



## CH04 조건문



# 이번 장에서 학습할 내용



- 조건문이란?
- if 문
- if, else 문
- 중첩 if 문
- switch 문
- break문
- continue문
- goto문

필요에 따라서  
조건이 만족되면  
문장의 실행 순서를  
변경할 수 있는  
기능이 제공됩니다.





- 만약 프로그램에 선택 구조가 없다면 프로그램은 항상 동일한 동작만을 되풀이 할 것이다.





# 제어문

## 제어문

### 조건문

연봉이 25000이상이면  
취업하고 아니면 고시  
준비해야지



### 반복문

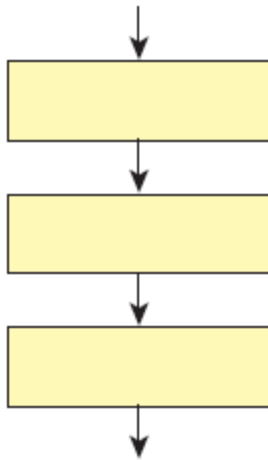
토익 점수가 600을  
넘을 때까지는  
반복해서 공부해야지



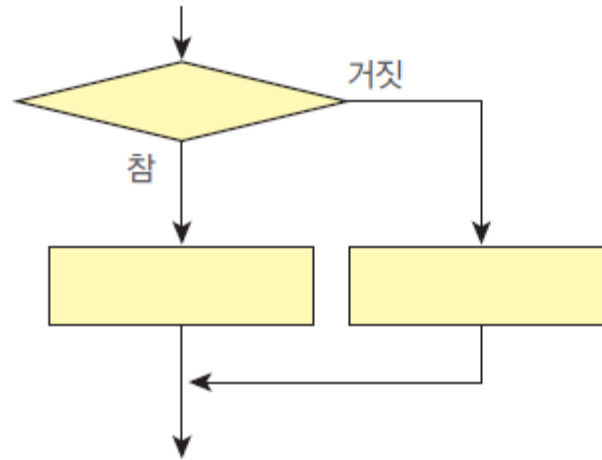


# 3가지의 제어구조

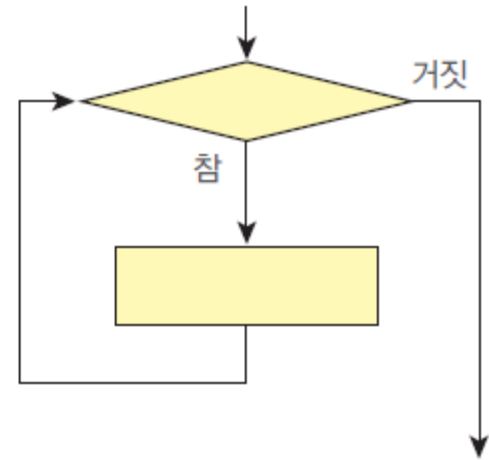
순차구조



선택구조

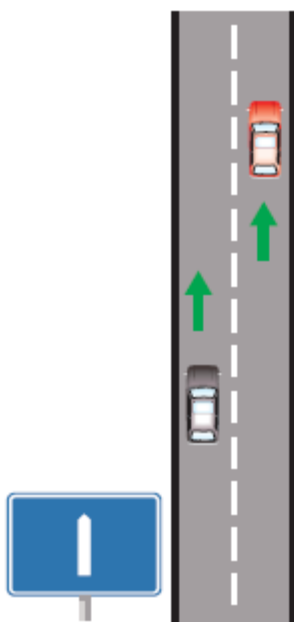


반복구조

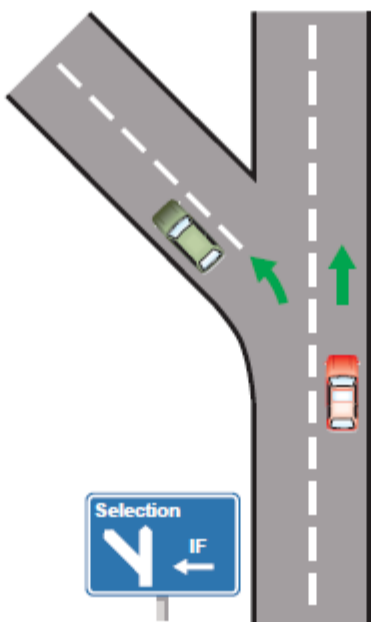




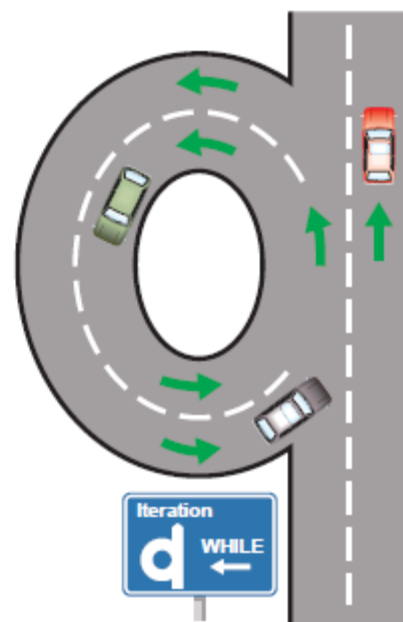
# 자동차(CPU)가 주행하는 도로로 생각하자.



순차 구조

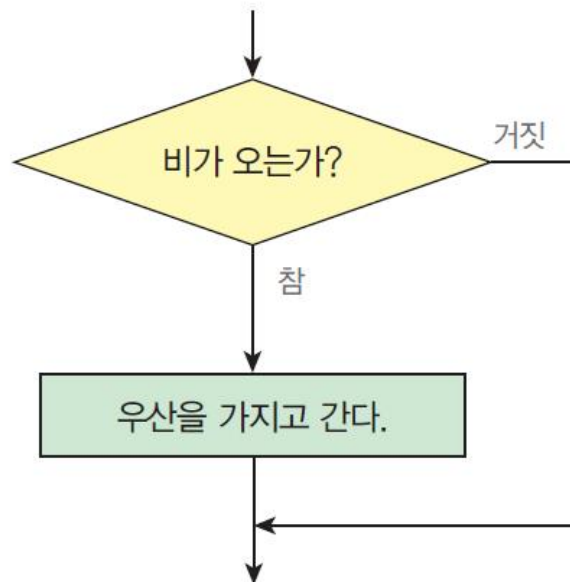


선택 구조



반복 구조

- 일상생활에서도 조건에 따라서 결정을 내려야 하는 경우는 많이 있다.





