

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

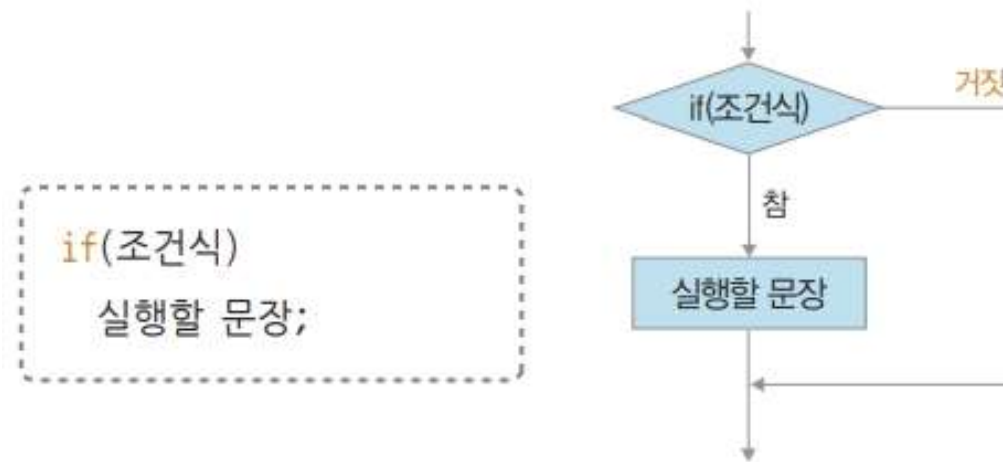


그림 5-1 if문의 형식과 순서도

1. if문

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

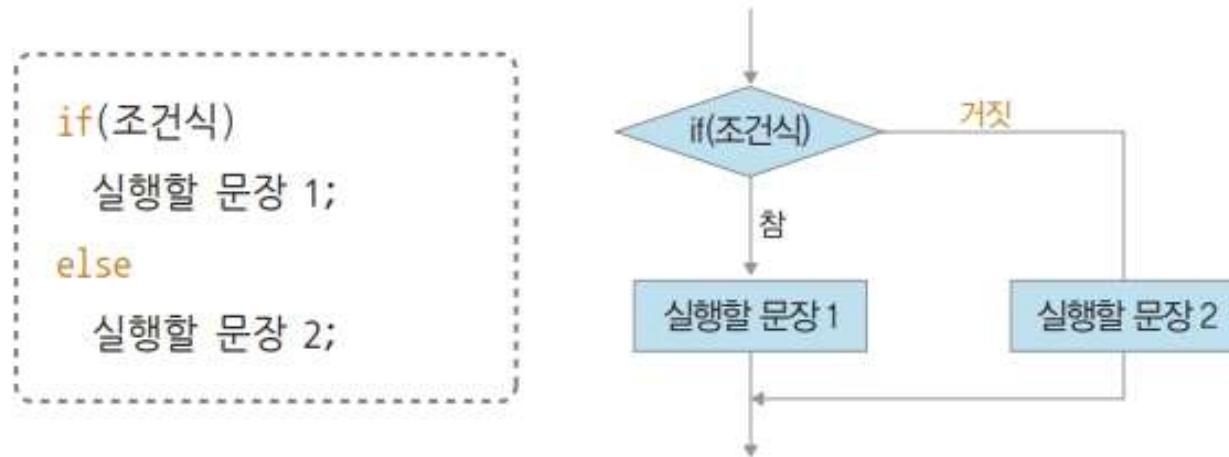


그림 5-3 if~else문의 형식과 순서도

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

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;  
}
```

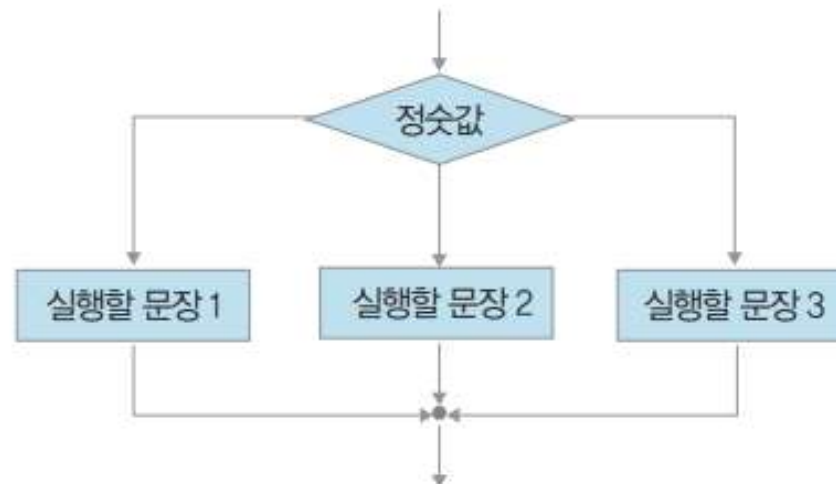


그림 5-7 switch~case문의 형식과 순서도

1. 단순 for문

2. for문의 개념과 활용

