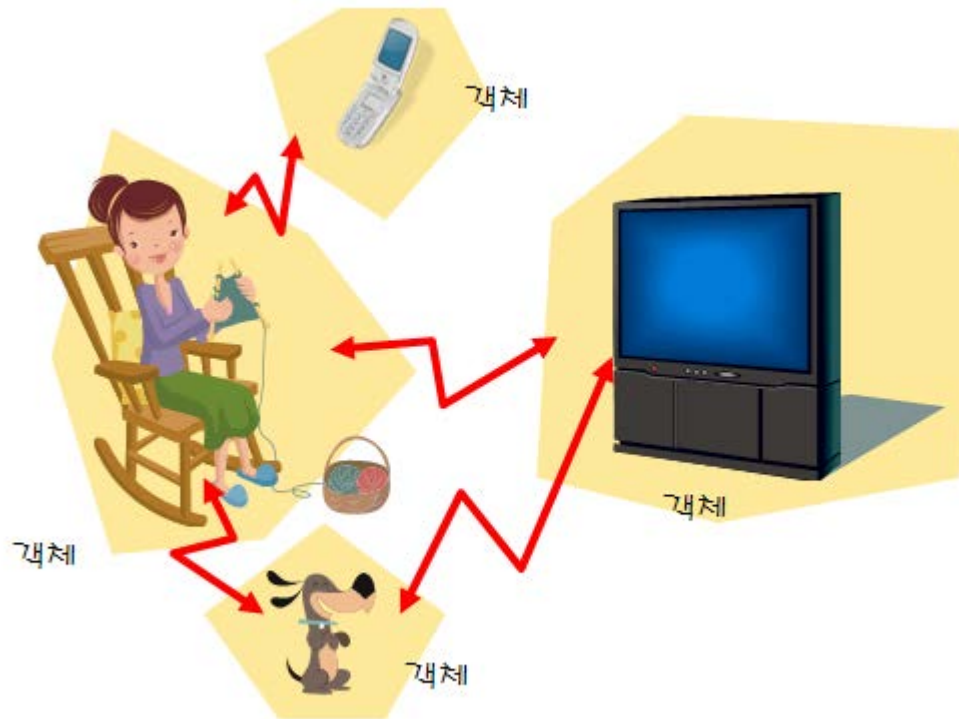




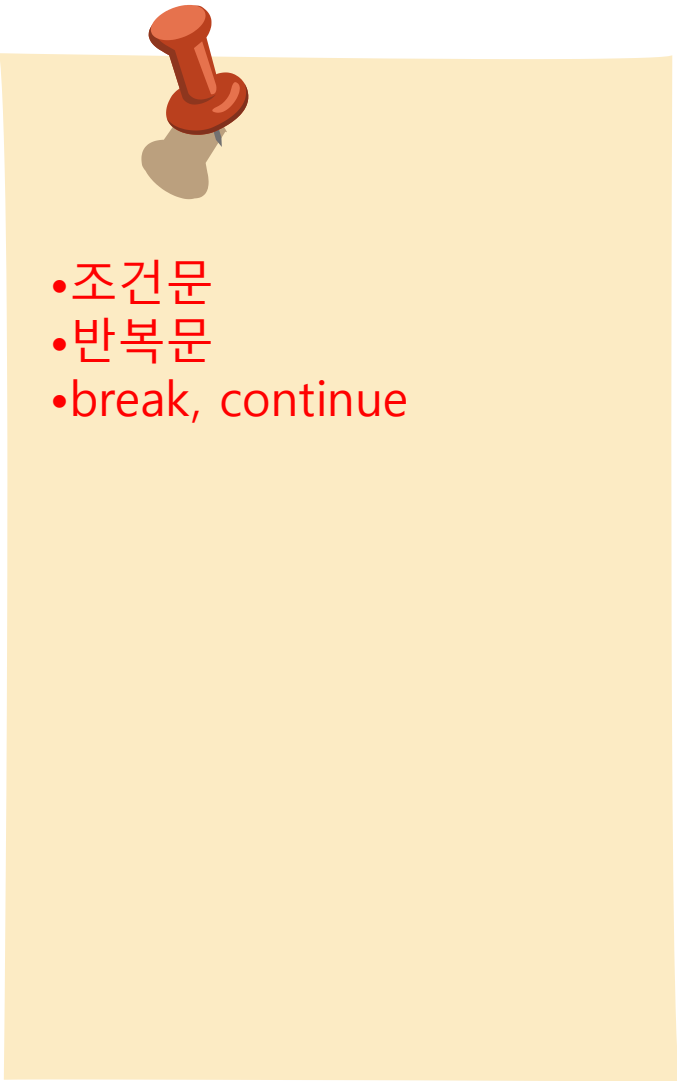
Power C++


제3장 선택과 반복





이번 장에서 학습할 내용

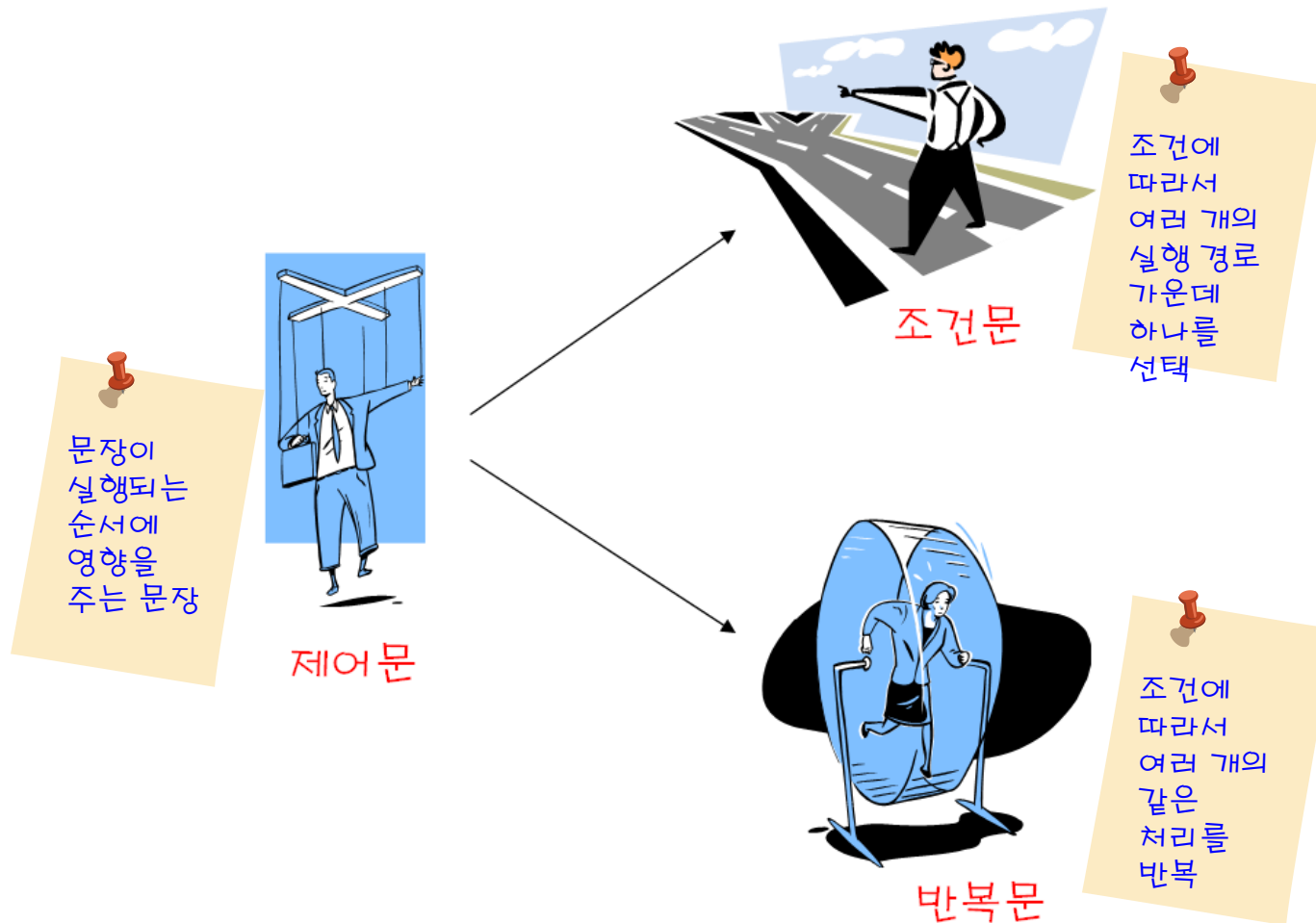
- 
- 조건문
 - 반복문
 - break, continue



이번 장에서는
선택과 반복
구조에 대하여
살펴봅니다.



