

1

-

# Conditional Statements

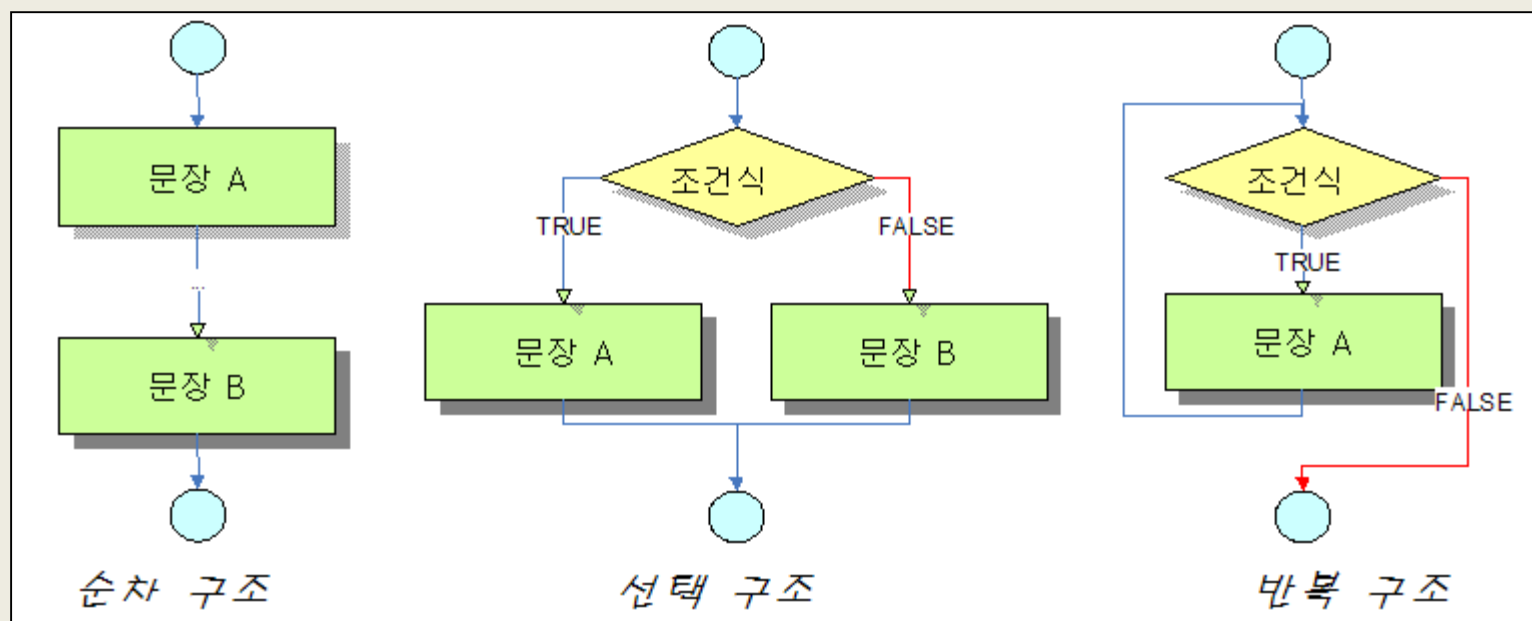


6

-



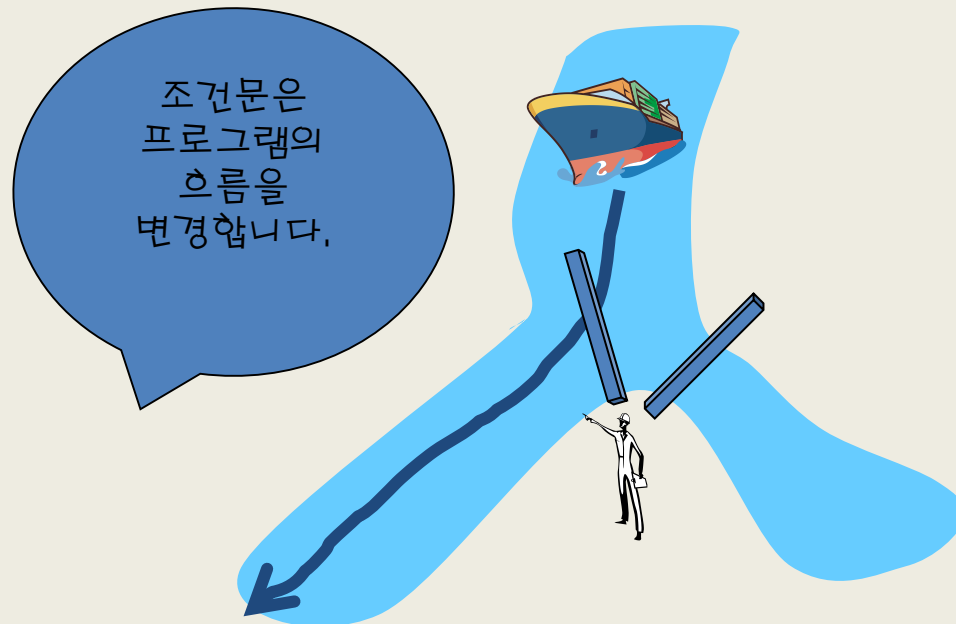
# 3 가





## 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.





# (conditional statement)

- ❑ 여러 상황의 조건들이 있는 경우, 각 조건에 맞춰 다른 작업을 진행하는 작업의 흐름 제어 방식
- ❑ 조건의존 흐름제어 문(conditional flow control statements)
  - ❑ `if(condition)`
  - ❑ `if(condition) – else`
  - ❑ `if(condition) – elseif(condition) – elseif(condition) -...-else`
  - ❑ `switch(condition) – case C1: - case C2: - .... – default:`
- ❑ 조건문 내부에 또 다른 조건문이 포함될 수 있다