# if문의 구조

형식

```
if(조건식)
```

```
{
```

```
    실행문;
```

```
    .
```

```
    .
```

```
    실행문;
```

```
}
```

```
실행문2;
```





# if문의 구조

조건식이 거짓일  
경우 {} 이후 코드  
실행

조건식이 참이면 {}  
사이의 코드 실행

```
if(조건식)
```

```
{
```

```
    실행문;
```

```
    .
```

```
    .
```

```
    실행문;
```

```
}
```

```
실행문2;
```

예

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

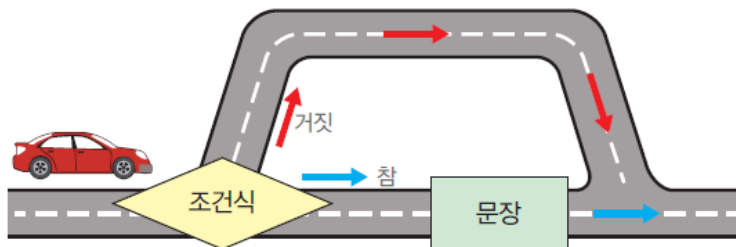
```
{
```

```
    printf("%d\n",number);
```

```
    printf("양수");
```

```
}
```

```
printf("if 종료");
```





# if문의 예

number 가 0보다 크면

```
if( number > 0 )  
{  
    printf("양수입니다\n");  
}
```

“양수입니다”를 출력한다.

```
if ( temperature < 0 )  
    printf("현재 영하입니다.\n");    // 조건이 참일 때만 실행
```

if 문이 끝나면 if 문 다음 문장이 실행된다.

