

기초프로그래밍

제8장 조건문

Sangsoo Lim

CSAI

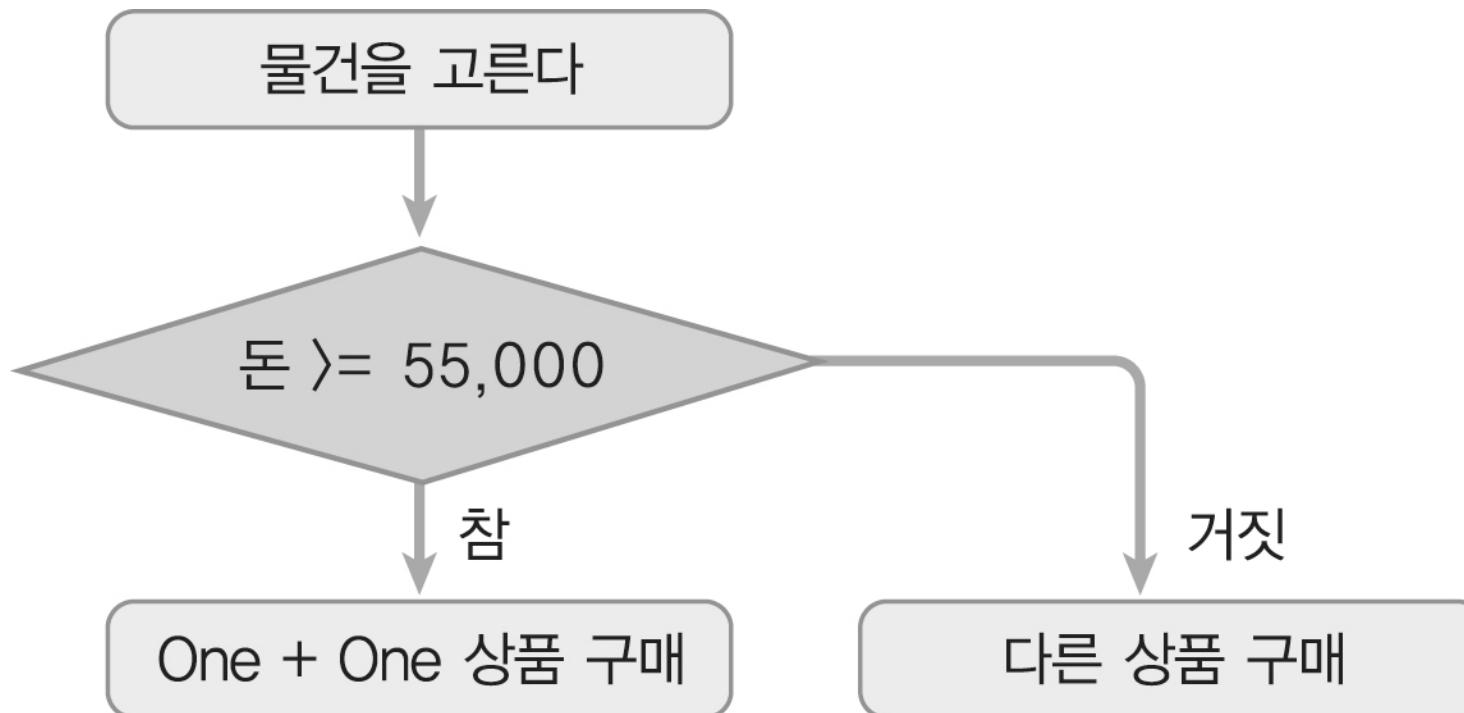
Dongguk University

차례

- 조건문이란
- 조건문을 만드는 방법 1 - if 문
- 조건문을 만드는 방법 2 - switch~case 문
- break와 continue

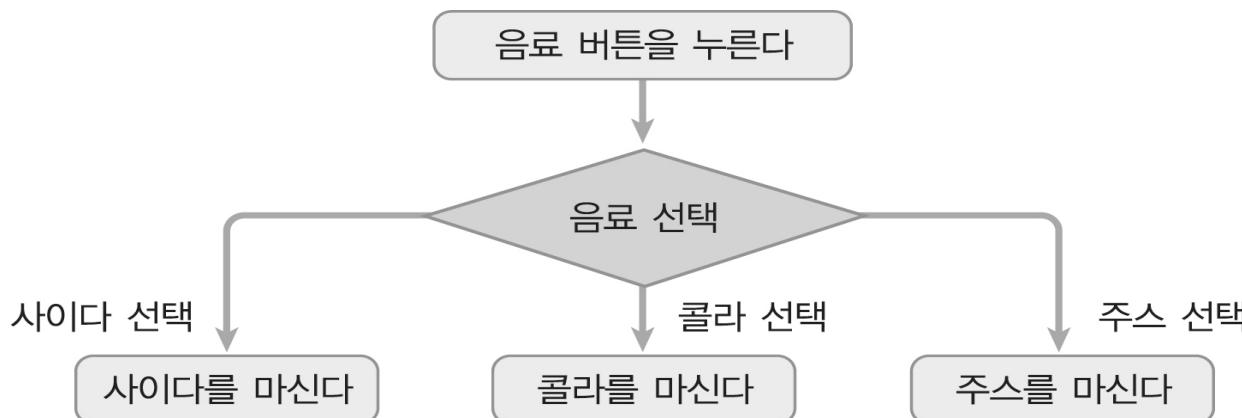
조건문이란

- 조건문은 프로그램의 흐름을 바꾼다.