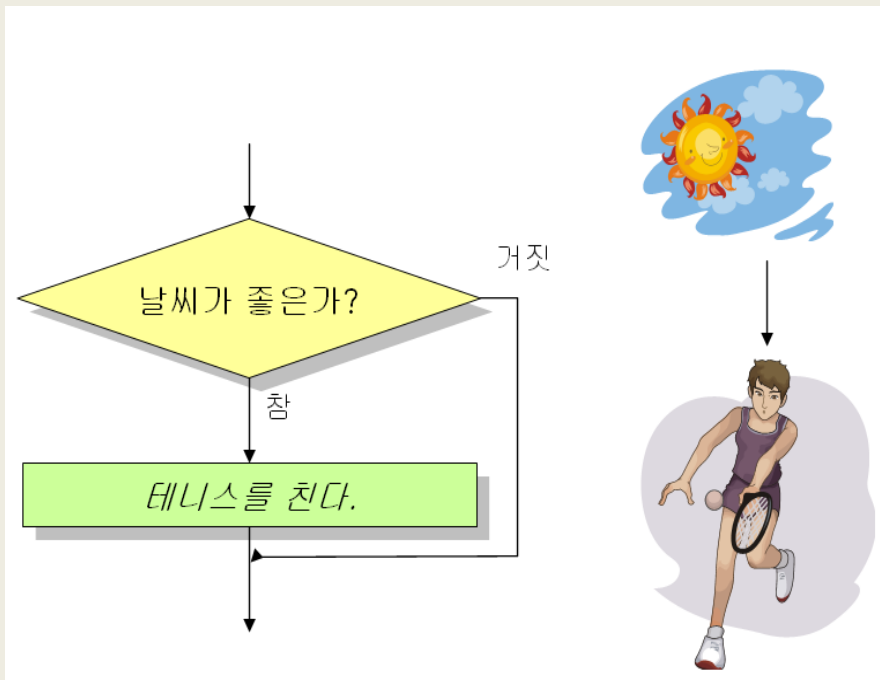
# if

- 주어진 조건에 따라서 해당 작업을 수행하는가, 하지 않는가 결정을 내리는 경우

**if( 조건식 )**  
**문장;**

조건식이 참으로 계산되면

문장이 실행된다.





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

```
    bonus = 200;
```

sales가 2000보다 크면

bonus에 200을 대입한다.

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

```
    printf("합격입니다.\n");
```

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

```
    printf("놀이기구를 탈 수 있습니다.\n");
```

❑ if 문이 끝나면 if 문 다음 문장이 실행된다.

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

```
    printf("현재 영하입니다.\n");           // 조건이 참일 때만 실행
```

```
printf("현재 온도는 %도 입니다.\n", temperature); // 항상 실행
```



- ❑ 복합문(compound statement) → 블록(block)
  - ❑ 2개 이상의 statement가 {, } 를 사용하여 블록화 한 것
  - ❑ 단일문 대신 들어 갈 수 있다
  - ❑ 조건이 참일 경우 수행하는 작업이 2개 이상의 statement로 구성된다면, 반드시 블록화 하여야 한다

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

조건식이 참이면 2개의  
문장이 묶여서 실행된다.



## 예제 #1

```
// if 문을 사용하여 절대값을 구하는 프로그램
#include <stdio.h>

int main(void)
{
    int number;

    printf("정수를 입력하시오:");
    scanf("%d", &number);

    if( number < 0 )
        number = -number;

    printf("절대값은 %d 입니다.\n", number);

    return 0;
}
```

정수를 입력하시오:-5  
절대값은 5 입니다.





## 예제 #2

```
// if 문을 사용하여 음수와 양수를 구별하는 프로그램  
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int number;
```

```
    printf("정수를 입력하시오:");
```

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

```
    if( number > 0 )
```

```
        printf("입력된 정수는 양수입니다.\n");
```

```
    if( number == 0 )
```

```
        printf("입력된 정수는 0입니다.\n");
```

```
    if( number < 0 )
```

```
        printf("입력된 정수는 음수입니다.\n");
```

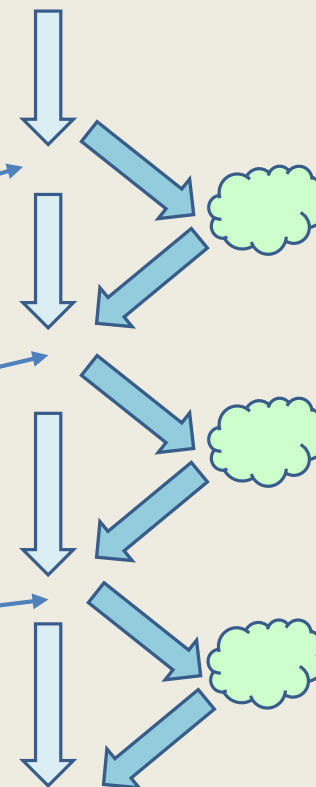
```
    return 0;
```

```
}
```

사용자가  
25를  
입력하였다  
면

25

입력된 정수는 양수입니다.





