

-Part1-

제8장 조건문이란 무엇인가

(교재 199페이지 ~ 224페이지)

학습목차

8.1 조건문이란

-교재 202페이지 -

8.2 조건문을 만드는 방법 1 - if 문

-교재 203페이지 -

8.3 조건문을 만드는 방법 2 - switch~case 문

-교재 213페이지 -

8.4 break와 continue

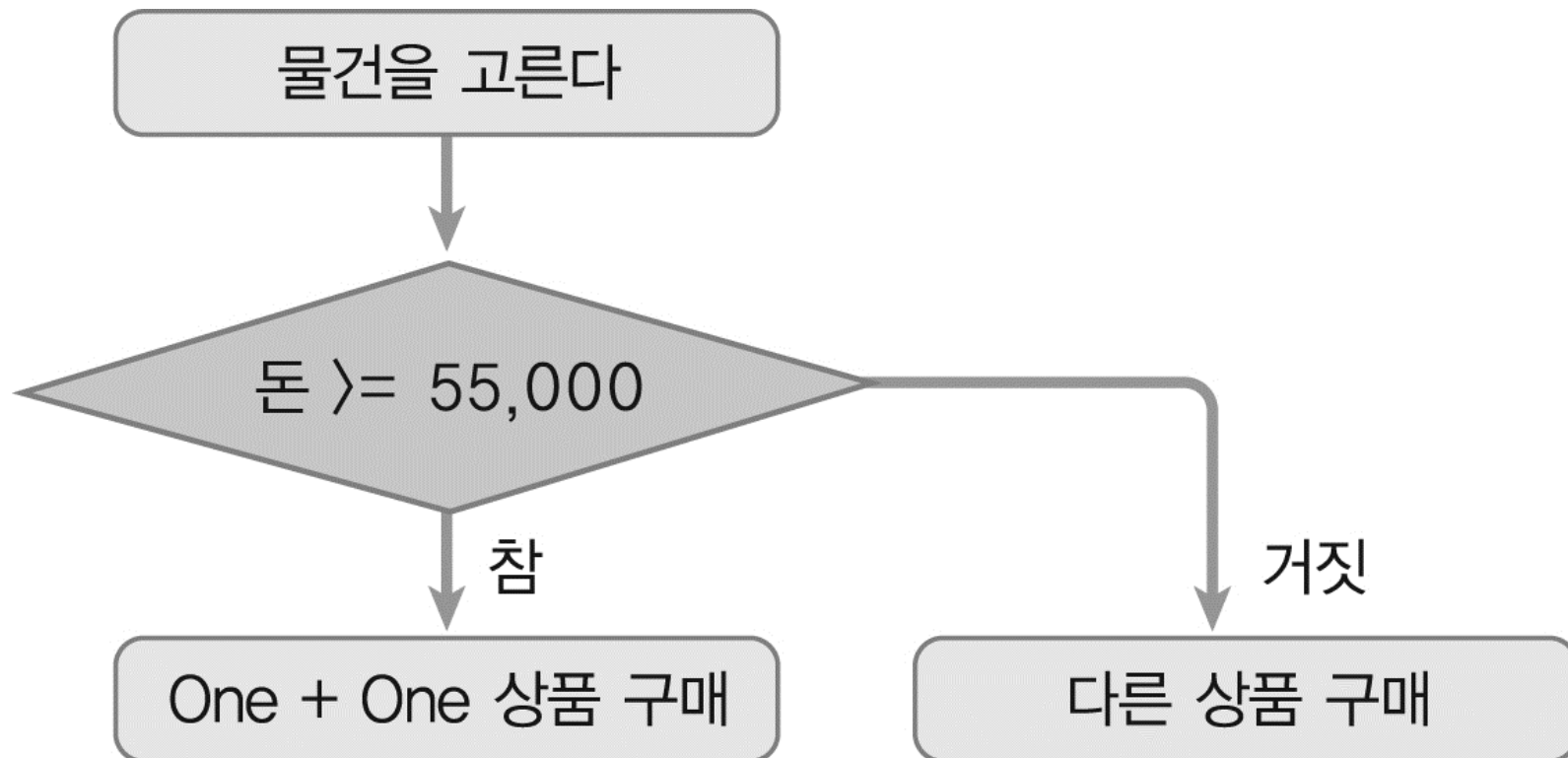
-교재 219페이지 -

8.1 조건문이란

-교재 202페이지 -

8.1 조건문이란 (1/2)

▶ '조건문은 프로그램의 흐름을 바꾼다'.