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

```
if ( temperature < 0 )  
{  
    printf("현재 영하입니다.\n");    // 조건이 참일 때만 실행  
}  
printf("현재 온도는 %도 입니다.\n", temperature); // 항상 실행
```



# 예제

ch04\_ex1.c

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main(void)
{
    int number;

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

    if( number > 0 )
    {
        printf("양수입니다.");
    }
    printf("입력된 값은 %d입니다.", number);

    return 0;
}
```

정수를 입력하시오:25  
양수입니다.  
입력된 값은 25입니다.



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

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

조건식이 참이면 2개의  
문장이 묶여서 실행된다.



## 오류 주의

### 경고: 오류 주의 #1

다음과 같이 if 문장의 조건식 뒤에 세미콜론을 찍으면 안 된다. if 문장은 조건식과 문장이 합쳐서 하나의 문장을 이룬다. 아래와 같이 작성하면 if 문은 `if( x > 0 );`로 끝나고 `printf` 문장은 조건에 관계없이 실행된다.

```
if( x > 0 );  
    printf("양수입니다.\n");
```

### 경고: 오류 주의 #2

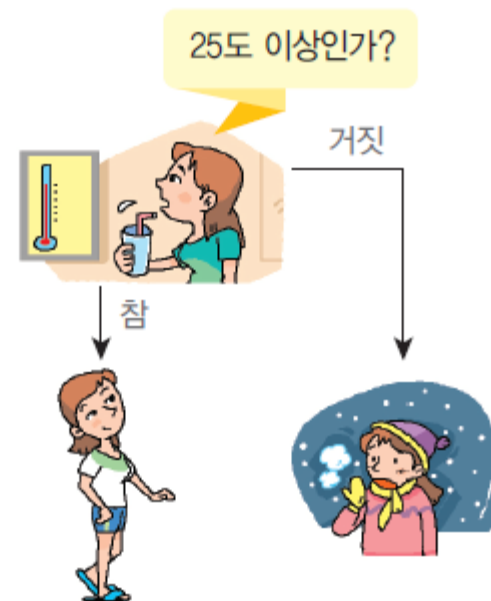
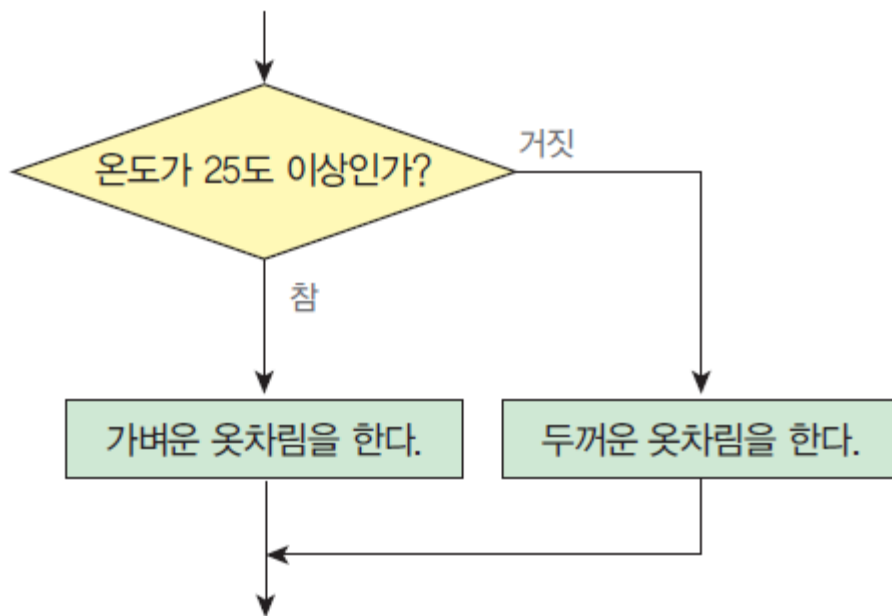
아주 많이 하는 오류가 두 값을 비교할 때 `==` 연산자를 사용하지 않고 `=` 연산자를 사용하는 것이다. 이 경우에는 비교가 되지 않고 값이 단순히 변수에 대입된다. 대입된 값에 따라서 참과 거짓이 결정된다.

