한빛아카데미 Hanbit Academy, Inc.

C 언어 프로그래밍

Part 02. C 언어 기본 학습

Chapter 05. 선택문

목차

1. 제어문이란?
2. if문
3. if~else문
4. swith문

[실전예제] 8의 배수? 4의 배수? 2의 배수?

학습목표

- **선택문의 if문과 switch문을 이해한다.**
- **if~else문과 if~else if문의 필요성과 사용법을 이해한다.**
- **break를 이해하고 switch문과 함께 사용해본다.**

01

제어문이란?

01. 제어문이란?

I. 실행 흐름의 분류

- **프로그램의 실행 흐름**
  - 코드의 실행이 순차적으로 진행되면서 이전 코드가 반복되기도 함
  - 조건에 따라서 실행되는 코드가 분기될 수도 있음
  - 어떤 사건을 기다리면서 멈출 수도 있음

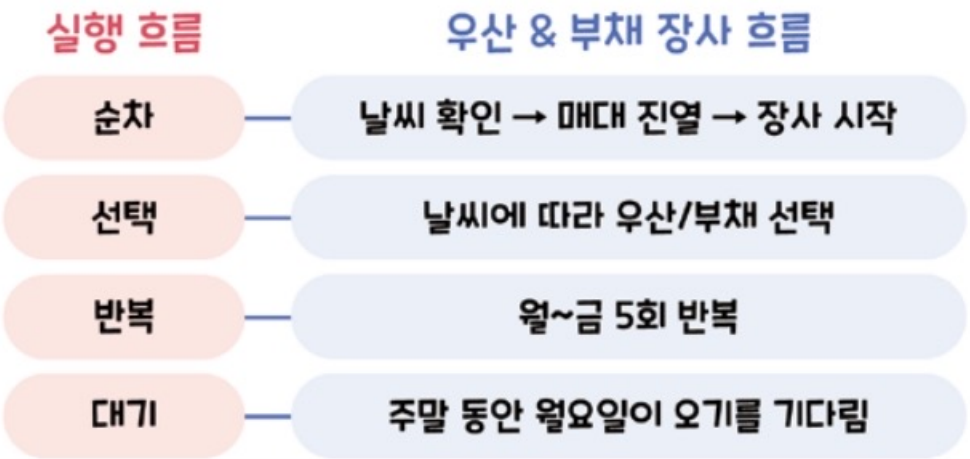


그림 5-1 실행 흐름과 우산 & 부채 장사의 흐름 비교

01. 제어문이란?

II. 선택문

- **선택문의 개념**
  - 실행 흐름이 조건식의 값에 따라서 갈라지는 것
  - if문 : 조건식의 값을 평가하여 참, 거짓으로 나눈 후 양 갈래로 분기함
  - switch문 : 조건식의 값에 따라 여러 갈래의 분기를 만듦

01. 제어문이란?

확인문제1

1. 다음 빈칸에 들어갈 단어를 채우시오.

선택문, 반복문과 같이 프로그램의 실행 흐름을 제어하는 명령문을 _____ 이라고 한다.

2. if문과 switch문의 흐름 분기의 차이를 서술하시오.

3. if문, switch문을 사용하지 않고, 실행 흐름을 분기하는 방법을 서술하시오.

02

if문

02. if문

I. if문의 조건식

- **if문**
  - 조건식의 값을 평가하여 참, 거짓으로 나누고 그에 따라서 흐름을 양 갈래로 분기함
- **if문의 사용 형식**

```
---------------------  ●—— if문 시작
if(조건식)
    명령문 ●———————— 조건식이 참이면 실행되는 문장
---------------------  ●—— if문 끝
```

02. if문

I. if문의 조건식

[코드 5-1] 조건식의 평가

```
01  #include <stdio.h>
02
03  int main()
04  {
05      if(0)
06          printf("0");
07
08      if(1)
09          printf("1");
10  }
```