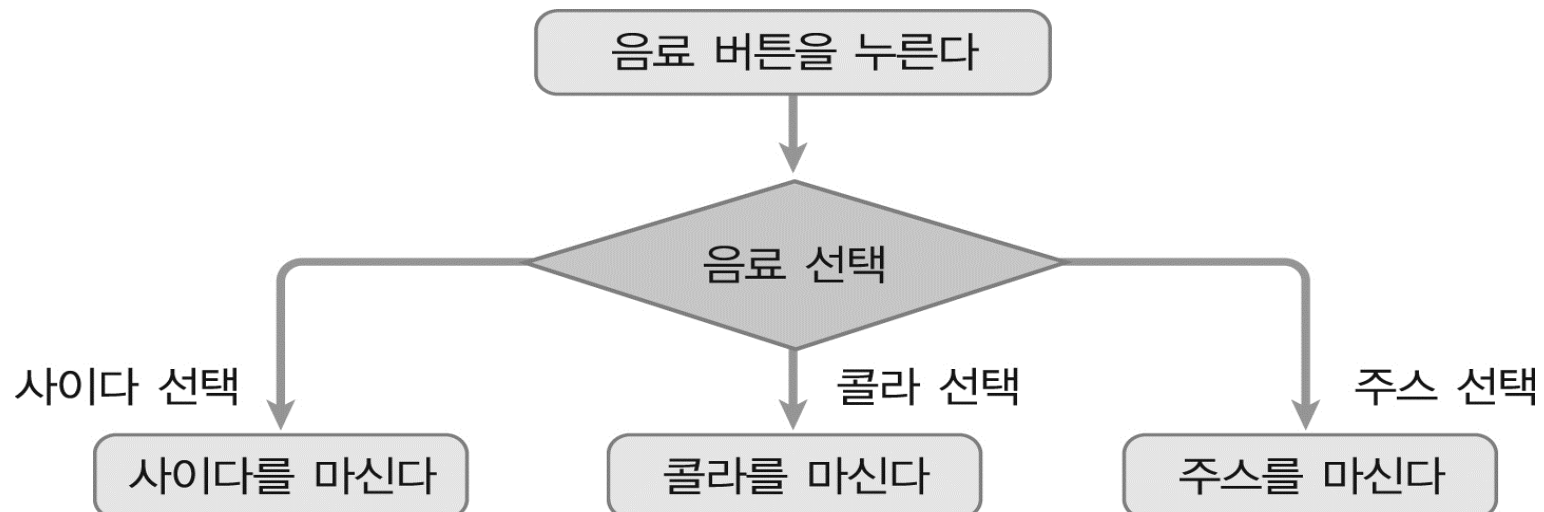


8.1 조건문이란 (2/2)

▶ '조건문은 선택의 기회를 제공한다.'

✓ 자판기에서 사용자가 마실 음료수를 선택할 때 하는 행동

- 조건문1 : 사이다 버튼이 '참'이라면 사이다 선택
- 조건문2 : 콜라 버튼이 '참'이라면 콜라 선택
- 조건문3 : 주스 버튼이 '참'이라면 주스 선택



8.2 조건문을 만드는 방법1 – if 문

-교재 203페이지 -

8.2 조건문을 만드는 방법1 - if문(1/12)