## 예제 #3

// if 문을 사용하여 홀수와 짝수를 구별하는 프로그램

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

```
int main(void)
```

```
{
```

```
    int number;
```

```
    printf("정수를 입력하시오:");
```

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

```
    if( number % 2 == 1 )
```

```
        printf("입력된 정수는 홀수입니다.\n");
```

```
    if( number % 2 == 0 )
```

```
        printf("입력된 정수는 짝수입니다.\n");
```

```
    return 0;
```

```
}
```

사용자가  
23를  
입력하였다면

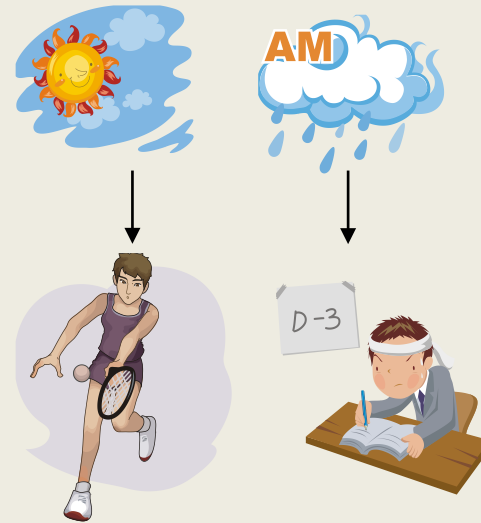
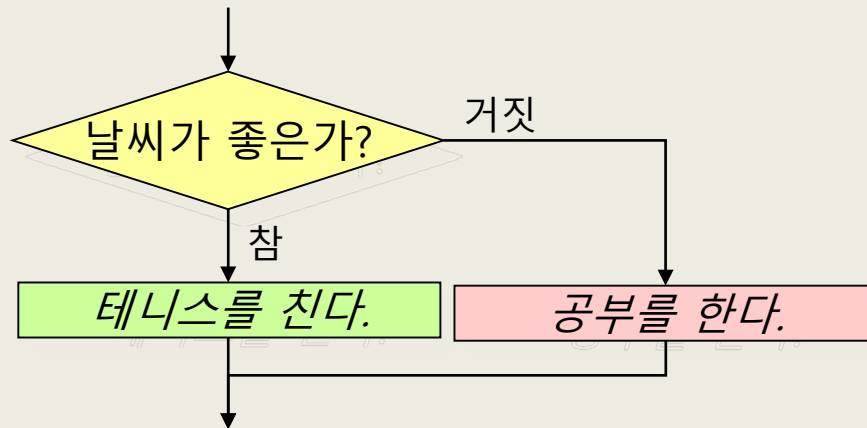
문제점이 보이는가?

정수를 입력하시오:23  
입력된 정수는 홀수입니다.



## if-else

- 주어진 조건에 따라서 서로 다른 작업을 수행하는 경우 사용하는 conditional statement



**if( 조건식 )**

문장 1;

**else**

문장 2;

조건식이 참이면 실행된다.

조건식이 거짓이면  
실행된다.

```
if ( score >= 60 )  
    printf("합격입니다.\n");  
else  
    printf("불합격입니다.\n");
```

score가 60이상이면 실행

score가 60미만이면 실행

```
if ( score >= 60 )  
{  
    printf("합격입니다.\n");  
    printf("장학금도 받을 수 있습니다.\n");  
}  
else  
{  
    printf("불합격입니다.\n");  
    printf("공부하세요.\n");  
}
```

score가 60이상이면 실행

score가 60미만이면 실행



## 예제 #1

// if-else 문을 이용하여 홀수와 짝수를 구분한다.

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

```
int main(void)
```

```
{
```

```
    int number;
```

```
    printf("정수를 입력하시오:");
```

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

```
    if( number % 2 == 0 )
```

```
        printf("입력된 정수는 짝수입니다.\n");
```

```
    else
```

```
        printf("입력된 정수는 홀수입니다.\n");
```

```
    return 0;
```

```
}
```

2로 나누어서 나머지가 0이면  
짝수이다.

정수를 입력하시오: 50  
입력된 정수는 짝수입니다.



## 예제 #2

// 나눗셈을 하기 전에 분모가 0인지를 if-else 문을 이용하여 검사

#include <stdio.h>

int main(void)

{

int n1, n2, n3;

printf("피젯수 정수를 입력하시오: ");

scanf(" %d ", &n1);

printf("젯수 정수를 입력하시오: ");

scanf("%d", &n2);

if( n2 == 0 )

{

printf("0으로 나눌 수는 없습니다.\n");

}

else

{

n3 = n1 / n2;

printf("결과는 %d입니다.\n", n3);

}

return 0;

}

정수를 입력하시오: 5

정수를 입력하시오: 4

결과는 1입니다.