```
if( x = 0 )  
    printf("x가 0이다.");
```

이 경우에는 `x`에 0이 대입되어서 항상 거짓이 된다. `x == 0`으로 작성하여야 한다. 이러한 오류를 방지하기 위하여 어떤 사람들은 `0 == x`와 같이 적는다. 만약 `0 = x`가 되면 문법 오류가 발생한다.



# if-else 문





# if-else 문

형식

```
if(조건식)
{
    실행문;
    .
    .
    실행문;
}
```

실행문2;

형식

```
if(조건식)
{
    실행문;
    .
    .
    실행문;
}
```

```
else
{
    실행문;
    .
    .
    실행문;
}
```

실행문2;



# if-else 문

형식

if(조건식)

{

실행문;

}

else

{

실행문;

}

실행문2;

조건식이 참이면 if  
다음의 {} 사이의  
코드 실행

예

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

```
{
```

```
    printf("양수");
```

```
}
```

```
else
```

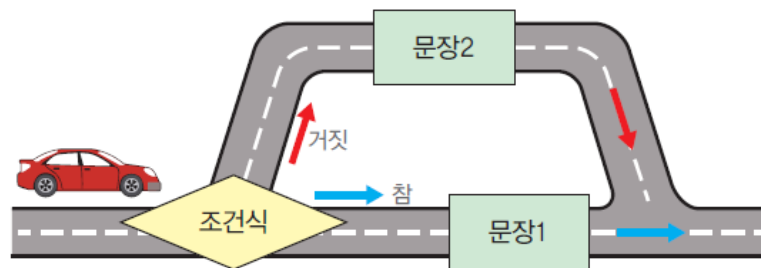
```
{
```

```
    printf("음수");
```

```
}
```

```
printf("조건문 종료");
```

조건식이 거짓이면  
else 다음의 {} 사이의  
코드 실행







# 예제 #1

ch04\_ex3.c

```
#define _CRT_SECURE_NO_WARNINGS
#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: 23  
입력된 정수는 홀수입니다.



## 예제 #3

ch04\_ex4.c

```
#define _CRT_SECURE_NO_WARNINGS
// 3의 배수인수 판단
#include <stdio.h>

int main(void)
{
    int num;
    printf("숫자를 입력하시오: ");
    scanf("%d", &num);

    if (num % 3 == 0)
    {
        printf("3의 배수임.\n", num);
    }
    else
    {
        printf("3의 배수 아님\n", num);
    }
    return 0;
}
```

숫자를 입력하시오: 6  
3의배수임



## 예제 #3

ch04\_ex5.c

```
#define _CRT_SECURE_NO_WARNINGS
// 2의 배수 이면서 3의 배수인수 판단
#include <stdio.h>
```

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

```
    int num;
    printf("숫자를 입력하시오: ");
    scanf("%d", &num);
```

```
    if (num % 2 == 0 && num % 3 == 0)
    {
```

```
        printf("2의 배수 이면서 3의 배수임.\n", num);
```

```
    }
```

```
    else
    {
```

```
        printf("2의 배수 이면서 3의 배수 아님\n", num);
```

```
    }
```

```
    return 0;
```

```
}
```

