



[6주차 2강] 반복문(2)



학습 내용

6.3 do-while 문

6.4 for 문



학습 목표

6.3 do-while 문의 사용법을 익힌다.

6.4 for 문의 사용법을 익힌다.





6.3 do-while 문

6.4 for 문

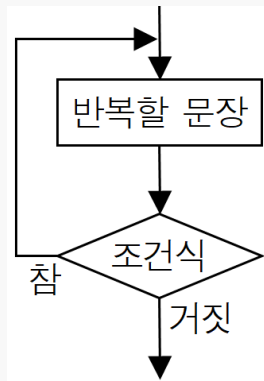




do-while 문

- 반복할 문장을 실행한 후에 **조건식** 검사
- 반복문 내에 있는 문장을 **최소한 한 번 실행하고자 할 때 유용**
- 주의) 마지막에 **세미콜론(;)을 반드시** 써야 함
- 구문

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



```
do {  
    sum += i;  
    i++;  
} while( i <= 5 );
```

```
while( i <= 5 ) {  
    sum += i;  
    i++;  
}
```

while 문과 비교

[프로그램 3] 프로그램 1을 do-while 문으로 작성하기

- do-while 문을 이용하여 1부터 5까지의 합 구하기

```
int i, sum;

sum = 0;           // 합
i = 1;             // 더하는 수

do {
    sum += i;       // i를 sum에 더하고
    i++;           // i값 1 증가
} while ( i <= 5 ); // 반복 조건 검사

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

[프로그램 4] 프로그램 2를 do-while 문으로 작성하기

- 0 또는 음수가 입력되기 전까지 정수를 입력 받아,
입력 받은 정수의 합(종료 조건 0 또는 음수는 합에서 제외) 구하기
- [프로그램 2]의 코드와 무엇이 다른지 비교해보자.

1 2 3 0 4

```
int x = 0, sum = 0;

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

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

```
int x, sum = 0;

printf("정수를...");
scanf("%d", &x);

while ( x > 0 ) {
    sum += x;
    printf("정수를... ");
    scanf("%d", &x);
}

printf("입력된...", sum);
```

[프로그램 2]

실습하기



[예제 6.5]

정수를 입력 받아, do-while 문을 이용하여
그 정수에 해당하는 구구단 출력하기



입력

5

출력

5 * 1 = 5

5 * 2 = 10

...

5 * 9 = 45

실습하기



[예제 6.6]
do-while 문을 이용하여 영어 소문자 순서대로 출력하기



실행 결과

abcdefghijklmnopqrstuvwxyz

실습하기



[예제 6.7]

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 6.8]

10 이상의 정수를 입력 받아, 각 자릿수의 합 출력하기



✓ 예) $6235 \rightarrow 6+2+3+5 = 16$

✓ hint) 나머지 연산자(%)와 나누기 연산자(/) 활용

입력 예시

6235

출력 예시

16



6.3 do-while 문

6.4 for 문



for 문 예제 훑어보기(while 문과 비교하여)

- 1부터 5까지의 합 계산

```
sum = 0;
for( i = 1 ; i <= 5 ; i++ ) {
    sum += i;
}
```

```
i = 1, sum = 0;
while( i <= 5 ) {
    sum += i;
    i++;
}
```

✓ for(i = 1 ; i <= 5 ; i++) 의미

- ① i의 값을 1부터 시작해서(초기식)
- ② i가 5보다 작거나 같은 동안(조건식)
- ③ i의 값을 1씩 증가시키면서(증감식), 반복

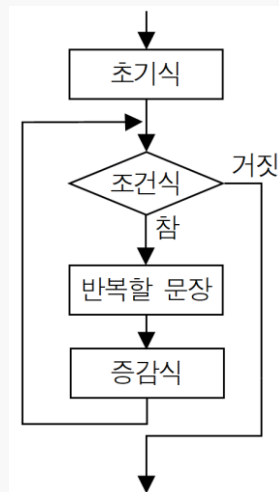
✓ 반복과 관련된 수식을 하나로 모아 코드의 가독성을 높인 형태

for 문

- 일반적으로 반복하는 횟수가 정해진 경우에 사용
- 초기식, 조건식, 증감식으로 구성되고, **세미콜론**으로 구분
 - ✓ 초기식은 처음 한 번만 수행
 - ✓ 이후, 조건식 → 반복할 문장 → 증감식이 반복적으로 수행됨
- 구문

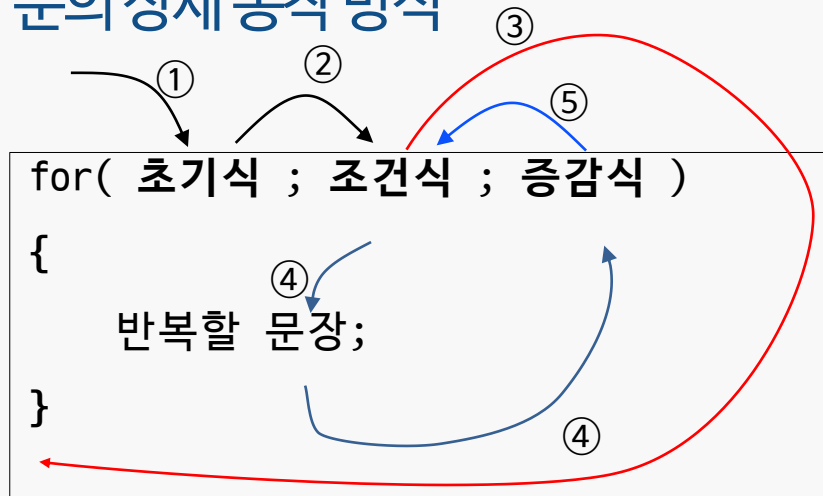
```
for(초기식 ; 조건식 ; 증감식) {  
    반복할 문장;  
}
```

```
for( i = 1 ; i <= 5 ; i++ ) {  
    sum += i;  
}
```





for 문의 상세 동작 방식



- ① 초기식을 수행한다.
- ② 조건식을 검사한다.
- ③ **조건식의 값이 거짓이면** for 문의 실행이 종료된다.
- ④ **조건식의 값이 참이면** 문장이 실행된다.
- ⑤ 증감을 실행하고 ②로 돌아간다.

