



# [6주차 1강] 반복문(1)



# 학습 내용

6.1 반복문 개요

6.2 while 문

6.3 do-while 문



# 학습 목표

- 6.1 반복문이 무엇인지 이해한다.
- 6.2 while 문의 사용법을 익힌다.
- 6.3 do-while 문의 사용법을 익힌다.





6.1 반복문 개요

6.2 while 문

6.3 do-while 문



## 제어문

- C 언어는 순차처리언어로, 특별한 지정이 없으면, 소스 코드 첫 줄부터 차례대로 처리
- 그러나 문제 해결 위해 처리 흐름 제어 필요 → 제어문
- C 언어에서는 조건문(5장), **반복문(6장)**과 같은 제어문 제공

## 제어문 종류

분류	종류
조건문	if 문, if~else 문, switch 문
반복문	for 문, while 문, do~while 문
기타	break 문, continue 문, goto 문, return 문



## 반복문

- 특정 조건을 만족하는 동안 계속 반복하여 실행하는 문장
- 루프(loop)라고도 불림
- while 문, do-while 문, for 문
- 반복문이 필요한 예
  - ✓ "Hello World"를 100번 출력하려면?
  - ✓ 1부터 100까지의 합을 구하려면?

```
// 동일한 내용을 반복할 경우  
printf("Hello World\n");  
printf("Hello World\n");  
printf("Hello World\n");
```

```
// 일정한 규칙으로 반복하는 작업을 수행하는 경우  
int sum = 1 + 2 + 3 + 4 + 5;
```



### while 문의 대략적인 형태

- "Hello World" 3번 출력하기

```
int i = 1;           ⇒ 반복문 진입 전에 i=1 로 초기 설정
while( i <= 3 ) {    ⇒ i<=3 인 동안(while) 아래 문장 반복
    printf("Hello World\n");    ⇒ 반복할 문장
    i++;               ⇒ i 값 1 증가
}                     ⇒ while 문의 끝
```

- ✓ 키워드 **while** : 조건을 만족하는 **동안~**
- ✓ 변수 **i** 는 **반복 횟수를 제어**하는 역할



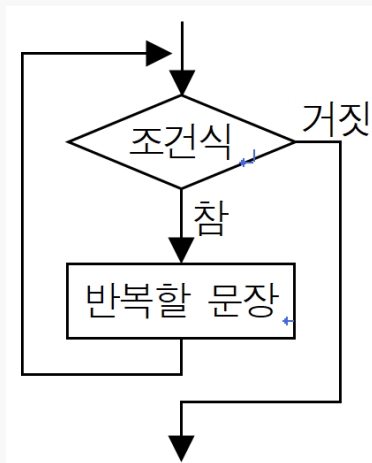
### while 문 대략적으로 살펴보기

- 조건식이 참인 동안 중괄호로 둘러 쌓인 블록 안의 문장 반복  
조건식이 거짓이 되면 while 문을 빠져 나옴
  - ✓ 반복할 문장이 하나이면 중괄호 생략 가능

- 구문

```
while( 조건식 )  
{  
    반복할 문장;  
}
```

```
while( i <= 3 ) {  
    printf("Hello World\n");  
    i++;  
}
```







### [프로그램 1] 반복 횟수가 정해져 있는 예시

- while 문을 이용하여 1부터 5까지의 합 구하기

```
int i, sum;

sum = 0;           // 합을 저장하는 변수
i = 1;             // 더하는 수를 저장하는 변수

while ( i <= 5 ) { // i가 5보다 같거나 작은 동안
    sum += i;      // i를 sum에 더하고
    i++;           // i값 1 증가
}

printf("1부터 5까지의 합은 %d 입니다.\n", sum);
```

- ✓ 프로그램의 실행됨에 따라 제어 흐름과 변수가 어떻게 변하는지 따져보자.  
(교재 p.148의 수행과정 표 참조)



### While 문 수행 과정 확인하기

- while 문의 끝에 **출력문**을 추가하여 수행과정을 눈으로 확인해보자.

```
while( i <= 5 ) {  
    sum += i;  
    i++;  
    printf("i: %d, sum: %d\n", i, sum);  
}
```

#### 실행 결과

```
i: 2, sum: 1  
i: 3, sum: 3  
i: 4, sum: 6  
i: 5, sum: 10  
i: 6, sum: 15
```

- ✓ **디버깅 팁** : 반복문의 처음이나 마지막에 출력문을 넣는 것은  
반복문이 의도한대로 동작하는지 확인하는 기본적인 방법 중 하나

# 실습하기



다음 각 문장에 해당하는 while 문을 작성하시오.



1. 변수  $i$ 가 10보다 작은 동안,  
“Hello World” 를 출력하고  $i$ 를 1만큼 증가
2. 변수  $i$ 가 0보다 크고 10보다 작은 동안,  
변수  $x$ 에 0을 대입하고  $i$ 를 1만큼 감소
3. 변수  $num$ 이 50보다 크기 전까지, (즉, 50보다 크면 반복 종료)  
정수를 읽어 들여  $num$ 에 저장

31 50 4 51



### [프로그램 2] 반복횟수가 정해져 있지 않은 예시

- 0 또는 음수가 입력되기 전까지 정수를 입력 받아,  
입력 받은 정수의 합(종료 조건 0또는 음수는 합에서 제외) 구하기

```
int x, sum = 0;

printf("정수를 입력하시오: ");           // 입력 안내문
scanf("%d", &x);                          // 첫 정수 입력

while ( x > 0 ) { // x가 양수이면 반복
    sum += x;
    printf("정수를 입력하시오: ");
    scanf("%d", &x);
}

printf("입력된 정수의 합은 %d 입니다.\n", sum);
```

# 실습하기



[예제 6.1]

정수를 입력 받아, 그 정수에 해당하는 구구단 출력하기



입력

5

출력

$$5 * 1 = 5$$

$$5 * 2 = 10$$

...

$$5 * 9 = 45$$

# 실습하기



## [예제 6.2] 영어 소문자 순서대로 출력하기



- ✓ hint) 영어 소문자의 아스키 코드 값이 연속한다는 성질 이용
- ✓ 영문자의 아스키 코드 값을 알 필요는 없음 (2장 문자 자료형)

실행 결과

abcdefghijklmnopqrstuvwxyz

# 실습하기



[예제 6.3]

1부터 시작하여 값을 1씩 증가시키면서 차례로 더한 합이 최초로 100 이상이 되는 때, 마지막에 더한 값 구하기



✓ 즉,  $1+2+\dots+n \geq 100$  인 가장 작은  $n$  구하기

실행 결과

14

# 실습하기



[예제 6.4]

0이 입력되기 전까지 정수를 입력 받아

(즉, 종료 조건은 정수 0 입력), 이 중에서 가장 작은 값 구하기



- ✓ 처음입력정수는 0이 아니라고 가정하고, 정수 0은 비교대상에서 제외
- ✓ hint) 반복문 안에서 조건문 사용

입력 예시 1

10 2 15 7 0

출력 예시 1

2

입력 예시 2

10 2 -15 7 0

출력 예시 2

-15



# 학습 정리

- **반복문**은 특정 조건을 만족할 때까지 계속 반복하여 실행하는 문장을 의미하며, 간혹 루프(loop)라고 불리우기도 함
- 일반적으로 **while 문**과 do-while 문은 반복 횟수가 정해지지 않는 경우에 사용하고, for 문은 반복하는 횟수가 정해진 경우에 사용함
- **while 문**은 조건식을 검사한 후, 참이면 코드(반복할 문장)을 실행함