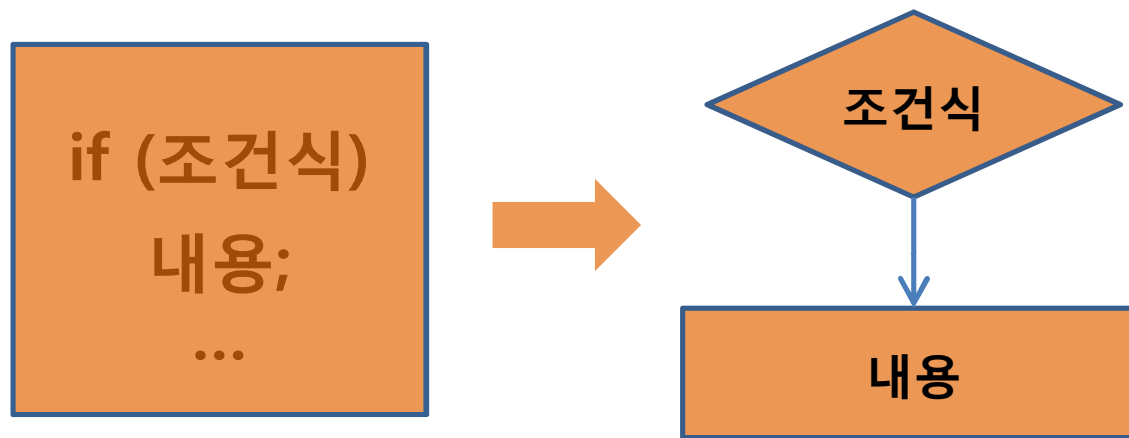
▶ 조건문을 만드는 방법1 - if문

- ① if문
- ② if ~ else문
- ③ 중첩 if ~ else문과 else if문

8.2 조건문을 만드는 방법1 - if문(2/12)

① if문

✓ 수행할 **내용**이 **한 문장**인 경우 **중괄호 생략**

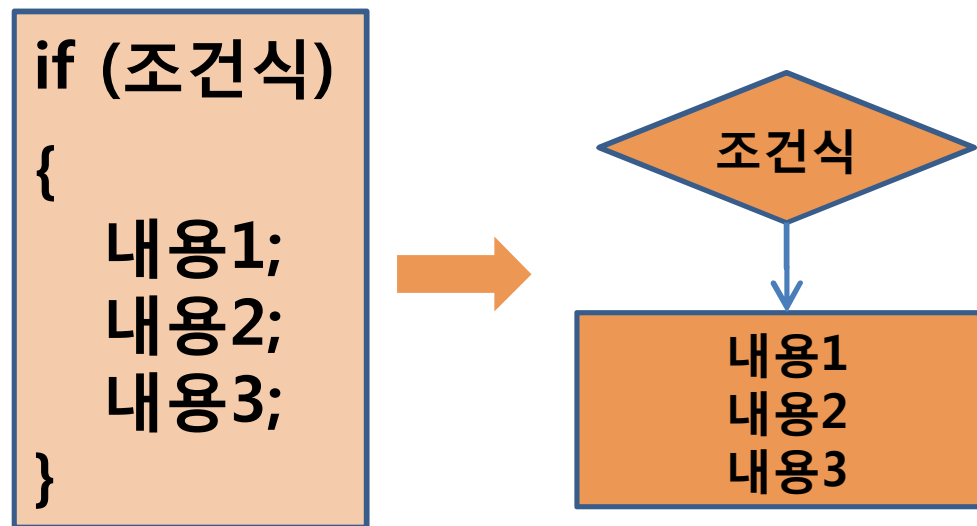


해석: **조건식**이 참이면 **내용**을 수행해라!

8.2 조건문을 만드는 방법1 - if문(3/12)

① if문

✓ 수행할 **내용**이 **여러 문장**인 경우 **중괄호**로 묶음



해석: **조건식**이 참이면 **내용1**, **내용2**, **내용3**을 수행해라!

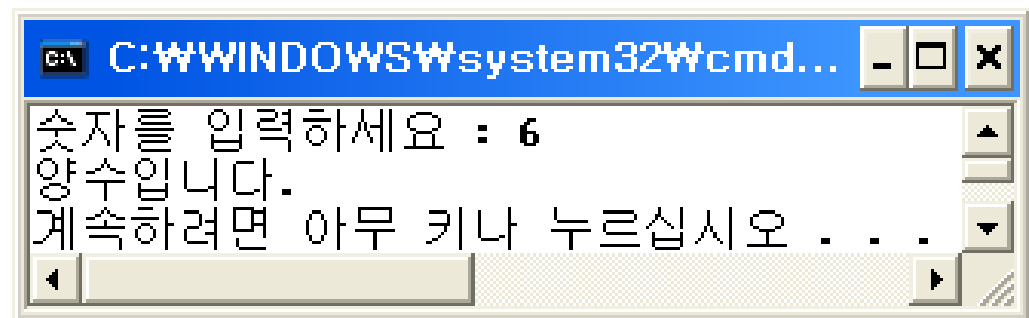
8.2 조건문을 만드는 방법1 - if문(4/12)---[8-1.c 실습]

```
#include <stdio.h>
int main(void)
{
    int num;
    printf("숫자를 입력하세요 : ");
    scanf("%d", &num);

    if(num >= 0)
        printf("양수입니다.\n");

    if(num < 0)
        printf("음수입니다.\n");

    return 0;
}
```



8.2 조건문을 만드는 방법1 - if문(5/12)---[8-2.c 실습]