조건식 자체가 거짓(false)로 평가됨

조건식 자체가 참(true)으로 평가됨

```
1
```

02. if문

I. if문의 조건식

[코드 5-2] 비교 연산자와 조건식

```
01  #include <stdio.h>
02
03  int main()
04  {
05     int a = 7;
06     int b = 3;
07
08     if(a > b)
09        printf("a > b");
10
11     if(a < b)
12        printf("a < b");
13  }
```

```
a > b
```

02. if문

II. if문의 명령문

- **if문의 특징**
  - If(조건식) 바로 다음에는 조건식이 참인 경우 수행될 명령문이 단 하나만 존재할 수 있음
  - 여러 개의 명령문을 실행해야 한다면 복합문을 만들어야 함

02. if문

II. if문의 명령문

[코드 5-3] if 단일 명령문

```
01  #include <stdio.h>
02
03  int main()
04  {
05      if(0)
06          printf("1\r\n");
07          printf("2\r\n");
08
09      printf("3");
10  }
```

```
2
3
```

02. if문

II. if문의 명령문

[코드 5-4] if 복합문

```
01  #include <stdio.h>
02
03  int main()
04  {
05      if(0)
06      {
07          printf("1\r\n");
08          printf("2\r\n");
09      }
10
11      printf("3");
12  }
```

```
3
```

02. if문

II. if문의 명령문

하나 더 알기 **If문의 명령문이 0개인 경우**

- 명령문이 하나도 없는 빈 블록도 복합문으로서 하나의 명령문이 됨
- If문을 사용할 경우, 조건식이 참일 때 명령문이 하나뿐이더라도 매크로 함수를 사용하여 복합문 실수를 줄일 수 있기 때문에 블록으로 감싸서 복합문을 만들기도 함

02. if문

확인문제2

1. 다음 빈칸에 들어갈 단어를 채우시오.

조건식은 0으로 계산될 경우 거짓으로 평가되며, [ ]으로 계산될 경우 참으로 평가된다.

2. 다음 프로그램의 실행 결과는 무엇인가?

```
01   #include <stdio.h>
02
03   int main()
04   {
05       if(0)
06           printf("A"); printf("B");
07       printf("C");
08   }
```

02. if문

LAB 5-1 **홀/짝 판별하기**

정수를 입력받아서 홀수, 짝수를 판별하여 출력하는 프로그램을 작성해봅시다.

```
정수를 입력하세요.
20
짝수입니다.
```

n이 짝수일 때 n % 2는 0이므로 거짓이고, !(n % 2)는 1이므로 참입니다.

02. if문

LAB 5-1 정답

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03
04 int main()
05 {
06    int n;
07    printf("정수를 입력하세요.\r\n");
08    scanf("%d", &n);
09
10    if(n % 2)
11       printf("홀수입니다.");
12
13    if(!(n % 2))
14       printf("짝수입니다.");
15 }
```

03

if~else문

03. if~else문

I. if~else문의 형식

- **if문과의 차이**
  - if문의 조건식이 거짓인 경우에도 else를 사용하여 실행될 명령문이 있음
  - if와 else 사이에는 오직 명령문(복합문) 하나만 있어야 함
- **if~else문의 사용 형식**

```
----------------------  if문 시작
if(조건식)
    명령문T  ----------  조건식이 '참'이면 실행되는 문장
else
    명령문F  ----------  조건식이 '거짓'이면 실행되는 문장
----------------------  if문 끝
```

03. if~else문

I. if~else문의 형식

[코드 5-5] else 오류

```
01  #include <stdio.h>
02
03  int main()
04  {
05      if(0)
06          printf("1\r\n");
07          printf("2");
08      else
09          printf("3");
10  }
```

오류 발생

03. if~else문

II. if~else if문

- **if~else if문의 개념**
  - 여러 조건식에 대해서 각각의 분기가 필요한 경우 사용됨
- **if~else문의 사용 형식**

```
---------------------  if문 시작
if(조건식1)
    명령문1
else if(조건식2)
    명령문2
...
else if(조건식N)
    명령문N
[else                  미사용 시 생략 가능함([]는 생략 가능 기호)
    명령문E]
---------------------  if문 끝
```

03. if~else문

II. if~else if문

[코드 5-6] if~else if

