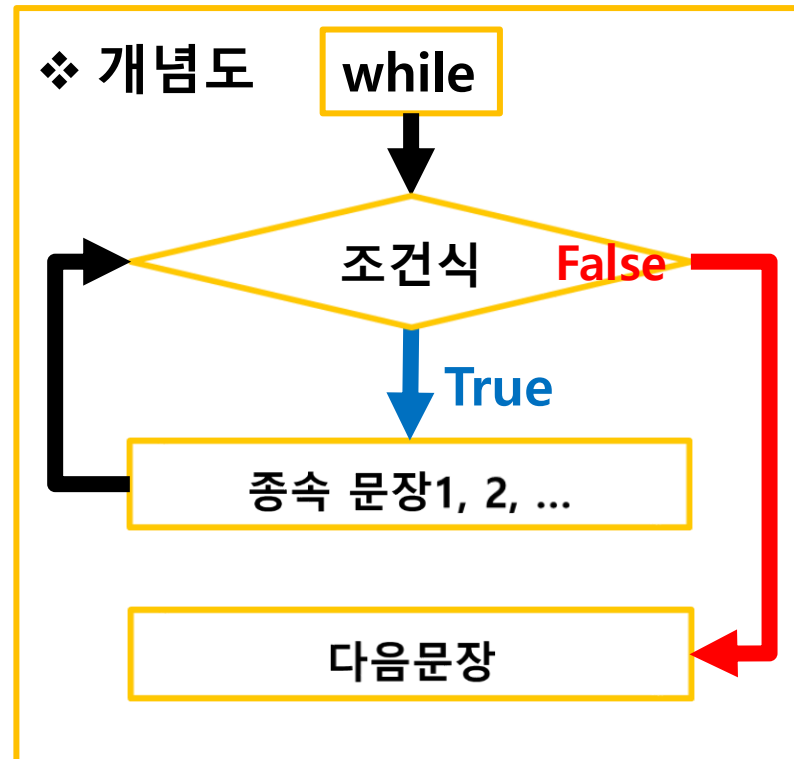
### ❖ 프로그램의 흐름을 제어하는 제어문종의 두번째

- 규칙성 있게 반복되는 내용의 불특정 횟수 실행을 처리
- 값을 이용하며 일반적으로 조건식 또는 값을 사용

#### ❖ 문법

```
이전문장;  
while (조건식) {  
    종속문장1;  
    종속문장2;  
    종속문장3;  
}  
다음문장;
```

#### ❖ 개념도



## 반복문( while )

### ❖ while의 사용방법 첫번째 : 무한반복

- 1) 언제 끝날지는 명확하게 존재한다
- 2) 무엇으로 시작하며, 몇 번 하는지는 명확하게 존재하지 않는다

#### ❖ 코드1. 반복 미사용

```
int count;
scanf_s("%d", &count);
if (count < 0) {
    scanf_s("%d", &count);
}
if (count < 0) {
    scanf_s("%d", &count);
}
if (count < 0) {
    scanf_s("%d", &count);
}
```

#### ❖ 코드2. 반복 사용

```
int count;
scanf_s("%d", &count);
while (count < 0) {
    scanf_s("%d", &count);
}
```

## 반복문( while )

< 파일이름 : whileEX1.c >

❖ 실습예제1. 아래의 조건을 달성하는 코드를 작성하세요.

### 조건

값을 입력을 받습니다.

입력을 받은 값이 10 이하이면 입력을 종료합니다.

### 결과

정수 입력 >> 11

정수 입력 >> 12

정수 입력 >> 5000

정수 입력 >> 1

10 이하가 입력이 되어 종료합니다.

