




누구나 즐기는 C언어 콘서트

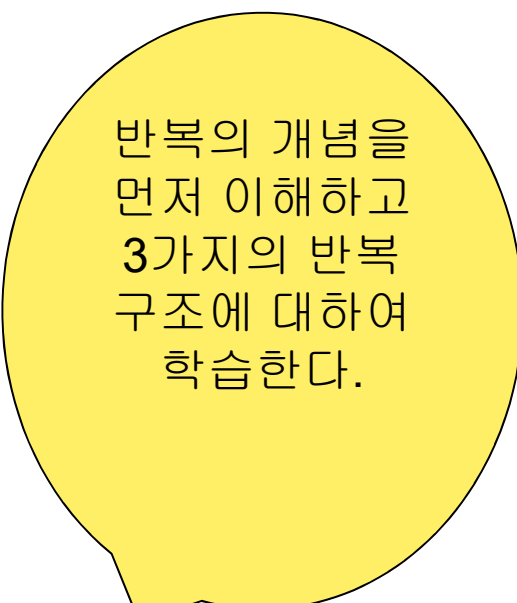
제6장 반복문





이번 장에서 학습할 내용

- 
- 반복의 개념 이해
 - while 반복문
 - do-while 반복문
 - for 반복문
 - break와 continue문



반복의 개념을
먼저 이해하고
3가지의 반복
구조에 대하여
학습한다.

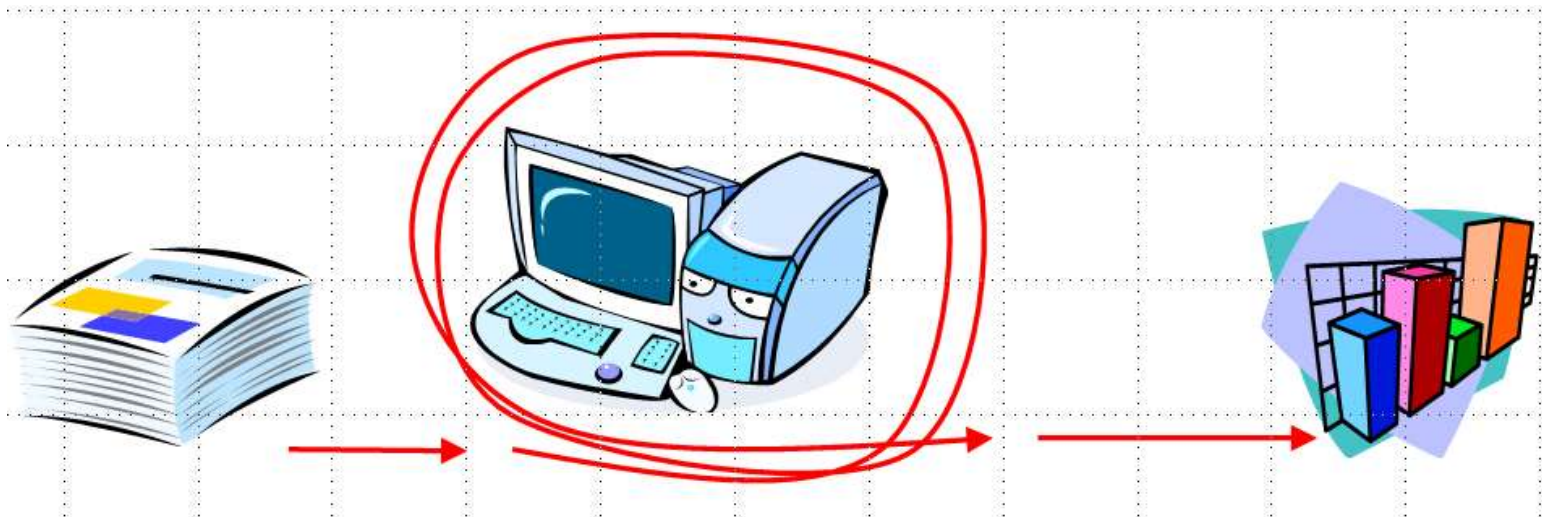




반복문

Q) 반복 구조는 왜 필요한가?

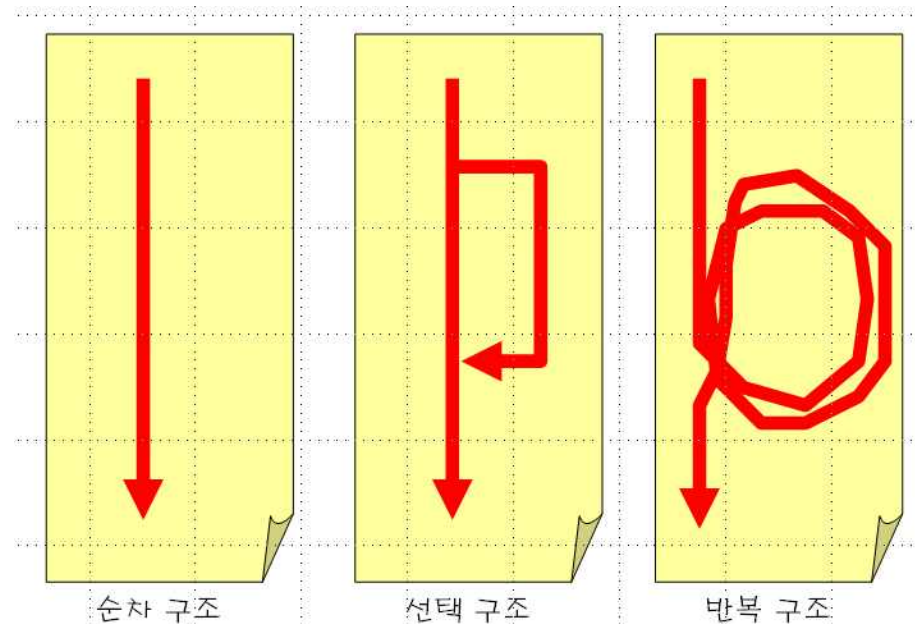
A) 같은 처리 과정을 되풀이하는 것이 필요하기 때문이다. 학생 30명의 평균 성적을 구하려면 같은 과정을 30번 반복하여야 한다.





프로그램의 흐름을 제어하는 방법

- 순차 구조
 - 차례대로 실행
- 선택 구조
 - 조건을 검사하여 여러 개의 실행 경로 중에서 하나를 선택
- 반복 구조
 - 조건이 만족될 때까지 반복





마일을 미터로 바꾸는 프로그램

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

```
int main(void)
```

```
{
```

```
    int meter;
```

```
    meter = 0 * 1609;
```

```
    printf("0 마일은 %d미터입니다\n", meter);
```

```
    meter = 1 * 1609;
```

```
    printf("1 마일은 %d미터입니다\n", meter);
```

```
    meter = 2 * 1609;
```

```
    printf("2 마일은 %d미터입니다\n", meter);
```

```
    return 0;
```

```
}
```

같은
처리
과정 #1

같은
처리
과정 #2

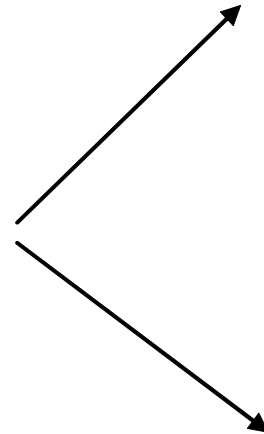
같은
처리
과정 #3



반복문의 종류



반복문



while



10kg이 빠질 때까지 반복하세요



for



100번 반복하세요,



중간 점검

1. 프로그램에 반복 구조가 필요한 이유는 무엇인가?
2. 반복문에는 _____, _____문이 있다.





이번 장에서 학습할 내용

- 반복의 개념 이해
- while** 반복문
- do-while 반복문
- for 반복문
- break와 continue문

반복의 개념을
먼저 이해하고
3가지의 반복
구조에 대하여
학습한다.