조건문이란

- 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



조건문을 만드는 방법1 - if문

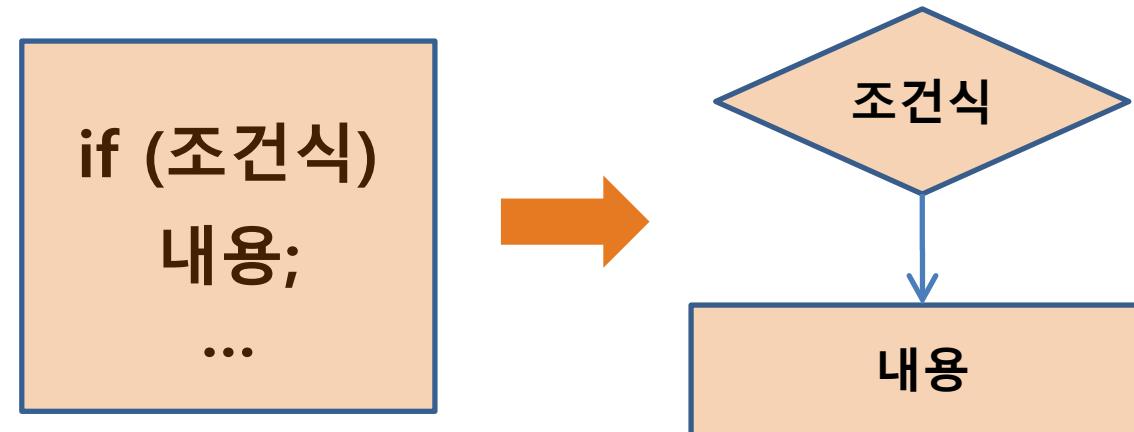
- 조건문을 만드는 방법1 - if문

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

조건문을 만드는 방법1 – if문

① if문

– 수행할 내용이 한 문장인 경우 중괄호 생략

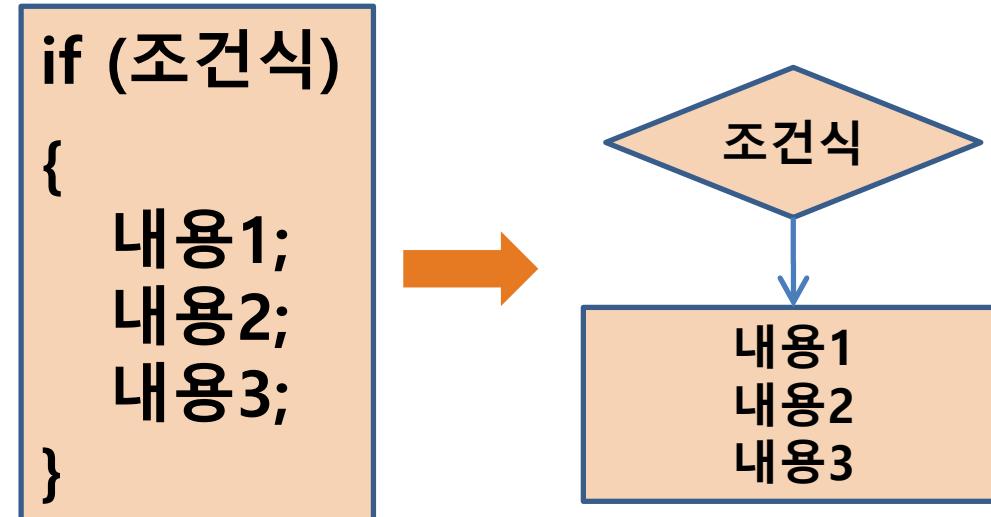


해석: 조건식이 참이면 내용을 수행해라!

조건문을 만드는 방법1 – if문

① if문

- 수행할 내용이 여러 문장인 경우 중괄호로 묶음



해석: 조건식이 참이면 내용1, 내용2, 내용3을 수행해라!

