

반복 실행하는 반복문(1)

- while 문, do ~ while 문 -

성공회대학교 IT융합자율학부
소프트웨어공학전공
홍 성 준



while 문에 의한 문장의 반복

◎ 반복문이란?

- 한 개 이상의 명령을 두 번 이상 반복 실행하기 위해 구성하는 문법
(예) 문자열 'hello world'를 출력하는 printf() 함수 호출을 총 열 번 실행해주세요.

◎ 반복문의 종류

- while
- do ~ while
- for

while 문에 의한 문장의 반복

◎ while 문

- 특정 조건을 주고, 그 조건이 만족하는 동안(참인 동안) 특정 명령을 반복해서 실행하는 구문
- SimpleWhile.c

```
int main(void)
{
    int num=0;
    while(num<5)
    {
        printf("Hello world! %d \n", num);
        num++;
    }
    return 0;
}
```

while 반복문

반복할 코드 블록

```
Hello world! 0
Hello world! 1
Hello world! 2
Hello world! 3
Hello world! 4
```

- 소괄호 안에 반복 조건을 명시하고, 중괄호 안에 반복할 구문을 입력
- 무한 루프를 만들지 않기 위해서는 반복 문을 벗어나기 위한 연산이 필요함



while 문에 의한 문장의 반복

◎ while 문 (cont.)

- 블록(중괄호)을 사용할 때는 코드의 가독성을 높이기 위해 일반적으로 들여쓰기를 사용

들여쓰기를 하지 않은 것

```
int main(void)
{
  int num=0;
  while(num<5)
  {
    printf("Hello world! %d \n", num);
    num++;
  }
  return 0;
}
```

들여쓰기를 한 것

```
int main(void)
{
  int num=0;
  while(num<5)
  {
    printf("Hello world! %d \n", num);
    num++;
  }
  return 0;
}
```

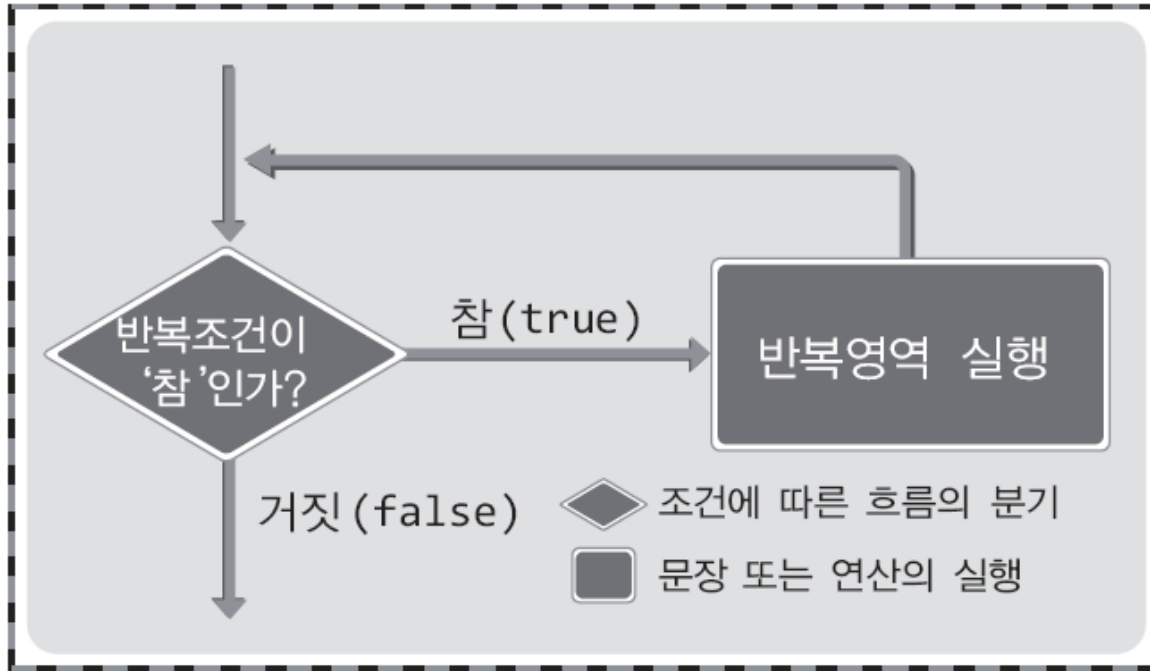
- 반복할 명령이 하나라면 중괄호를 생략 가능

```
while(num<5)
  printf("Hello world! %d \n", num++);
```

```
while(num<5)
  printf("Hello world! %d \n", num), num++;
```

while 문에 의한 문장의 반복

◎ while 문 (cont.)



```
int main(void)
{
    int num=0;
    while(num<3) // 3회 반복
    {
        printf("Hello world! %d \n", num);
        num++;
    }
    . . .
}
```



while 문에 의한 문장의 반복

◎ 구구단 출력 하기

- NineNineDan.c

```
int main(void)
{
    int dan=0, num=1;
    printf("몇 단? ");
    scanf("%d", &dan);

    while(num<10)
    {
        printf("%d×%d=%d \n", dan, num, dan*num);
        num++;
    }
    return 0;
}
```

