

K G 아 이 티 뱅 크

C 언 어

C L A N G U A G E

반복문 1. while

반복문(while)

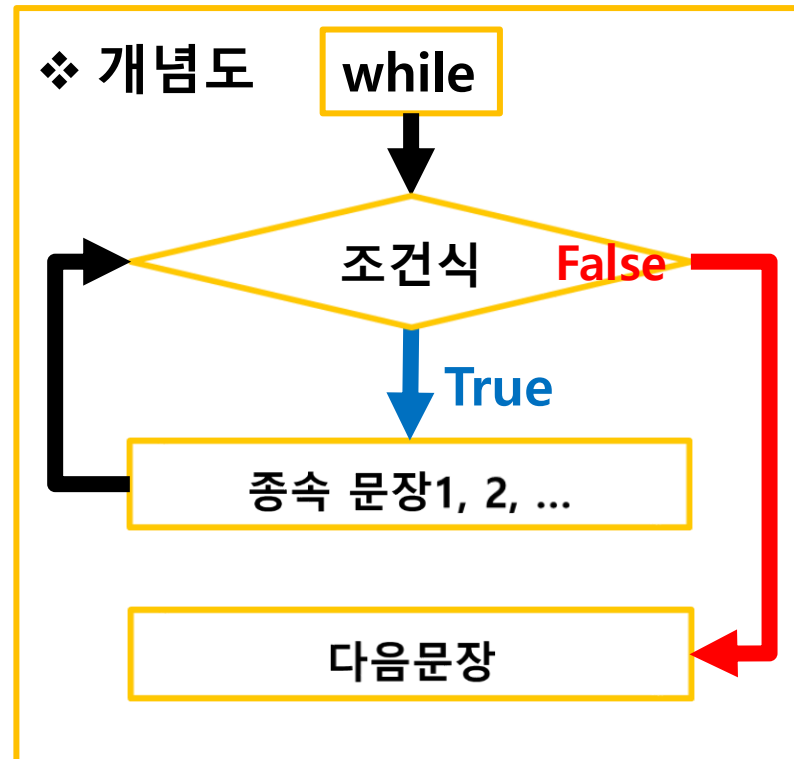
❖ 프로그램의 흐름을 제어하는 제어문종의 두번째

- 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) 왜 반복을 시작하며, 왜 끝나는지 명확하게 존재한다
- 2) 몇 번 하는지는 모른다

❖ 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 = -1;
while (count < 0) {
    scanf_s("%d", &count);
}
```

반복문(while)

< 파일이름 : whileEX1.c >

❖ 실습예제1. 아래의 조건을 달성하는 코드를 작성하세요.

조건

값을 입력을 받고, 입력을 받은 값이 10 이하이면
반복을 종료합니다.

10보다 크면 계속 입력을 받습니다.

결과

- 반복구간 -

정수 입력 >> 11

정수 입력 >> 12

정수 입력 >> 5000

정수 입력 >> 1

- 반복종료후 출력 -

10 이하가 입력이 되어 종료합니다.

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

- 반복구간 -

정수 입력 >> 12

정수 입력 >> 8

정수 입력 >> 121

- 반복종료후 출력 -

11의 배수가 입력이 되어 종료합니다.

반복문(while)

❖ while의 사용방법 두번째 : 특정 횟수만큼만 반복

1) 규칙성을 기반으로 움직인다

- 규칙성을 위한 조건식과 그에 대응되는 증감식이 있음

2) 규칙성이 없으면 대부분의 경우 1씩 증가/감소한다

- 제일 어려운 것은 아무런 변화가 없는 반복
- 그럴 경우 1씩 증가/감소를 통해 해결

❖ 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 >

❖ 실습예제3. 아래의 조건을 달성하는 코드를 작성하세요.

조건

Hello! 를 10번 출력합니다.

결과

Hello!(10번 출력)

반복문(while)

< 파일이름 : whileEX4.c >

❖ 실습예제4. 아래의 조건을 달성하는 코드를 작성하세요.

조건

1부터 30까지의 정수 중 2의 배수만 출력

결과

2 4 6 8 10 30

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 : whileEX5.c >

❖ 실습예제5. 아래의 조건을 달성하는 코드를 작성하세요.

조건

1부터 30까지의 수에 대하여
입력을 받은 수가 양의 정수일 경우에만
배수를 출력합니다.
0이하면 재입력을 받습니다.

결과

출력할 배수 입력 >> 5
5 10 15 20 25 30

반복문(while)

< 파일이름 : whileEX6.c >

❖ 실습예제6. 아래의 조건을 달성하는 코드를 작성하세요.

조건

1부터 10까지의 정수중

내가 입력한 0보다 큰 양의 정수의 배수의 합을
구합니다.

0이하이면 재입력을 받습니다.

결과

배수 지정 >> 3

배수의 합 : 18

반복문(while)

< 파일이름 : whileEX7.c >

❖ 실습예제7. 아래의 조건을 달성하는 코드를 작성하세요.

조건

내가 입력한 양의 정수의 약수를 출력하세요.

결과

- 값의 입력 -
약수를 구하려고 하는 양의 정수 >> 15

- 결과 출력 -
약수 목록
1 3 5 15