조건문을 만드는 방법1 - if문

```
/* 8-1.c */
#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;
}
```

숫자를 입력하세요 : 5
양수입니다.

조건문을 만드는 방법1 - if문

```
/* 8-2.c */
#include <stdio.h>
int main(void)
{
    int num;

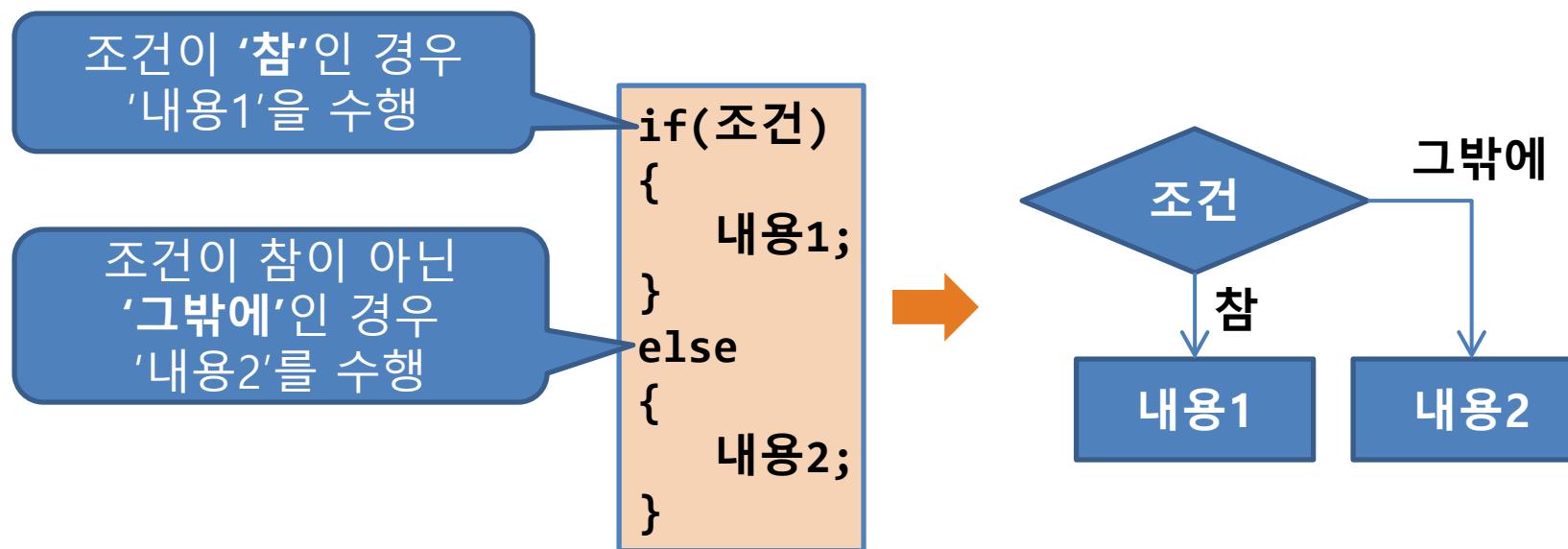
    printf("5의 배수 : ");
    for(num=1; num<=100; num++)
    {
        if(num%5==0)
            printf("%3d \n",num);
    }
    return 0;
}
```