// 분모가 0인지 검사

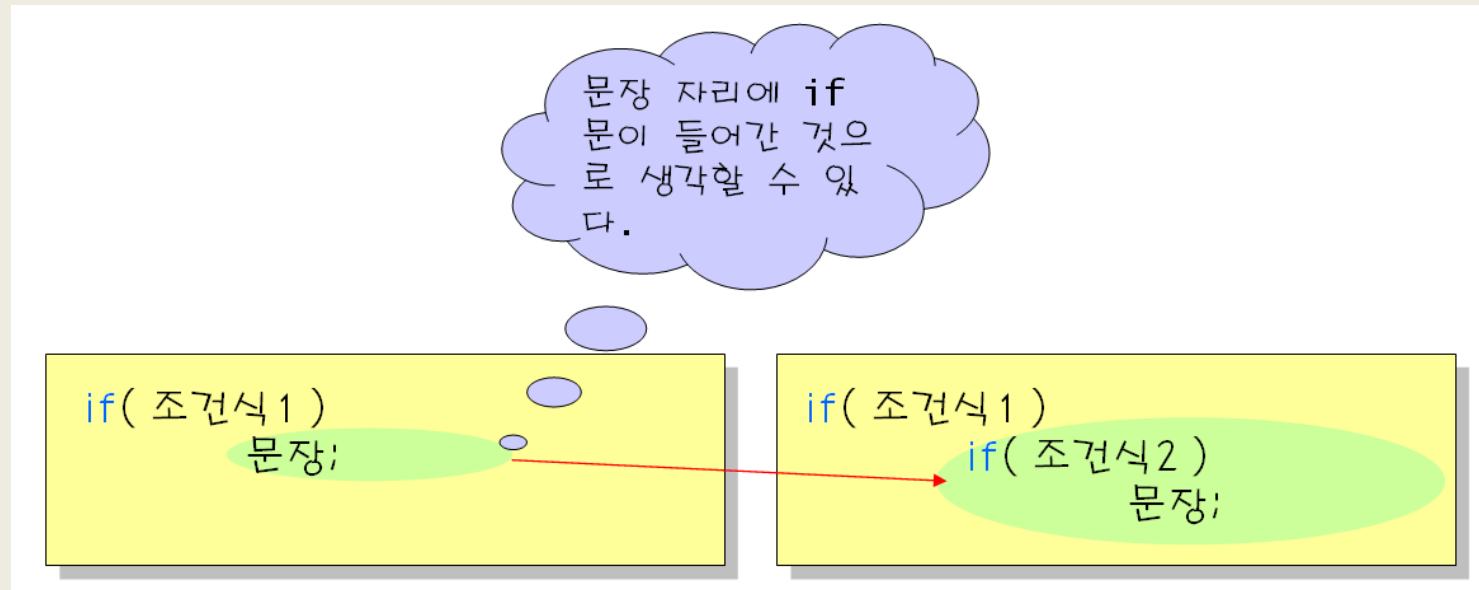
// 나눗셈 실행



## 중첩 if (nested if)

- if 조건문 내부에 또 다른 if 조건문을 statement로서 포함

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





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

if 문안의 문장자리에 if문이  
들어간 경우

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

if 문안의 문장자리에 if-else  
문이 들어간 경우





## if와 else의 대응 관계

else 절은 가장 가까운 if절과 매치된다.

if(score > 80)

if( score > 90)

printf("당신의 학점은 A입니다\n");

else

printf("당신의 학점은 B입니다\n")

if( score > 80 )

{

if( score > 90 )

printf("당신의 학점은 A입니다.\n");

}

else

printf("당신의 학점은 A나 B가 아닙니다.\n");

만약 다른 if절과 else 절을 매치시키려면 중괄호를 사용하여 블록으로 묶는다.

## 정수 3개중의 최소값 찾기

```
#include <stdio.h>
int main(void)
{
    int n1, n2, n3, min;

    printf("정수 3개를 입력하시오:");
    scanf("%d %d %d", &n1, &n2, &n3);

    if( n1 < n2 )
    {
        if( n1 < n3 )
            min = n1;
        else
            min = n3;
    }
    else
    {
        if( n2 < n3 )
            min = n2;
        else
            min = n3;
    }

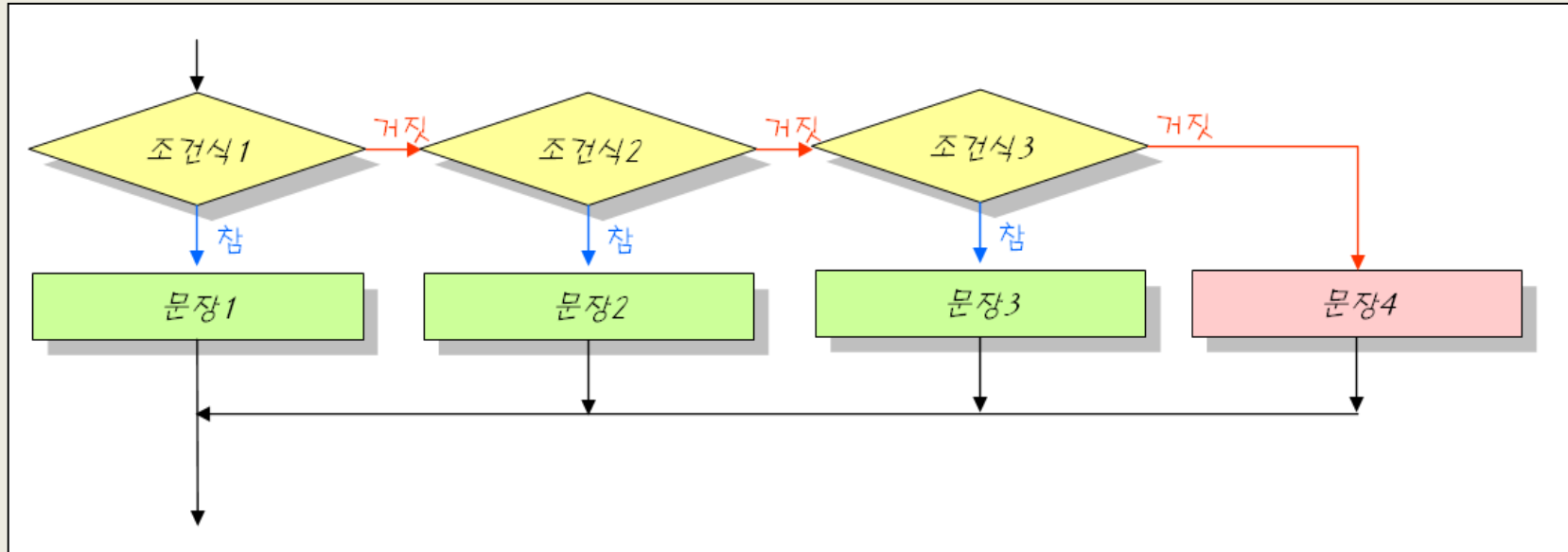
    printf("제일 작은 수는 %d입니다\n", min);
    return 0;
}
```

If-else 절 안에 다른 if-else절이 포함될 수 있다.

