

제어문(1) - 조건문

#01. if문

1) 구문형식

괄호 안에 주어진 조건이 참인 경우 **블록 {}** 안을 수행.

조건은 **참**, **거짓** 을 판단할 수 있는 형태면 어떤 구문이건 사용 가능하다.

```
if (조건) {  
    // ... 조건이 참인 경우 실행할 명령 ...  
}
```

구분	연산
비교식	<code>==, !=, >, >=, <, <=</code>
논리식	<code>&&, </code>
논리값	<code>true, false</code>

2) if ~ else문

조건이 참인 경우 if 블록이 실행, 그렇지 않은 경우 else 블록이 실행 된다.

```
if (조건) {  
    // ... 조건이 참인 경우 실행할 명령 ...  
} else {  
    // ... 조건이 참이 아닌 경우 실행할 명령 ...  
}
```

3) if ~ else if ~ else 문

여러 개의 조건을 나열하여 그 중 **가장 처음 만나는 참인 조건의 블록을 수행**하고 그 외의 블록은 실행하지 않고 빠져 나간다.

else if 블록은 필요한 만큼 나열 가능하다.

else 블록은 필요하지 않다면 생략 가능하다.

```
if (1차 조건) {  
    ...  
} else if (2차 조건) {  
    ...  
} else if (3차 조건) {  
    ...  
}
```

```
    ...  
} else {  
    ...  
}
```

#02. switch문

1) 구문 형식

하나의 변수값에 대한 여러 가지 경우의 수를 나열하고 그 중 일치하는 case 블록을 **break 명령을 만나기 전까지 실행함**.

만약 **break;** 명령이 없다면 다음 case 블록도 함께 실행된다.

마지막의 **default** 블록은 일치하는 case를 찾지 못한 경우 실행되며 필요하지 않다면 생략 가능하다.

```
switch (변수) {  
    case '값1':    // 경우의 수는 필요한 만큼 나열  
        ...  
        break;  
    case '값2':  
        ...  
        break;  
    case '값n':  
        ...  
        break;  
    default: // 일치하는 경우가 없을 경우 수행할 블록(생략가능)  
        ...  
        break;  
}
```

2) 의도적으로 break절을 조절하는 경우

조건에 맞는 case가 두 개 이상인 경우.

grade값이 A, B, C인 세 개의 경우가 모두 하나의 문장을 실행할 수 있도록 블록 구조를 구성한다.