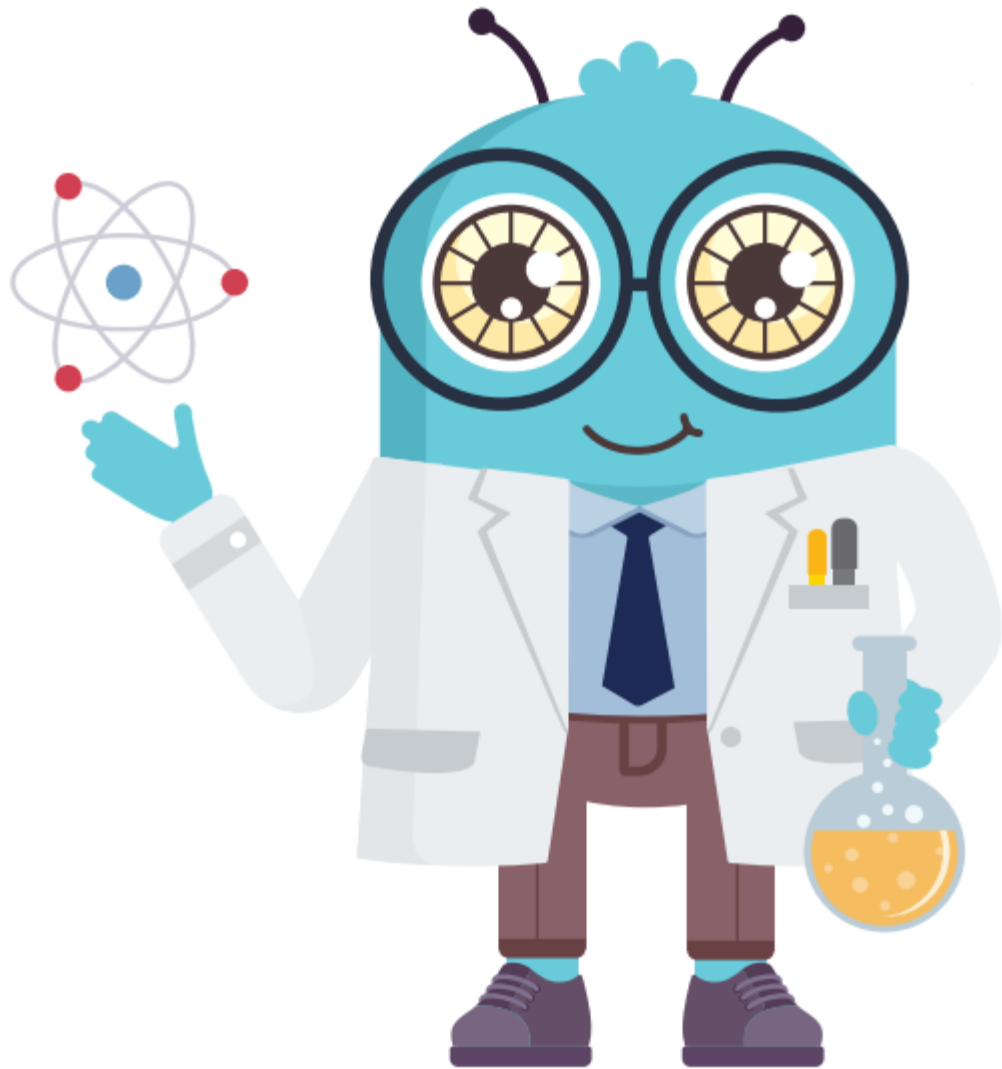


Chapter 05

조건문



목차

1. if문
2. 중첩 if문
3. switch~case문

01

if문

1. if문

1. 기본 if문

- 조건문(선택 제어문)에는 if문과 switch문이 있는데 둘 다 조건에 따라서 다른 내용이 실행되도록 실행의 흐름을 제어하는 명령문
- 'if'라는 단어의 사전상 의미는 '만약 ~라면'으로, C에 서도 '만약'이라는 어떤 조건을 내세울 때 if문을 사용
- 조건식이 참이면 '실행할 문장'을 실행하고 조건식이 거짓이면 실행하지 않음