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

    printf("5의 배수 : ");
    for(num=1; num<=100; num++)
    {
        if(num%5==0)
            printf("%3d \n",num);
    }
    return 0;
}
```

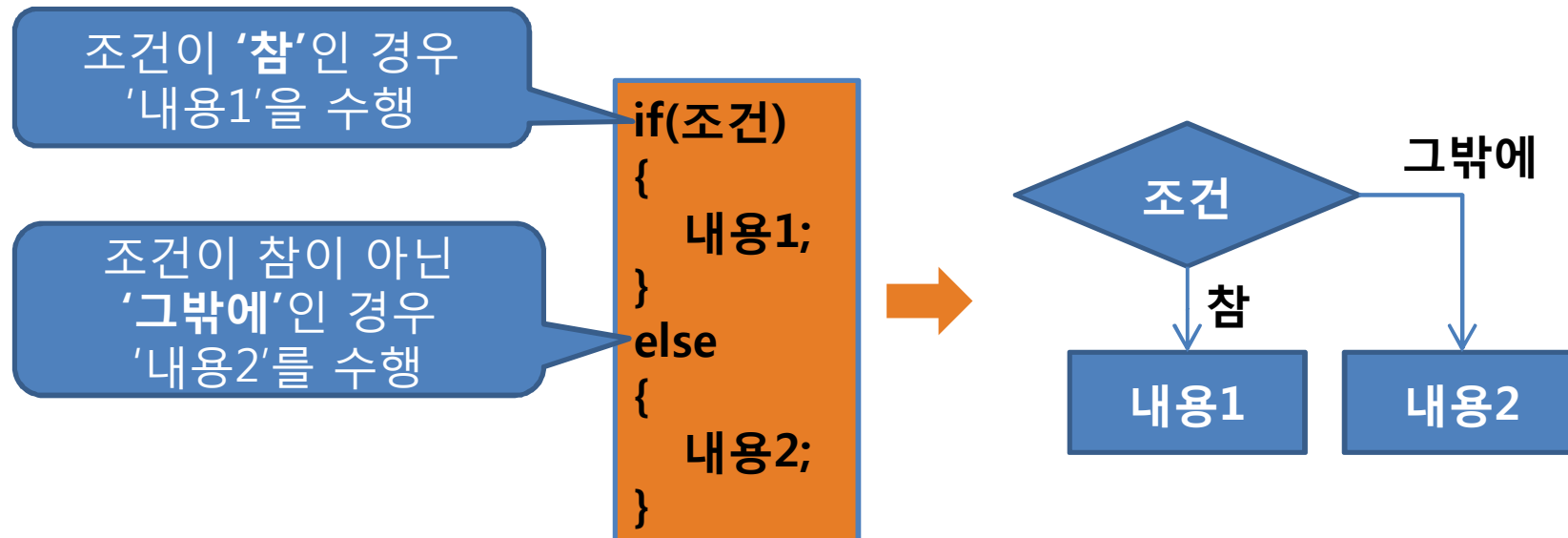
```
C:\WINDOWS\system32\cmd.exe
5의 배수:
  5
 10
 15
 20
 25
 30
 35
 40
 45
 50
 55
 60
 65
 70
 75
 80
 85
 90
 95
100
계속하려면 아무 키나 누르십시오
```

8.2 조건문을 만드는 방법1 - if문(6/12)

② if~else 문

✓ if의 의미: '만약에~'

✓ else의 의미: '그밖에~', 'else만을 따로 사용하지 못한다.'



8.2 조건문을 만드는 방법1 - if문(7/12)---[8-3.c 실습]

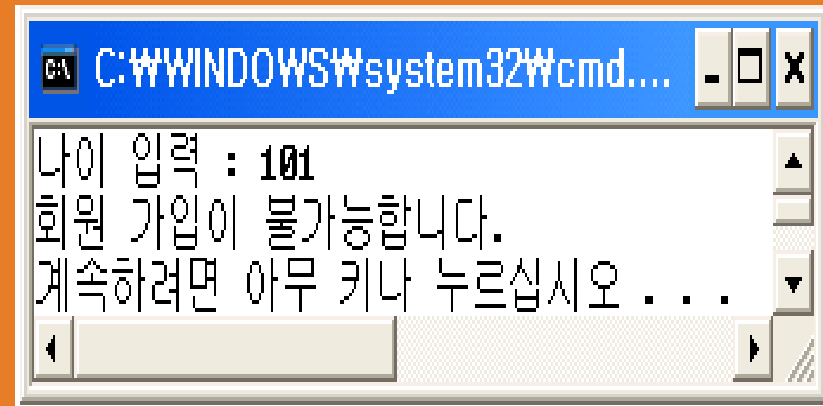
```
#include <stdio.h>
int main(void)
{
    int age;

    printf("나이 입력 : ");
    scanf("%d", &age);

    if(age >= 15 && age <= 100)
        printf("회원 가입이 가능합니다.\n");

    else
        printf("회원 가입이 불가능합니다.\n");

    return 0;
}
```



8.2 조건문을 만드는 방법1 - if문(8/12)---[8-4.c 실습]

③ 중첩 if~else 문과 else if 문

✓ else문의 한계