## 반복문( while )

### ❖ while의 사용을 보조 : break

1) 원하는 시점에서 반복을 종료하기 위해 사용

#### ❖ 코드1. break 미사용

```
int count;
scanf_s("%d", &count);
while (count < 0) {
    scanf_s("%d", &count);
}
```

1. 입력을 받음
2. 조건식을 점검
  - 참/거짓을 확인
3. 참이라면 다시 입력을 받음
4. 거짓이라면 반복을 종료

#### ❖ 코드2. break 사용

```
int count;
while (1) {
    scanf_s("%d", &count);
    if (count >= 0) break;
}
```

1. 조건식을 점검 : 참이니 통과
2. 입력을 받는다
3. 입력을 받은 값을 확인
  - 참이라면 반복종료

## 반복문( while )

### ❖ while의 사용을 보조 : continue

1) 원하는 시점에서 반복을 다시 수행하기 위해 사용

#### ❖ 코드1. continue 없음

```
int count;
while (1) {
    scanf_s("%d", &count);
    if (count >= 0) break;
}
```

1. 조건식을 점검 : 참이니 통과

2. 입력을 받는다

3. 입력을 받은 값을 확인

➤ 참이라면 반복종료

#### ❖ 코드2. continue 있음

```
int count;
while (1) {
    scanf_s("%d", &count);
    if (count < 0) continue;
    break;
}
```

1. 조건식을 점검 : 참이니 통과

2. 입력을 받는다

3. 입력을 받은 값을 확인

➤ 참이라면 처음으로 이동

4. 반복 종료

## 반복문( while )

< 파일이름 : whileEX2.c >

❖ 실습예제2. 아래의 조건을 달성하는 코드를 작성하세요.

### 조건

1. 입력한 값이 11의 배수가 아니라면 계속 입력을 받습니다.
2. 입력한 값이 11의 배수면 입력을 종료합니다.

### 결과

```
정수 입력 >> 12  
정수 입력 >> 8  
정수 입력 >> 121
```

11의 배수가 입력이 되어 종료합니다.



## 반복문( while )

### ❖ while의 사용방법 두번째 : 특정 횟수만큼만 반복

#### 1) 규칙성을 찾아야만 한다

- 특정 내용이 같은 모양으로 반복이 되는 것을 확인
- 이를 위해서 반드시 전체/일부 내용의 작성이 필요

#### 2) break와 continue를 활용할 수 있음

- 모든 반복문에서 사용할 수 있는 명령어
- 단, 사용하는 시점과 필요성 등을 판단해야 함

#### ❖ 코드1. 원본

```
printf("1\n");  
printf("2\n");  
printf("3\n");  
printf("4\n");  
printf("5\n");  
printf("6\n");
```

#### ❖ 코드2. 반복문으로 변환

```
int num = 1  
while (num<=6) {  
    printf("%d ", num);  
    num++;  
}
```

## 반복문( while )

< 파일이름 : whileEX3.c, whileEX4.c >

❖ 실습예제3. 아래의 조건을 달성하는 코드를 작성하세요.

**조건**

Hello! 를 10번 출력합니다.

**결과**

Hello!(10번 출력)

❖ 실습예제4. 아래의 조건을 달성하는 코드를 작성하세요.

**조건**

1부터 30까지의 정수 중 2의 배수만 출력

**결과**

2 4 6 8 10 . . . . 30

## 반복문( while )

< 파일이름 : whileEX5.c, whileEX6.c >

❖ 실습예제5. 아래의 조건을 달성하는 코드를 작성하세요.

### 조건

1부터 30까지의 수 중 입력을 받은 정수의  
배수만 출력합니다.

### 결과

출력할 배수 입력 >> 5  
5 10 15 20 25 30

❖ 실습예제6. 아래의 조건을 달성하는 코드를 작성하세요.

### 조건

1부터 내가 지정한 범위까지의 수만 출력합니다.

### 결과

출력할 범위 지정 >> 5  
1 2 3 4 5

## 반복문( while )

< 파일이름 : whileEX7.c >

❖ 실습예제7. 아래의 조건을 달성하는 코드를 작성하세요.

### 조건1

1부터 10까지의 합을 반복으로 구해서  
출력합니다.

### 결과1

1부터 10까지의 합 : 55

### 조건2

내가 지정한 배수의 합만 구하세요.

### 결과2

배수 지정 >> 3  
배수의 합 : 18

## 반복문( while )

< 파일이름 : whileEX8.c >

❖ 실습예제8. 아래의 조건을 달성하는 코드를 작성하세요.

### 조건1

내가 입력한 양의 정수의 약수를 출력하세요.