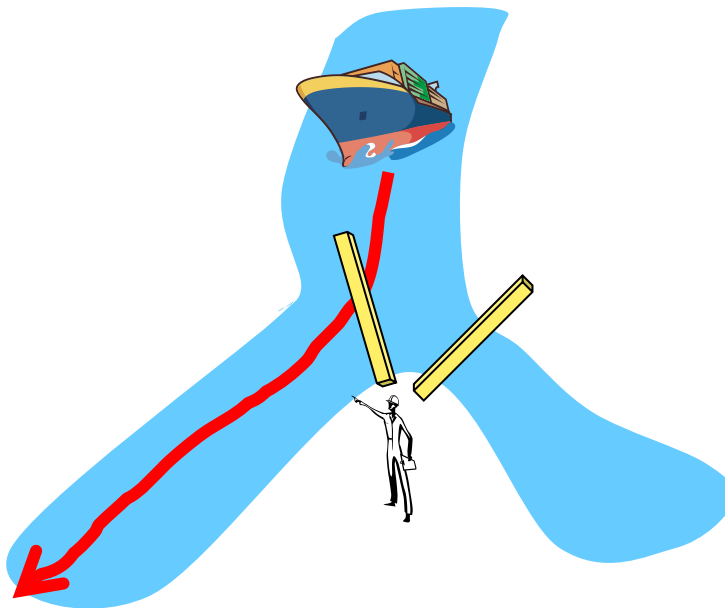
조건문





일상 생활에서의 조건문의 예

- 만약 비가 오지 않으면 테니스를 친다.
- 만약 결석이 1/3이상이면 F학점을 받는다.
- 만약 시간이 없는 경우에는 택시를 탄다.
- 만약 날씨가 좋고 공휴일이면 공원에 산책을 간다.
- 점수가 60점 이상이면 합격이고 그렇지 않으면 불합격이다.



조건문은
프로그램의
흐름을
변경합니다.





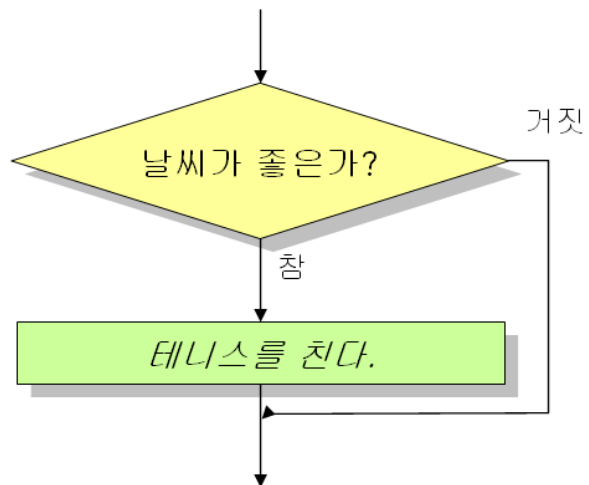
if문

- 조건에 따라서 결정을 내리는 경우에 사용

```
if( 조건식 )  
    문장 ;
```

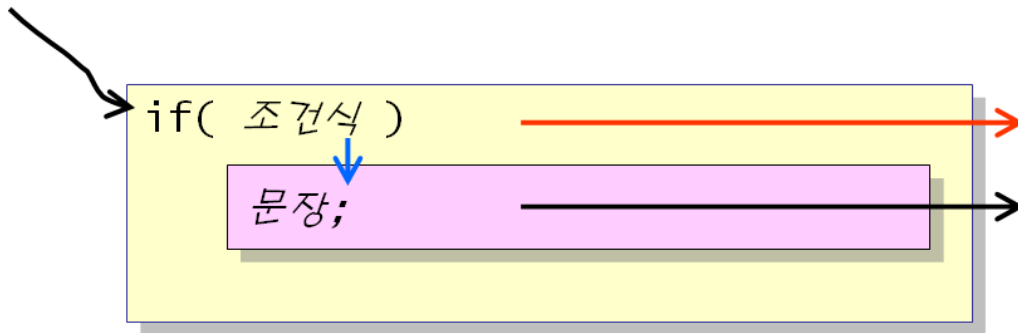
조건식이 참으로 계산되면

문장이 실행된다.

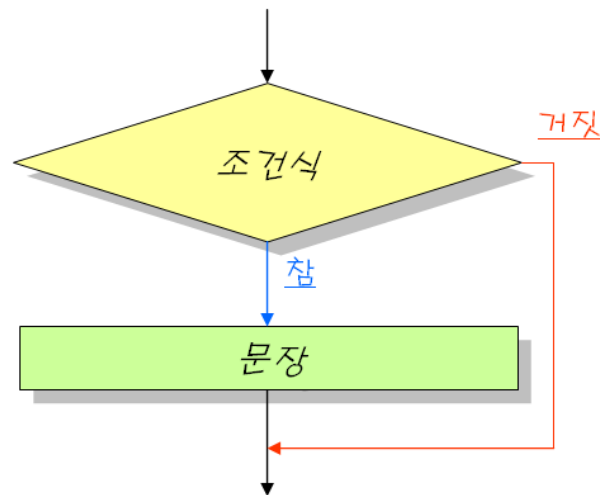




if문



파란 화살표는 조건이 참인 경우이고 빨간색 화살표는 조건이 거짓인 경우이다.





If문의 예

```
if( sales > 2000 )
```

```
    bonus = 200;
```

sales가 2000보다 크면

bonus에 200을 대입한다.

```
if( score >= 60 )
```

```
    cout << "합격입니다.\n";
```

```
if( height >= 130 && age >= 10 )
```

```
    cout << "놀이기구를 탈 수 있습니다.\n";
```

- if 문이 끝나면 if 문 다음 문장이 실행된다.

```
if ( temperature < 0 )
```

```
    cout << "현재 영하입니다.\n";           // 조건이 참일 때만 실행
```

```
    cout <<"현재 온도는 " << temperature <<"도 입니다.\n"; // 항상 실행
```



복합문

- 복합문(compound statement)
 - 중괄호를 사용하여 문장들을 그룹핑하는 것,
 - 블록(block)이라고도 한다.
 - 단일문 대신 들어 갈 수 있다.

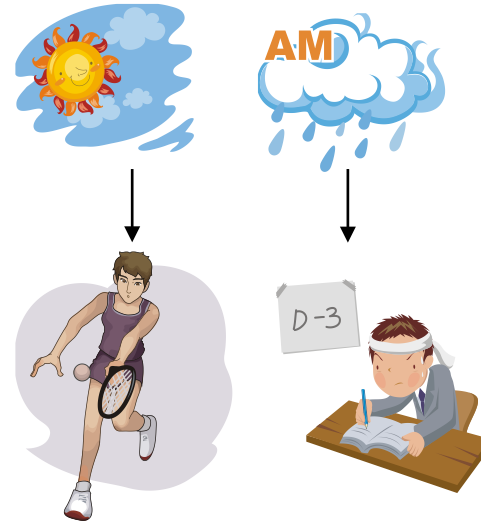
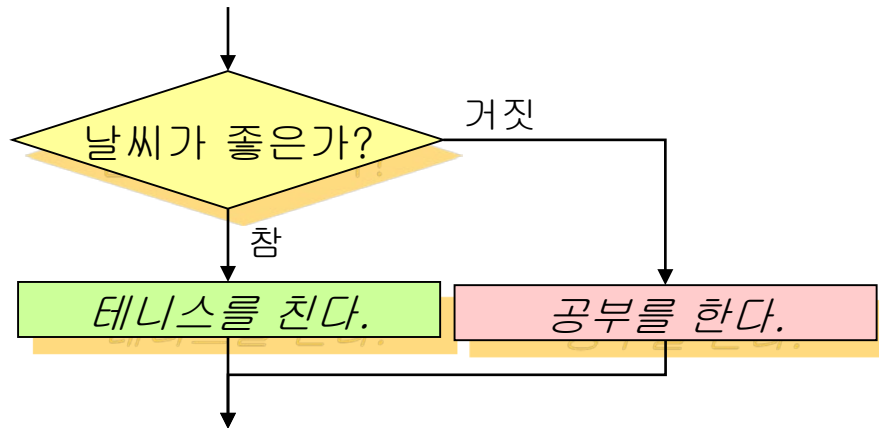


```
if( score >= 60 )  
{  
    cout << "합격입니다.\n";  
    cout << "장학금도 받을 수 있습니다.\n";  
}
```

조건식이 참이면 2개의
문장이 묶여서 실행된다.



if-else 문



if(조건식)