- for문의 형식을 보면 괄호 안에 초깃값, 조건식, 증감식이 세미콜론(;)으로 구분됨
- 그리고 중괄호({ }) 안에 반복할 문장들이 있음
- 반복할 문장이 하나뿐이라면 중괄호를 생략해도 무관
- [그림 6-1]을 통해 반복되는 순서를 보면 ❶, ❷를 수행한 뒤 ❸, ❹, ❷ 순서로 반복할 문장들을 계속 수행

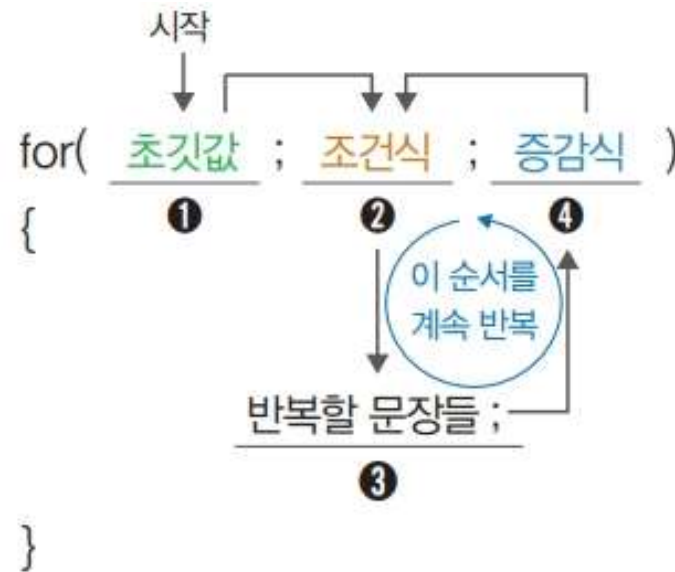


그림 6-1 for문의 기본 형식

1. 단순 for문

2. for문의 개념과 활용



그림 6-2 for문의 개념과 실제 사용

- ❶ for문을 사용하려면 무조건 변수를 하나 준비
- ❷ 사용할 변수의 초기값을 0으로 설정
- ❸ 5가 될 때까지를 검사하기 위해 조건을 `i < 5`로 설정
- ❹ `i++`는 `i=i+1`과 동일한 역할
- ❺ 실제로 반복되는 내용

1. while문

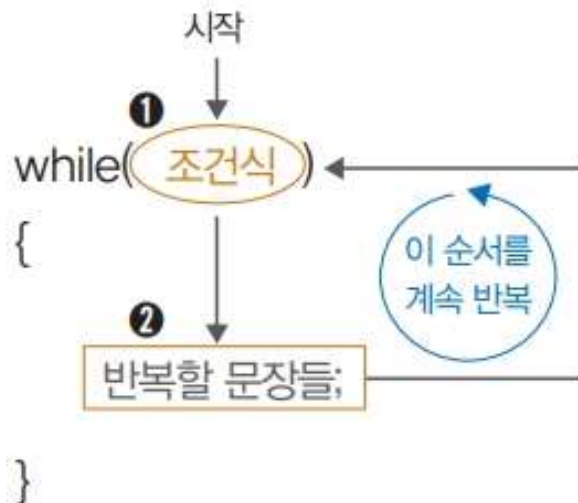
1. for문과 while문의 비교

▪ For문