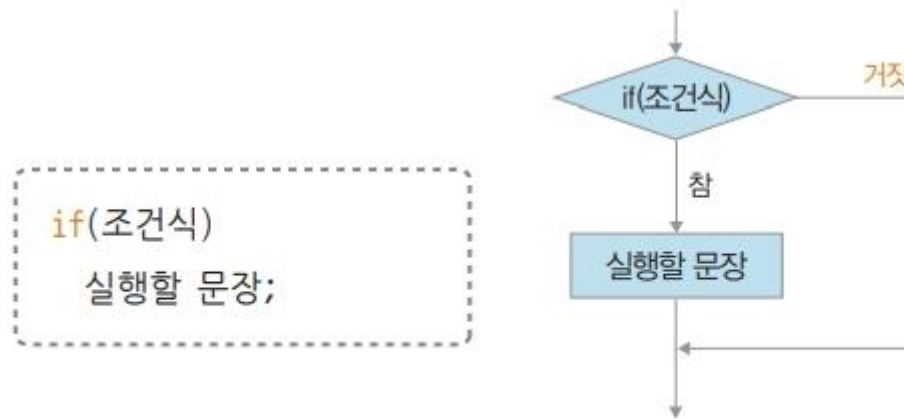


그림 5-1 if문의 형식과 순서도

1. if문

1. 기본 if문

기본 5-1 기본 if문 사용 예 1

5-1.c

```
01 #include <stdio.h>
02
03 void main( )
04 {
05     int a = 99;
06
07     if(a < 100) ----- a가 100보다 작으므로 참이다.
08         printf("100보다 작군요..\n");
09 }
```

실행 결과

100보다 작군요..

1. if문

1. 기본 if문

- 현재 a에 99가 들어 있어 조건식 ' $a < 100$ '이 참이므로 if문 안의 문장을 실행

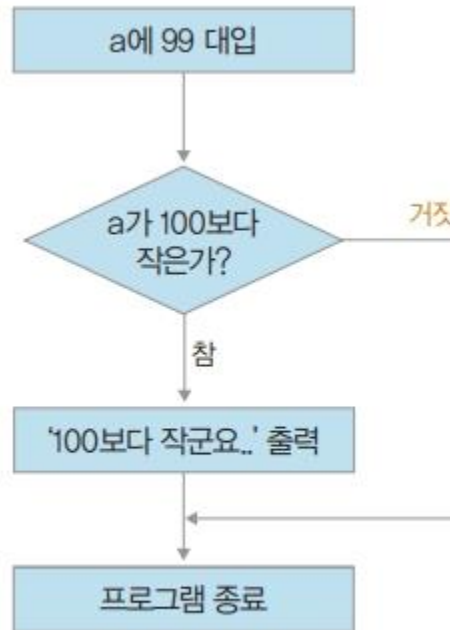


그림 5-2 [기본 5-1]의 실행 과정

1. if문

1. 기본 if문

기본 5-2 기본 if문 사용 예 2

5-2.c

```
01 #include <stdio.h>
02
03 void main( )
04 {
05     int a = 200;
06
07     if(a < 100)
08         printf("100보다 작군요..\n");
09         printf("거짓이므로 이 문장은 안 보이겠죠?\n");
10
11     printf("프로그램 끝! \n");
12 }
```

----- 7행이 참일 경우에 수행할 것으로 예상된다.

----- 7행이 거짓이면 8~9행을 수행하지 않고 11행만 수행할 것으로 예상된다.

실행 결과

거짓이므로 이 문장은 안 보이겠죠?
프로그램 끝!

1. if문

1. 기본 if문

- 그런데 5행의 a가 200이므로 7행의 조건식은 거짓이 되어 8~9행 을 건너뛰고 11행 을 실행할 것으로 예상했는데 9행도 실행
- 이는 바로 '행 바꿈의 함정' 때문, 다음과 같이 행 바꿈을 수정해서 다시 실행

```
01  #include <stdio.h>
02
03  void main( )
04  {
05      int a = 200;
06
07      if(a < 100)
08          printf("100보다 작군요..\n");
09
10      printf("거짓이므로 이 문장은 안 보이겠죠?\n");
11      printf("프로그램 끝! \n");
12  }
```


1. if문

1. 기본 if문

기본 5-3 기본 if문 사용 예 3

5-3.c

```
01 #include <stdio.h>
02
03 void main( )
04 {
05     int a = 200;
06
07     if(a < 100)
08     {
09         printf("100보다 작군요..\n");
10         printf("거짓이므로 앞의 문장은 안 보이겠죠?\n");
11     }
12
13     printf("프로그램 끝! \n");
14 }
```

—— 7행이 참이면 블록으로 감싼 부분이 모두 수행된다.