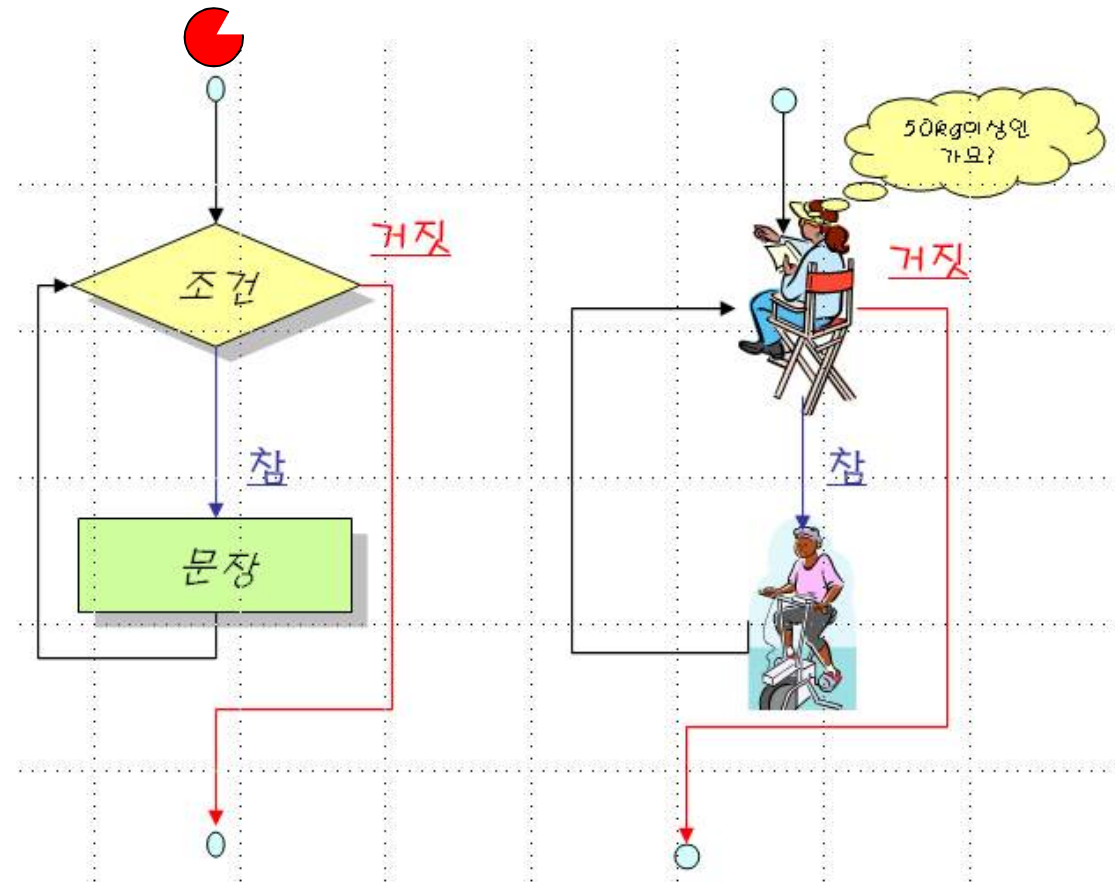


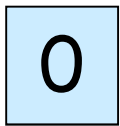


while 문

- 주어진 조건이 만족되는 동안 문장들을 반복 실행한다.

while(조건식)
문장;





i

예제

```
#include <stdio.h>

int main(void)
{
    int meter;
    int i = 0;

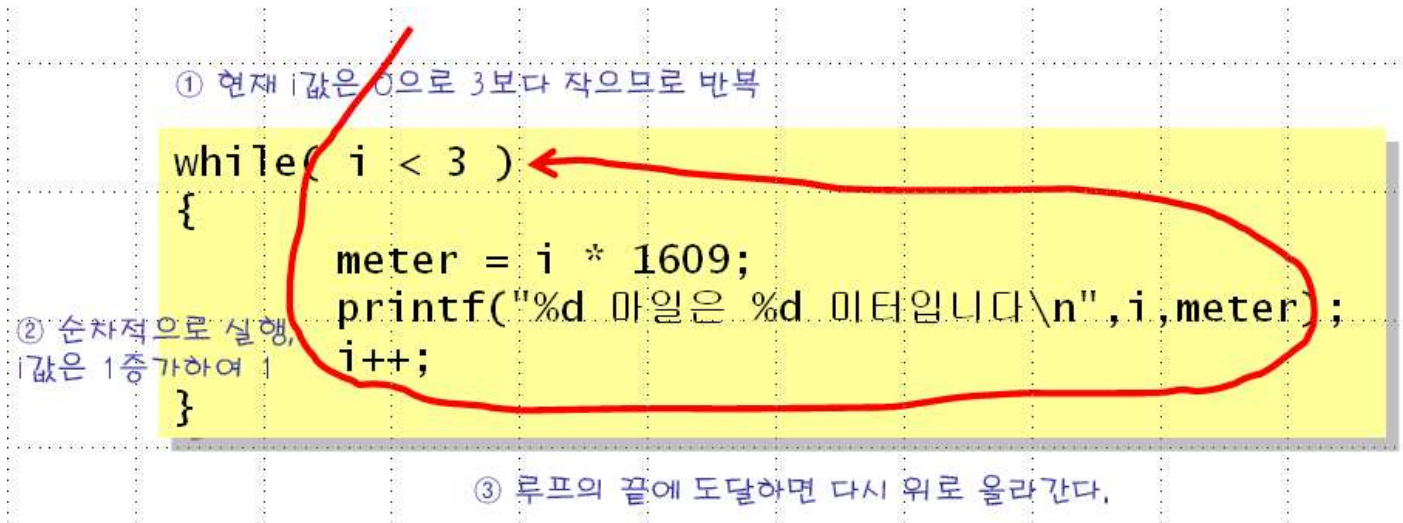
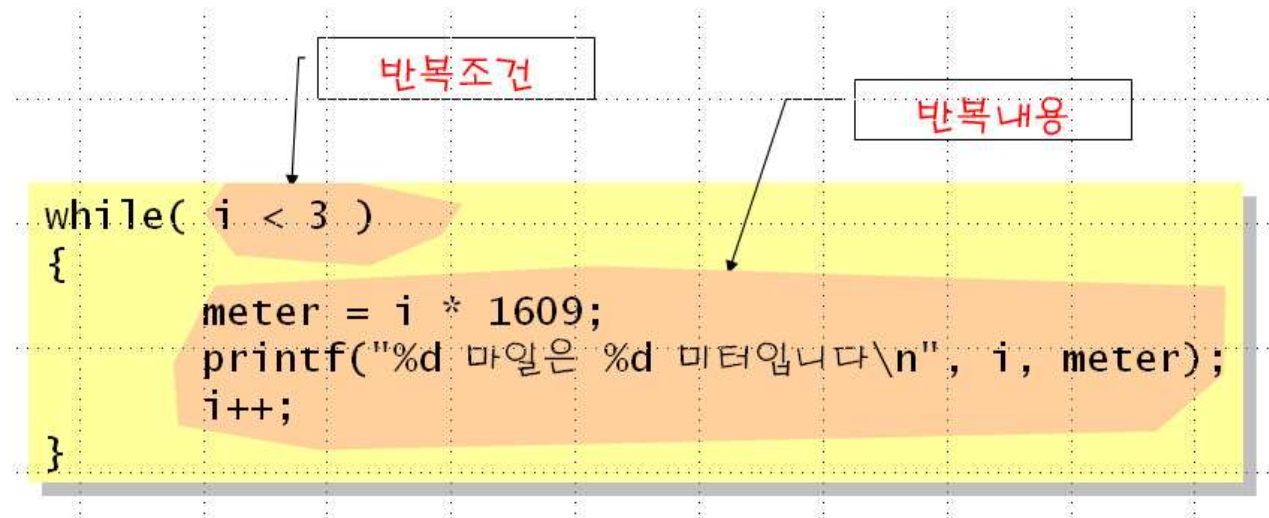
    while(i < 3)
    {
        meter = i * 1609;
        printf("%d 마일은 %d 미터입니다\n", i, meter);
        i++;
    }
    return 0;
}
```



0 마일은 0 미터입니다
1 마일은 1609 미터입니다
2 마일은 3218 미터입니다



반복문의 실행 과정





반복문의 실행 과정

④ 현재 i값은 1로 3보다 작으므로 반복

```
while( i < 3 )  
{  
    meter = i * 1609;  
    printf("%d 마일은 %d 미터입니다\n",i,meter);  
    i++;  
}
```

⑤ 순차적으로 실행,
i값은 1증가하여 2

⑥ 루프의 끝에 도달하면 다시 위로 올라간다.