문장 1;

else

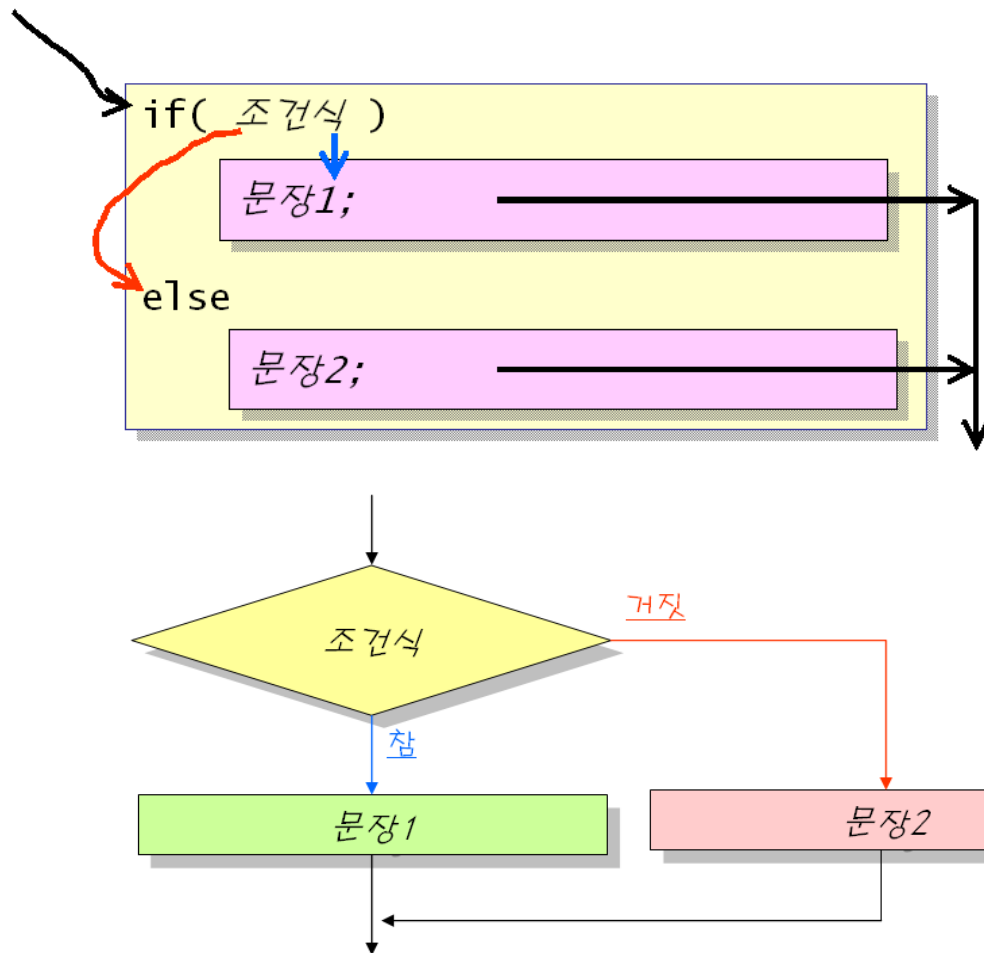
문장 2;

조건식이 참이면 실행된다.

조건식이 거짓이면 실행된다.



If-else 문





If-else 문

```
if ( score >= 60 )
```

```
    cout << "합격입니다.\n";
```

```
else
```

```
    cout << "불합격입니다.\n";
```

score가 60이상이면 실행

score가 60미만이면 실행

```
if ( score >= 60 )
```

```
{
```

```
    cout << "합격입니다.\n";
```

```
    cout << "장학금도 받을 수 있습니다.\n";
```

```
}
```

```
else
```

```
{
```

```
    cout << "불합격입니다.\n";
```

```
    cout << "공부하세요.\n";
```

```
}
```

score가 60이상이면 실행

score가 60미만이면 실행



예제 #1

```
#include <iostream>
using namespace std;

int main() {
    const int RATE = 5000;
    int pay;                // 임금
    int hours;              // 시간

    cout << "시간을 입력하시오: "; // 입력 안내 출력
    cin >> hours;

    if (hours > 8)
        pay = RATE * 8 + (int) (1.5 * RATE * (hours - 8));
    else
        pay = RATE * hours;
    cout << "임금은 " << pay << "입니다.\n";
    return 0;
}
```



시간을 입력하시오: 8
임금은 40000입니다.



중첩 if

- if 문에 다시 if 문이 포함

```
if( 조건식1 )  
    if( 조건식2 )  
        문장;
```

문장 자리에 if
문이 들어간 것으
로 생각할 수 있
다.

```
if( 조건식1 )  
    문장;
```

```
if( 조건식1 )  
    if( 조건식2 )  
        문장;
```



중첩 if

```
if( score > 80 )  
    if( score > 90 )  
        cout << "당신의 학점은 A입니다.\n";
```

If 문안의 문장자리에 if문이
들어간 경우

```
if( score > 80 )  
    if( score > 90 )  
        cout << "당신의 학점은 A입니다.\n";  
    else  
        cout << "당신의 학점은 B입니다.\n";
```

If 문안의 문장자리에 if-else
문이 들어간 경우



if와 else의 매칭 문제

else 절은 가장 가까운 if절과 매치된다.

```
if(score > 80)
    if( score > 90)
        cout << "당신의 학점은 A입니다\n";
    else
        cout << "당신의 학점은 B입니다\n";
```

Diagram illustrating the matching of the `else` statement to the `if(score > 90)` statement, as it is the closest one.

```
if( score > 80 )
{
    if( score > 90 )
        cout << "당신의 학점은 A입니다.\n";
}
else
    cout << "당신의 학점은 A나 B가 아닙니다.\n";
```

만약 다른 if절과 else 절을 매치시키려면 중괄호를 사용하여 블록으로 묶는다.



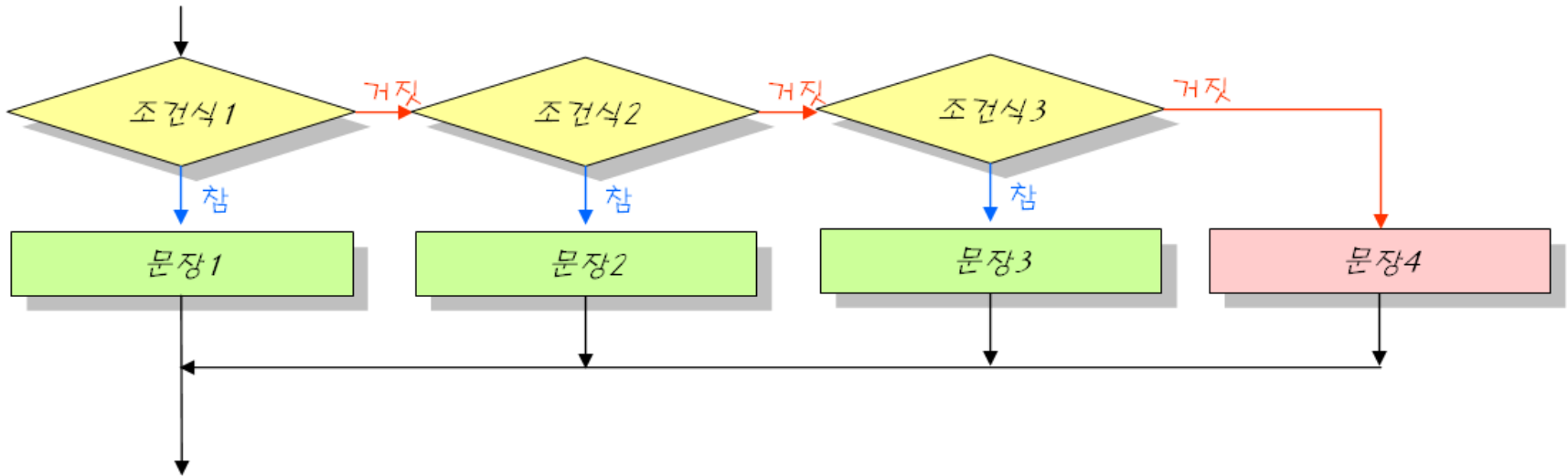
중첩 if

```
if( score > 80 )
{
    if( score > 90 )
        cout << "당신의 학점은 A입니다.\n";
    else
        cout << "당신의 학점은 B입니다.\n";
}
else
{
    if( score > 70 )
        cout << "당신의 학점은 C입니다.\n";
    else
        cout << "당신의 학점은 D 또는 F입니다.\n";
}
```

If-else 절 안에 다른 if-else절이 포함될 수 있다.



연속적인 if



```
if( 조건식1 )  
    문장1;  
else if( 조건식2 )  
    문장2;  
else if( 조건식3 )  
    문장3;  
else  
    문장4;
```



연속적인 if

- 성적을 입력받아서 연속적인 if를 사용하여 학점을 매기는 코드

```
if (score >= 90)
    cout << "합격: 학점 A\n";
else if (score >= 80)
    cout << "합격: 학점 B\n";
else if (score >= 70)
    cout << "합격: 학점 C\n";
else if (score >= 60)
    cout << "합격: 학점 D\n";
else
    cout << "불합격: 학점 F\n";
```

score >= 80 && score < 90
와 같이 쓸 필요는 없음



소득세 계산 프로그램



```
#include <iostream>
using namespace std;
int main()
{
    int income; // 과세표준
    int tax; // 세금
    cout << "과세 표준 금액을 입력하시오:";
    cin >> income;
    if (income <= 1000)
        tax = (int) (0.09 * income);
    else if (income <= 4000)
        tax = (int) (0.18 * income);
    else if (income < 8000)
        tax = (int) (0.27 * income);
    else
        tax = (int) (0.36 * income);
    cout << "소득세는 " << tax << "입니다.\n";
    return 0;
}
```

과세 표준 금액을 입력하시오: **5000**
소득세는 **1350**입니다.