정수 3개를 입력하시오:10 20 30  
제일 작은 수는 10입니다.



## 연속적인 if



```
if( 조건식 1 )  
    문장 1;  
else if( 조건식 2 )  
    문장 2;  
else if( 조건식 3 )  
    문장 3;  
else  
    문장 4;
```



- ❑ 성적을 받아서 연속적인 if를 사용하여 학점을 매기는 코드

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

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



// 문자들을 분류하는 프로그램

#include <stdio.h>

int main(void)

{

    char ch;

    printf("문자를 입력하시오: ");

    scanf("%c", &ch);

    if( ch >= 'A' && ch <= 'Z' )

        printf("%c는 대문자입니다.\n", ch);

    else if( ch >= 'a' && ch <= 'z' )

        printf("%c는 소문자입니다.\n", ch);

    else if( ch >= '0' && ch <= '9' )

        printf("%c는 숫자입니다.\n", ch);

    else

        printf("%c는 기타문자입니다.\n", ch);

    return 0;

}

문자를 입력하시오: c  
c는 소문자입니다.



## 단순 계산기 예제

// 간단한 산술 계산기 프로그램

#include <stdio.h>

int main(void)

{

char op;

int x, y;

printf("수식을 입력하시오: ");

scanf("%d %c %d", &x, &op, &y);

if( op == '+' )

printf("%d %c %d = %d\n", x, op, y, x + y);

else if( op == '-' )

printf("%d %c %d = %d\n", x, op, y, x - y);

else if( op == '\*' )

printf("%d %c %d = %d\n", x, op, y, x \* y);

else if( op == '/' )

printf("%d %c %d = %d\n", x, op, y, x / y);

else if( op == '%' )

printf("%d %c %d = %d\n", x, op, y, x % y);

else

printf("지원되지 않는 연산자입니다. \n");

return 0;