```
#include <stdio.h>
int main(void)
{
    int num;
    printf("숫자 입력 : ");
    scanf("%d", &num);
```

```
    if(num>0)
```

```
        printf("0보다 큰 수가 num에 저장\n");
```

```
    else
```

```
        printf("0 또는 0보다 작은 수가 num에 저장\n");
    return 0;
```

```
}
```

else문은
num==0, num== -1, num== -2와 같은
다양한 경우들을 제어하지 못함

8.2 조건문을 만드는 방법1 - if문(9/12)---[8-5.c 실습]

```
#include <stdio.h>
int main(void)
{
    int num;
    printf("숫자 입력 : ");
    scanf("%d", &num);

    if(num>0)
        printf("0보다 큰 수가 num에 저장\n");

    else
    {
        if(num==0)
            printf("0이 num에 저장\n");

        else
            printf("0보다 작은 수가 num에 저장\n");
    }
    return 0;
}
```

else문 내부에
if~else문이 존재

else문에서 중괄호를 지우면
else if 문이 된다.

8.2 조건문을 만드는 방법1 - if문(10/12)---[8-6.c 실습]

```
#include <stdio.h>
int main(void)
{
    int num;
    printf("숫자 입력 : ");
    scanf("%d", &num);
```

<else if문의 필요성>
세 가지 이상의 선택 사항이 주어질 경우 사용

```
    if(num>0)
```

```
        printf("0보다 큰 수가 num에 저장\n");
```

```
    else if(num==0)
```

```
        printf("0이 num에 저장\n");
```

```
    else
```

```
        printf("0보다 작은 수가 num에 저장\n");
```

```
    return 0;
```

```
}
```


8.2 조건문을 만드는 방법1 - if문(11/12)---[8-7.c 와 8-8.c 비교]

```
if(num >= 95)
    printf("A+ 입니다. \n");
else
{
    if(num >= 90)
        printf("A 입니다. \n");
    else
    {
        if(num >= 85)
            printf("B+ 입니다. \n");
        else
        {
            if (num >= 80)
                printf("B 입니다. \n");
            else
                printf("F 입니다. \n");
        }
    }
}
```

/* 8-7.c */ --- 가독성이 떨어짐

```
if(num >= 95)
    printf("A+ 입니다. \n");

else if (num >= 90)
    printf("A 입니다. \n");

else if (num >= 85)
    printf("B+ 입니다. \n");

else if (num >= 80)
    printf("B 입니다. \n");

else
    printf("F 입니다. \n");
```

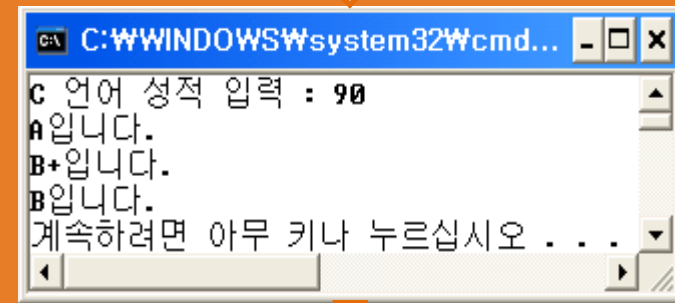
/* 8-8.c */ --- 가독성이 좋아짐

8.2 조건문을 만드는 방법1 - if문(12/12)---[8-9.c 실습]

▶ else if문이 필요한 코드