중간 점검 문제

1. 변수 n 의 값이 100보다 크거나 같으면 "large", 100보다 작으면 "small"을 출력하는 if-else 문을 작성하라.
2. k 의 값이 각각 3, 0, -1인 경우에 다음의 코드에 의하여 생성되는 출력은 무엇인가?

```
if( k != 0 )  
    cout <<"A";  
else if( k > 3 )  
    cout <<"B";  
else  
    cout <<"C";
```



3. 컵의 사이즈를 받아서 100ml미만은 small, 100ml이상 200ml미만은 medium, 200ml 이상은 large라고 출력하는 연속적인 if-else 문을 작성하시오.

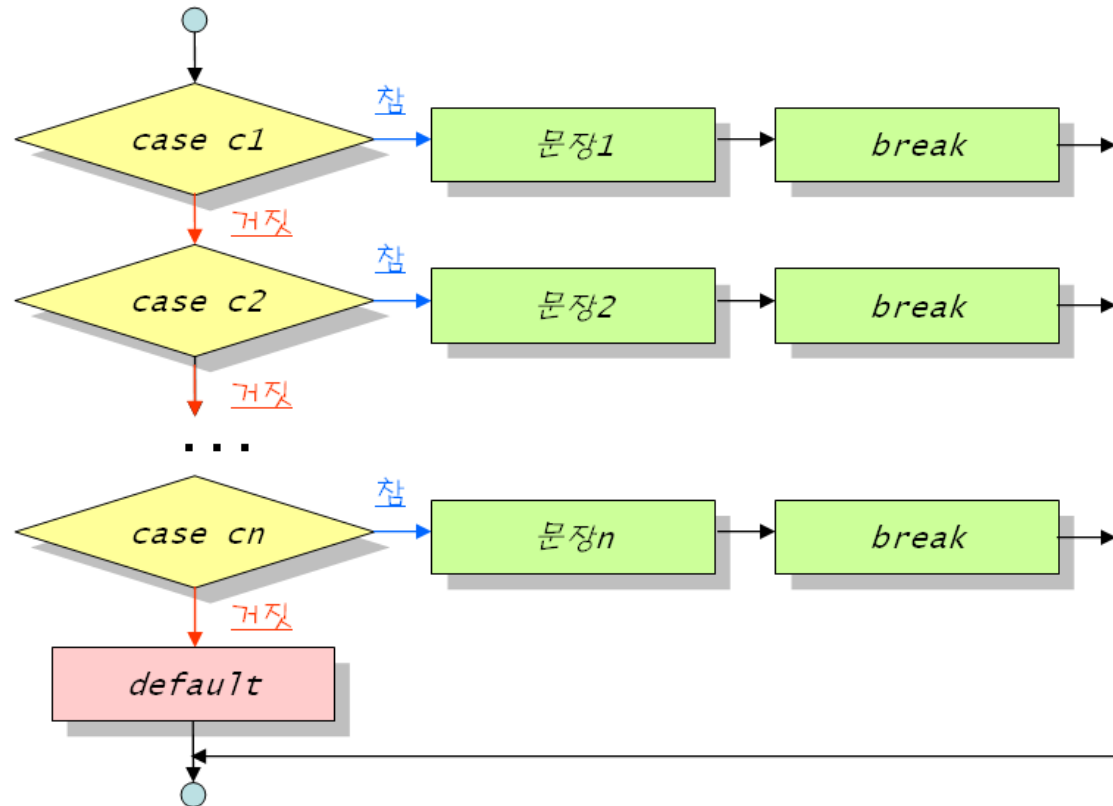


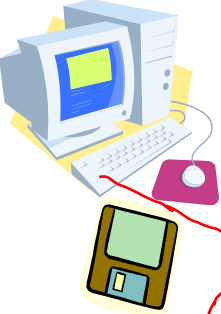
switch 문

- 여러가지 경우 중에서 하나를 선택하는데 사용

switch(조건식)

```
{  
  case c1:  
    문장1;  
    break;  
  case c2:  
    문장2;  
    break;  
  ...  
  case cn:  
    문장n;  
    break;  
  default:  
    문장d;  
    break;  
}
```





예제



정수를 입력하시오: 1
하나

```
int main()
{
    int number;

    cout << "정수를 입력하시오:";
    cin >> number;
    switch(number)
    {
        case 0:
            cout << "없음\n";
            break ;
        case 1:
            cout << "하나\n";
            break ;
        case 2:
            cout << "둘\n";
            break ;
        default:
            cout << "많음\n";
            break;
    }
}
```



의도적인 break생략

```
switch(number)
{
    case 0:
        cout << "없음\n";
        break;
    case 1:
        cout << "하나\n";
        break;
    case 2:
    case 3:
        cout << "두서너개\n";
        break;
    default:
        cout << "많음\n";
        break;
}
```

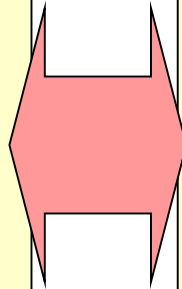
2개의 경우를
하나로 묶어서
처리하기
위하여 이러한
기법을 사용





switch 문과 if-else 문

```
if( number == 0 )  
    cout << "없음\n";  
else if( number == 1 )  
    cout << "하나\n";  
else if( number == 2 )  
    cout << "둘\n";  
else  
    cout << "많음\n";
```



```
switch(number)  
{  
    case 0:  
        cout << "없음\n";  
        break;  
    case 1:  
        cout << "하나\n";  
        break;  
    case 2:  
        cout << "둘\n";  
        break;  
    default:  
        cout << "많음\n";  
        break;  
}
```




예제

```
// 달의 일수를 계산하는 프로그램
```

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

```
int main(void)
```

```
{
```

```
    int month, days;
```

```
    cout << "일수를 알고 싶은 달을 입력하시오: ";
```

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

```
    switch(month)
```

```
    {
```

```
        case 2:
```

```
            days = 28;
```

```
            break;
```

```
        case 4:
```

```
        case 6:
```

```
        case 9:
```

```
        case 11:
```

```
            days = 30;
```

```
            break;
```

```
        default:
```

```
            days = 31;
```

```
            break;
```

```
    }
```

```
    cout << "%d월의 일수는 %d입니다.\n", month, days);
```

```
    return 0;
```

```
}
```



일수를 알고 싶은 달을 입력하시오: 12
12월의 일수는 31입니다.



예제

```
#include <iostream>
using namespace std;
```

```
int main()
{
```

```
    int month;
    int year = 2009;
    int days = 0;
```

```
    cout << "일수를 알고 싶은 달을 입력하시오:";
    cin >> month;
```

```
    switch (month) {
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12:
        days = 31;
        break;
```



일수를 알고 싶은 달을 입력하시오: 12
12월의 일수는 31입니다.



예제



case 4:

case 6:

case 9:

case 11:

days = 30;

break;

case 2:

if (((year % 4 == 0) &&!(year % 100 == 0)) || (year % 400 == 0)) //윤년

days = 29;

else

days = 28;

// 평년

break;

default:

cout <<"월이 잘못 입력되었습니다.";

break;

}

cout <<"월의 날수는 " << days << endl;

return 0;

}

일수를 알고 싶은 달을 입력하시오: 12
12월의 일수는 31입니다.



중간 점검 문제

1. case 절에서 break 문을 생략하면 어떻게 되는가?
2. 변수 fruit의 값이 각각 1, 2, 5일 때, 다음의 코드의 출력을 쓰시오.

```
switch(fruit) {  
    case 1: cout <<"사과";  
            break;  
    case 2: cout <<"배";  
    case 3: cout <<"바나나";  
            break;  
    default:cout <<"과일";  
}
```