실행 결과

프로그램 끝!

1. if문

2. if~else문

- 참일 때 실행하는 문장과 거짓일 때 실행하는 문장이 다른 경우 if~else문을 사용
- 조건식이 참이라면 '실행할 문장 1'을 실행하고, 그렇지 않으면 '실행할 문장 2' 실행

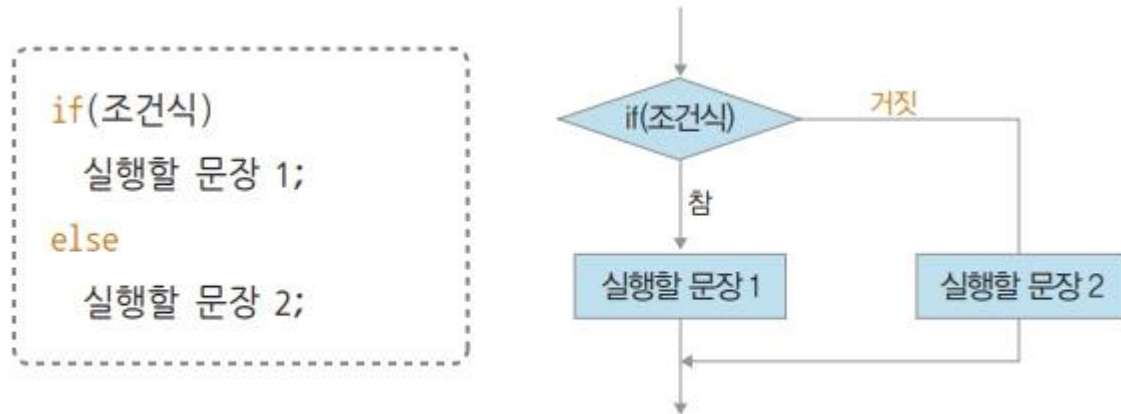


그림 5-3 if~else문의 형식과 순서도

- [그림 5-3]을 보면 조건이 참일 때 실행할 문장과 거짓일 때 실행할 문장이 다르다는 것을 알 수 있음

1. if문

2. if~else문

기본 5-4 if~else문 사용 예

5-4.c

```
01 #include <stdio.h>
02
03 void main( )
04 {
05     int a = 200;
06
07     if(a < 100)
08         printf("100보다 작군요..\n"); —— 7행이 참이면(a가 100보다 작으면) 실행된다.
09     else
10         printf("100보다 크군요..\n"); —— 7행이 거짓이면(a가 100보다 크거나 같으면) 실행된다.
11 }
```

실행 결과

100보다 크군요..

1. if문

2. if~else문

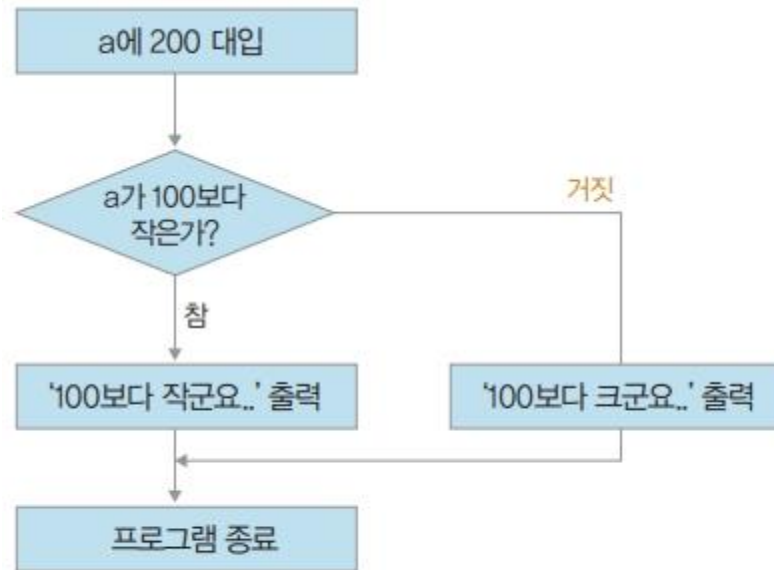


그림 5-4 [기본 5-4]의 실행 과정

- 만약 참일 때 실행할 문장도 여러 개이고 거짓일 때 실행할 문장도 여러 개라면 블록({ })으로 묶도록 함

1. if문

2. if~else문

기본 5-5 블록을 활용한 if~else문 사용 예 1

5-5.c

```
01 #include <stdio.h>
02
03 void main( )
04 {
05     int a = 200;
06
07     if(a < 100)
08     {
09         printf("100보다 작군요..\n");
10         printf("참이면 이 문장도 보이겠죠?\n");
11     }
12     else
13     {
14         printf("100보다 크군요..\n");
15         printf("거짓이면 이 문장도 보이겠죠?\n");
16     }
17
18     printf("프로그램 끝! \n");
19 }
```