```
01  #define _CRT_SECURE_NO_WARNINGS
02  #include <stdio.h>
03
04  int main()
05  {
06      int age;
07      printf("나이를 입력하세요!\r\n");
08      scanf("%d", &age);
09
10      if(age >= 20)
11          printf("성인입니다.");
12      else if(age >= 12)
13          printf("청소년입니다.");
14      else
15          printf("아동입니다.");
16  }
```

```
나이를 입력하세요!
18
청소년입니다.
```

03. if~else문

II. if~else if문

[코드 5-7] if~else if문에서 오직 한 갈래 분기

```
01  #include <stdio.h>
02
03  int main()
04  {
05      int age = 45;
06
07      if(age >= 20)
08          printf("20대 이상입니다.");
09      else if(age >= 30)
10          printf("30대 이상입니다.");
11      else if(age >= 40)
12          printf("40대 이상입니다.");
13      else
14          printf("50대 이상입니다.");
15  }
```

```
20대 이상입니다.
```

03. if~else문

II. if~else if문

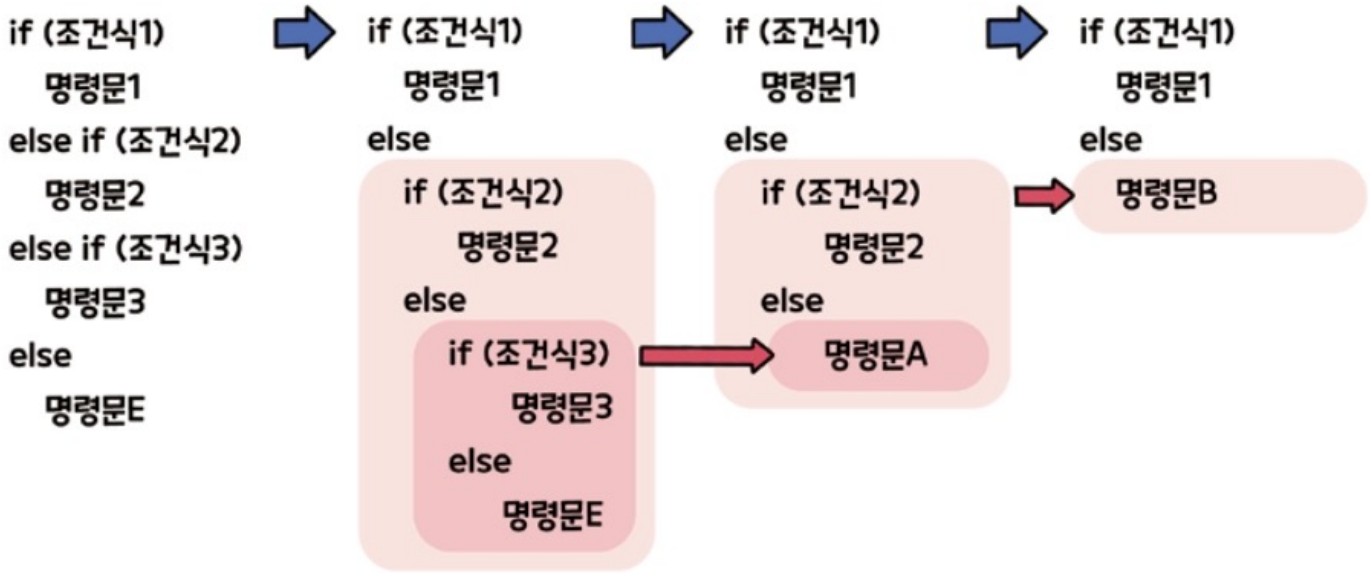


그림 5-2 if~else if문의 축소화

03. if~else문

확인문제3

1. 다음의 코드는 'if와 짝을 이루지 않는 잘못된 else문입니다.'라는 오류가 나온다. 몇 행이 잘못되었고, 그 이유는 무엇인지 서술하시오.