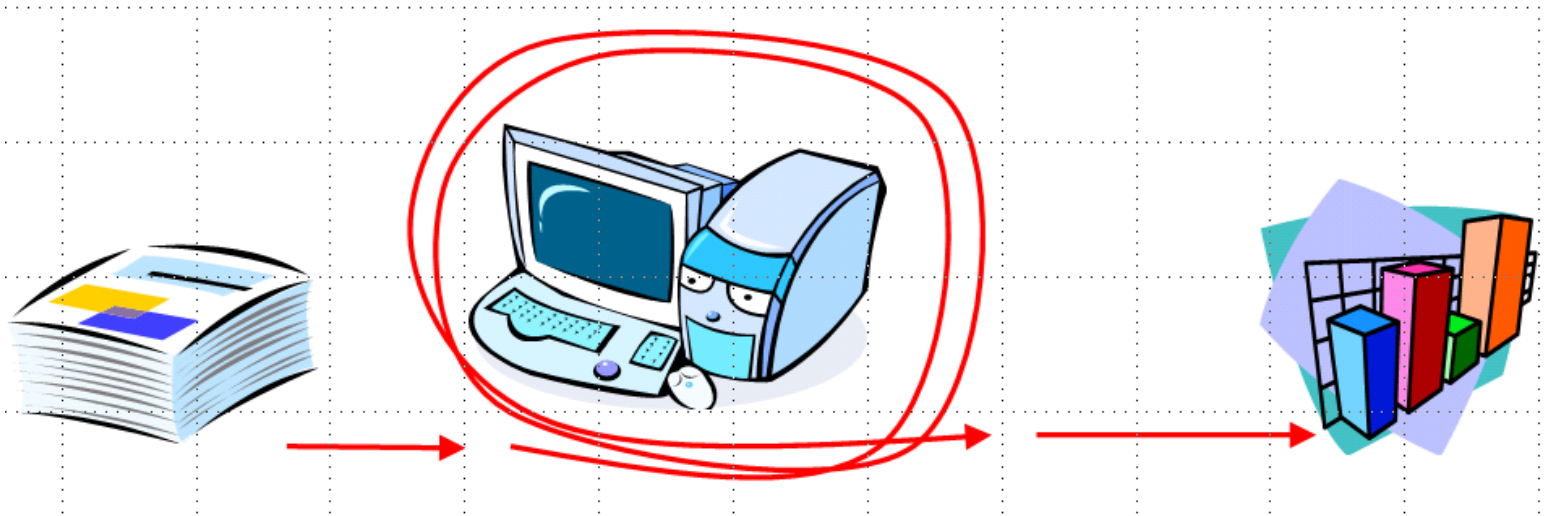




반복문

Q) 반복 구조는 왜 필요한가?

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

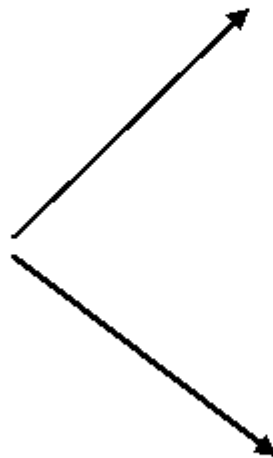




반복문의 종류



반복문



while



10kg이 빠질
때까지 반복
하세요



for



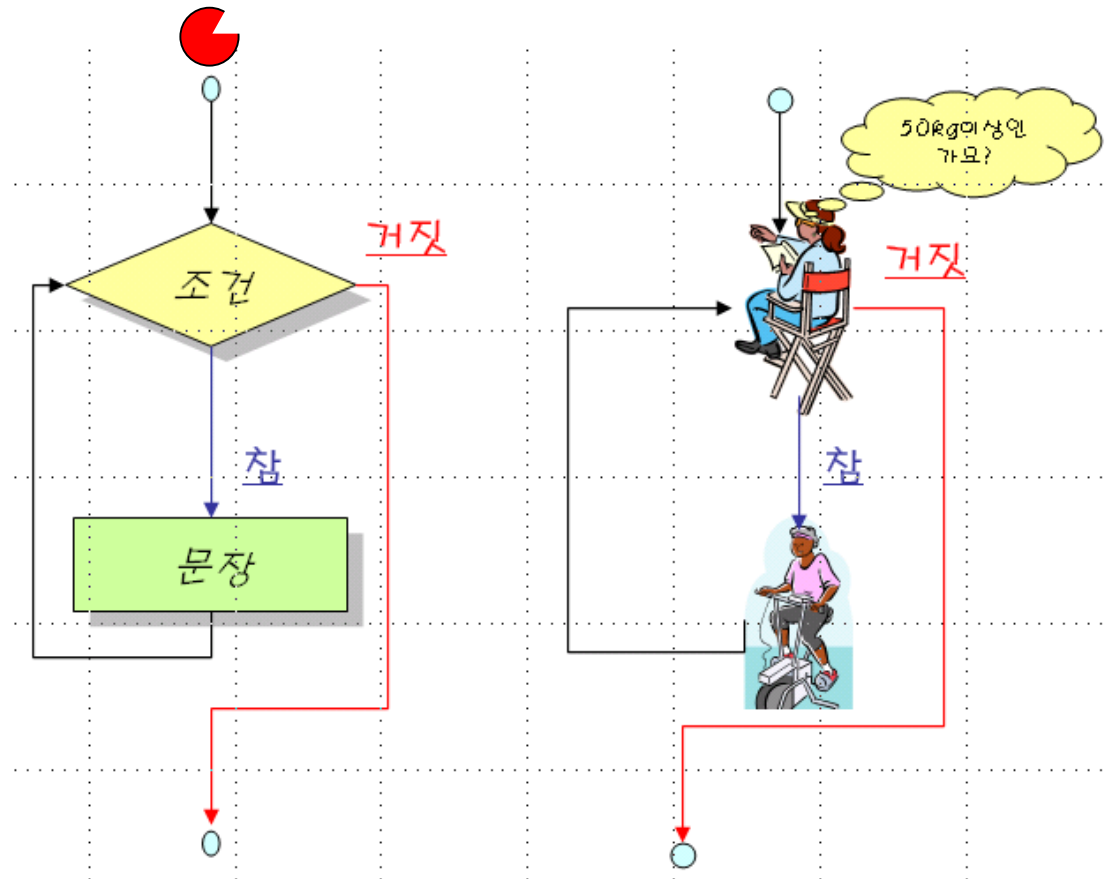
100번 반복
하세요



while 문

- 주어진 조건이 만족되는 동안 문장들을 반복 실행한다.

while(조건식)
문장;





예제

```
int main()
{
    int i = 0;
    while (i < 5){
        cout <<"정수: " << i << endl;
        i++;
    }
    return 0;
}
```



정수: 0
정수: 1
정수: 2
정수: 3
정수: 4



예제

```
// while 문을 이용한 구구단 출력 프로그램
#include <iostream>
using namespace std;

int main()
{
    int n;
    int i = 1;

    cout << "구구단 중에서 출력하고 싶은 단을 입력하시오: ";
    cin >> n;
    while (i <= 9){
        cout << n << "*" << i << "=" << n * i << endl;
        i++;
    }
    return 0;
}
```



구구단 중에서 출력하고 싶은 단을 입력하시오: 9

9*1 = 9
9*2 = 18
9*3 = 27
....
9*9 = 81



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

```
#include <iostream>
using namespace std;
```

```
int main()
{
```

```
    int x, y, r;
```

```
    cout << "두개의 정수를 입력하시오(큰 수, 작은 수): ";
```

```
    cin >> x;
```

```
    cin >> y;
```

```
    while (y != 0) {
```

```
        r = x % y;
```

```
        x = y;
```

```
        y = r;
```

```
    }
```

```
    cout << "최대 공약수는 " << x << "입니다.\n";
```

```
    return 0;
```

```
}
```

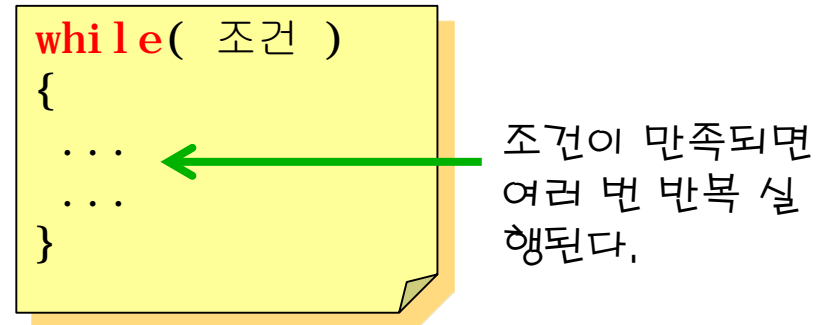
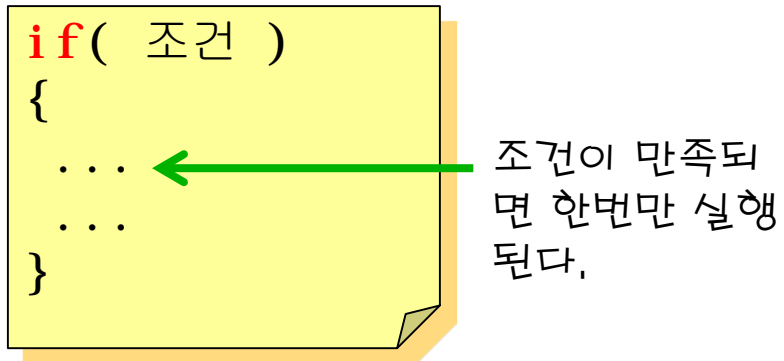


두개의 정수를 입력하시오(큰 수, 작은 수): 12 8
최대 공약수는 4입니다.