몇 단? 7

$7 \times 1 = 7$

$7 \times 2 = 14$

$7 \times 3 = 21$

$7 \times 4 = 28$

$7 \times 5 = 35$

$7 \times 6 = 42$

$7 \times 7 = 49$

$7 \times 8 = 56$

$7 \times 9 = 63$

while 문에 의한 문장의 반복

◎ 무한 루프

- 반복문의 탈출 조건이 성립하지 않아 무한 반복 실행하는 반복문

```
while( 1 )  
{  
    printf("%d×%d=%d \n", dan, num, dan*num);  
    num++;  
}
```

- 숫자 1은 '참'을 의미하므로 반복 조건이 1이면, 항상 '참'으로 무한 반복을 시행
- 실수로 무한 루프를 만들기도 하지만, break 문과 함께 유용하게 사용함



while 문에 의한 문장의 반복

◎ while 문의 중첩

- while 문 안에 while 문을 작성하는 것과 같이 반복문을 중첩하여 구현
- TwoNineNine.c

```
int main(void)
{
    int cur=2;
    int is=0;

    while(cur<10) // 2단부터 9단까지 반복
    {
        is=1; // 새로운 단의 시작을 위해서
        while(is<10) // 각 단의 1부터 9의 곱을 표현
        {
            printf("%d×%d=%d \n", cur, is, cur*is);
            is++;
        }
        cur++; // 다음 단으로 넘어가기 위한 증가
    }
    return 0;
}
```

바깥쪽 while문

안쪽 while문



do ~ while 문에 의한 문장의 반복

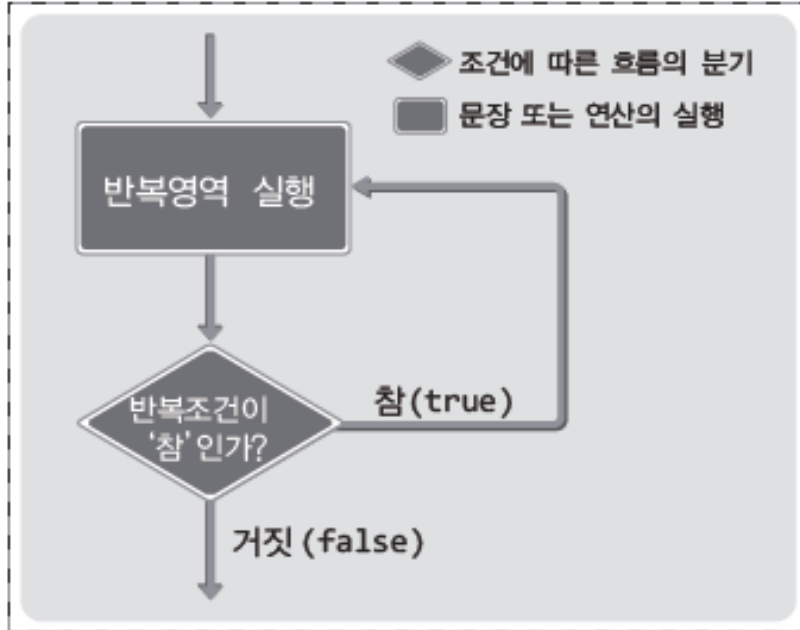
◎ do ~ while 문

- 반복 영역을 최소한 한 번은 실행하고, 특정 조건이 만족하는 동안 특정 명령을 반복해서 실행하는 구문
- 반복 조건이 앞에 있는 while 문과는 달리 반복 조건을 뒤에서 검사함

```
do
{
    printf("Hello world! \n");
    num++;
} while(num<3);
```

do ~ while 문에 의한 문장의 반복

◎ do ~ while 문 (cont.)



- while 문과는 반복 조건의 검사 위치가 다르며, 이로 인해 반복 영역을 최소한 1회 이상 실행한다는 점이 while문과 다름

do ~ while 문에 의한 문장의 반복

◎ do ~ while 문 (cont.)

```
while(num<10)
{
    printf("%d×%d=%d \n", dan, num, dan*num);
    num++;
}
```



동일한 횟수를 반복하는 반복문들

```
do
{
    printf("%d×%d=%d \n", dan, num, dan*num);
    num++;
} while(num<10);
```

do ~ while 문에 의한 문장의 반복

◎ do ~ while 문이 자연스러운 상황

- 적어도 한 번 이상 반복 구문을 실행해야 하는 경우 do ~ while 문으로 구현
- UsefulDoWhile.c

```
int main(void)
{
    int total=0, num=0;
    do
    {
        printf("정수 입력(0 to quit): ");
        scanf("%d", &num);
        total += num;
    }while(num!=0);
    printf("합계: %d \n", total);
    return 0;
}
```

```
정수 입력(0 to quit): 1
정수 입력(0 to quit): 2
정수 입력(0 to quit): 3
정수 입력(0 to quit): 4
정수 입력(0 to quit): 5
정수 입력(0 to quit): 0
합계: 15
```