⑦ 현재 i값은 2로 3보다 작으므로 반복

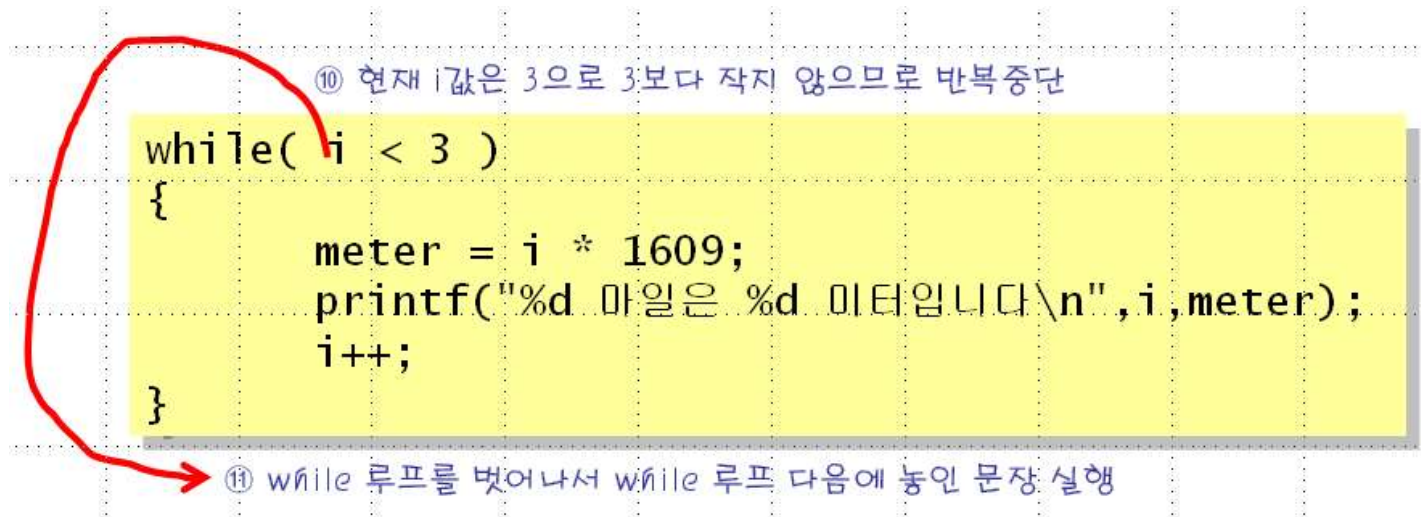
```
while( i < 3 )  
{  
    meter = i * 1609;  
    printf("%d 마일은 %d 미터입니다\n",i,meter);  
    i++;  
}
```

⑧ 순차적으로 실행,
i값은 2증가하여 3

⑨ 루프의 끝에 도달하면 다시 위로 올라간다.



반복문의 실행 과정



반복횟수	i의 값	i < 3	반복여부
첫번째 반복(시작위치)	0	참(1)	반복
두번째 반복(시작위치)	1	참(1)	반복
세번째 반복(시작위치)	2	참(1)	반복
네번째 반복(시작위치)	3	거짓(0)	중지



예제 #1

```
// while 문을 이용한 구구단 출력 프로그램
```

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

```
int main(void)
```

```
{
```

```
    int n;
```

```
    int i = 1;
```

```
    printf("구구단 중에서 출력하고 싶은 단을 입력하시오: ");
```

```
    scanf("%d", &n);
```

```
    while (i <= 9)
```

```
    {
```

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

```
        i++;
```

```
    }
```

```
    return 0;
```

```
}
```



구구단 중에서 출력하고 싶은 단을 입력하시오: 9

9*1 = 9

9*2 = 18

9*3 = 27

....

9*9 = 81



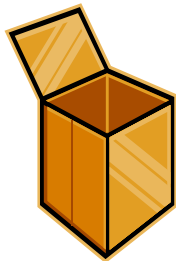
예제 #2

- 1부터 n 까지의 합을 계산하는 프로그램

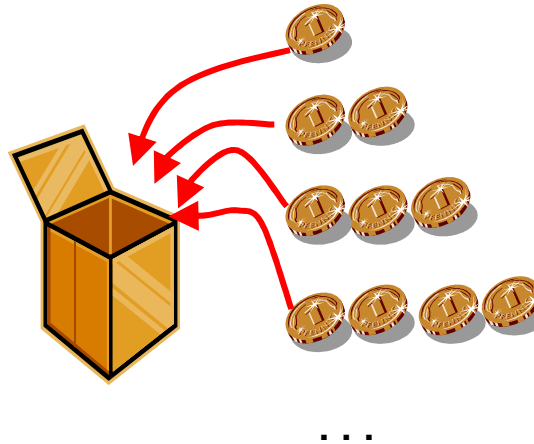
$$1 + 2 + 3 + \dots + n$$

- n 이 무엇이 될지 모르는 경우라면 다음과 같이 계산

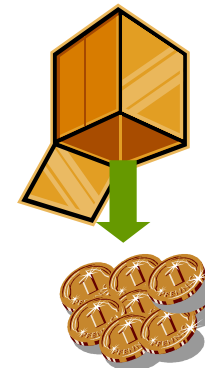
① 빈통을 준비한다.



② 통에 1부터 n 까지를 넣는다.



③ 통에 들어 있는 동전의 개수를 출력한다.





예제

```
#include <stdio.h>

int main(void)
{
    int i, sum;
    i = 1;
    sum = 0;
    while (i <= 1000)
    {
        sum += i;
        i++;
    }
    printf("합은 %d입니다.\n", sum);
    return 0;
}
```