if 문과 while 문의 비교

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



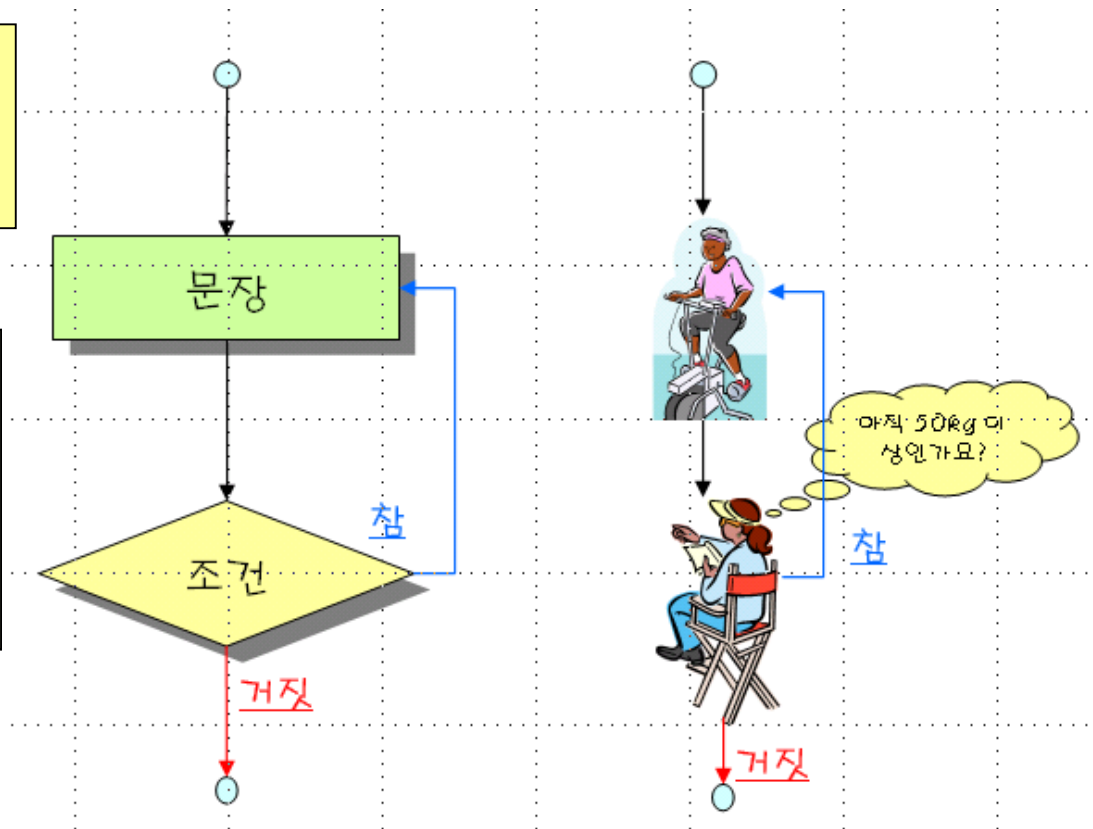


do...while문

- 반복 조건을 루프의 끝에서 검사

do
문장
while(조건)

- ① 문장들이 실행된다.
- ② 조건식이 계산된다.
- ③ 결과가 참이면 ①로 돌아간다.
- ④ 결과가 거짓이면 종료된다.





예제

```
int main()
{
    int i = 10;
    do {
        cout << "i의 값: " << i << endl;
        i++;
    } while (i < 3);
}
```

i의 값: 10



게임 예제

// 반복을 이용한 게임 프로그램

```
#include <iostream>
using namespace std;
```

```
int main()
{
```

```
    int answer =59;    // 정답
    int guess;
    int tries = 0;
    // 반복 구조
    do {
```

```
        cout << "정답을 추측하여 보시오: ";
        cin >> guess;
        tries++;
```

```
        if (guess >answer)    // 사용자가 입력한 정수가 정답보다 높으면
            cout <<"제시한 정수가 높습니다.\n";
        if (guess <answer)    // 사용자가 입력한 정수가 정답보다 낮으면
            cout <<"제시한 정수가 낮습니다.\n";
```

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

```
    cout << "축하합니다. 시도 횟수=" << tries << endl;
```

```
    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 문제

1. 다음 코드의 출력을 쓰시오.

```
int n = 10;  
while (n > 0) {  
    cout << n;  
    n = n - 3;  
}
```

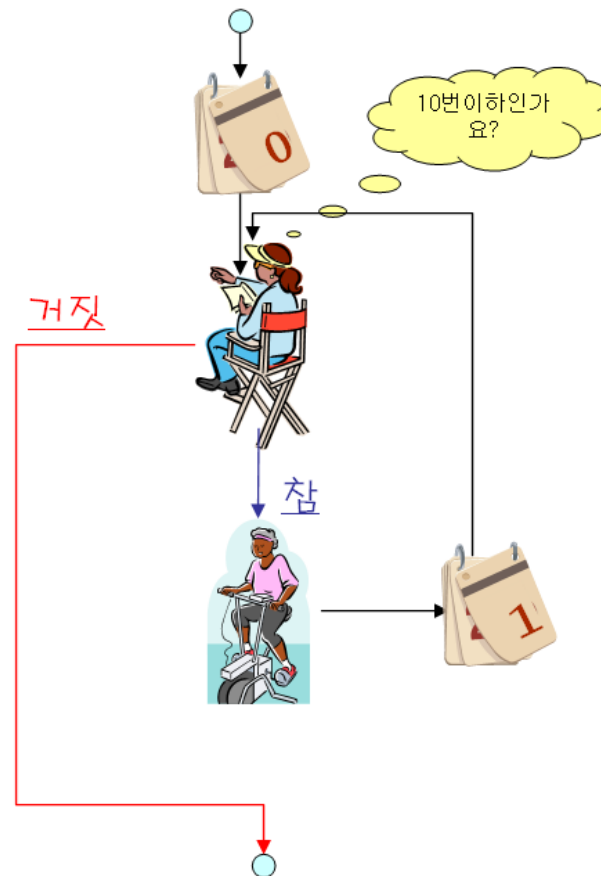
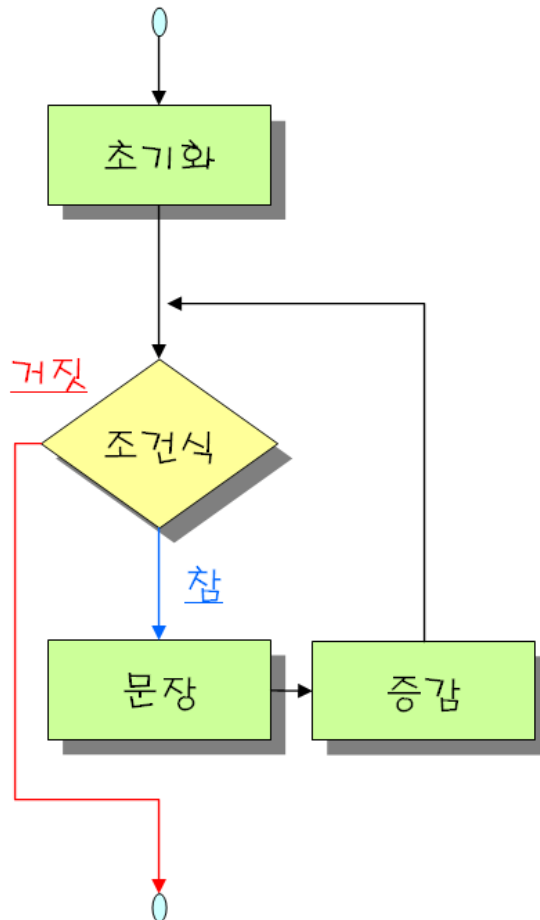
2. 1번 문제의 반복 구조를 do-while로 변경하면 출력이 어떻게 변화되는가?





for 루프

- 정해진 횟수만큼 반복하는 구조

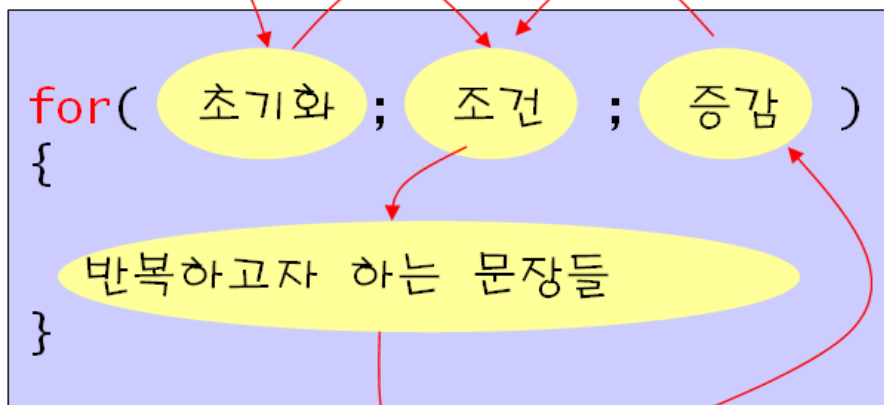




for 문의 구조

for (초기화; 조건식; 증감식)
문장;

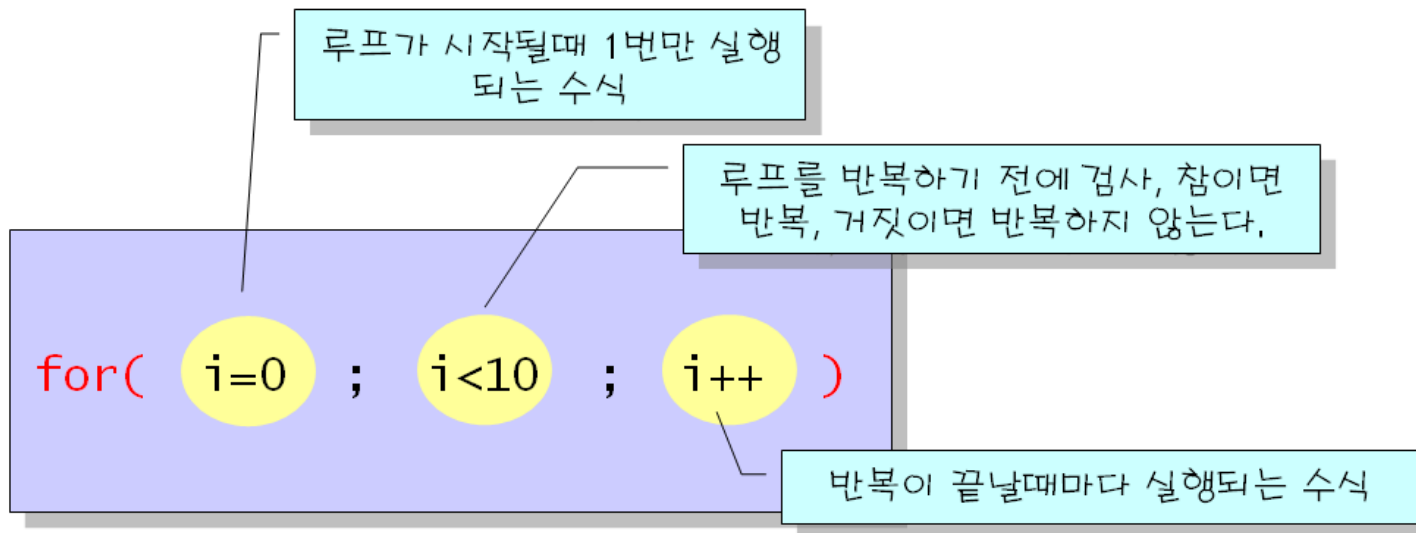
- ① 초기화를 실행한다.
- ② 반복 조건을 나타내는 조건식을 계산한다.
- ③ 수식의 값이 거짓이면 for 문의 실행이 종료된다.
- ④ 수식의 값이 참이면 문장이 실행된다.
- ⑤ 증감을 실행하고 ②로 돌아간다.





for 문의 구조

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

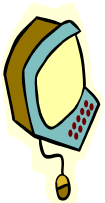




for문의 실행과정