```
01  #include <stdio.h>
02
03  int main()
04  {
05      if(0)
06          printf("1");;
07      else
08          print("2");
09  }
```

2. 다음 코드의 실행 결과를 기술하고, 그 이유를 설명하시오.

```
01  #include <stdio.h>
02
03  int main()
04  {
05      if(0)
06          if(0)
07              printf("A");
08      else
09          printf("B");
10  }
```

03. if~else문

LAB 5-2 **2의 배수, 3의 배수, 2와 3의 공배수 판별하기**

정수를 입력받아 2의 배수인지, 3의 배수인지, 2와 3의 공배수인지를 출력하는 프로그램을 작성해봅시다.

```
정수를 입력하세요.
6
2와 3의 공배수입니다.
```

```
정수를 입력하세요.
22
2의 배수입니다.
```

!(n % 2) && !(n % 3)은 2의 배수이고 3의 배수일 때 참이 된다. 따라서 2와 3의 공배수를 먼저 찾은 후 이어지는 else if를 통해서 2의 배수와 3의 배수를 확인하여 출력한다.

03. if~else문

LAB 5-2 정답

```
01  #define _CRT_SECURE_NO_WARNINGS
02  #include <stdio.h>
03
04  int main()
05  {
06      int n;
07      printf("정수를 입력하세요.\r\n");
08      scanf("%d", &n);
09
10      if(!(n % 2) && !(n % 3))
11          printf("2와 3의 공배수입니다.");
12      else if(!(n % 2))
13          printf("2의 배수입니다.");
14      else if(!(n % 2))
15          printf("3의 배수입니다.");
16      else
17          printf("2와 3의 공배수가 아닙니다.");
18  }
```

04

switch문

04. switch문

I. switch문의 형식

- **switch문의 개념**
  - 조건식이 계산된 값에 따라서 다중 분기를 할 수 있는 제어문
- **switch문의 사용 형식**

```
----------------------  switch문 시작
switch(조건식)
{                       라벨(Label)
    case 값1: 명령절1
    case 값2: 명령절2
    ...
    case 값N: 명령절N
    default: 명령절D    미사용 시 생략 가능함
}
----------------------  switch문 끝
```

04. switch문

I. switch문의 형식

- **switch문과 if문의 차이점**
  - switch문은 조건식의 참, 거짓 여부를 따지지 않고, 오직 조건식의 계산 값과 case 값(Label)의 일치 여부만을 따짐
- **switch문의 구성과 실행 흐름**
  - 조건식을 계산하여 특정한 명령절로 실행 흐름이 변경된 후에는 특별한 명령이 없는 한 switch문은 탈출하지 않음

하나 더 알기 **명령절이란?**

- 저자가 만든 개념으로, 0개 이상의 명령문이 나열된 것을 의미함
- 하나의 case에 여러 개의 명령문을 사용할 수 있음

04. switch문

I. switch문의 형식

[코드 5-8] switch

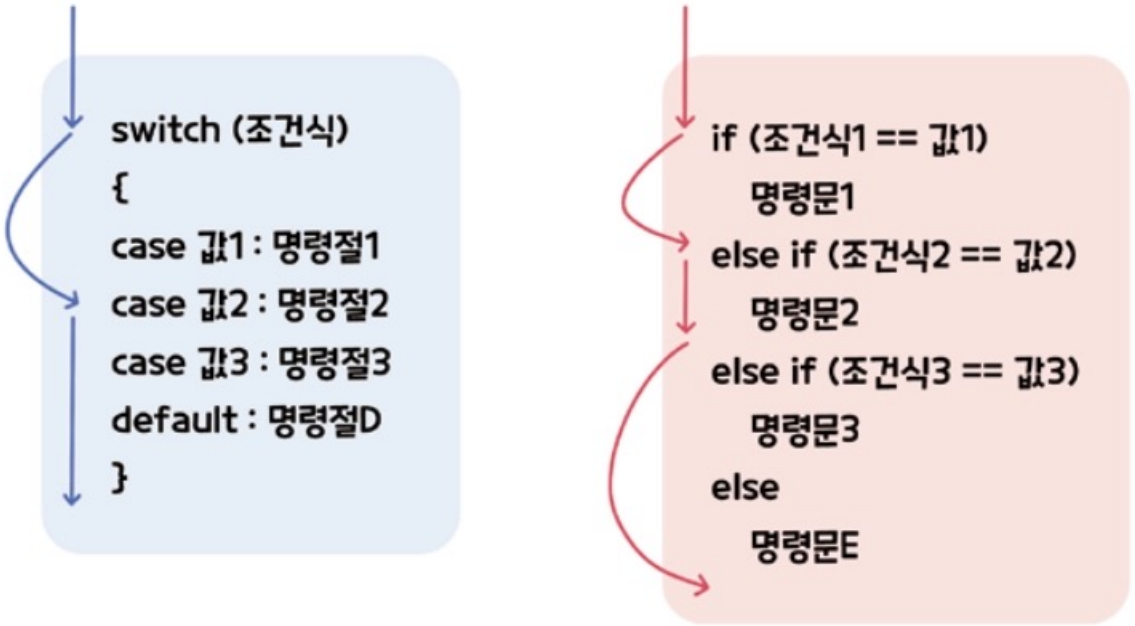
```
01  #include <stdio.h>
02
03  int main()
04  {
05      switch(1 + 1)                  // case 2로 실행 흐름 분기
06      {
07      case 1: printf("1\r\n");
08      case 2: printf("2\r\n");
09              printf("2A\r\n");
10      case 3: printf("3\r\n");
11          {
12              printf("3A\r\n");
13              printf("3B\r\n");
14          }
15      default: printf("D");
16      }
17  }
```