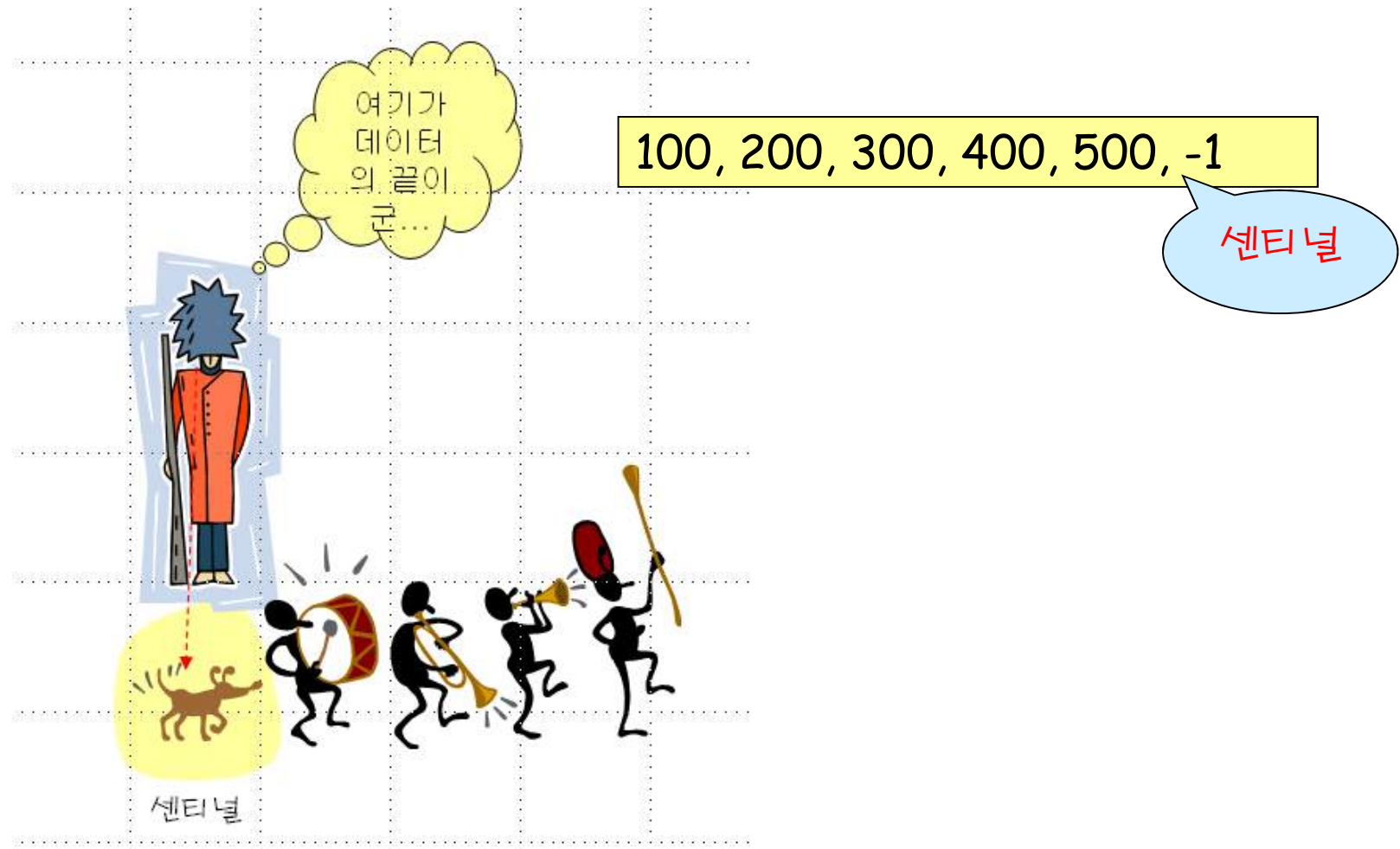


합은 500500입니다.



센티널(보초값의 이용)

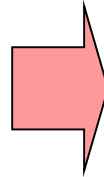
- 센티널: 입력되는 데이터의 끝을 알리는 특수한 값





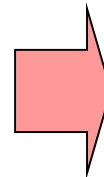
성적들의 평균을 구하는 문제

- 성적의 평균을 구한다.



1. 필요한 변수들을 초기화한다.
2. 성적을 입력받아서 합계를 구하고 성적의 개수를 센다.
3. 평균을 계산하고 화면에 출력한다.

1. 필요한 변수들을 초기화한다.

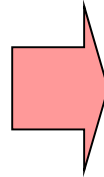


- (1) **sum**을 0으로 초기화한다.
- (2) **n**을 0으로 초기화한다.
- (3) **grade**를 0으로 초기화한다.



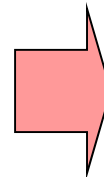
성적들의 평균을 구하는 문제

2. 성적을 입력받아서 합계를 구하고 성적의 개수를 센다.



while 성적이 0보다 작지 않으면
(1) 사용자로부터 성적을 읽어서 **grade**에 저장한다.
(2) **sum**에 이 점수를 누적한다.
(3) **n**을 하나 증가한다.

3. 평균을 계산하고 화면에 출력한다.



(1) **sum**을 **n**으로 나누어서 **average**에 저장한다.
(2) **average**를 화면에 출력한다.



센티넬 예제 1/2

```
// while 문을 이용한 성적의 평균 구하기 프로그램
#include <stdio.h>

int main(void)
{
    int grade, n;
    float sum, average;

    // 필요한 변수들을 초기화한다.
    n = 0;
    sum = 0;
    grade = 0;

    printf("성적 입력을 종료하려면 음수를 입력하시오\n");
```



센티넬 예제 2/2

// 성적을 입력받아서 합계를 구하고 학생 수를 센다.

```
while (grade >= 0)
```

```
{
```

```
    printf("성적을 입력하시오: "),  
    scanf("%d", &grade);
```



```
    sum += grade;  
    n++;
```

```
}
```

```
sum = sum - grade; // 마지막 데이터를 제거한다.
```

```
n--; // 마지막 데이터를 제거한다.
```

```
// 평균을 계산하고 화면에 출력한다.
```

```
average = sum / n;
```

```
printf("성적의 평균은 %f입니다.\n", average);
```

```
return 0;
```

```
}
```

성적 입력을 종료하려면 음수를
입력하시오

성적을 입력하시오: 10

성적을 입력하시오: 20

성적을 입력하시오: 30

성적을 입력하시오: 40

성적을 입력하시오: 50

성적을 입력하시오: -1

성적의 평균은 30.000000입니다.



예제

- 두수의 최대 공약수 구하기
- 유클리드 알고리즘

- ① 두 수 가운데 큰 수를 x , 작은 수를 y 라 한다.
- ② y 가 0이면 공약수는 x 와 같다.
- ③ $r \leftarrow x \% y$
- ④ $x \leftarrow y$
- ⑤ $y \leftarrow r$
- ⑥ 단계 ②로 되돌아간다.



예제

// while 문을 이용한 최대 공약수 구하기 프로그램

#include <stdio.h>

int main(void)
{

int x, y, r;

printf("두개의 정수를 입력하시오(큰수, 작은수): ");
scanf("%d%d", &x, &y);

while (y != 0)

{

r = x % y;

x = y;

y = r;

}

printf("최대 공약수는 %d입니다.\n", x);

return 0;

}



두개의 정수를 입력하시오(큰 수, 작은 수): 12 8
최대 공약수는 4입니다.



if 문과 while 문의 비교

- if문은 while 문으로 변경할 수 있다.
- while 문도 if문으로 변경할 수 있다.

```
if( 조건 )  
{  
    ...  
    ...  
}
```

조건이 만족되면
한번만 실행
된다.

```
while( 조건 )  
{  
    ...  
    ...  
}
```

조건이 만족되면
여러 번 반복 실행
된다.



while 문에서 주의할 점

```
int i = 1;
while(i < 10)
{
    printf("반복중입니다\n");
    i--;
}
```

변수가 증가 아니라 감소

```
int i = 0;
while(i < 3)
    printf("반복중입니다\n");
    i++;
```

반복 루프에 포함되어 있지 않다.

```
int i = 0;
while(i < 3) ;
{
    printf("반복중입니다\n");
    i++;
}
```