```
int i;  
for(i = 0; i < 10; i++)  
    cout << "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





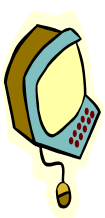
예제

```
#include <iostream>
using namespace std;

int main()
{
    int sum = 0;

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

    cout << "1부터 10까지의 정수의 합 = " << sum << endl;
    return 0;
}
```



1부터 10까지의 정수의 합 = 55



예제

```
#include <iostream>
using namespace std;

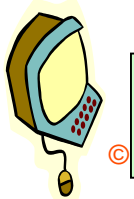
int main()
{
    long fact = 1;
    int n;

    cout << "정수를 입력하시요:";
    cin >> n;

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

    cout << n << "!은 " << fact << "입니다.\n";

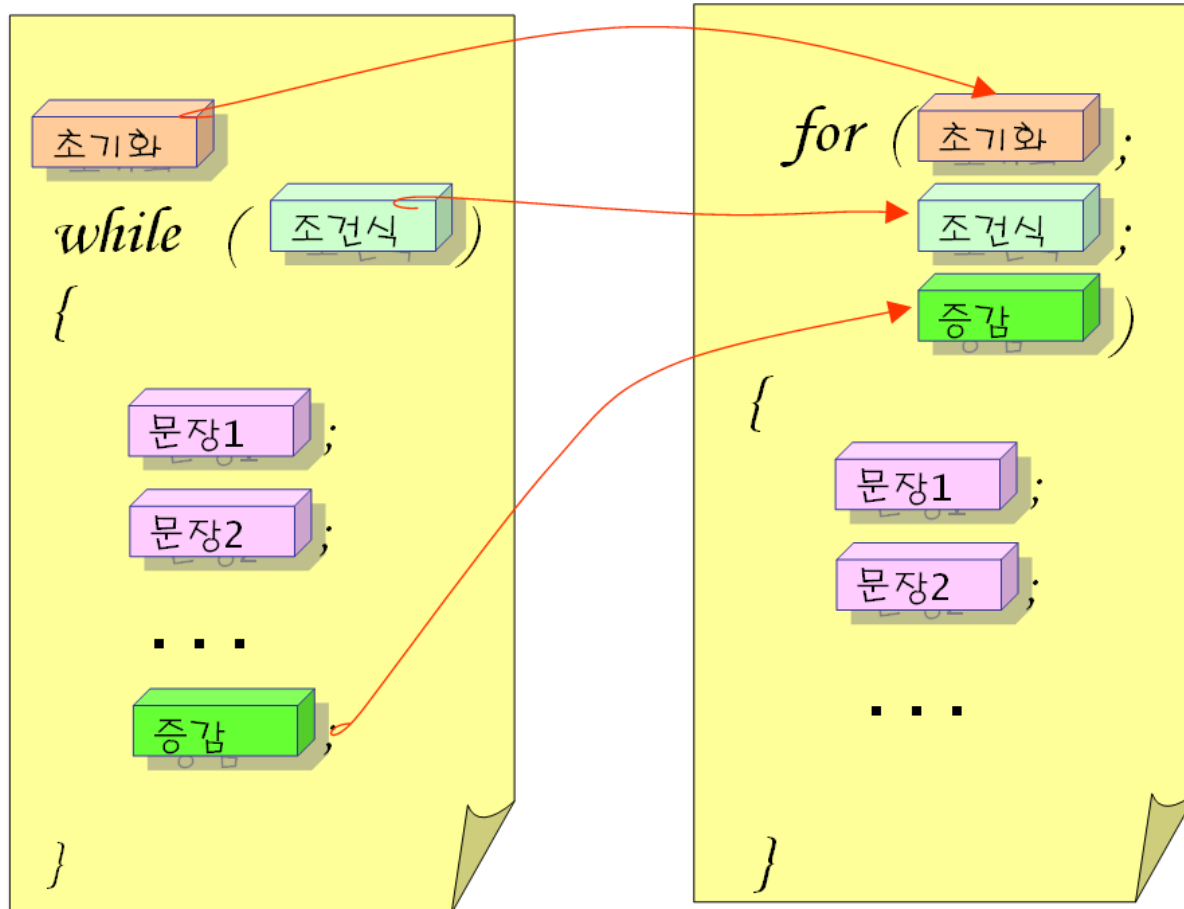
    return 0;
}
```



정수를 입력하시요: 10
10!은 3628800입니다.



while 루프와 for 루프와의 관계





다양한 증감수식의 형태

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

백셈 사용

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

2씩 증가

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

2를 곱한다.

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

어떤 수식이라도 가능

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

한부분이 없을 수도 있다.

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

2개 이상의 변수 초기화

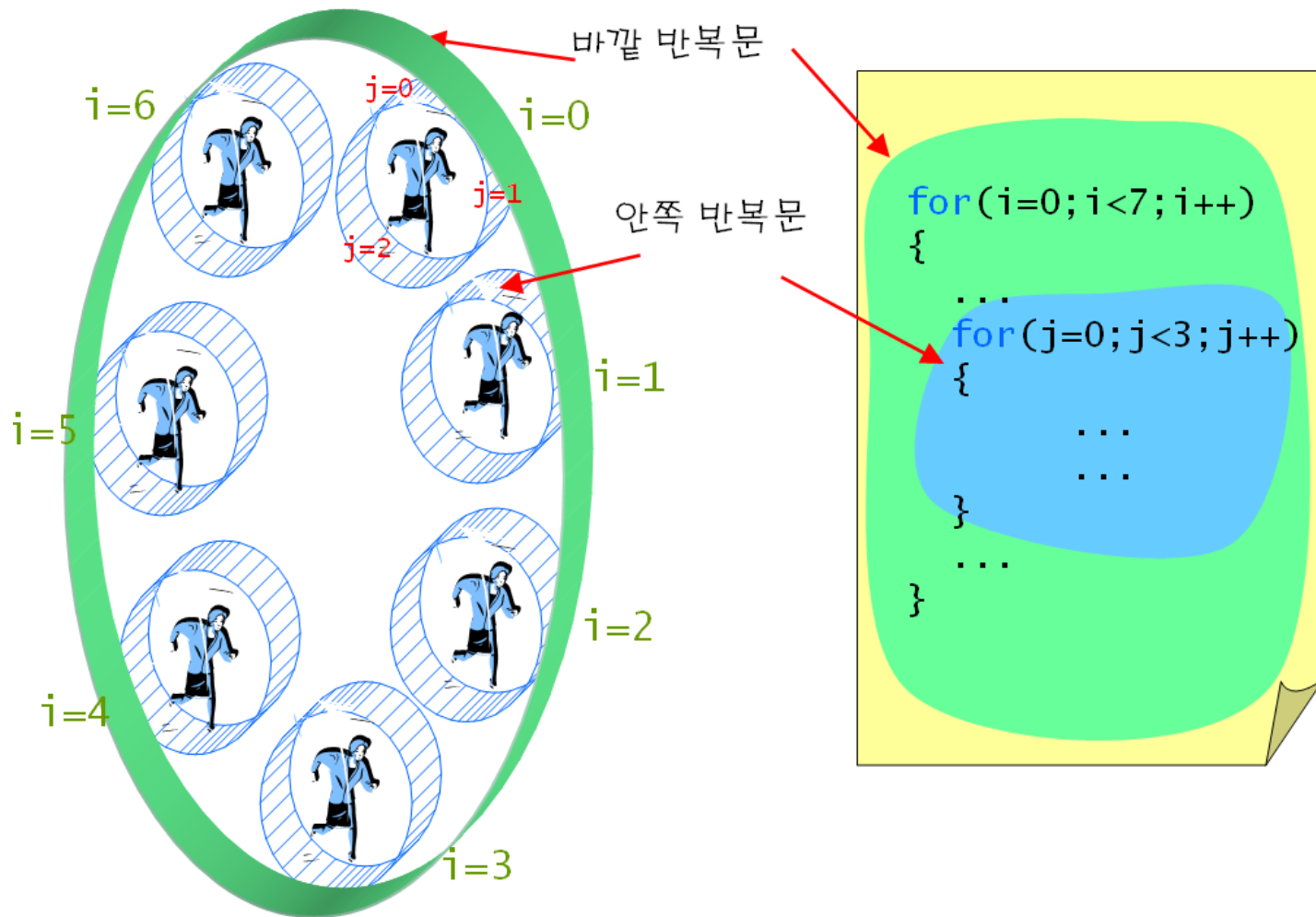
```
for (cout << "반복시작"), i = 0; i < 100; i++ )  
    cout << "Hello World!\n";
```