```
int num;  
printf("C 언어 성적 입력 : ");  
scanf("%d", &num);  
  
if(num >= 95)  
    printf("A+ 입니다. \n");  
  
if(num >= 90)  
    printf("A 입니다. \n");  
  
if(num >= 85)  
    printf("B+ 입니다. \n");  
  
if(num >= 80)  
    printf("B 입니다. \n");  
  
else  
    printf("F 입니다. \n");
```

<'예상한 결과가 출력되지 않는다.'>
이유: if문의 중복 실행



'else if 문으로 수정해야 한다.'

8.3 조건문을 만드는 방법 2 – switch~case 문

-교재 213페이지 -

8.3 조건문을 만드는 방법2 – switch~case문(1/7)

▶ switch~case문

- ✓ '조건문을 표현하는 방식이다.'
- ✓ 'if~else문'을 간결하게 표현 한다.'
- ✓ 'if~else문'에 비해 가독성이 높다.'
- ✓ 'if~else문'에 비해 약간의 제약이 있다.'

▶ '다음을 가정하자!'

- ✓ '우리 집에는 스위치가 3개 있다.'
 - 1번 스위치를 누른 경우: 전등이 켜짐
 - 2번 스위치를 누른 경우: 전등이 꺼짐
 - 3번 스위치를 누른 경우: 고장으로 작동 안함

8.3 조건문을 만드는 방법2 – switch~case문(2/7)---[8-10.c 실습]

```
int num;
printf("몇 번 스위치를 누르시겠습니까? ");
scanf("%d", &num);

switch(num)
{
    case 1:
        printf("전등이 켜짐 %n");
        break;

    case 2:
        printf("전등이 꺼짐 %n");
        break;

    case 3:
        printf("고장 %n");
        break;

    default:
        printf("스위치 오류 : 우리집 스위치는 1번 ~ 3번까지만 있습니다.%n");
}
```

8.3 조건문을 만드는 방법2 – switch~case문(3/7)

▶ switch 문에 사용되는 default와 break

✓ default문

- switch 문에서 정의한 case에 해당되는 조건이 없는 경우 수행

✓ break문

- switch문을 종료하는 역할
- break를 만나면 그 이후 내용은 무시되고 switch문 종료

8.3 조건문을 만드는 방법2 – switch~case문(4/7)---[8-11.c 실습]

```
#include <stdio.h>
int main(void)
{
    char ch;

    printf("(T)hursday, (F)riday, (S)aturday\n");
    printf("문자 입력(T, F, S) : ");
    scanf("%c", &ch);

    switch(ch)
    {
        case 'T':
        case 't':
            printf("Thursday\n");
            break;
        case 'F':
        case 'f':
            printf("Friday\n");
            break;
```

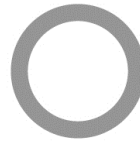
```
        case 'S':
        case 's':
            printf("Saturday\n");
            break;

        default:
            printf("잘못 입력되었습니다.\n");
    }
    return 0;
}
```

8.3 조건문을 만드는 방법2 – switch~case문(5/7)

▶ switch~case 문과 if~else문과의 관계

```
int main(void)
{
    ...
    switch(num)
    {
        case 1:
            내용 1;
            break;
        case 2:
            내용 2;
            break;
        case 3:
            내용 3;
            break;
        default:
            내용 4;
    }
    ...
}
```



상호 코드
전환 가능

```
int main(void)
{
    ...
    if(num==1)
        내용 1;

    else if(num==2)
        내용 2;