조건뒤에 ;이 있음



중간 점검

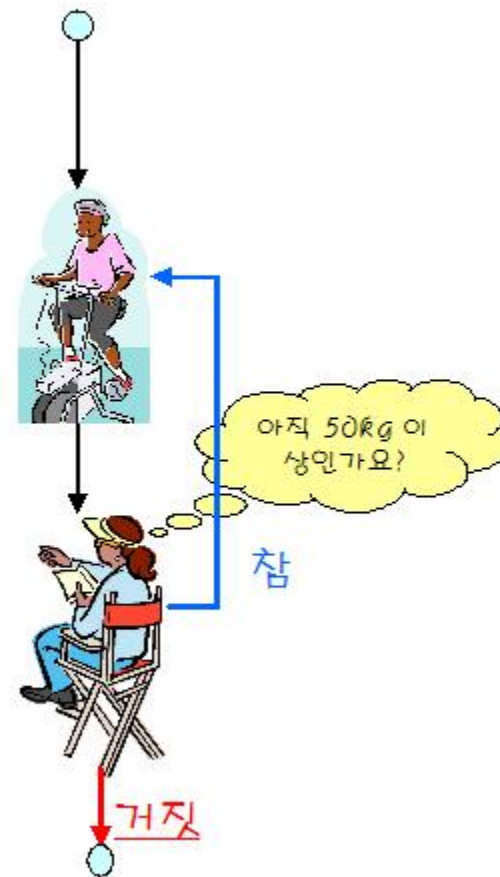
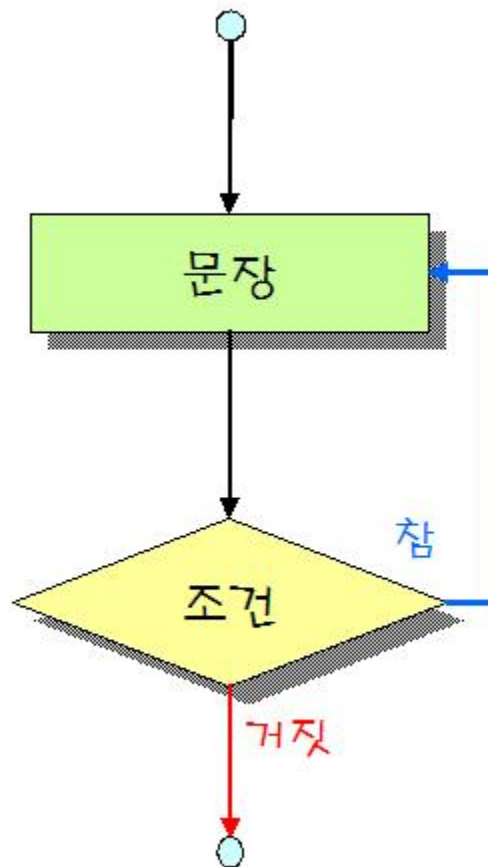
1. **if** 문과 **while** 문을 비교하여 보라. 똑같은 조건이라면 어떻게 동작하는가?
2. **while** 루프를 이용하여 무한 루프를 만들어 보라.
3. 다음 코드의 출력을 쓰시오.

```
int n = 10;
while (n > 0) {
    printf("%d\n", n);
    n = n - 3;
}
```





do-while문





do...while문

do

문장;

while (조건식);



일단 문장을
실행한 후에
조건을
검사하여 반복
여부를
결정한다..



예제

// do..while 문을 이용한 메뉴
#include <stdio.h>



```
int main(void)
{
```

```
    int i = 0;
```

```
    do
```

```
    {
```

```
        printf("1---새로만들기\n");
```

```
        printf("2---파일열기\n");
```

```
        printf("하나를 선택하시요.\n");
```

```
        scanf("%d", &i);
```

```
    } while(i < 1 || i > 2);
```

```
    printf("선택된 메뉴=%d\n", i);
```

```
    return 0;
```

```
}
```

1---새로만들기
2---파일열기
하나를 선택하시요.
1
선택된 메뉴=1



글자 게임 예제

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

```
int main(void)
```

```
{
```

```
    int answer =59; // 정답
```

```
    int guess;
```

```
    int tries = 0;
```

```
    // 반복 구조
```

```
    do {
```

```
        printf("정답을 추측하여 보시오: ");
```

```
        scanf("%d", &guess);
```

```
        tries++;
```

```
        if (guess > answer) // 사용자가 입력한 정수가 정답보다 높으면
```

```
            printf("제시한 정수가 높습니다.\n");
```

```
        if (guess < answer) // 사용자가 입력한 정수가 정답보다 낮으면
```

```
            printf("제시한 정수가 낮습니다.\n");
```

```
    } while (guess != answer);
```

```
    printf("축하합니다. 시도횟수=%d\n", tries);
```

```
    return 0;
```

```
}
```



정답을 추측하여 보시오: 10
제시한 정수가 낮습니다.
정답을 추측하여 보시오: 30
제시한 정수가 낮습니다.
정답을 추측하여 보시오: 60
제시한 정수가 높습니다.
정답을 추측하여 보시오: 59
축하합니다. 시도횟수=4



중간 점검


1. 다음 코드의 출력을 쓰시오.

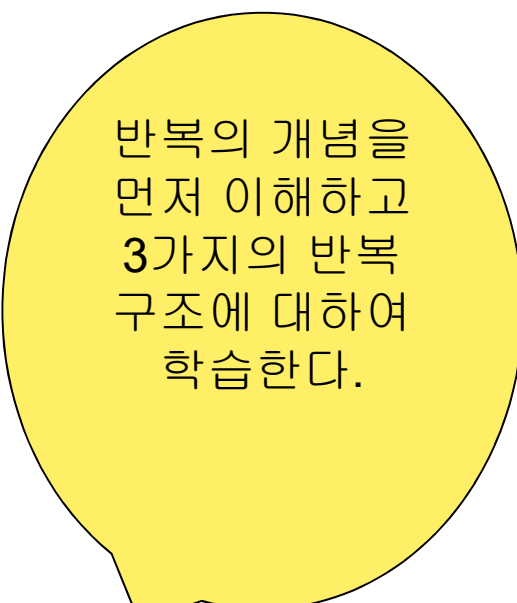
```
int n = 0;  
do {  
    printf("%d\n", n);  
    n = n + 3;  
} while(n < 10);
```





이번 장에서 학습할 내용

- 
- 반복의 개념 이해
 - while 반복문
 - do-while 반복문
 - for 반복문
 - break와 continue문



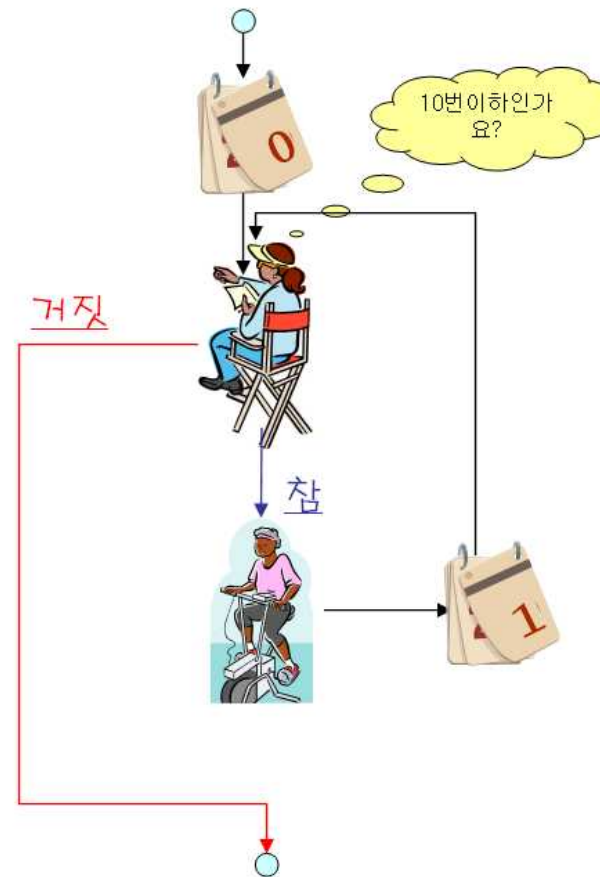
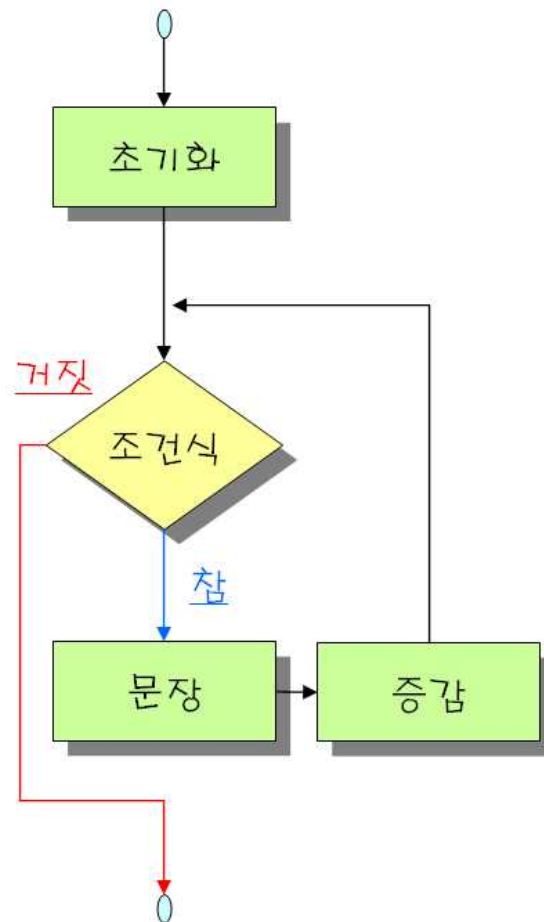
반복의 개념을
먼저 이해하고
3가지의 반복
구조에 대하여
학습한다.