숫자를 입력하시오: 6  
2의 배수이면서 3의 배수임



# 연속적인 if

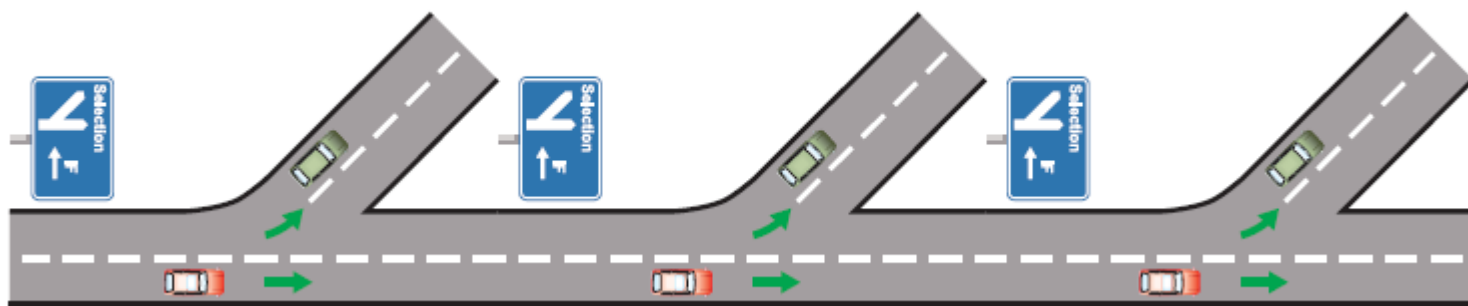
```
if(조건식)
{
    실행문;
    .
    .
    실행문;
}
else
{
    실행문;
    .
    .
    실행문;
}
실행문2;
```

Syntax: 연속적인 if문

문법

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

만약 조건식1이 참이면 문장1이 실행된다.  
그렇지 않고 조건식2가 참이면 문장2가 실행된다.  
그렇지 않고 조건식3가 참이면 문장3이 실행된다.  
그렇지 않으면 문장4이 실행된다.





```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main(void)
{
    int score;

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

    if (score >= 90){
        printf("합격: 학점A\n");
    }
    else if (score >= 80){
        printf("합격: 학점B\n");
    }
    else if (score >= 70){
        printf("합격: 학점C\n");
    }
    else if (score >= 60){
        printf("합격: 학점D\n");
    }
    else{
        printf("불합격: 학점F\n");
    }
    return 0;
}
```

성적을 입력하시오: 88  
학점 B



## 중첩 if

- if 문에 다시 if 문이 포함

```
if( 조건식1 )  
    문장;
```

문장 자리에 if문이  
들어간 것으로  
생각할 수 있다.

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



# 중첩 if

```
if( score >= 80 )  
{
```

```
    if( score >= 90 )  
    {
```

```
        printf("당신의 학점은 A입니다.\n");
```

```
    }
```

```
}
```

if 문안의 문장 자리에  
if문이 들어간 경우



# 중첩 if

```
if( score >= 80 )
```

```
{
```

```
    if( score >= 90 )
```

```
    {
```

```
        printf("당신의 학점은 A입니다.\n");
```

```
    }
```

```
    else
```

```
    {
```

```
        printf("당신의 학점은 B입니다.\n");
```

```
    }
```

```
}
```

if 문안의 문장 자리에  
if-else 문이 들어간 경우





# if와 else의 매칭 문제

else 절은 가장 가까운 if절  
과 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");