}

수식을 입력하시오: 1 + 2

1 + 2 = 3

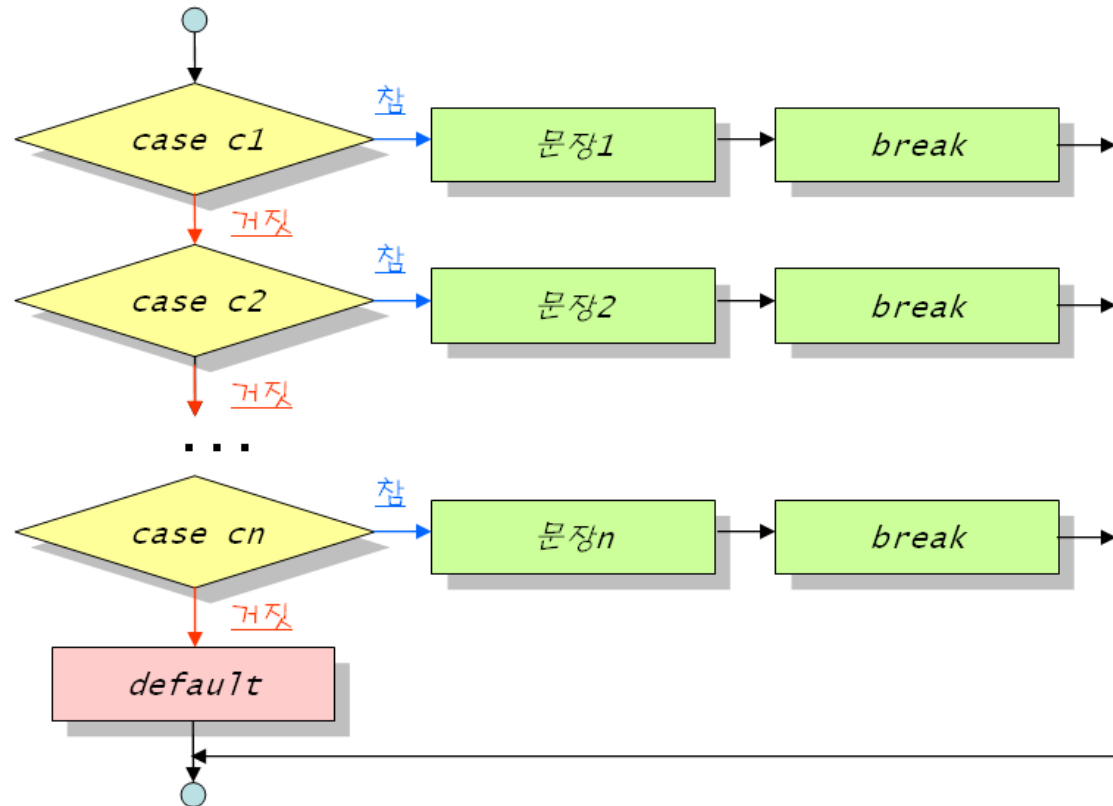


## switch 문

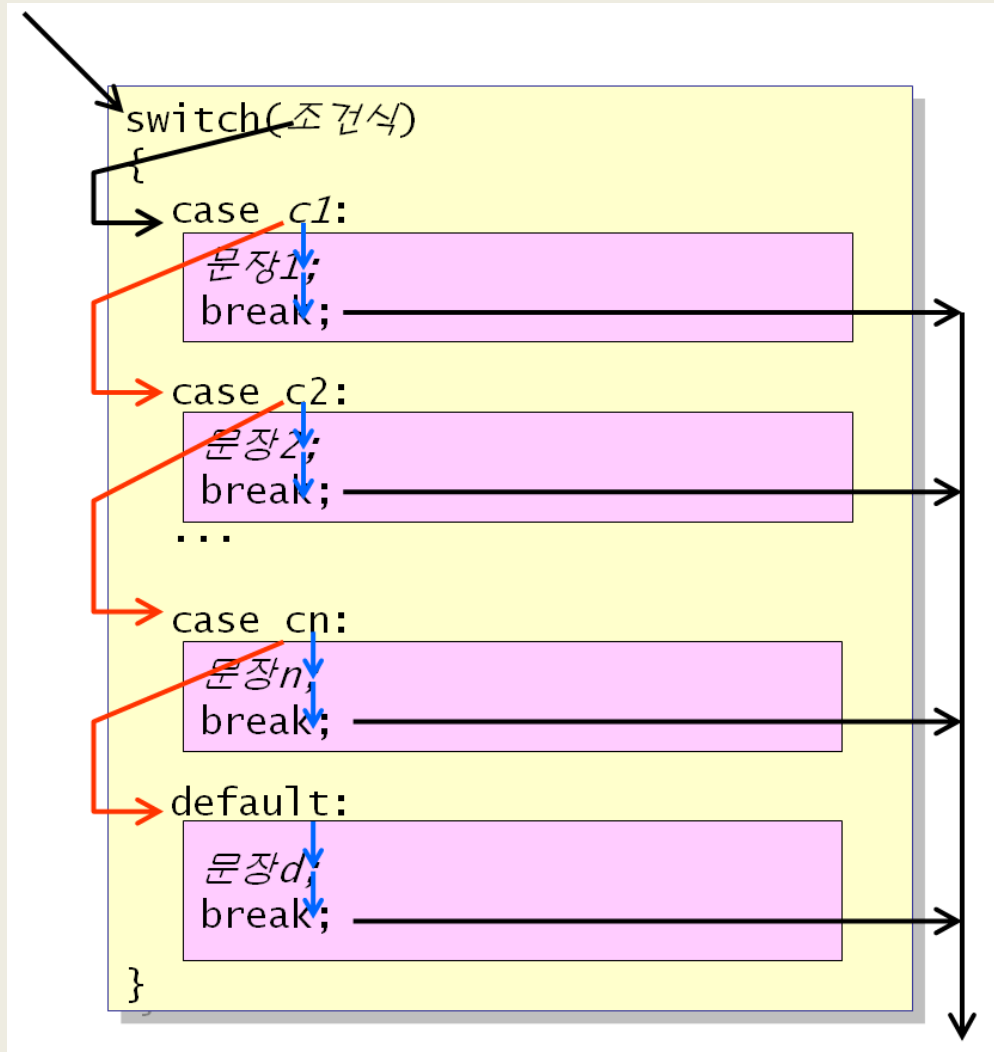
- 여러가지 경우 중에서 하나를 선택하는데 사용

**switch**(조건식)

```
{  
  case c1:  
    문장 1;  
    break;  
  case c2:  
    문장 2;  
    break;  
  ...  
  case cn:  
    문장 n;  
    break;  
  default:  
    문장 d;  
    break;  
}
```