```
2
2A
3
3A
3B
D
```

04. switch문

I. switch문의 형식

- switch문과 if문의 실행 흐름 비교



(a) switch문의 실행 흐름 (b) if~else if문의 실행 흐름

그림 5-3 switch문과 if~else if문의 실행 흐름 비교

04. switch문

II. break

- **break의 개념**
  - 자신을 포함하고 있는 switch문을 강제로 빠져나오게 하는 명령
- **break의 사용 형식**

```
---------------------  switch문 시작
switch(조건식)
{                      라벨(Label)
    case 값1: 명령절1 break;
    case 값2: 명령절2 break;
    ...
    case 값N: 명령절N break;
    default: 명령절D   미사용 시 생략 가능함
}
---------------------  switch문 끝
```

04. switch문

II. break

[코드 5-9] break를 활용한 다중 분기문

```
01  #include <stdio.h>
02
03  int main()
04  {
05     switch(2)                          // case 2로 실행 흐름 분기
06     {
07     case 1: printf("1"); break;
08     case 2: printf("2"); break;        // 실행 후 switch문 탈출
09             printf("3"); break;
10     default: printf("D");
11     }
12  }
```

```
2
```

04. switch문

II. break

- **break의 활용**
  - switch문에서 break 없이 사용하는 경우가 상당히 많음
  - **예제)** 초등학교 저학년에서 수학 시험을 볼 때, 열 문제에 개당 10점씩 100점 만점일 경우 점수에 따라 메시지를 출력하는 프로그램
    - » 100점 : 매우 잘함
    - » 80~90점 : 잘함
    - » 50~70점 : 보통
    - » 0~40점 : 노력이 필요함

04. switch문

[코드 5-10] break의 선별 사용

```
01  #define _CRT_SECURE_NO_WARNINGS
02  #include <stdio.h>
03
04  int main()
05  {
06     printf("수학 점수를 입력하세요\r\n");
07
08     int math;
09     scanf("%d", &math);
10
11     switch(math)
12     {
13     case 100: printf("매우 잘함"); break;
14     case 90:
15     case 80: printf("잘함"); break;
16     case 70:
17     case 60:
18     case 50: printf("보통"); break;
19     default: printf("노력이 필요함"); break;
20     }
21  }
```