}

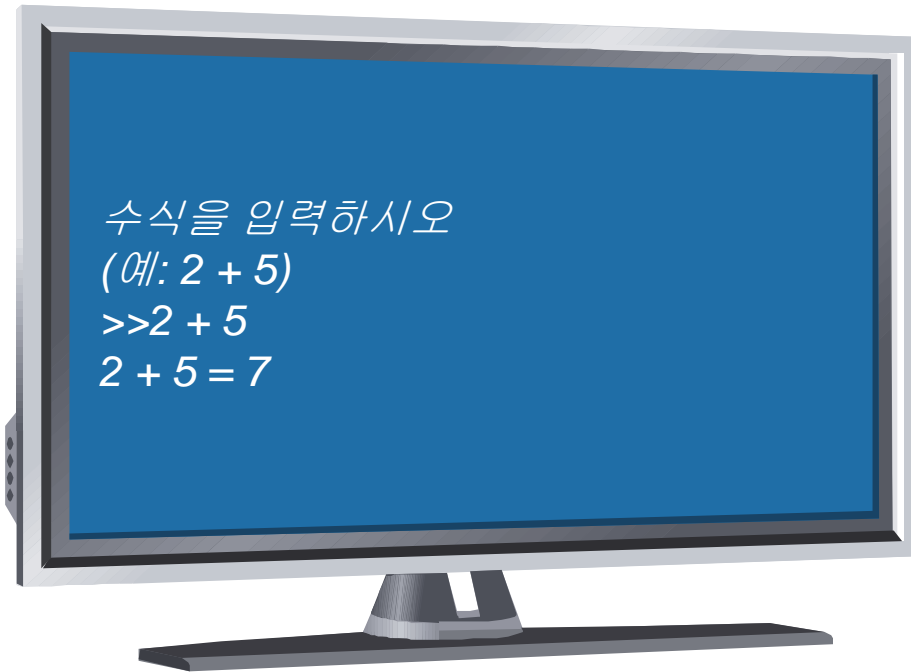
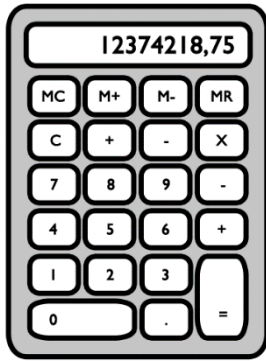
else

printf("당신의 학점은 A나 B가 아닙니다.\n");

만약 다른 if절과 else 절을 매치시키려면  
중괄호를 사용하여 블록으로 묶는다.



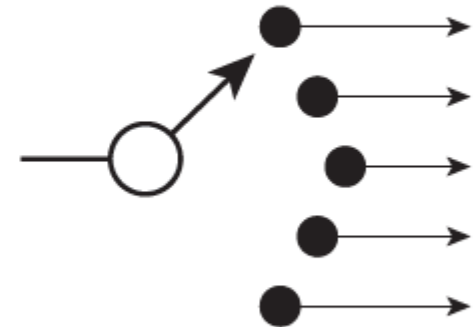
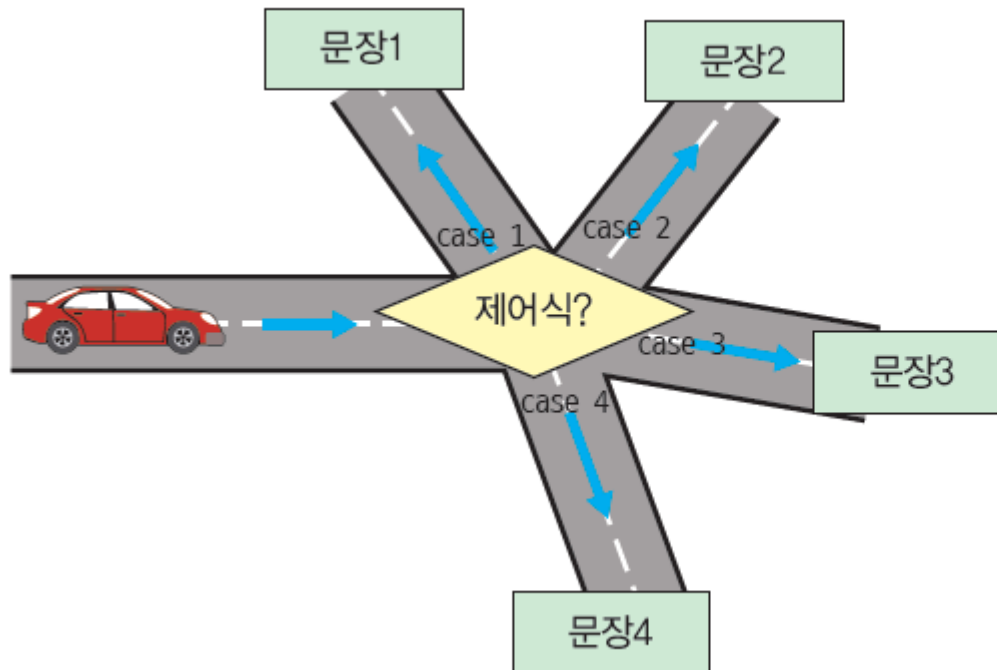
# Lab 1: 산술 계산기