### 결과1

약수를 구하려고 하는 양의 정수 >> 15

약수 목록

1 3 5 15

### 조건2

내가 입력한 양의 정수의 약수의 합을 구하세요.

### 결과2

약수의 합을 구하려고 하는 양의 정수 >> 10

약수의 합 : 18

반복문 2. for

## 반복문( for )

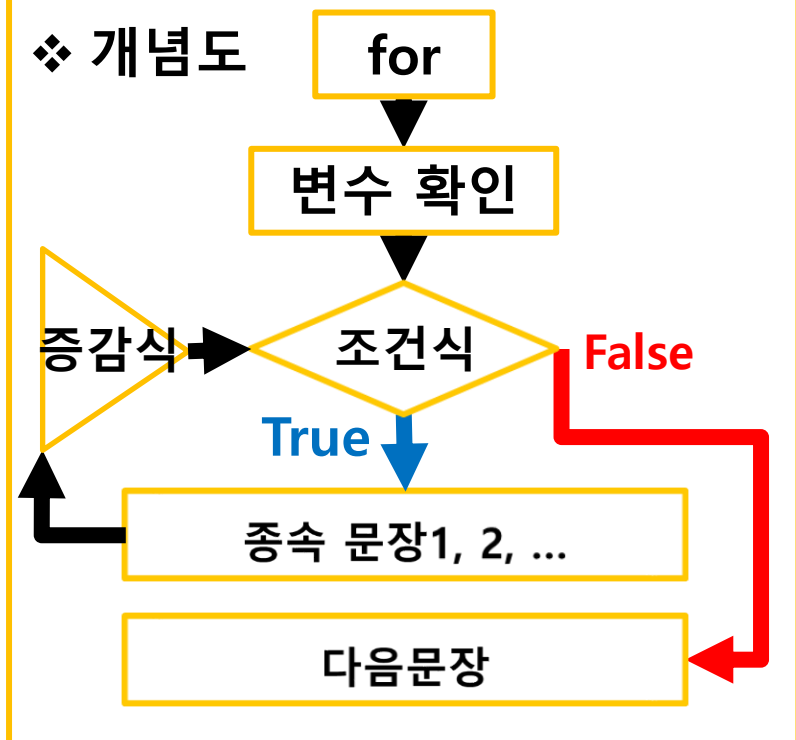
### ❖ 프로그램의 흐름을 제어하는 제어문 중의 세번째

- 규칙성 있게 반복되는 내용의 특정 횟수 실행을 처리
- 변수, 조건식, 증감식을 전부 미리 준비하여 사용

#### ❖ 코드

```
// 최초 구성시 미리 설정  
// 변수를 선언할 수 있음  
for (변수준비;조건식;증감식) {  
    종속문장1;  
    종속문장1;  
    종속문장1;  
}  
다음문장;
```

#### ❖ 개념도



## 반복문( for )

### ❖ for 반복문은 반복횟수가 정해져 있는 반복문

#### ➤ 기본적인 운용은 while과 동일하게 사용이 가능

① while : 제어용 변수를 가공하여 결과를 만들 수 있음

② for : 제어용 변수는 제어 또는 종속문 보조용으로 사용

#### ➤ for문은 최초 구성단계부터 반복할 횟수를 미리 설정

① 지정횟수 미만으로 처리시 **break**의 활용이 가능

② 특정 값의 처리가 필요가 없다면 **continue**로 넘기기 가능

#### ❖ 코드1. break를 통한 중단

// 반복문은 break로 중단이 가능

```
int num=1;
for (int i=1;i<=10;i++) {
    num*=i;
    if (num>=100) break;
}
printf("num : %d", num);
```

#### ❖ 코드2. continue로 넘기기

# while과 달리 증감식은 별도처리

```
int num=1;
for (int i=1;i<=10;i++) {
    if (i%2==0) continue;
    num*=i;
}
printf("num : %d", num);
```



## 반복문( for )

< 파일이름 : forEX1.c >

❖ 실습예제1. 아래의 조건을 달성하는 코드를 작성하세요.

### 조건

지정한 횟수만큼 아래의 내용을 출력하세요.