for 문과 while 문의 형식 비교

```
①  
for(초기식; ② 조건식; ③ 증감식) {  
    ④ 반복할 문장;  
}
```

```
①  
초기식; ②  
while( 조건식 ) {  
    ④ 반복할 문장;  
    ③ 증감식;  
}
```

- ✓ 위 비교는 for 문의 동작을 이해시키기 위함이다.
- ✓ for 문을 여러 번 연습하면 자연스럽게 체득된다.



[프로그램 5] 프로그램 1을 for 문으로 작성하기

- for 문을 이용하여 1부터 5까지의 합 구하기

```
int i, sum;

sum = 0;           // 합

for( i = 1 ; i <= 5 ; i++ ) { // i를 1~5까지
                                // 1씩 증가시키면서
    sum += i;           // i를 sum에 더하기
}

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

- 프로그램의 실행됨에 따라 제어 흐름과 변수가 어떻게 변하는지 따져보자.
(교재 p.157의 수행과정 표 참조)

- ## for 문 수행과정 확인하기
- for 문의 끝에 **출력문**을 추가하여 수행과정을 눈으로 확인해보자.

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

실행 결과

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

- ✓ while 문의 수행과정 출력 결과와 동일한가?
- ✓ 다르다면 왜 다른지 이해할 수 있는가?

실습하기



다음 각 문장에 해당하는 for 문의 첫 줄을 작성해보자.



1. 변수 i 를 2부터 10까지, 2씩 증가시키면서 반복
2. 변수 i 를 10부터 시작하여,
1씩 감소시키면서 i 가 0보다 큰 동안 반복
3. 변수 i 의 초기값은 0이고, i 값을 $i*i+2$ 로 바꾸면서,
 i 가 50보다 작은 동안 반복



for 문의 자주 활용되는 형태

- ✓ for(; i <= 5 ; i++) ⇨ 빈 초기식
- ✓ for(int i = 1 ; i <= 5 ; i++) ⇨ 변수 선언 및 초기화
단, i는 for 문 안에서만 사용 가능
- ✓ for(sum=0, i=1 ; i <= 5 ; i++) ⇨ 여러 개의 초기식 (콤마로 구분)
- ✓ for(i=0 ; i <= 5 ;) ⇨ 빈 증감식
- ✓ for(i=0, j=0 ; i <= 5 ; i++, j++) ⇨ 여러 개의 초기식과 증감식
- ✓ for(i=0 ; ; i++) ⇨ 빈 조건식 (무한 루프)
이 경우 조건식의 결과는 항상 참으로 간주
- ✓ (문법적으로는 초기식, 조건식, 증감식에는 어떤 수식도 가능)

실습하기



[예제 6.9]
정수 n 을 입력 받아, $n! = 1*2*\dots*n$ 계산하기



입력

5

출력

120

실습하기



[예제 6.10]
문자와 정수를 입력 받아, 문자를 정수 개수 만큼 출력하기



입력

a 6

출력

☐ a a a a a a

☐는 공백을 의미

실습하기



[예제 6.11]

1부터 10까지의 홀수의 합, 짝수의 합 각각 출력하기



✓ hint) 모든 홀수를 구하려면,
1부터 시작하여 몇 씩 증가시키면 될까?

실행 결과

25

30

실습하기



[예제 6.12]

0~6사이의 정수를 입력 받아, 그 정수에 해당하는 요일 (0이면 일요일, 1이면 월요일, ..., 6이면 토요일)에 1일이 시작하는 달력 출력하기



- 한 달은 31일로 가정
- 각 날짜는 3개의 칸에 출력 (3장 printf의 서식 플래그 활용)

입력

4

출력

	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	26	27	28	29	30	31		

실습하기



[예제 6.12]
달력 출력하기 hint



- ✓ hint 1) 1일 앞의 빈 칸은 따로 출력
- ✓ hint 2) 각 주의 줄 바꿈
 - ✓ 매주 토요일에 해당하는 날짜 출력 후 개행 문자 출력
 - ✓ 토요일인지는 어떻게 판단? 몇 가지 예를 통해 규칙 찾기

Tip: 문제가 복잡할 땐 단계적으로 해결하자.

학습 정리

- **do-while 문**은 먼저 코드(반복할 문장)를 한번 실행시키고 그 다음 조건식을 검사함(조건식 다음에 **세미콜론**을 반드시 써야 함)
- **for 문**은 초기식, 조건식, 증감식으로 구성되고, 세미콜론으로 구분함
- **초기식**은 반복 조건에 관련된 식을 초기화하는 문장임
- **조건식**은 반복 여부를 검사하는 문장임
- **증감식**은 제어 변수의 값을 변경하는 문장임