# switch 문

- 제어식의 값에 따라서 여러 경로 중에서 하나를 선택할 수 있는 제어 구조





# switch 문

Syntax: switch 문

c1, c2, c3 에 해당하는 부분에는  
정수, 상수(1,2,3등), 문자 상수('c', 'd')만 사용 가능

문법

```
switch(제어식)
```

```
{
```

```
case c1:
```

```
문장1;
```

```
break;
```

제어식의 값이 c1이면 실행된다.

```
case c2:
```

```
문장2;
```

```
break;
```

제어식의 값이 c2이면 실행된다.

```
...
```

```
default:
```

```
문장d;
```

```
break;
```

일치하는 값이 없으면 실행된다.

```
}
```



## 예제

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

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



사용자가 number에 1을 입력한다.

1

```
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;
```

```
}
```

값이 0이면 이쪽으로 오세요!

값이 1이면 이쪽으로 오세요!

값이 2이면 이쪽으로 오세요!

일치하는 값이 없으면 이쪽으로 오세요!



# break가 생략되는 경우



1

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



# 의도적인 break 생략



2

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





# default



5

```
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;
}
```

값이 0이면 이쪽으로 오세요!

값이 1이면 이쪽으로 오세요!

값이 2이면 이쪽으로 오세요!

일치하는 값이 없으면 이쪽으로 오세요!



# switch 문과 if-else 문

```
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;  
}
```



# switch 문에서 주의할 점

```
switch(number)
{
    case x:                                // 변수는 사용할 수 없다.
        printf("x와 일치합니다. \n ");
        break;
    case (x+2):                            // 변수가 들어간 수식은 사용할 수 없다.
        printf("수식과 일치합니다. \n ");
        break;
    case 0.001:                            // 실수는 사용할 수 없다.
        printf("실수 \n ");
        break;
    case 'a':                              // 문자는 사용할 수 있다.
        printf("문자 \n ");
        break;
    case "001":                            // 문자열은 사용할 수 없다.
        printf("문자열 \n ");
        break;
}
```



# 정수의 범위를 나타낼 때

```
switch (score) {  
    case 100:  
    case 99:  
    case 98:  
    ...  
    case 90:  
        printf("A학점입니다.\n");  
        break;  
    ...  
}
```



```
if( score >= 90 && score <= 100 )  
    printf("A학점입니다.\n");
```

정수의 범위도 표현할 수 있으나 번거롭다.



# 정수의 범위를 나타낼 때

```
int iscore;
...
iscore = score/10;           // 정수 나눗셈의 경우, 나머지는 없어진다.
switch (iscore) {
    case 9: grade = 'A'; break; // 90-100은 A 학점
    case 8: grade = 'B'; break; // 80-89은 B 학점
    case 7: grade = 'C'; break; // 70-79은 C 학점
    case 6: grade = 'D'; break; // 60-69은 D 학점
    default: grade = 'F'; break; // 59점 이하는 F 학점
}
```



switch 문과 if/else 체인 중에서 어떤 것이 더 효율적인가?

차이는 미소하다. 하지만 switch 문은 간략한 점프 테이블로 효율적으로 구현이 가능하도록 설계되었다. 따라서 대부분의 경우 switch를 사용하는 것이 좋다. 코드가 간결하고 아마 약간은 효율적이다.



# 예제

ch04\_ex10.c

```
// 달의 일수를 계산하는 프로그램
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void)
{
    int month, days;

    printf("달을 입력하시오: ");
    scanf("%d", &month);
```



## 예제

```
switch(month)
{
    case 2:
        days = 28;
        break;
    case 4:
    case 6:
    case 9:
    case 11:
        days = 30;
        break;
    default:
        days = 31;
        break;
}
printf("%d월의 일수는 %d입니다.\n", month, days);
return 0;
}
```

달을 입력하시오: 12  
12월의 일수는 31입니다.