“C Language is SIMPLE”

### 결과( 입력 횟수 3로 가정 )

--값의 입력--

횟수 입력 :

--반복 출력--

“C Language is SIMPLE”

“C Language is SIMPLE”

“C Language is SIMPLE”

## 반복문( for )

< 파일이름 : forEX2.c >

❖ 실습예제2. 아래의 조건을 달성하는 코드를 작성하세요.

### 조건

1부터 20까지의 수 중 홀수만 출력합니다.  
아래의 결과를 참고하여 코드를 작성합니다.

### 결과

홀수만 출력하는 반복문

1회차: 1

2회차: 3

3회차: 5

4회차: 7

5회차: 9

...

10회차: 19

## 반복문( for )

< 파일이름 : forEX3.c >

❖ 실습예제3. 아래의 조건을 달성하는 코드를 작성하세요.

### 조건

구구단을 출력하세요.

단, 출력할 단과 출력할 범위를 지정합니다.

### 결과( 3 6을 입력했을 경우)

-- 값의 입력 --

단수, 범위 설정(띄어쓰기로 구분) :

구구단 3단

3 x 1 = 3

3 x 2 = 6

3 x 3 = 9

3 x 4 = 12

3 x 5 = 15

3 x 6 = 18

## 반복문( for )

< 파일이름 : forEX4.c >

❖ 실습예제4. 아래의 조건을 달성하는 코드를 작성하세요.

### 조건

for 문의 제어 변수로 실수 변수를 이용합니다.

값은 0.0으로 설정합니다.

for 문의 증감식으로 0.1씩 증가하도록 설정합니다.

위의 두 조건을 이용하여 10회 반복이 되도록  
설정하세요.