어떤 수식도 가능



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++)
            cout << "x";
        cout << endl;
    }

    return 0;
}
```



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



중간 점검 문제

1. 다음 코드의 출력을 쓰시오.

```
for(int i = 1; i < 5; i++)  
    cout << (2 * i) << " ";
```

2. 다음 코드의 출력을 쓰시오.

```
for(int i = 10; i > 0; i = i - 2)  
    cout <<"Student" << i << endl;
```

3. 다음 코드의 출력을 쓰시오.

```
for(int i = 1; i < 6; i++)  
    for(int j = 5; j >= 1; i--)  
        cout << i << "곱하기 " << j << "은 " << i*j << endl;
```





break 문

- break 문은 반복 루프를 빠져 나오는데 사용된다.

```
int main()
{
    int i;
    for(i=1; i<100; i++)
    {
        cout << i << endl;
        if( i == 10 ) break;
    }
}
```

나를 감싸고 있는 loop만 깨는 것



예제

// break를 이용하여 무한루프를 탈출한다.

```
#include <iostream>
#include <cmath>
using namespace std;
```

```
int main()
```

```
{
```

```
    double v;
```

```
    while(1)
```

```
    {
```

```
        cout << "실수값을 입력하시오: ";
```

```
        cin >> v;
```

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

```
            break;
```

```
        cout << v << "의 제곱근은 " << sqrt(v) << "입니다.\n";
```

```
    }
```

```
    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

```
#include <iostream>
#include <cmath>
using namespace std;
```

```
int main()
{
```

```
    float grade, sum = 0.0, average;
    int count = 0;
    while(1)
    {
```

```
        cout << "학생 성적을 입력하시오: ";
        cin >> grade;
```

// 정수도 받을 수 있다.

```
        if( grade < 0.0 )
            break;
```

// 음수가 입력되면
// 반복을 빠져나간다.

```
        count++;
        sum += grade;
```

```
    }
```

```
    average = sum / count;
```

```
    cout << "학생들의 성적의 평균은 " << average << "입니다.\n";
```

```
    return 0;
```

```
}
```

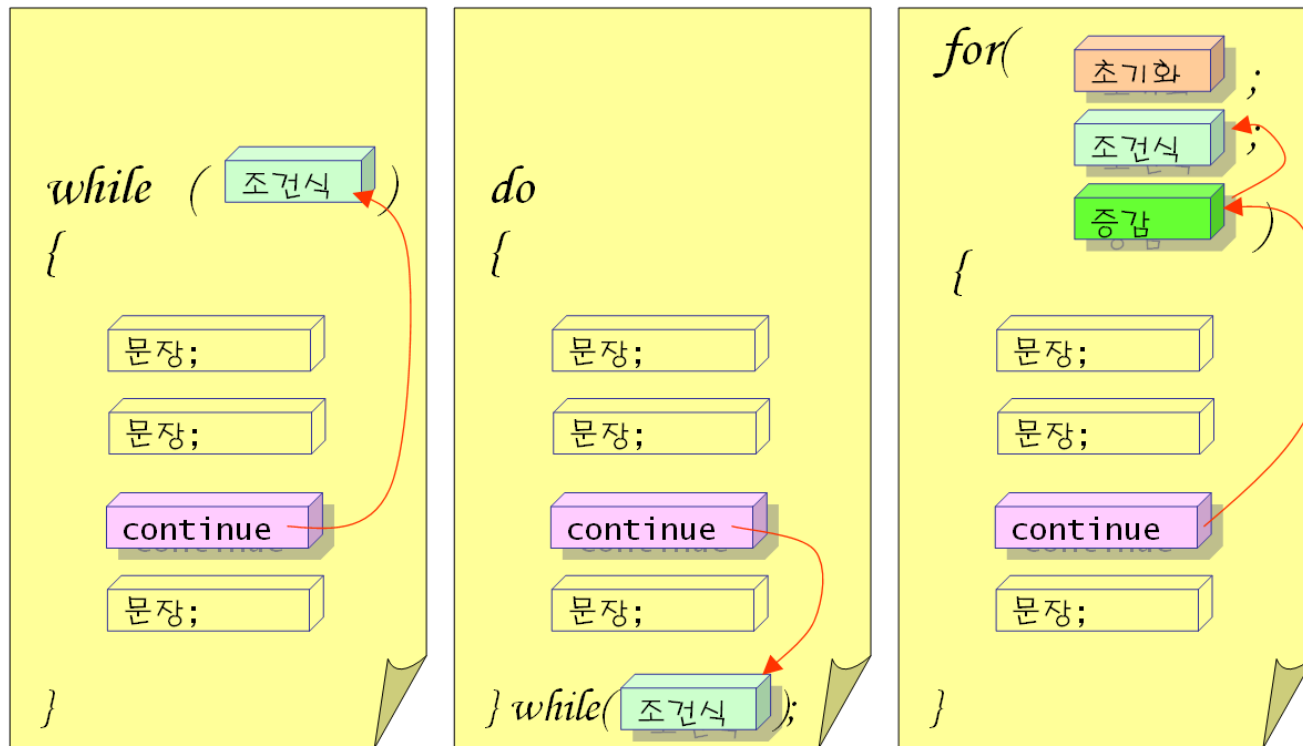


학생 성적을 입력하시오: 90
학생 성적을 입력하시오: 90
학생 성적을 입력하시오: 80
학생 성적을 입력하시오: 70
학생 성적을 입력하시오: -1
학생들의 성적의 평균은 82.500000입니다.



continue 문

- 현재의 반복을 중단하고 다음 반복을 시작하게 한다.





continue 문

- 현재의 반복을 중단하고 다음 반복을 시작하게 한다.

```
int main()
{
    int i=0, sum = 0;
    while( i < 100 ) {
        if( i%2 == 1 )
        {
            i++;
            continue;
        }
        sum += i;
        i++;
    }
    cout << "sum = " << sum << endl;
    return 0;
}
```




예제



// 소문자를 대문자로 변경한다.

```
#include <iostream>
using namespace std;
```

```
int main()
{
```

```
    char letter;
```

```
    while(1)
    {
```

```
        cout << "소문자를 입력하시오: ";
        cin >> letter;          // 공백 문자 제외
```

```
        if( letter == 'Q' )
            break;
```

```
        if( letter < 'a' || letter > 'z' )
            continue;
```

```
        letter -= 32;                // 소문자 -> 대문자
        cout << "변환된 대문자는 " << letter << "입니다.\n";
```

```
    }
```

```
    return 0;
```

소문자를 입력하시오: **a**
변환된 대문자는 **A**입니다.
소문자를 입력하시오: **b**
변환된 대문자는 **B**입니다.
소문자를 입력하시오: **c**
변환된 대문자는 **C**입니다.
소문자를 입력하시오: **Q**



중간 점검 문제

1. 다음 코드의 출력을 쓰시오.

```
int n = 12;  
while (n > 0) {  
    n = n - 2;  
    if( n == 6 )  
        break;  
    cout << n;  
}
```



2. 1번 문제에서 `break`를 `continue`로 변경하면 어떻게 되는가?



Q & A