5의 배수 : 5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100

조건문을 만드는 방법1 – if문

② if~else 문

- if의 의미: 만약에~
- else의 의미: 그밖에~, else만을 따로 사용하지 못한다.



조건문을 만드는 방법1 - if문

```
/* 8-3.c */
#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;
}
```

나이 입력 : 101
회원 가입이 불가능합니다.

조건문을 만드는 방법1 – if문

③ 중첩 if~else 문과 else if 문

– else문의 한계

```
/* 8-4.c */
#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

조건문을 만드는 방법1 – if문

```
/* 8-5.c */
#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.

조건문을 만드는 방법1 – if문

```
/* 8-6.c */
#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;
}
```

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

조건문을 만드는 방법 1 - if문

```
if(num>=95)
    printf("A+입니다.\n");
else
{
    if(num>=90)
        printf("A입니다.\n");
    else
    {
        if(num>=85)
            printf("B+입니다.\n");
        else
        {
            if (num>=80)
                printf("B입니다. \n");
            else
                printf("F입니다. \n");
        }
    }
}
/* 8-7.c */ --- 가독성이 떨어짐
```

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

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

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

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

else
    printf("F입니다. \n");
```

/* 8-8.c */ --- 가독성이 좋아짐

조건문을 만드는 방법1 – if문

– 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("A입니다. \n");
if(num>=85)
    printf("B+입니다. \n");
if(num>=80)
    printf("B입니다. \n");
else
    printf("F입니다. \n");
```

조건문을 만드는 방법2 – switch~case문

- **switch~case문**
 - 조건문을 표현하는 방식이다.
 - if~else문을 간결하게 표현 한다.
 - if~else문에 비해 가독성이 높다.
 - if~else문에 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

조건문을 만드는 방법2 – switch~case문

```
/* 8-10.c */
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");
}
```

조건문을 만드는 방법2 – switch~case문

- **switch** 문에 사용되는 **default**와 **break**
 - **default**문
 - **switch** 문에서 정의한 **case**에 해당되는 조건이 없는 경우 수행
 - **break**문
 - **switch**문을 종료하는 역할
 - **break**를 만나면 그 이후 내용은 무시되고 **switch**문 종료