—— 7행이 참이면(a가 100보다 작으면) 실행된다.

—— 7행이 거짓이면(a가 100보다 크거나 같으면) 실행된다.

실행 결과

100보다 크군요..
거짓이면 이 문장도 보이겠죠?
프로그램 끝!

1. if문

2. if~else문

응용 5-6 블록을 활용한 if~else문 사용 예 2

5-6.c

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 void main( )
04 {
05     int a;
06
07     printf("정수를 입력하세요 : ");
08     scanf("%d", &a);      —— 정수를 입력받는다.
09
10     if(a % 2 == 0)        —— 입력된 값을 2로 나눈 나머지가 0일 경우
11     {                    —— 짝수이면(2로 나눈 나머지가 0이면) 실행된다.
12         printf("짝수를 입력했군요..\n");
13     }
14     else
15     {                    —— 홀수이면(2로 나눈 나머지가 0이 아니면) 실행된다.
16         printf("홀수를 입력했군요..\n");
17     }
18 }
```

실행 결과

실행 결과

정수를 입력하세요 : 125
홀수를 입력했군요..

02

중첩 if문

2. 중첩 if문

- if문을 한 번 실행하고 그 결과에 다시 if문을 실행하는 것을 '중첩 if문(또는 중복 if문)'이라고 함
- '조건식 1'이 참이면 'if(조건식 2)'를 실행하고 '조건식 2'가 참이면 '실행할 문장 1'을 실행하는 식

```
if(조건식 1){  
    if(조건식 2)  
        실행할 문장 1;  
    else  
        실행할 문장 2;  
}  
else  
    실행할 문장 3;
```

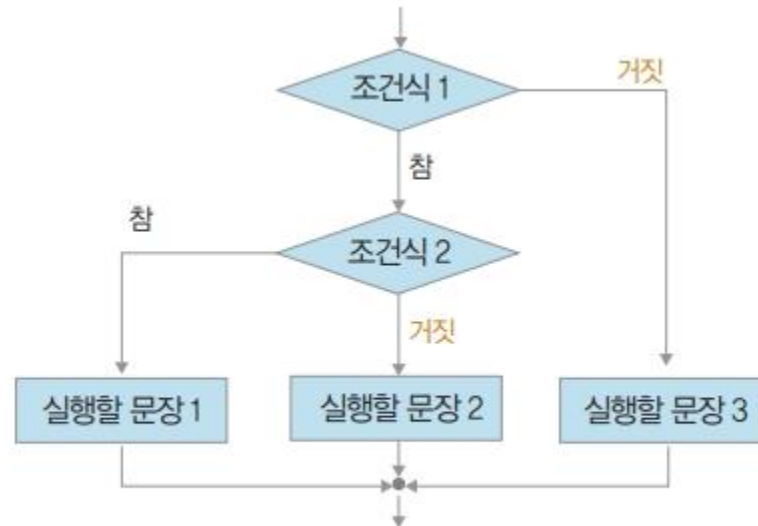


그림 5-5 중첩 if문의 형식과 순서도

2. 중첩 if문

기본 5-7 중첩 if문 사용 예 1

5-7.c

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

```
02
```

```
03 void main( )
```

```
04 {
```

```
05     int a = 75;
```

```
06
```

```
07     if(a > 50)
```

```
08     {
```

```
09         if(a < 100)
```

```
10         {
```

```
11             printf("50보다 크고 100보다 작군요..\n");
```

```
12         }
```

```
13     else
```

```
14     {
```

```
15         printf("와~ 100보다 크군요..\n");
```

```
16     }
```

```
17 }
```

7행이 참이면(a가 50보다 크면)
실행된다.

7행이 참이고(a가 50보다 크고)
9행이 참이면(a가 100보다 작으면)
실행된다.

7행이 참이고(a가 50보다 크고)
9행이 거짓이면(a가 100보다 크거나
같으면) 실행된다.

2. 중첩 if문