## switch 문의 실행 순서



break는 문장의  
실행을 끝내고  
switch 문을  
탈출하는데  
사용





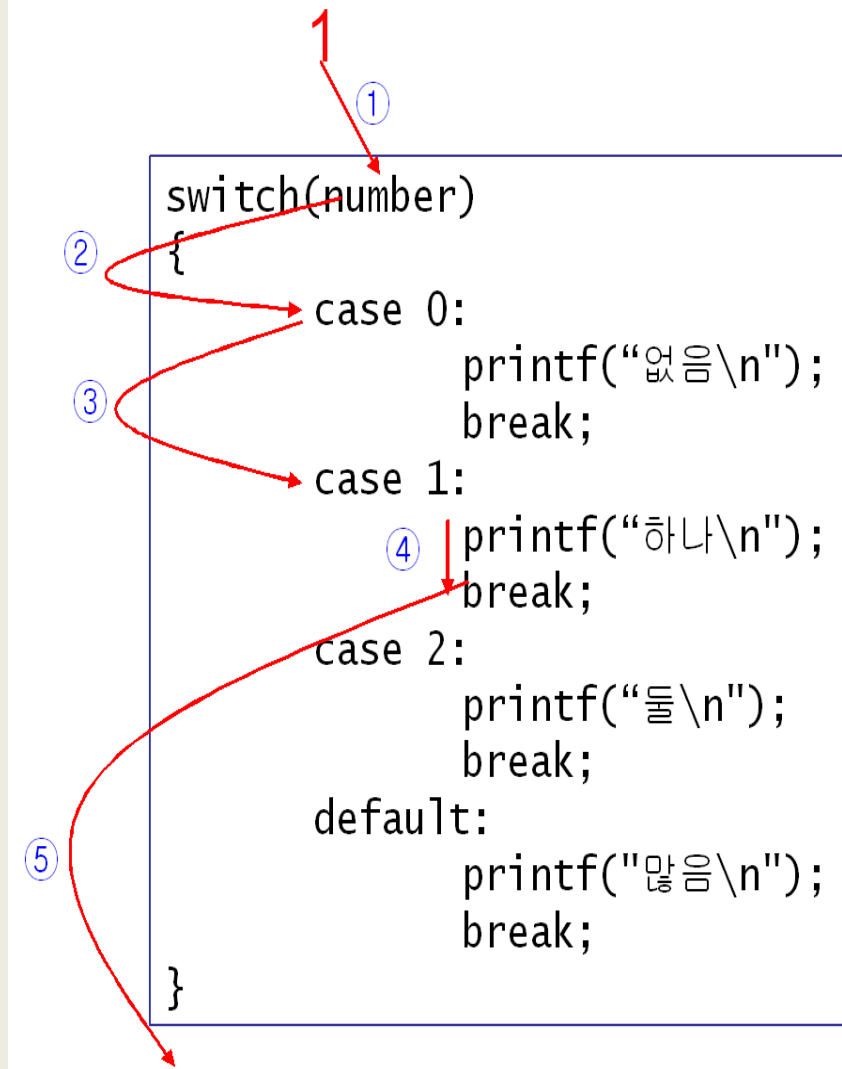
## Break의 역할

```
int main(void)
{
    int number;

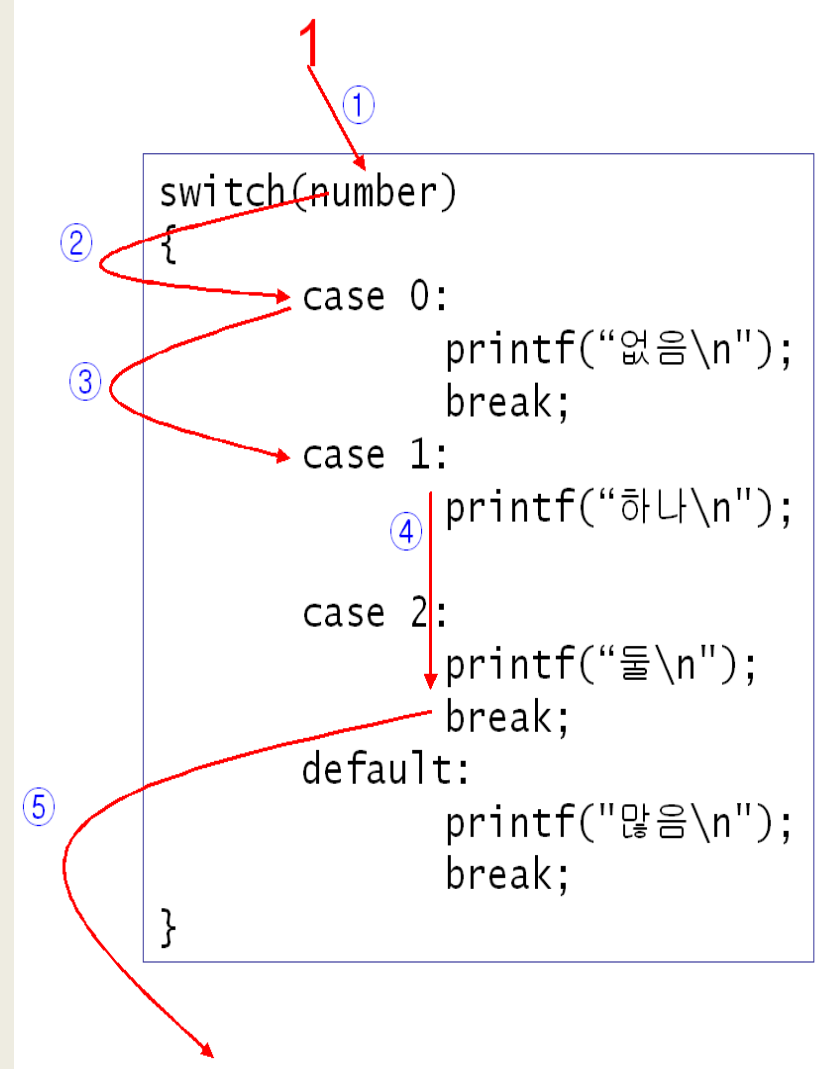
    printf("정수를 입력하시오:");
    scanf("%d", &number);
    switch(number)
    {
        case 0:
            printf("없음\n");
            break ;
        case 1:
            printf("하나\n");
            break ;
        case 2:
            printf("둘\n");
            break ;
        default:
            printf("많음\n");
            break;
    }
}
```

정수를 입력하시오: 1  
하나

사용자가 1을 입력하는 경우



break를 빼먹으면...