조건문을 만드는 방법2 – switch~case문

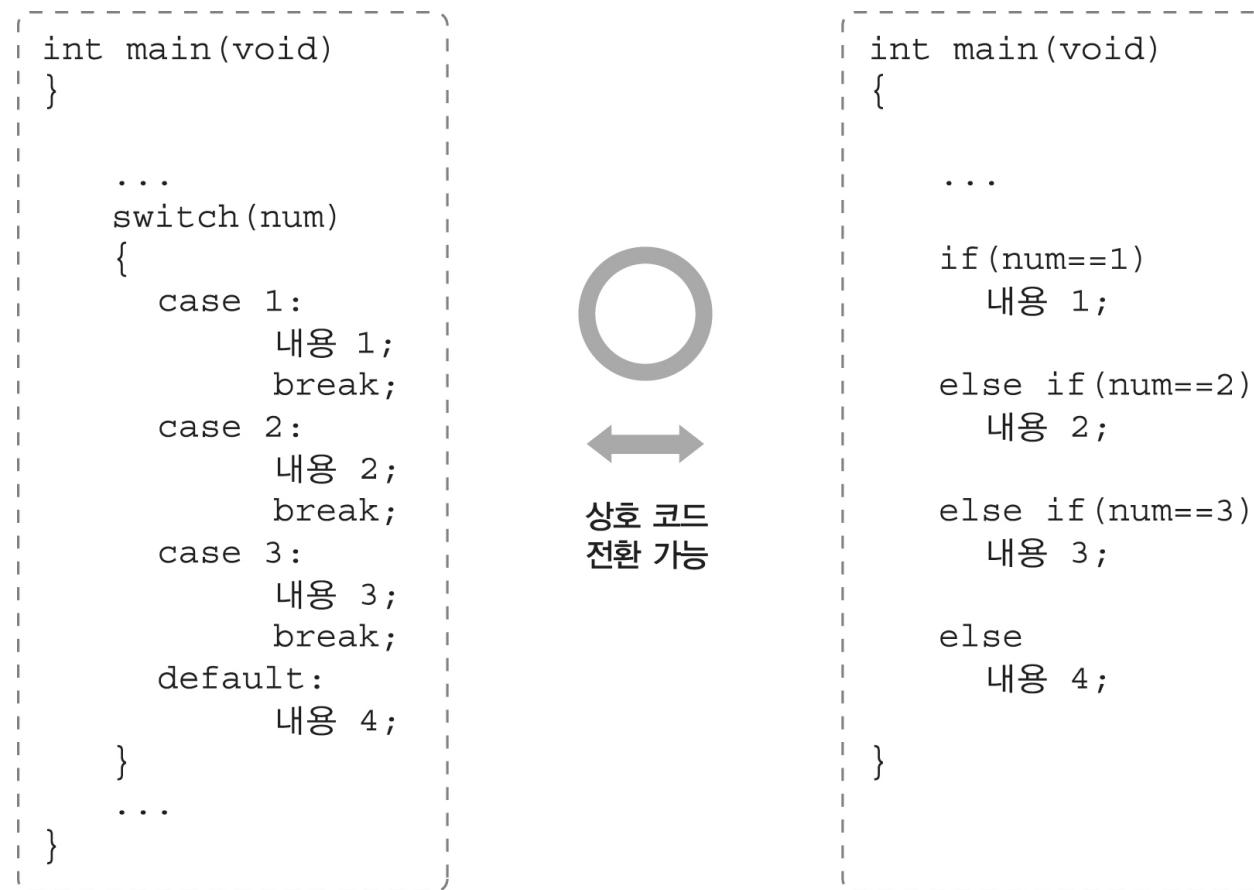
```
/* 8-11.c */
#include <stdio.h>
int main(void)
{
    char ch;

    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;
}
```

조건문을 만드는 방법2 – switch~case문

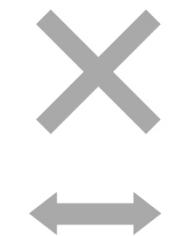
- switch~case 문과 if~else문과의 관계



조건문을 만드는 방법2 – switch~case문

- **switch~case 문의 한계**
 - 관계 연산이 올 수 없다.