```
18  else
19  {
20      printf("에게~ 50보다 작군요..\n");
21  }
22 }
```

—— 7행이 거짓이면(a가 50보다 작거나
같으면) 실행된다.

실행 결과

50보다 크고 100보다 작군요..

- [기본 5-7]의 7행에서 a가 50보다 크면 참이므로 블록 안(8~17행)의 내용을 실행
- 블록 안에서 a가 100보다 작으면 11행을 출력하고 그렇지 않으면 15행을 실행
- 만약 7행에서 a가 50보다 작거나 같으면 18행으로 바로 넘어가서 20행을 실행

2. 중첩 if문

응용 5-8 중첩 if문 사용 예 2

5-8.c

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 void main( )
04 {
05     int a;
06
07     printf("점수를 입력하세요 : ");
08     scanf("%d", &a);      ----- 점수를 입력한다(100점 만점).
09
```

2. 중첩 if문

```
10  if(__1__)
11      printf("A");
12  else
13      if(__2__)
14          printf("B");
15      else
16          if(a >= 70)
17              printf("C");
18          else
19              if(a >= 60)
20                  printf("D");
21      else
22          printf("F");
23
24  printf(" 학점 입니다. \n");
25 }
```

입력한 점수가 90점 이상이면 A 학점을 출력한다.

입력한 점수가 80점 이상이면 B 학점, 70점 이상이면 C 학점, 60점 이상이면 D 학점을 출력한다.

입력한 점수가 60점 미만이면 F 학점을 출력한다.

08 =< e 06 =< e 1 135

실행 결과

점수를 입력하세요 : 77
C 학점 입니다.

2. 중첩 if문

- [응용 5-8]의 10행에서 a가 90점 이상이면 11행을 실행한 후 바로 23행으로 빠져나옴
- 그렇지 않으면(a가 90점 미만이면) 13행으로 이동
- 여기서 a가 80점 이상이면 14행을 실행한 후 바로 23행으로 빠져나옴
- 그렇지 않으면(a가 80점 미만이면) 16행으로 가서 바로 23행으로 빠져나옴

