



*switch ~ case*

## *switch ~ case*

- switch ~ case

- if ~ else if ~ else를 대체하여 사용 가능
- 단, 조건이 데이터 비교일 때 사용
- 즉, if 문에서 단순 데이터 비교에 의한 else if 문이 다수 일 때 유용
- break; 적절히 사용

- switch ~ case 문의 기본 문법

```
switch(데이터) {  
    case (비교데이터1):    // 데이터가 비교데이터1이면 여기서부터 실행  
        ...  
    case (비교데이터2):    // 데이터가 비교데이터2이면 여기서부터 실행  
        ...  
    case (비교데이터3):    // 데이터가 비교데이터3이면 여기서부터 실행  
        ...  
    default:               // 해당하는 case가 없으면 여기서부터 실행  
        ...  
}
```

## *switch ~ case*

```
class SwitchBasic {  
    public static void main(String[] args) {  
        int n = 1;  
  
        switch(n) {  
            case 1:  
                System.out.println("Simple Java");  
            case 2:  
                System.out.println("Funny Java");  
            default:  
                System.out.println("The best programming language");  
        }  
    }  
}
```

## 연산자

- switch ~ case 문에서 break의 활용

```
class SwitchBasic {  
    public static void main(String[] args) {  
        int n = 1;  
  
        switch(n) {  
            case 1:  
                System.out.println("Simple Java");  
                break;  
            case 2:  
                System.out.println("Funny Java");  
                break;  
            default:  
                System.out.println("The best programming language");  
                break;  
        }  
    }  
}
```

## 연산자

1. SwitchBasic class를 switch~case 문이 아닌 if 문으로 변경해 보시오.
2. 아래의 코드를 if ~ else 문이 아닌 switch 문을 이용하는 형태로 변경해 보시오.

```
class NumberRange {  
    public static void main(String[] args) {  
        int n = 5;  
        if(n >= 0 && n < 3) {  
            System.out.println("0이상 3미만의 수");  
        } else if (n >= 3 && n < 6) {  
            System.out.println("3이상 6미만의 수");  
        } else {  
            System.out.println("음수 또는 6이상의 수")  
        }  
    }  
}
```

3. 두 개의 양의 정수와 사칙 연산 기호(+, -, \*, /)를 하나 입력 받아 계산 결과를 출력하는 계산기 프로그램을 작성하시오.