# 작업 점검

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

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

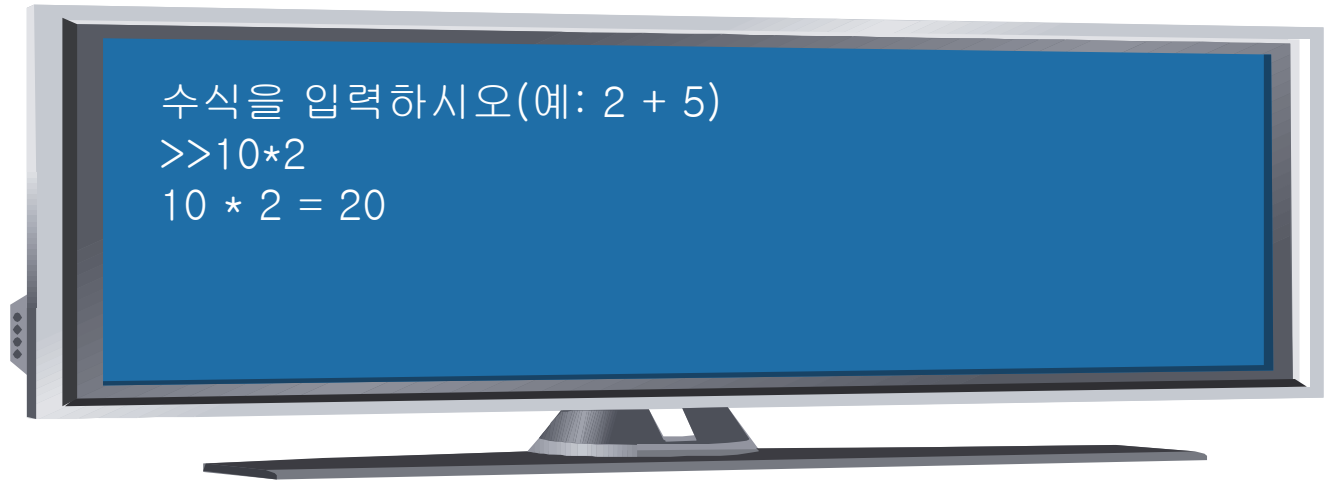






## Lab2: 산술 계산기

- 앞의 산술 계산기 예제를 **switch** 문을 이용하여 다시 작성하여 보자.





# Lab 1

ch04\_lab1.c

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main(void)
{
    char op;
    int x, y, result;
    printf("수식을 입력하세요\n ");
    printf("(예: 2 + 5) \n");
    printf(">>\n");
    scanf("%d %c %d", &x, &op, &y);
```



# Lab 1

ch04\_lab1.c

```
if (op == '+')
    result = x + y;
else if (op == '-')
    result = x - y;
else if (op == '*')
    result = x * y;
else if (op == '/')
    result = x / y;
else if (op == '%')
    result = x % y;
else
    printf("지원되지 않는 연산자입니다. ");

printf("%d %c %d = %d\n ", x, op, y, result);
return 0;

return 0;
}
```



## Lab2: 산술 계산기

ch04\_ex11.c

```
#define _CRT_SECURE_NO_WARNINGS
// 간단한 산술 계산기 프로그램

#include <stdio.h>
int main(void)
{

    char op;
    int x, y, result;

    printf("수식을 입력하시오\n");
    printf("(예: 2 + 5) \n");
    printf(">>");
    scanf("%d %c %d", &x, &op, &y);
```



## Lab2: 산술 계산기

```
switch (op)
{
    case '+':
        result = x + y;
        break;
    case '-':
        result = x - y;
        break;
    case '*':
        result = x * y;
        break;
    case '/':
        result = x / y;
        break;
}
```



## Lab2: 산술 계산기

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

수식을 입력하시오(예: 2 + 5)

>>10\*2

10 \* 2 = 20