### 결과

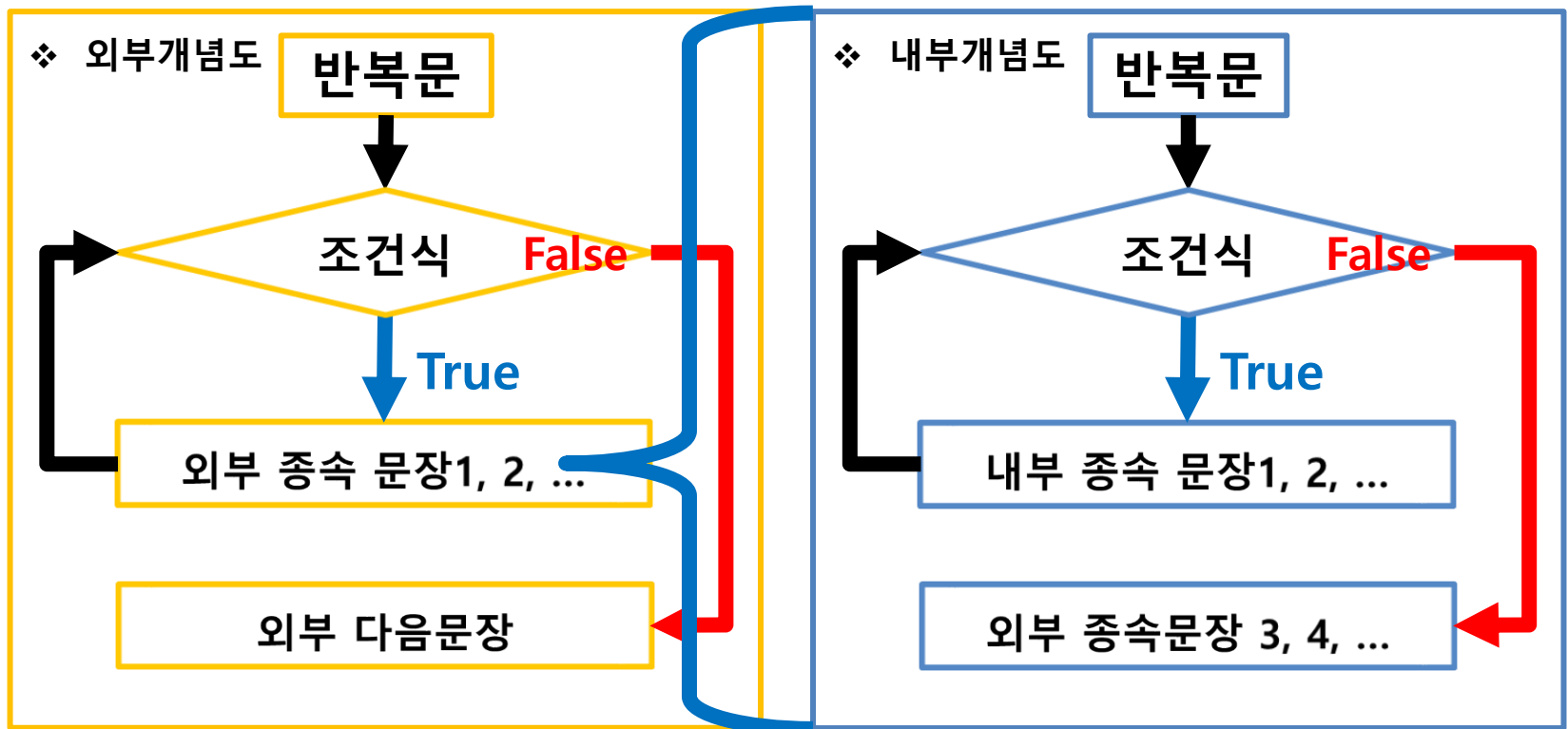
임의의 출력 10회 달성

다중 반복문

## 다중 반복문

### ❖ 반복문의 종속문장으로 반복문이 들어간 코드

- while과 for 둘 다 가능하나, 일반적으로 for문으로 구성
- 내부의 반복문의 반복이 다 끝나야 다음 코드를 진행
  - 동시반복, 병렬반복이 아닌 별개의 반복으로 처리됨



## 다중 반복문

❖ 다중 반복문은 종속문의 반복되는 내용을 반복시키는 것

1) 코드의 간략화가 가능하며, 좀 더 가독성이 높아짐

- 비슷한 내용의 반복은 검토할 때 피로를 유발
- 반복되는 비슷한 내용은 사소한 실수를 유발하게 됨

2) while은 같은 변수를 이용하게 되니 주의해야 함

- for : 지역변수로 선언된 변수를 이용하여 처리
- while : 지역변수가 아닐 경우 값이 누적되니 바뀌어야 함

### ❖ 코드1. 단일반복문

```
for (int i=1;i<=3;i++) {  
    printf("%d ",i+1);  
    printf("%d ",i+2);  
    printf("%d ",i+3);  
    printf("%d ",i+4);  
    printf("%d ",i+5);  
    printf("\n");  
}
```

### ❖ 코드2. 이중반복문

```
for (int i=1;i<=3;i++) {  
    for (int j=1;j<=5;j++) {  
        printf("%d ",i+j);  
    }  
    printf("\n");  
}
```

## 다중 반복문

< 파일이름 : dualEX1.c >

❖ 실습예제1. 아래의 조건을 달성하는 코드를 작성하세요.

**조건**

크기와 문자를 입력을 받아 큐브를 만드세요.

**결과( 크기 3, 문자열 □으로 가정)**

크기 입력 :

문자 지정 :

□ □ □

□ □ □

□ □ □



## 다중 반복문

< 파일이름 : dualEX2.c >

❖ 실습예제2. 아래의 조건을 달성하는 코드를 작성하세요.

### 조건

지정한 단수의 계단을 만드세요.

### 결과( 4를 입력했을 경우 )

계단 단수 :

□

□ □

□ □ □

□ □ □ □

## 다중 반복문

< 파일이름 : dualEX3.c >

❖ 실습예제3. 아래의 조건을 달성하는 코드를 작성하세요.

**조건**

구구단을 출력해보세요.

**결과**

1 x 1 = 1

...

3 x 6 = 18

...

9 x 9 = 81

## 다중 반복문

< 파일이름 : dualEX4.c >

❖ 실습예제4. 아래의 조건을 달성하는 코드를 작성하세요.

### 조건

1부터 25까지의 숫자를 아래와 같이 출력합니다.  
임의로 구성합니다.

### 결과

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25