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

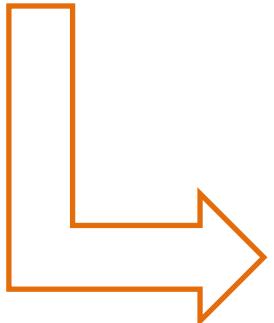


상호 코드
전환 불가능

```
int main(void)
{
    ...
    if (num>0)
        내용 1;
    else if (num<0)
        내용 2;
    else
        내용 3;
    ...
}
```

조건문을 만드는 방법2 – switch~case문

- **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;

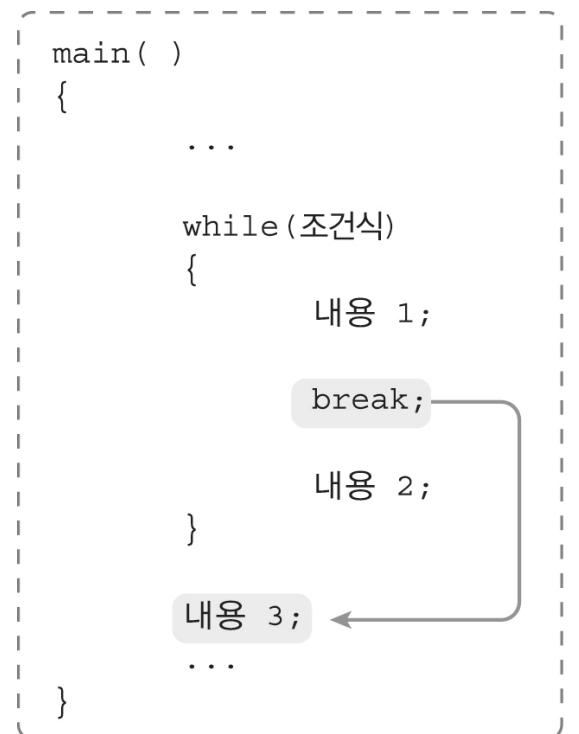
        case 4.14:
            내용 2;
            break;
    }
    return 0;
}
```

break와 continue

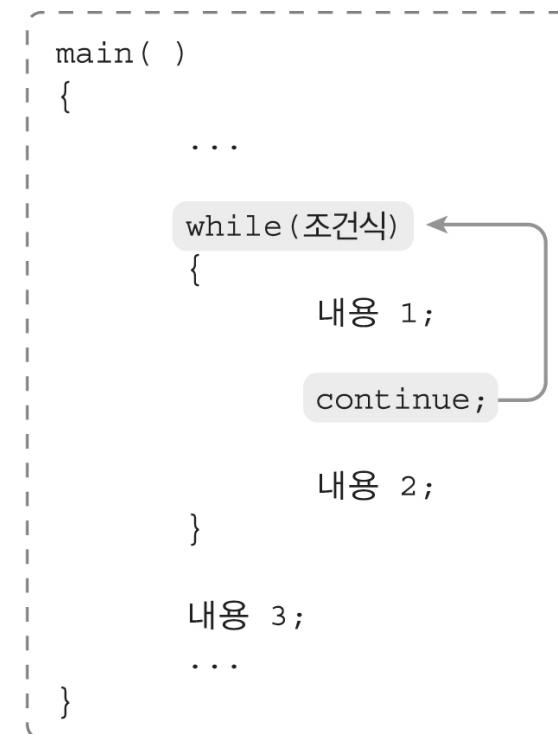
- break문과 continue문은 반복문에서 유용
 - break문의 기능
 - 조건문을 종료
 - 반복문을 종료
 - continue의 기능
 - 반복문을 생략
 - while문, for문, do~while문과 같은 반복문과 사용
 - switch문에는 사용할 수 없음

break와 continue

- break 와 continue 비교



반복문에서 break를 만나면 무조건
반복문을 종료하고 다음 문장을 수행



반복문에서 continue를 만나면 무조건
조건식을 수행

break와 continue

```
/* 8-12.c */
#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;
}
```

```
문자를 입력하세요(q를 입력하면 종료) : a
s
d
e
q
반복문을 종료합니다.
```

break와 continue

```
/* 8-13.c */
#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;
}
```

Summary

- 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

1. 사용자 권한 검사 (if ~ else if ~ else)

- 사용자가 입력한 사용자 등급(예: 0: 게스트, 1: 일반 사용자, 2: 관리자)에 따라 접근 권한을 부여.
- 관리자일 경우 시스템 설정 메뉴까지 허용, 일반 사용자면 일부 제한, 게스트는 읽기 전용.