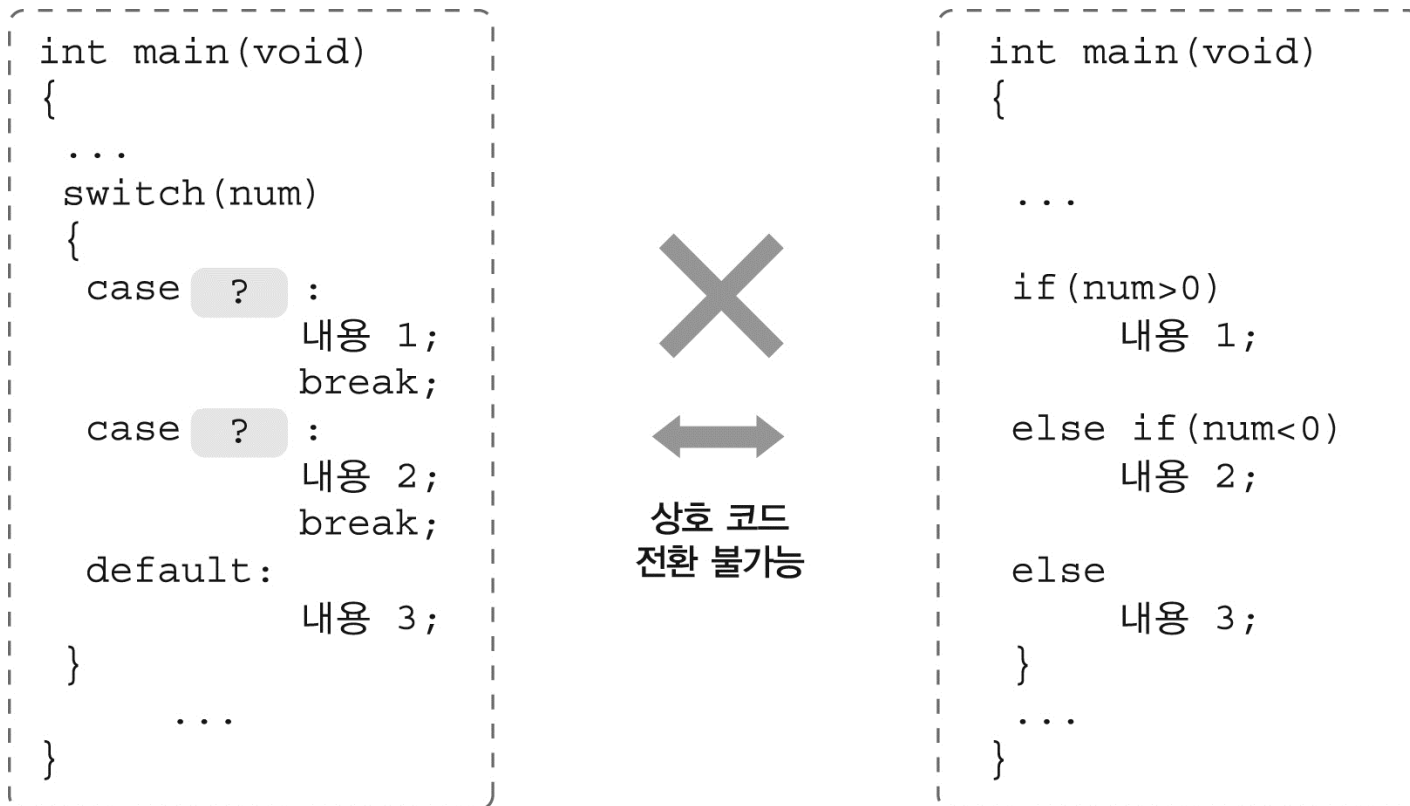
    else if(num==3)
        내용 3;

    else
        내용 4;
}
```


8.3 조건문을 만드는 방법2 – switch~case문(6/7)

▶ switch~case 문의 한계

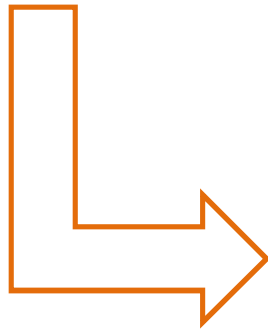
✓ '관계 연산이 올 수 없다.'



8.3 조건문을 만드는 방법2 – switch~case문(7/7)

▶ switch~case 문의 한계

✓ '실수' 자료형을 사용할 수 없다.



```
int main(void)
{
    double num;
    printf("실수 입력: ");
    scanf("%lf", num);

    switch(num)
    {
        case 3.14:
            내용 1;
            break;