```
for(초깃값; 조건식; 증감식)
```

▪ while 문의 실행 순서

- 조건식이 참인 동안 반복할 문장 수행
- 종괄호가 끝나는 곳에서 조건식으로 돌아와 같은 동작 반복



1. while문

2. 무한루프를 위한 while문

- 조건식이 무조건 참이어야 함
- `for(; ;)`와 동일한 역할
- `while(1)` 로 표현

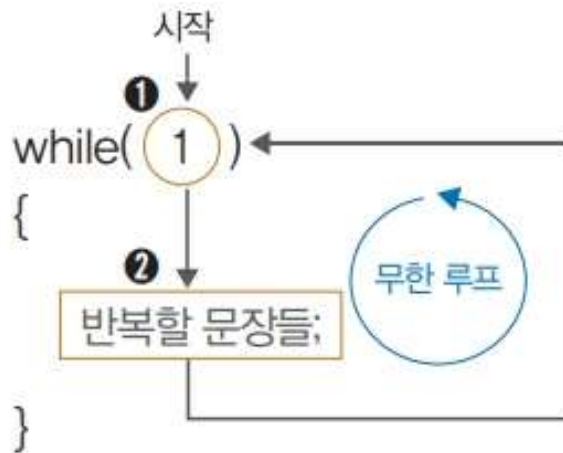


그림 7-3 while문을 이용한 무한 루프

2. do~while문

1. do~while문과 while문의 차이

- 조건식을 확인하기 전에 '반복할 문장'을 수행됨, 무조건 한 번은 실행됨
- 형식은 while 문과 동일하지만, 조건식이 아래에 위치



그림 7-4 do~while문의 형식과 실행 순서

- while문의 경우 처음 의 조건식이 거짓이면 '반복할 문장들'을 한 번도 실행하지 않음
- 하지만 do~while문에서는 조건식을 확인하기 전에 일단 '반복할 문장들'을 실행하므로 조건식이 거짓이든 참이든 무조건 한 번은 실행

3. 기타 제어문

1. 반복문을 탈출하는 break문

- for, while, do~while과 같은 반복문을 탈출할 때 사용
- if 문과 결합하여 무한루프 안에 사용
 - 무한루프를 돌다 특정 조건을 만족하면 프로그램을 종료하는 역할

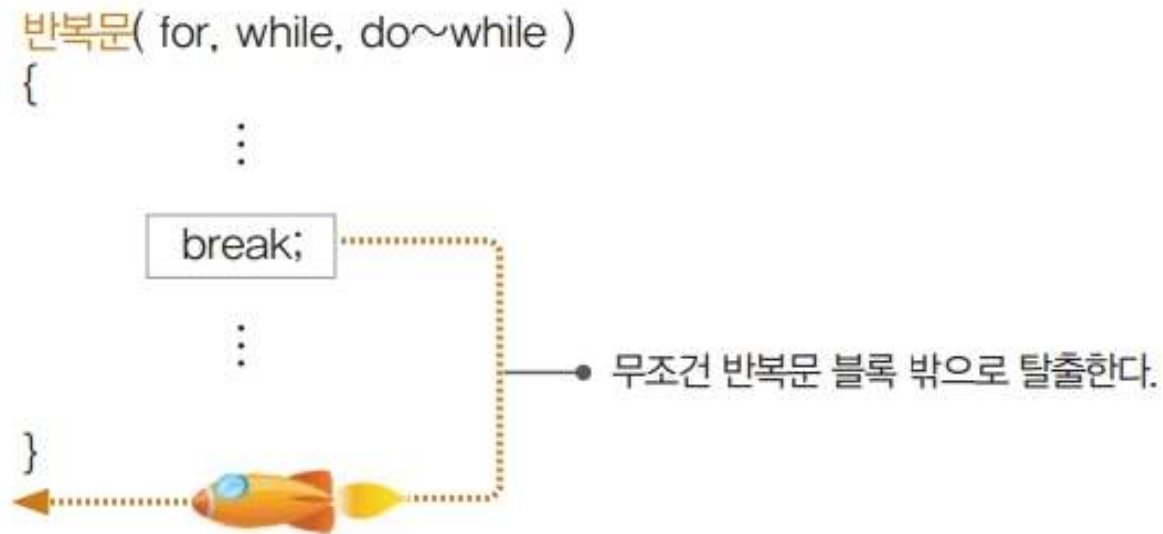


그림 7-5 break문의 작동