for 루프

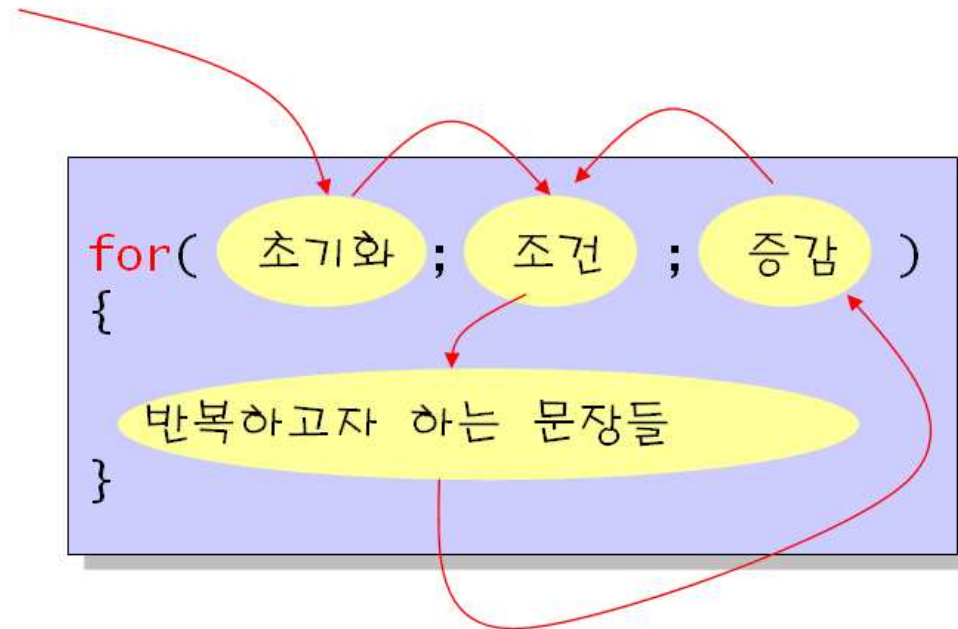
- 정해진 횟수만큼 반복하는 구조





for 문의 구조

```
for ( 초기식; 조건식; 증감식 )  
문장;
```





for 문의 예

```
int i;  
for(i = 0; i < 10; i++)  
    printf("Hello World!\n");
```



for문의 실행과정

```
for( i=0 ; i<10 ; i++ )  
    printf("Hello world!\n");
```

1번째 루프
i값은



```
for( i=0 ; i<10 ; i++ )  
    printf("Hello world!\n");
```

2-10번째 루프
i값은



...

...

```
for( i=0 ; i<10 ; i++ )  
    printf("Hello world!\n");
```

11번째 루프
i값은





for문의 실행과정

```
int i;  
for(i = 0; i < 10; i++)  
    printf("Hello World!\n");
```

반복횟수	i의 값	i<10	반복여부
1번째 반복	0	참	반복
2번째 반복	1	참	반복
3번째 반복	2	참	반복
4번째 반복	3	참	반복
5번째 반복	4	참	반복
6번째 반복	5	참	반복
7번째 반복	6	참	반복
8번째 반복	7	참	반복
9번째 반복	8	참	반복
10번째 반복	9	참	반복
11번째 반복	10	거짓	중지





예제

```
// 반복을 이용한 정수합 프로그램
#include <stdio.h>

int main(void)
{
    int i, sum;

    sum = 0;
    for(i = 1; i <= 10; i++)
        sum += i; // sum = sum + i;와 같음

    printf("1부터 10까지의 정수의 합 = %d\n", sum);

    return 0;
}
```



1부터 10까지의 정수의 합 = 55



예제

// 반복을 이용한 네모 그리기

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

```
int main(void)
```

```
{
```

```
    int i;
```

```
    printf("*****\n");
```

```
    for(i = 0; i < 5; i++)
```

```
        printf("*          *\n");
```

```
    printf("*****\n");
```

```
    return 0;
```

```
}
```



```
*****
```

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

```
*****
```



예제

```
// 반복을 이용한 팩토리얼 구하기
#include <stdio.h>

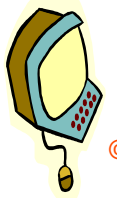
int main(void)
{
    long fact=1;
    int i, n;

    printf("정수를 입력하시요:");
    scanf("%d", &n);

    for(i = 1; i <= n; i++)
        fact = fact * i;

    printf("%d!은 %d입니다.\n", n, fact);

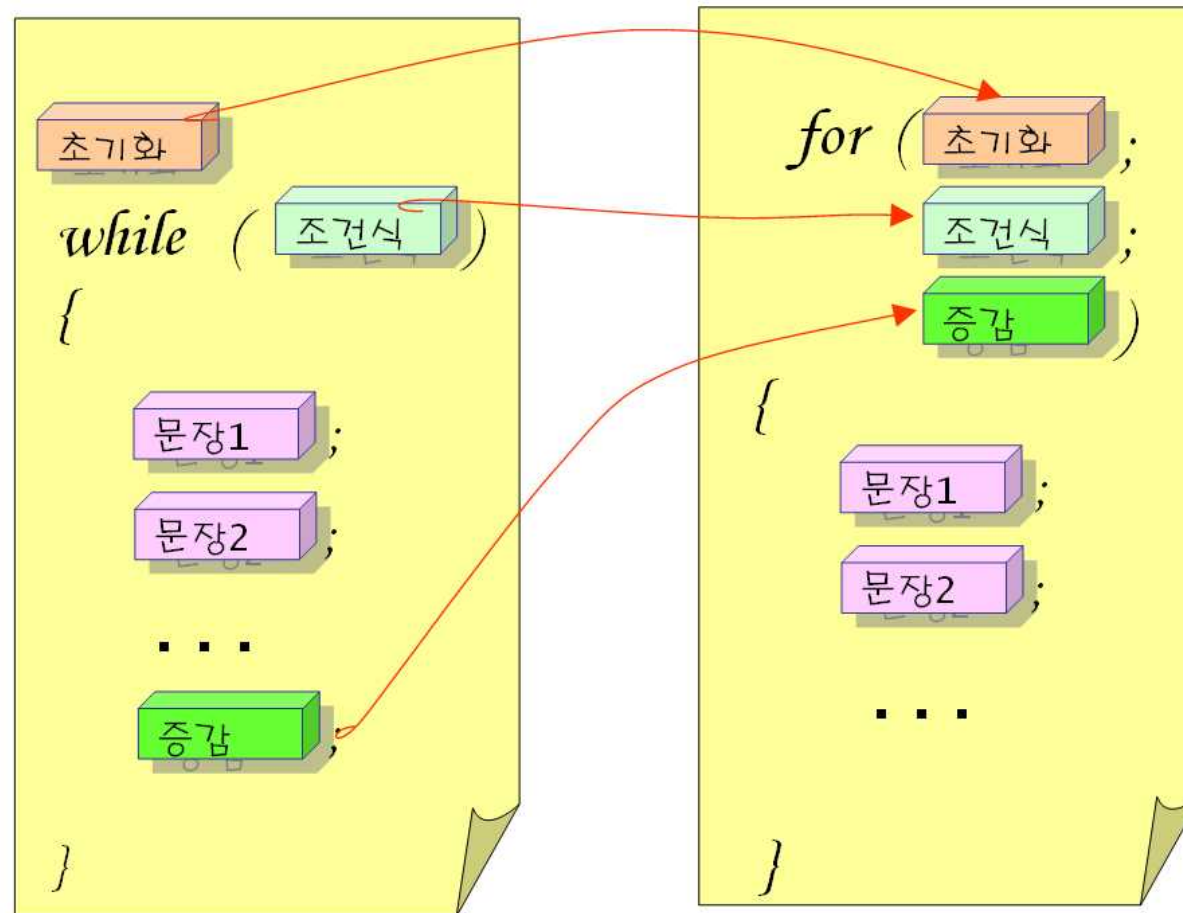
    return 0;
}
```