```
#include <stdio.h>

int main(void)
{
    int userLevel;

    printf("사용자 등급을 입력하세요 (0:게스트, 1:일반, 2:관리자): ");
    scanf("%d", &userLevel);

    if (userLevel == 2) {
        printf("관리자 권한입니다. 모든 메뉴에 접근 가능합니다.\n");
    } else if (userLevel == 1) {
        printf("일반 사용자 권한입니다. 제한된 메뉴에만 접근 가능합니다.\n");
    } else if (userLevel == 0) {
        printf("게스트 권한입니다. 읽기 전용으로 접근 가능합니다.\n");
    } else {
        printf("잘못된 등급입니다.\n");
    }

    return 0;
}
```

2. 할인 계산 프로그램 (if ~ else, 조건 연산자)

- 쇼핑몰에서 구매 금액에 따라 할인율을 적용한다.
- 예) 10만 원 이상이면 10% 할인, 그렇지 않으면 3% 할인.
- 최종 결제 금액을 출력하고, 0 이하 입력하면 오류 처리.

```
#include <stdio.h>

int main(void)
{
    double purchase; // 구매 금액
    double finalPrice;

    printf("구매 금액을 입력하세요: ");
    scanf("%lf", &purchase);

    if (purchase <= 0) {
        printf("잘못된 금액입니다.\n");
        return 0;
    }

    // 10만 원 이상 10% 할인, 그렇지 않으면 3% 할인
    if (purchase >= 100000) {
        finalPrice = purchase * 0.9; // 10% 할인
    } else {
        finalPrice = purchase * 0.97; // 3% 할인
    }

    printf("최종 결제 금액: %.2f원\n", finalPrice);
    return 0;
}
```

3. 다중 분기: 시험 성적 총평 (else if + 복합 조건)

- 국어, 영어, 수학 점수를 입력받은 뒤, 세 과목 평균으로 성적을 나눈다:
 - 90 이상: "우수" 70 이상: "보통" 그 외: "부족"
- 단, 한 과목이라도 60 미만이면 "재시험" 문구를 추가 출력.

```
#include <stdio.h>

int main(void)
{
    int kor, eng, math;
    printf("국어, 영어, 수학 점수를 입력하세요: ");
    scanf("%d %d %d", &kor, &eng, &math);

    double avg = (kor + eng + math) / 3.0;

    // 기본 등급 판정
    if (avg >= 90) {
        printf("성적: 우수\n");
    } else if (avg >= 70) {
        printf("성적: 보통\n");
    } else {
        printf("성적: 부족\n");
    }

    // 재시험 판정
    // 세 과목 중 하나라도 60 미만이면 "재시험" 출력
    if (kor < 60 || eng < 60 || math < 60) {
        printf("재시험 대상입니다.\n");
    }

    return 0;
}
```

4. 식당 메뉴 선택 (switch ~ case, 복합 case)

- 숫자(1~5)를 입력받아 메뉴를 고른다.
- 1 또는 2면 "한식" 메뉴, 3이면 "중식", 4면 "일식", 5면 "분식", 그 외는 "오류".

```
#include <stdio.h>

int main(void)
{
    int choice;
    printf("메뉴 선택 (1:김치찌개, 2:된장찌개, 3:짜장면, 4:초밥, 5:떡볶이): ");
    scanf("%d", &choice);

    switch (choice) {
        case 1:
        case 2:
            printf("한식입니다.\n");
            break;
        case 3:
            printf("중식입니다.\n");
            break;
        case 4:
            printf("일식입니다.\n");
            break;
        case 5:
            printf("분식입니다.\n");
            break;
        default:
            printf("잘못된 선택입니다.\n");
    }

    return 0;
}
```

5. 비밀번호 3회 입력 제한 (break, continue)

- 올바른 비밀번호를 맞출 때까지 최대 3번 시도.
- 맞추면 "로그인 성공" 후 반복 종료, 틀리면 시도 횟수 누적.
- 3회 모두 틀리면 "로그인 실패" 출력.

