

반복문과 선택문

제주대학교 컴퓨터공학과

변영철 교수

(ycb@jejunu.ac.kr)

이 장을 공부하면

- 반복문과 선택문을 이해할 수 있다.
- C 프로그램을 작성 시 반복문과 선택문을 활용하여 응용 프로그램을 작성할 수 있다.

while 문

```
while (a < 10) ← 조건식
{
    a = a * 2; ← 실행문
}
```

- 조건식 먼저 검사
- 참이면 아래 실행문 실행
- 다시 위로 올라가서 **반복**

while 문

```
int main(void)
{
    int a = 1;

    while (a < 10)
    {
        a = a * 2;
    }
    printf("a : %d\n", a);

    return 0;
}
```

while 문

조건식이 참이 아니면 while문 끝나고
아래 문장 실행

조건식이 거짓이면
반복문 이후의
문장으로
건너뛴다.

```
while (a < 10)   조건식
{
    a = a * 2;   반복할 실행문
}
printf("a : %d\n", a);
```

while 문

들여쓰기 하면 코드를 이해하기 쉬움 (가독성)

```
while (a < 10)
{
    들여쓰기 → a = a * 2;
}
```

for 문

아래와 같이 초기식, 조건식, 증감식으로 반복

```
for ( i = 0; i < 3; i ++ )  
{  
    a = a * 2;  
}
```

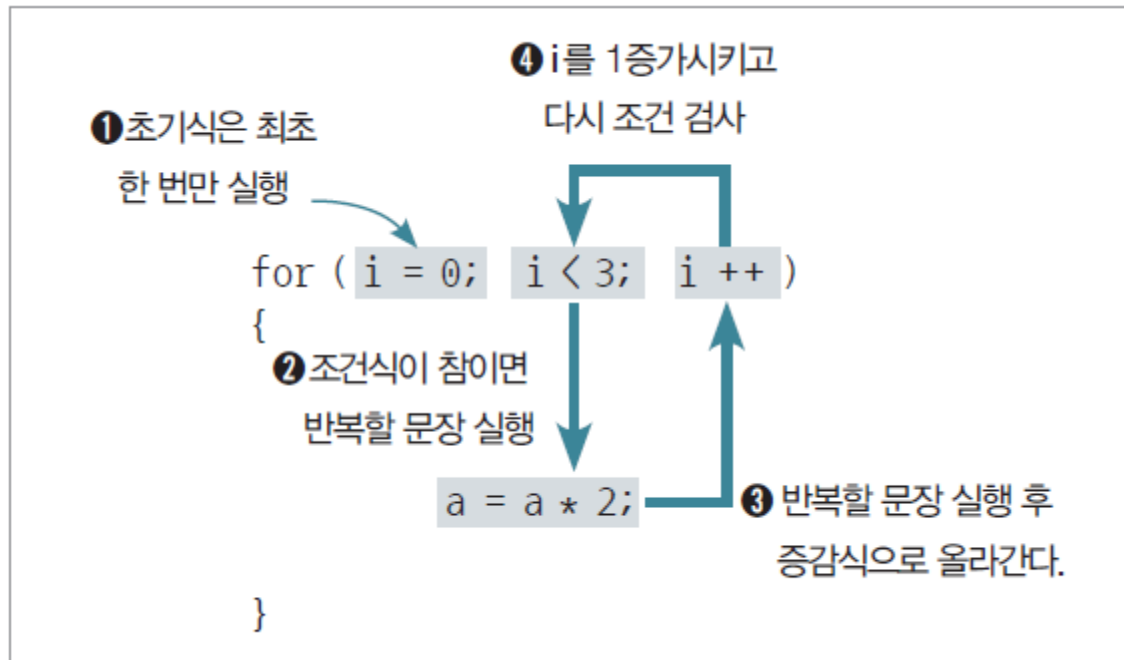
The diagram illustrates the components of a for loop. The code is as follows:

```
for ( i = 0; i < 3; i ++ )  
{  
    a = a * 2;  
}
```

Labels and arrows indicate the parts of the loop:

- 초기식** (Initialization): Points to `i = 0;`
- 조건식** (Condition): Points to `i < 3;`
- 증감식** (Increment/Decrement): Points to `i ++`
- 실행문** (Statement): Points to `a = a * 2;`

for 문



for 문

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
    int a = 1;
```

```
    for (int i = 0; i < 3; i++) {
```

```
        a = a * 2;
```

```
    }
```

```
    printf("a = %d\n", a);
```

```
    getchar();
```

```
}
```

1. 초기식 - 변수 i 초기화. 딱 한 번 실행
2. 조건식을 검사하여 결과가 참이면 블록 안으로
3. 반복문 $a = a * 2$ 실행
4. 증감식으로 i값 1 증가
5. 다시 조건식 검사하여 참이면 블록 안으로, 거짓이면 빠져나옴

실행
결과 a : 8

for 문



```
for ( i = 0; i < 10; i++ )  
{  
    printf("Be happy!\n");  
}
```

```
i = 0;  
while ( i < 10 )  
{  
    printf("Be happy!\n");  
    i++;  
}
```

do~while 문

```
do  
{  
    a = a * 2; ← 반복할 문장  
} while (a < 10); ← 반복 조건
```

일단 반복할 문장 수행 후 조건은 나중에 검사

do~while 문

```
#include <stdio.h>

int main(void)
{
    int a = 1;

    do
    {
        a = a * 2;
    } while(a < 10);
    printf("a : %d\n", a);

    return 0;
}
```

반복문 중첩



중첩 반복문 예

```
for ( i=0; i<10; i++ )  
{  
    for ( j=0; j<10; j++ )  
    {  
        반복할 문장;  
    }  
}
```

i-for문이 10번 반복되고 j-for문이
10번 반복되므로 반복할 문장은
100번 반복된다.

분기문 사용 예

```
while ( 1 )  
{  
    if ( 조건식 1 ) break;  
    if ( 조건식 2 ) continue;  
    반복할 문장;  
}
```

조건식 1이 참이면 반복문을 끝낸다.
조건식 2가 참이면 반복할 문장을
건너뛰고 처음부터 다시 반복한다.

반복문 중첩

```
int main(void)
{
    int i, j;

    for(i = 0; i < 3; i++)
    {
        for(j = 0; j < 5; j++)
        {
            printf("*");
        }
        printf("\n");
    }

    return 0;
}
```

실행
결과

```
*****
*****
*****
```

반복문 활용 (구구단)



$$2 * 1 = 2$$

$$2 * 2 = 4$$

$$2 * 3 = 6$$

$$2 * 4 = 8$$

$$2 * 5 = 10$$

$$2 * 6 = 12$$

$$2 * 7 = 14$$

$$2 * 8 = 16$$

$$2 * 9 = 18$$

반복문 활용 (구구단)



```
for ( j = 1; j <= 9; j++ )  
{  
    printf("2 * %d = %d\n", j , 2 * j);  
}
```


반복문 활용 (구구단)

```
for ( i = 0; i < 8; i++ )           // 8번 반복
{
    for ( j = 1; j <= 9; j++ )
    {
        printf("2 * %d = %d\n", j, 2 * j); ← 2단만 출력
    }
}
```

반복문 활용 (구구단)

```
for ( i = 2; i <= 9; i++ )
{
    for ( j = 1; j <= 9; j++ )
    {
        printf("%d * %d = %d\n" , i, j, i * j ) ;
    }
}
```

반복문 제어 (break, continue)

```
int main(void)
{
    int i;                // 반복 횟수를 세기 위한 제어 변수
    int sum = 0;          // 1부터 10까지의 합을 누적할 변수


    for(i = 1; i <= 10; i++) // i는 1부터 10까지 증가하면서 10번 반복
    {
        sum += i;           // i값을 sum에 누적
        if(sum > 30) break;  // 누적한 값이 30 보다크면 반복문을 끝낸다.
    }
    printf("누적한 값 : %d\n", sum);
    printf("마지막으로 더한 값 : %d\n", i);

    return 0;
}
```

실행 결과
누적한 값 : 36
마지막으로 더한 값 : 8

반복문 제어 (break, continue)

```
for ( ... )  
{  
    for ( ... )  
    {  
        ...  
        if ( 조건식 ) break;  
    }  
}
```



안쪽 for문 하나만 탈출

반복문 제어 (break, continue)

```
for ( i = 1; i <= 100; i++)  
{  
    if ( (i % 3) == 0 )  
    {  
        continue;  
    }  
    sum += i;  
}
```

i가 3의 배수면 `sum += i` 문장을 건너뛰고
블록 끝으로 간 후에 다시 반복한다.

무한 반복



```
while ( 1 )  
{  
    printf("Be happy!\n");  
}
```

```
for ( ; ; )  
{  
    printf("Be happy!\n");  
}
```