```
수학 점수를 입력하세요!
70
보통
```

04. switch문

III. switch문의 제약 사항

- 조건식은 반드시 정수값으로 계산되어야 함
- case의 값은 정수로서 상수여야 함
- 위의 제약을 어길 경우 컴파일 오류가 발생함

04. switch문

III. switch문의 제약 사항

[코드 5-11] switch 제약

```
01  #include <stdio.h>
02
03  int main()
04  {
05     switch(3.14)                        // 정수 아님
06     {
07     case 3.14: printf("3.14");          // 정수 상수 아님
08     }
09
10     switch("C언어")                     // 정수 아님
11     {
12     case "C언어": printf("C언어");      // 정수 상수 아님
13     }
14     int i = 3;
15     switch(i)
16     {
17     case i: printf("3");                // 정수 상수 아님
18     }
19  }
```

04. switch문

확인문제4

1. 다음 코드의 실행 결과를 기술하고, 그 이유를 설명하시오.

```
01  #include <stdio.h>
02
03  int main()
04  {
05      switch(3)
06      {
07      case 1: printf("1");
08      case 2: printf("2");
09      case 3: printf("3");
10      default: printf("D");
11      }
12  }
```

2. 다음 코드의 실행 결과를 기술하고, 그 이유를 설명하시오.

```
01  #include <stdio.h>
02
03  int main()
04  {
05      switch(1)
06      {
07      case 1: printf("1");
08      case 2: printf("2"); break;
09      case 3: printf("3");
10      default: printf("D");
11      }
12  }
```

04. switch문

3. 다음 코드의 잘못된 점과 고치는 방법을 서술하시오.

```
01   int main()
02   {
03       const int L1 = 1, L2 = 2;
04       switch(1)
05       {
06       case L1:
07       case L2:
08       }
09   }
```

[실전예제]

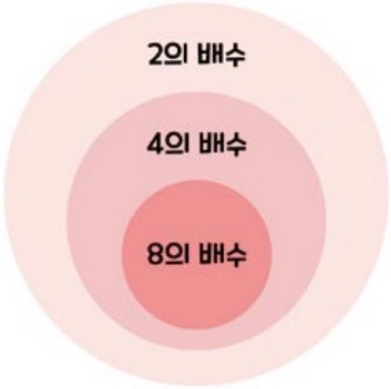
8의 배수? 4의 배수? 2의 배수?

[실전예제] 8의 배수? 4의 배수? 2의 배수?

[문제]

정수를 입력받아 8의 배수, 4의 배수, 2의 배수 중 어디에 해당하는지 출력하는 프로그램을 작성해봅시다.

단, 중복해서 해당될 수 있습니다.



실행 결과

```
정수를 입력하세요!
32
8의 배수, 4의 배수, 2의 배수
```

[실전예제] 8의 배수? 4의 배수? 2의 배수?

[해결]

1. 8의 배수는 4의 배수이자 2의 배수이고, 4의 배수는 2의 배수이다.

2. 따라서 입력받은 수가 8의 배수인지, 4의 배수인지, 2의 배수인지를 순서대로 따져서 변수 c에 8, 4, 2를 대입한다.

3. switch문의 case는 8의 배수, 4의 배수, 2의 배수를 묶음 처리하고 마지막에 break를 한다.

4. 흐름이 6개가 되는지 확인한다.

[실전예제] 8의 배수? 4의 배수? 2의 배수?

[해결]

```
01  #define _CRT_SECURE_NO_WARNINGS
02  #include <stdio.h>
03
04  int main()
05  {
06     int n;
07     printf("정수를 입력하세요!\r\n");
08     scanf("%d", &n);
09
10     int c = 0;
11     if(n % 8 == 0)
12         c = 8;
13     else if(n % 4 == 0)
14         c = 4;
15     else if(n % 2 == 0)
16         c = 2;
17
18   switch(c)
19     {
20     case 8: printf("8의 배수, ");
21     case 4: printf("4의 배수, ");
22     case 2: printf("2의 배수"); break;
23     default: printf("2, 4, 8의 배수 아님");
24     }
25  }
```

Thank you!

한빛아카데미 Hanbit Academy, Inc.