정수를 입력하시요: 10
10!은 3628800입니다.



while 루프와 for 루프와의 관계





다양한 증감수식의 형태

```
for (i = 10; i > 0; i-- )  
    printf("Hello World!\n");
```

백셈 사용

```
for (i = 0; i < 10; i += 2 )  
    printf("Hello World!\n");
```

2씩 증가

```
for (i = 1; i < 10; i *= 2 )  
    printf("Hello World!\n");
```

2를 곱한다.

```
for (i = 0; i < 100; i = (i * i) + 2 )  
    printf("Hello World!\n");
```

어떤 수식이라도 가능



다양한 증감수식의 형태

```
for ( ; i<100; i++ )  
    printf("Hello World!\n");
```

한부분이 없을 수도 있다.

```
for (i = 0, k = 0; i < 100; i++ )  
    printf("Hello World!\n");
```

2개 이상의 변수 초기화

```
for (printf("반복시작"), i = 0; i < 100; i++ )  
    printf("Hello World!\n");
```

어떤 수식도 가능



중간 점검

1. 다음 코드의 출력을 쓰시오.

```
for(i = 1; i < 5; i++)  
    printf("%d ", 2 * i);
```

2. 다음 코드의 출력을 쓰시오.

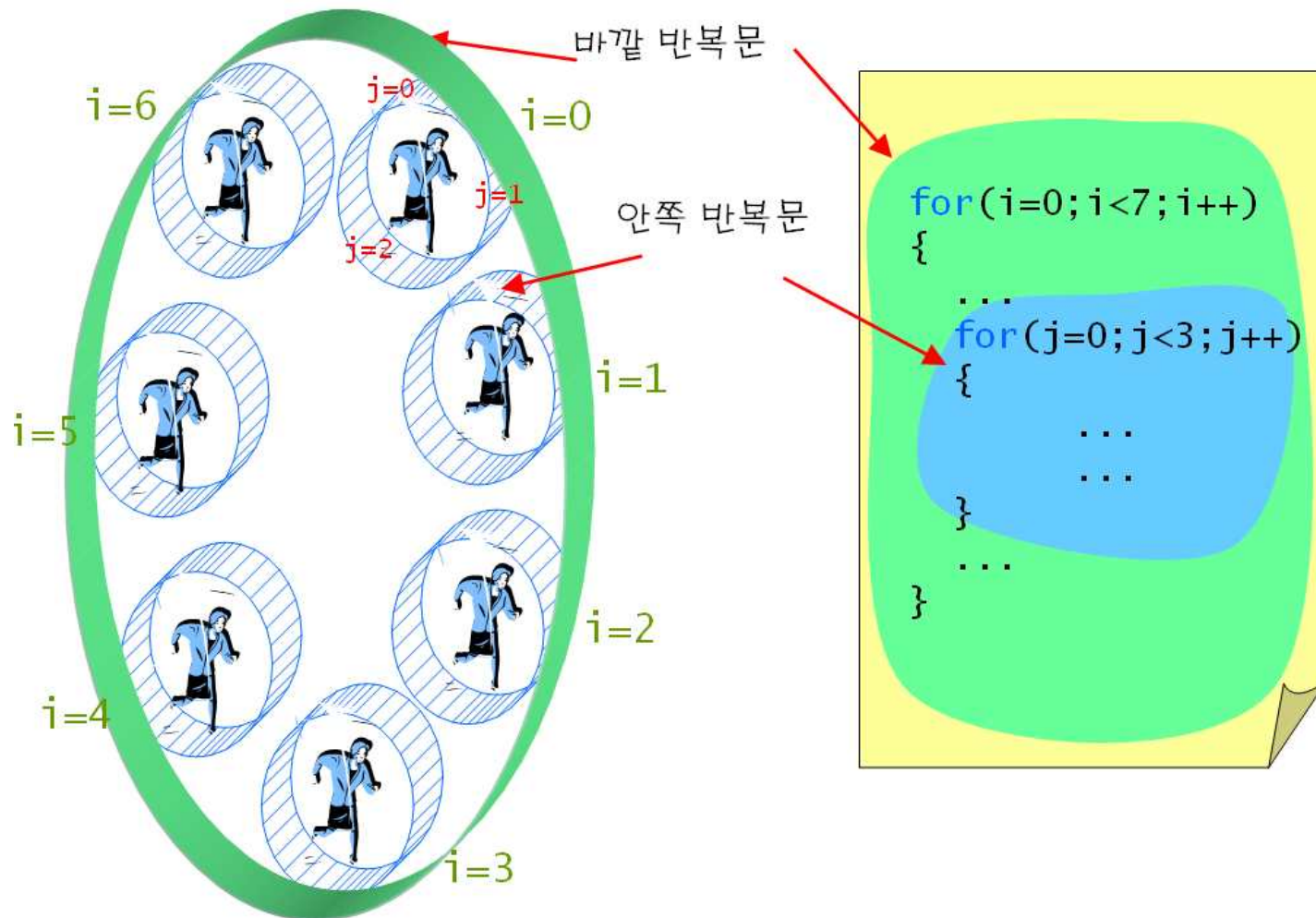
```
for(i = 10; i > 0; i = i - 2)  
    printf("Student%d\n", i);
```





중첩 반복문

- 중첩 반복문(nested loop): 반복문 안에 다른 반복문이 위치





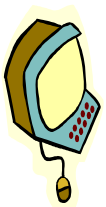
예제

// 중첩 for 문을 이용하여 *기호를 사각형 모양으로 출력하는 프로그램
`#include <stdio.h>`

```
int main(void)
{
    int x, y;

    for(y = 0; y < 5; y++)
    {
        for(x = 0; x < 10; x++)
            printf("*");
        printf("\n");
    }

    return 0;
}
```



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



예제

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

```
int main(void)
```

```
{
```

```
    int x, y;
```

```
    for(y = 1; y <= 5; y++)
```

```
    {
```

```
        for(x = 0; x < y; x++)
```

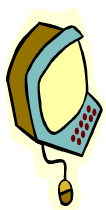
```
            printf("*");
```

```
        printf("\n");
```

```
    }
```

```
    return 0;
```

```
}
```



```
*
```

```
**
```

```
***
```

```
****
```

```
*****
```



중간 점검


1. 다음 코드의 출력을 쓰시오.