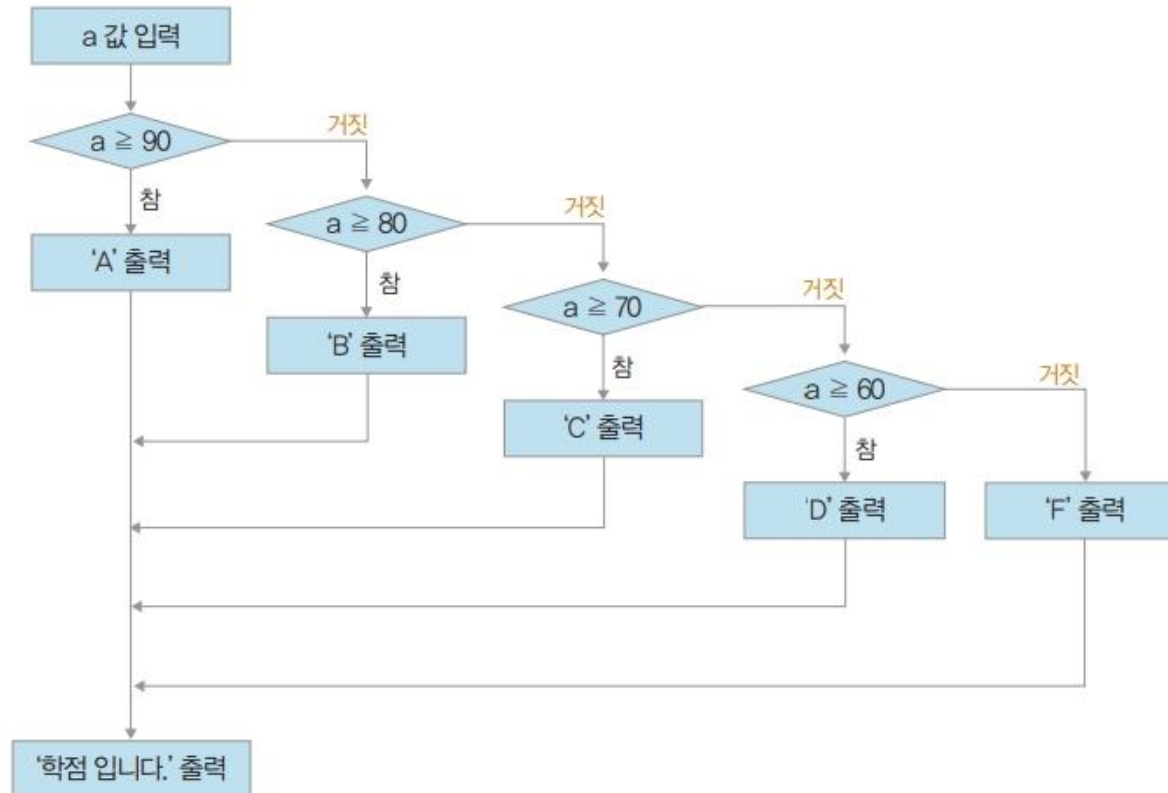


그림 5-6 [응용 5-8]의 실행 과정

03

switch~case문

3. switch~case문

- 여러 개 중에 하나를 선택을 '다중 분기' 라고 함
- 이 구문은 switch의 조건값(정숫값)에 따라 case문을 실행
- 정숫값이 1이면 1의 값에 해당하는 문장, 정숫값이 2이면 2의 값에 해당하는 문장을 실행하는 방식

```
switch(정숫값){  
  case 정숫값 1:  
    실행할 문장 1;  
    break;  
  case 정숫값 2:  
    실행할 문장 2;  
    break;  
  default:  
    실행할 문장 3;  
    break;  
}
```