```
#include <stdio.h>
#include <string.h> // strcmp 함수 등 사용

int main(void)
{
    const char correctPwd[] = "pass123";
    char input[50];
    int i;

    for (i = 1; i <= 3; i++) {
        printf("비밀번호를 입력하세요(%d/3): ", i);
        scanf("%s", input);

        if (strcmp(input, correctPwd) == 0) {
            printf("로그인 성공!\n");
            break; // for문 탈출
        } else {
            printf("틀렸습니다.\n");
        }
    }

    if (i > 3) {
        // 3번 기회 모두 소진
        printf("로그인 실패. 기회가 모두 소진되었습니다.\n");
    }
}

return 0;
}
```

6. 검색어 필터링 (continue)

- 여러 개의 문자열(검색어 후보)을 받다가, 만약 exit라는 단어가 나오면 입력을 중지(break).
- 받는 검색어 중에서 길이가 2 이하인 경우는 무시(continue)하고, 그 외만 출력.

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

int main(void)
{
    char word[100];

    printf("검색어를 입력하세요(종료하려면 'exit'): \n");
    while (1) {
        scanf("%s", word);

        if (strcmp(word, "exit") == 0) {
            printf("프로그램을 종료합니다.\n");
            break; // 반복 종료
        }

        if (strlen(word) <= 2) {
            continue; // 너무 짧은 검색어는 무시
        }

        printf("검색어로 '%s'를 선택했습니다.\n", word);
    }

    return 0;
}
```

1. BMI 지수 계산 (if ~ else if ~ else)

- 문제

- 사용자에게 몸무게(kg)와 키(m)를 입력받는다.
- BMI(Body Mass Index)를 계산:

$$BMI = \frac{\text{몸무게}(kg)}{(\text{키}(m))^2}$$

- BMI 수치에 따라 다음과 같이 분류하여 결과를 출력한다.
 - 18.5 미만 → 저체중
 - 18.5 이상 25 미만 → 정상
 - 25 이상 30 미만 → 과체중
 - 30 이상 → 비만

- 요구사항

- if ~ else if ~ else를 사용해 분기 처리.
- 키나 몸무게가 0 이하로 입력되면 "잘못된 입력입니다."라고 출력 후 종료.

몸무게(kg): 70
키(m): 1.75
BMI = 22.857143
결과: 정상

2. 월(Month) 입력받아 계절 출력 (switch ~ case)

- 문제
 1. 정수(1~12)를 입력받아, 해당 월이 어느 계절인지 출력한다.
 2. 분류 예시(유럽/북반구 기준):
 - 3~5 → 봄
 - 6~8 → 여름
 - 9~11 → 가을
 - 12,1,2 → 겨울
- 요구사항
 - switch ~ case 문을 사용한다.
 - 분기 후 break 문을 적절히 배치.
 - 1~12 범위를 벗어난 입력은 “잘못된 입력” 처리.

월 입력(1~12): 3
봄입니다.

3. 1~N 중 3의 배수만 출력 (break & continue)

- 문제
 1. 사용자에게 양의 정수 N을 입력받는다.
 2. 1부터 N까지 순회하면서, 3의 배수인 숫자만 출력한다.
 3. 만약 N이 0 이하이면 “잘못된 입력”을 출력 후 곧바로 반복문을 종료(또는 프로그램 전체 종료).
- 요구사항
 1. 반복문 중간에 continue를 사용하여 3의 배수가 아닌 숫자는 건너뛰도록 한다.
 2. $N \leq 0$ 일 경우 break나 return 등으로 반복문(또는 프로그램) 즉시 탈출.

```
N을 입력하세요: 15  
3 6 9 12 15  
완료!
```