## 의도적인 break생략

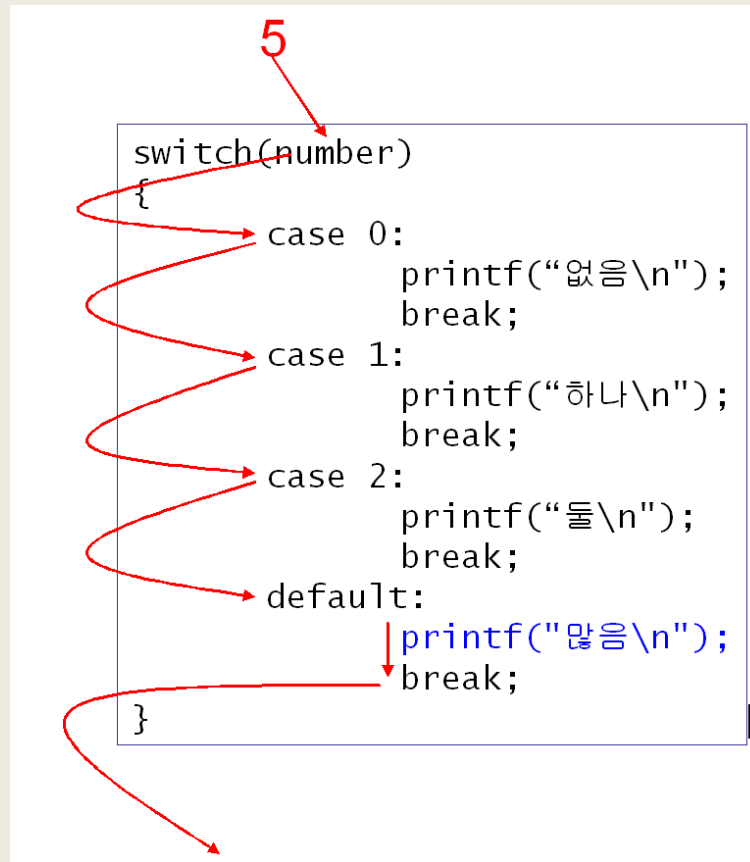
```
switch(number)
{
    case 0:
        printf("없음\n");
        break;
    case 1:
        printf("하나\n");
        break;
    case 2:
    case 3:
        printf("두서너개\n");
        break;
    default:
        printf("많음\n");
        break;
}
```

2개의 경우를  
하나로 묶어서  
처리하기  
위하여 이러한  
기법을 사용



## default 문

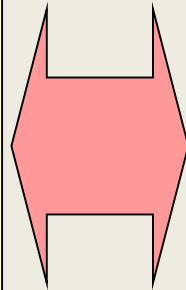
- 어떤 case문과도 일치되지 않는 경우에 선택



## switch 문과 if-else 문

```
int main(void)
{
    int number;
    scanf("%d", &number);

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



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



## 계산기 예제

```
#include <stdio.h>
int main(void)
{
    char op;
    int x, y;

    printf("수식을 입력하시오: ");
    scanf("%d %c %d", &x, &op, &y);

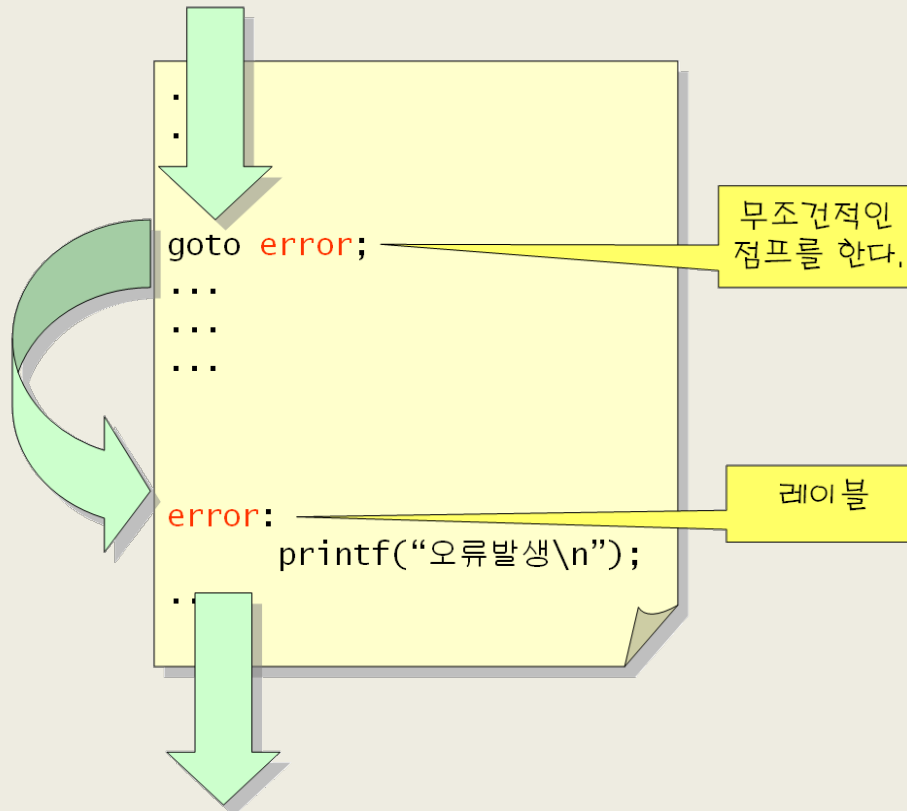
    switch(op)
    {
        case '+':
            printf("%d %c %d = %d \n", x, op, y, x + y);
            break;
        case '-':
            printf("%d %c %d = %d \n", x, op, y, x - y);
            break;
        case '*':
            printf("%d %c %d = %d \n", x, op, y, x * y);
            break;
        case '/':
            printf("%d %c %d = %d \n", x, op, y, x / y);
            break;
        case '%':
            printf("%d %c %d = %d \n", x, op, y, x % y);
            break;
        default:
            printf("지원되지 않는 연산자입니다. \n");
            break;
    }
    return 0;
}
```

수식을 입력하시오: 10 % 3  
10 % 3 = 1



# Goto문

- ❑ 조건없이 어떤 위치로 점프(unconditional jump)
- ❑ 사용하지 않는 것이 좋음





## Goto문을 사용하여 반복을 시키는 예제

```
// 구구단출력프로그램
#include <stdio.h>

int main(void)
{
    int i = 1;

loop:
    printf("%d * %d = %d \n", 3, i, 3 * i);
    i++;
    if( i == 10 ) goto end;
    goto loop;

end:
    return 0;
}
```

```
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
```