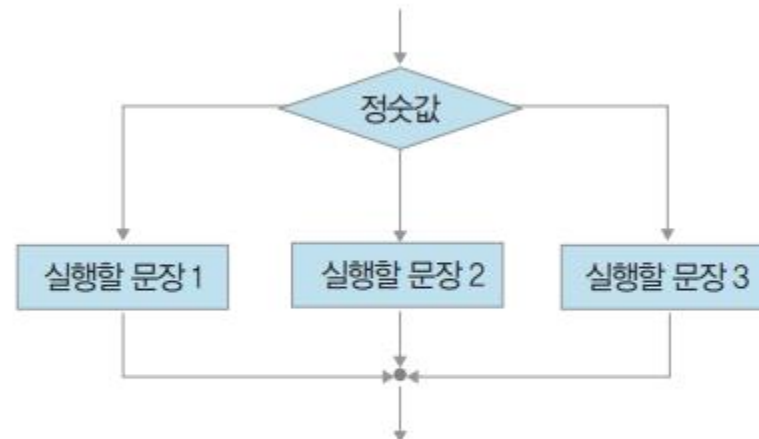


그림 5-7 switch~case문의 형식과 순서도

3. switch~case문

기본 5-9 switch~case문 사용 예 1

5-9.c

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 void main( )
04 {
05     int a;
06
07     printf("1 ~ 4 중에 선택하세요 : ");
08     scanf("%d", &a);
09
10     switch(a)          ----- 입력한 a 값에 따라서 분기한다.
11     {
12         case 1:          ----- a가 1이면 13행을 수행하고 14행에서 switch 블록을
13             printf("1을 선택했다\n");          빠져나간다.
14             break;
15         case 2:
16             printf("2를 선택했다\n");
17             break;
18         case 3:
19             printf("3을 선택했다\n");
20             break;
21         case 4:
22             printf("4를 선택했다\n");
23             break;
24         default:          ----- a가 1, 2, 3, 4 중 어디에도 해당되지 않을 때
25             printf("이상한 걸 선택했다.\n");      수행된다.
26     }
27 }
```

실행 결과

1 ~ 4 중에 선택하세요 : 3
3을 선택했다

3. switch~case문

- [기본 5-9]의 8행에서 입력한 숫자가 10행의 switch문으로 넘어감
- 만약 1이면 12행으로, 2이면 15행으로, 3이면 18행으로, 4이면 21행으로 넘어감
- 입력한 숫자가 1, 2, 3, 4가 아니면 24행의 default 부분이 실행

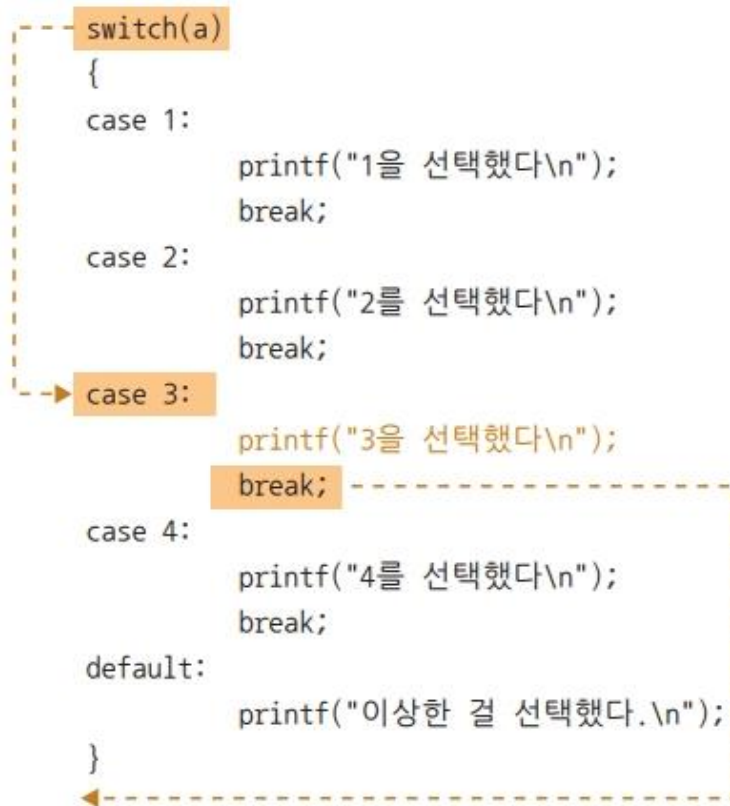


그림 5-8 a가 3일 때의 switch~case문 흐름도

3. switch~case문

- 각각의 실행문 마지막에 반드시 break문을 써야 한다는 것이 중요
- 그렇지 않으면 switch~case문을 빠져나가지 못한 채 다음 코드를 계속 수행

```
10 switch(a)
11 {
12 case 1:
13     printf("1을 선택했다\n");
14 case 2:
15     printf("2를 선택했다\n");
16 case 3:
17     printf("3을 선택했다\n");
18 case 4:
19     printf("4를 선택했다\n");
20 default:
21     printf("이상한 걸 선택했다.\n");
22 }
```

실행 결과

1 ~ 4 중에 선택하세요 : 2
2를 선택했다
3을 선택했다
4를 선택했다
이상한 걸 선택했다.

3. switch~case문

응용 5-10 switch~case문 사용 예 2

5-10.c

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 void main( )
04 {
05     int year;
06
07     printf("출생연도를 입력하세요 : ");
08     scanf("%d", &year);
09
10     switch( __1__ )
11     {
12         case 0 : printf("원숭이띠\n"); break;
13         case 1 : printf("닭띠\n"); break;
14         case 2 : printf("개띠\n"); break;
15         case 3 : printf("돼지띠\n"); break;
16         case 4 : printf("쥐띠\n"); break;
17         case 5 : printf("소띠\n"); break;
18         case 6 : printf("호랑이띠\n"); break;
19         case 7 : printf("토끼띠\n"); break;
20         case 8 : printf("용띠\n"); break;
21         case 9 : printf("뱀띠\n"); break;
22         case 10 : printf("말띠\n"); break;
23         case 11 : printf("양띠\n"); break;
24     }
25 }
```

입력한 연도를 12로 나눈 나머지값에
따라서 결과를 출력한다.

원숭이띠\nyear % 12

실행 결과

출생연도를 입력하세요 : 2003
양띠

*

예제 모음

[예제모음 11] 단순 if문을 활용한 간단한 계산기

예제 설명 단순 if문을 활용하여 두 수의 +, -, *, /, % 연산을 수행하는 프로그램이다.

실행 결과

첫 번째 수를 입력하세요 : 5
계산할 연산자를 입력하세요 : *
두 번째 수를 입력하세요 : 7
5 * 7 = 35 입니다.