        case 4.14:
            내용 2;
            break;
    }
    return 0;
}
```

8.4 break와 continue

-교재 219페이지 -

8.4 break와 continue (1/4)

▶ break문과 continue문은 반복문에서 유용

✓ break문의 기능

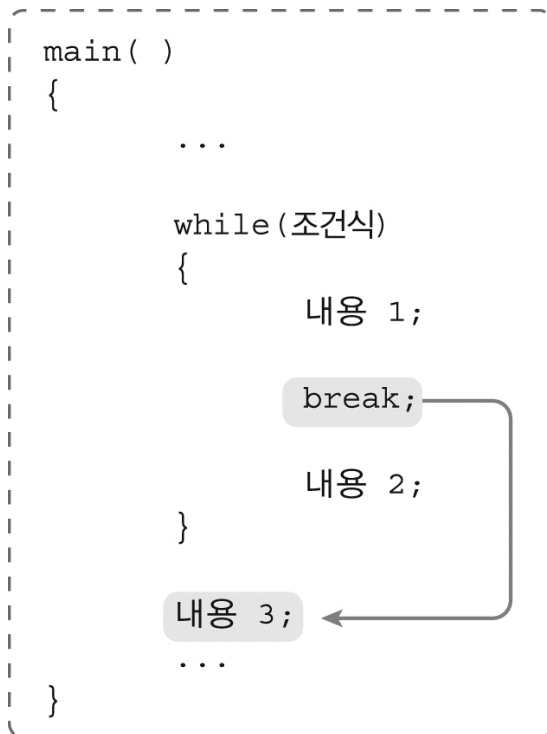
- 조건문을 종료
- 반복문을 종료

✓ continue의 기능

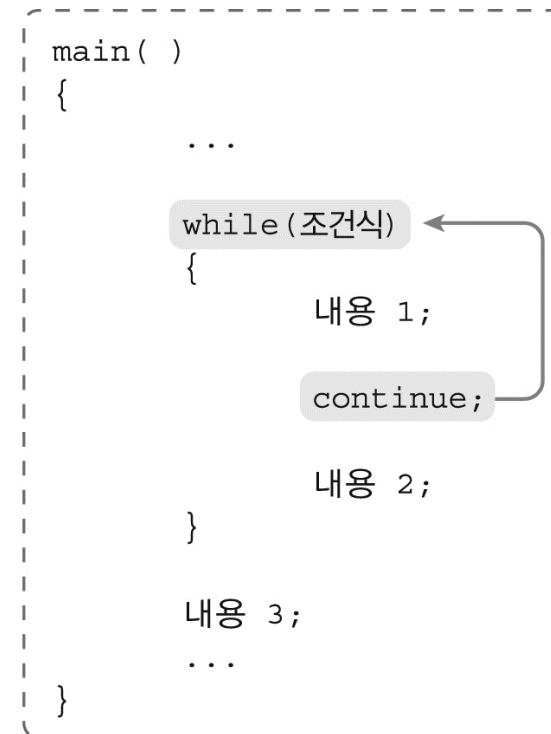
- 반복문을 생략
- while문, for문, do~while문과 같은 반복문과 사용
- switch문에는 사용할 수 없음

8.4 break와 continue (2/4)

▶ break 와 continue 비교



반복문에서 **break**를 만나면 무조건
반복문을 종료하고 다음 문장을 수행



반복문에서 **continue**를 만나면 무조건
조건식을 수행

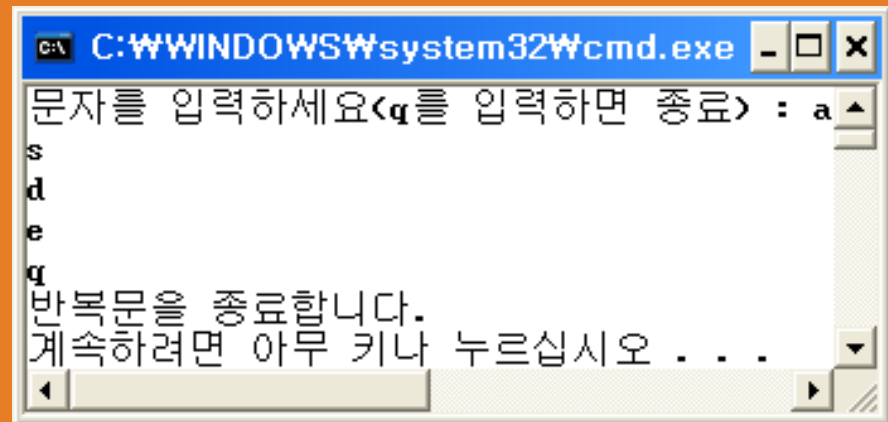
8.4 break와 continue (3/4)---[8-12.c 실습]

```
#include <stdio.h>
int main(void)
{
    char ch;

    printf("문자를 입력하세요(q를 입력하면 종료) : ");
    for ( ; ; )
    {
        scanf("%c", &ch);

        if(ch=='q')
            break;
    }
    printf("반복문을 종료합니다.\n");

    return 0;
}
```



8.4 break와 continue (4/4)---[8-13.c 실습]

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

    for(num=0; num<101; num++)
    {
        if (num%2==1)
            continue;

        printf("%d\\n", num);
    }
    return 0;
}
```

공부한 내용 떠올리기

▶ 조건문의 필요성

▶ 조건문을 만드는 방법 1

✓ if 문, if~else 문, else if 문을 이용한 조건문의 표현 방법

▶ 조건문을 만드는 방법 2

✓ switch~case 문을 이용한 조건문의 표현 방법

▶ break 문과 continue 문

단어 QUIZ의 출생의 비밀 (출처: 사랑과 지혜의 탈무드)