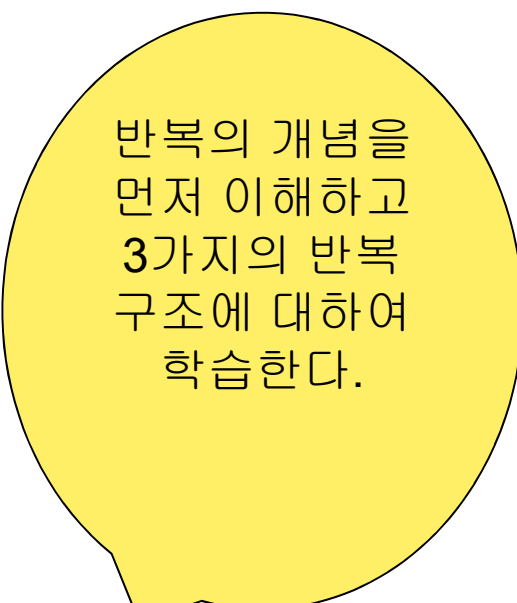
```
for(i = 1; i < 6; i++)  
    for(j = 5; j >= 1; j--)  
        printf("%d 곱하기 %d은 %d\n", i, j, i*j);
```





이번 장에서 학습할 내용

- 
- 반복의 개념 이해
 - while 반복문
 - do-while 반복문
 - for 반복문
 - break와 continue문



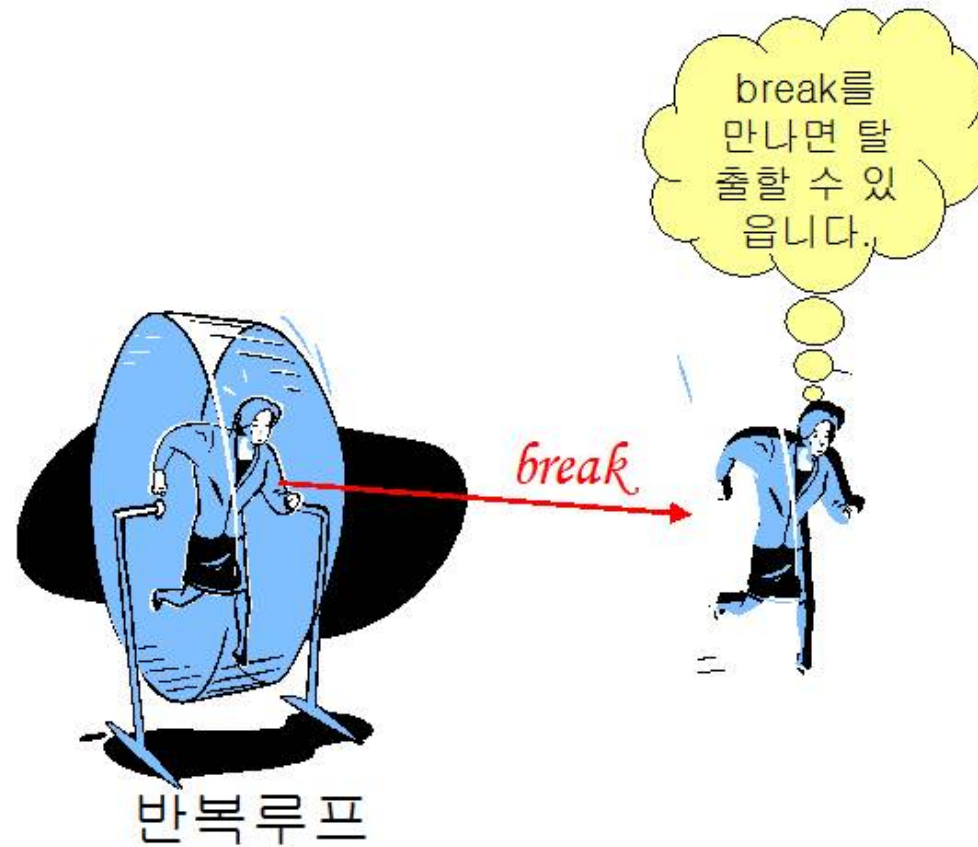
반복의 개념을
먼저 이해하고
3가지의 반복
구조에 대하여
학습한다.





break 문

- break 문은 반복 루프를 빠져 나오는데 사용된다.





break 문

- break 문은 반복 루프를 빠져 나오는데 사용된다.

```
#include <stdio.h>

int main(void)
{
    int i;

    for(i=1; i<100; i++)
    {
        printf("%d\n", i);
        if( i == 5 ) break;
    }

    return 0;
}
```

for 반복 루프

1
2
3
4
5



예제

// break를 이용하여 무한루프를 탈출한다.

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

```
#include <math.h>
```

```
int main(void)
```

```
{
```

```
    double v;
```

```
    while(1)
```

```
    {
```

```
        printf("실수값을 입력하시오: ");
```

```
        scanf("%lf", &v);
```

```
        if( v < 0.0 )
```

```
            break;
```

```
        printf("%f의 제곱근은 %f입니다.\n", v, sqrt(v));
```

```
    }
```

```
    return 0;
```

```
}
```



실수값을 입력하시오: 9.0

9.000000의 제곱근은 3.000000입니다.

실수값을 입력하시오: 12.0

12.000000의 제곱근은 3.464102입니다.

실수값을 입력하시오: 25.0

25.000000의 제곱근은 5.000000입니다.

실수값을 입력하시오: -1



예제

// break를 이용하여 무한루프를 탈출한다.

#include <stdio.h>

int main(void)

{

float grade, sum = 0.0, average;

int count = 0;

while(1)

{

printf("학생 성적을 입력하시오: ");

scanf("%f", &grade);

if(grade < 0.0)

break;

count++;

sum += grade;

}

average = sum / count;

printf("학생들의 성적의 평균은 %f입니다.\n", average);

return 0;

© }



학생 성적을 입력하시오: 90

학생 성적을 입력하시오: 90

학생 성적을 입력하시오: 80

학생 성적을 입력하시오: 70

학생 성적을 입력하시오: -1

학생들의 성적의 평균은 82.500000입니다.



중첩 반복의 경우에는 goto로 탈출

```
#include <stdio.h>

int main(void)
{
    int x, y;

    for(y = 1; y < 10000; y++)
    {
        for(x = 1; x < 50; x++)
        {
            if( _kbhit() ) goto OUT;
            printf("*");
        }
        printf("\n");
    }
    OUT:
    return 0;
}
```



continue 문

- 현재의 반복을 중단하고 다음 반복을 시작하게 한다.

```
while(i < 10)
{
    if( i % 2 == 1 ){
        i++;
        continue;
    }
    i++;
}
```



continue 문

- 현재의 반복을 중단하고 다음 반복을 시작하게 한다.

```
int i = 0;
int sum = 0;
for(i = 0; i < 10; i++)
{
    if( i % 2 == 1 ){
        continue;
    }
    sum += i;
}
```




예제



// 소문자를 대문자로 변경한다.

#include <stdio.h>

```
int main(void)
{
    char letter;

    while(1)
    {
        printf("소문자를 입력하시오: ");
        scanf(" %c", &letter);

        if( letter == 'Q' )
            break;
        if( letter < 'a' || letter > 'z' )
            continue;

        letter -= 32;
        printf("변환된 대문자는 %c입니다.\n", letter);
    }

    return 0;
}
```

소문자를 입력하시오: a
변환된 대문자는 A입니다.
소문자를 입력하시오: b
변환된 대문자는 B입니다.
소문자를 입력하시오: c
변환된 대문자는 C입니다.
소문자를 입력하시오: Q



중간 점검

1. _____ 문이 반복문에서 실행되면 현재의 반복을 중단하고 다음번 반복 처리가 시작된다.
2. _____ 문이 반복문에서 실행되면 반복문을 빠져 나온다.
3. 다음 코드의 출력을 쓰시오.

```
int i;  
for(i = 1; i < 10; i++) {  
    if( i % 3 == 0 ) break;  
    printf("%d\n", i);  
}
```





Q & A