[예제모음 11] 단순 if문을 활용한 간단한 계산기

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 void main( )
04 {
05     int a, b;
06     char ch;
07
08     printf("첫 번째 수를 입력하세요 : ");
09     scanf("%d", &a);
10     printf("계산할 연산자를 입력하세요 : ");
11     scanf(" %c", &ch);
12     printf("두 번째 수를 입력하세요 : ");
13     scanf("%d", &b);
14
15     if(ch == '+')
16         printf("%d + %d = %d 입니다. \n", a, b, a+b);
17
18     if(ch == '-')
19         printf("%d - %d = %d 입니다. \n", a, b, a-b);
20
21     if(ch == '*')
22         printf("%d * %d = %d 입니다. \n", a, b, a*b);
23
24     if(ch == '/')
25         printf("%d / %d = %f 입니다. \n", a, b, a/(float)b);
26
27     if(ch == '%')
28         printf("%d %% %d = %d 입니다. \n", a, b, a%b);
29 }
```

입력받을 정수 2개와 연산자 문자 1개를 선언한다.

계산할 첫 번째 숫자를 입력한다.

연산자를 입력한다.

계산할 두 번째 숫자를 입력한다.

기본 if문을 사용한 연산을 수행한다.

[예제모음 12] 중복 if문을 활용한 간단한 계산기

예제 설명 중복 if문을 활용하여 두 수의 +, -, *, /, % 연산을 수행하는 프로그램이다.

실행 결과

첫 번째 수를 입력하세요 : 12
계산할 연산자를 입력하세요 : /
두 번째 수를 입력하세요 : 5
12 / 5 = 2.400000 입니다.

첫 번째 수를 입력하세요 : 88
계산할 연산자를 입력하세요 : &
두 번째 수를 입력하세요 : 77
연산자를 잘못 입력했습니다.

[예제모음 12] 중복 if문을 활용한 간단한 계산기

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 void main( )
04 {
05     int a, b;
06     char ch;
07
08     printf("첫 번째 수를 입력하세요 : ");
09     scanf("%d", &a);
10     printf("계산할 연산자를 입력하세요 : ");
11     scanf(" %c", &ch);
12     printf("두 번째 수를 입력하세요 : ");
13     scanf("%d", &b);
14
15     if(ch == '+')
16         printf("%d + %d = %d 입니다. \n", a, b, a+b);
17     else if(ch == '-')
18         printf("%d - %d = %d 입니다. \n", a, b, a-b);
19     else if(ch == '*')
20         printf("%d * %d = %d 입니다. \n", a, b, a*b);
21     else if(ch == '/')
22         printf("%d / %d = %f 입니다. \n", a, b, a/(float)b);
23     else if(ch == '%')
24         printf("%d %% %d = %d 입니다. \n", a, b, a%b);
25     else
26         printf("연산자를 잘못 입력했습니다. \n");
27 }
```

중복 if문을 사용한 연산을 수행한다.

+, -, *, /, % 외의 문자를 입력하면 오류 메시지를 보여준다.

[예제모음 13]switch~case문을 활용한 간단한 계산기

예제 설명 수식을 띄어쓰기로 한 줄에 입력받고 switch~case문을 활용하여 두 수의 +, -, *, /, % 연산을 수행하는 프로그램이다.

실행 결과

수식을 한 줄로 띄어쓰기로 입력하세요 : 99 - 22
99 - 22 = 77 입니다.

수식을 한 줄로 띄어쓰기로 입력하세요 : 55 # 99
연산자를 잘못 입력했습니다.

[예제모음 13]switch~case문을 활용한 간단한 계산기

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 void main( )
04 {
05     int a, b;
06     char ch;
07
08     printf("수식을 한 줄로 띄어쓰기로 입력하세요 : ");
09     scanf("%d %c %d", &a, &ch, &b);
10
11     switch(ch)
12     {
13     case '+':
14         printf("%d + %d = %d 입니다. \n", a, b, a+b);
15         break;
16     case '-':
17         printf("%d - %d = %d 입니다. \n", a, b, a-b);
18         break;
19     case '*':
20         printf("%d * %d = %d 입니다. \n", a, b, a*b);
21         break;
```

—— 일반 수식처럼 띄어쓰기를 해서
한 줄에 변수 3개를 입력한다.

—— switch~case문을 사용한 연산을
수행한다.

[예제모음 13]switch~case문을 활용한 간단한 계산기

```
22  case '/':
23      printf("%d / %d = %d 입니다. \n", a, b, a/b);
24      break;
25  case '%':
26      printf("%d * %d = %d 입니다. \n", a, b, a*b);
27      break;
28  default:
29      printf("연산자를 잘못 입력했습니다. \n");
30  }
31 }
```

switch~case문을 사용한 연산을 수행한다.

감사합니다!

