

어서와 *Java*는 처음이지!

제3장 선택과 반복

○ 조건문

◎ if

◎ switch

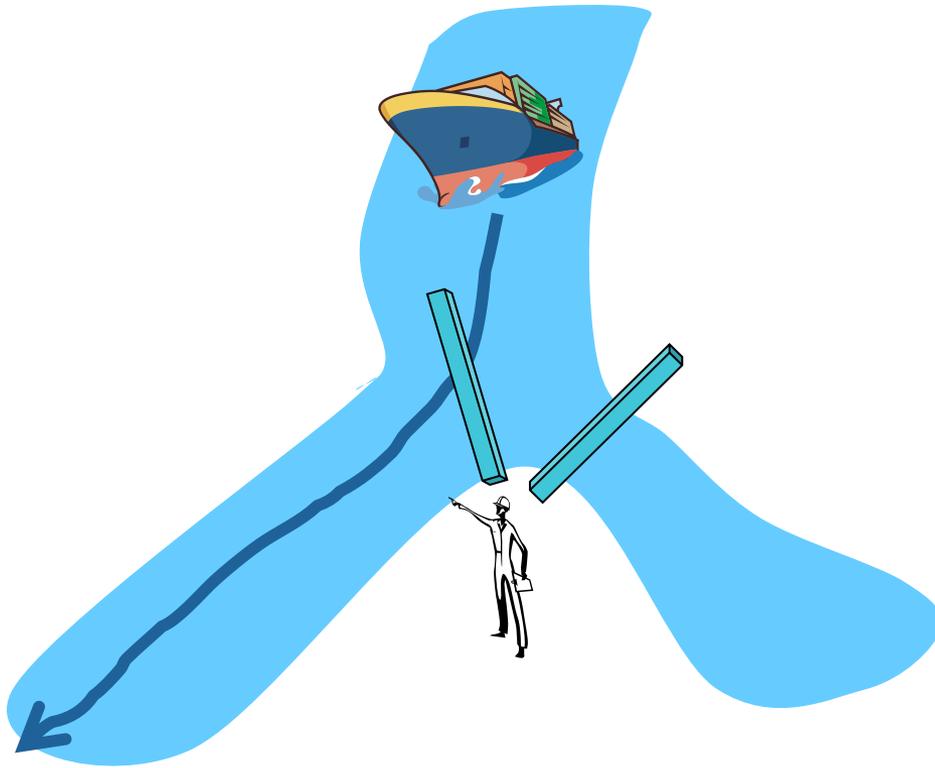
자바에서는 조건을 따져서
문장을 수행시키나봐요.

네, 조건은 프로그래밍에서는
무척 중요합니다. 조건을 따져서
문장을 수행시키기 때문에
컴퓨터가 스마트한 것이죠.





조건문



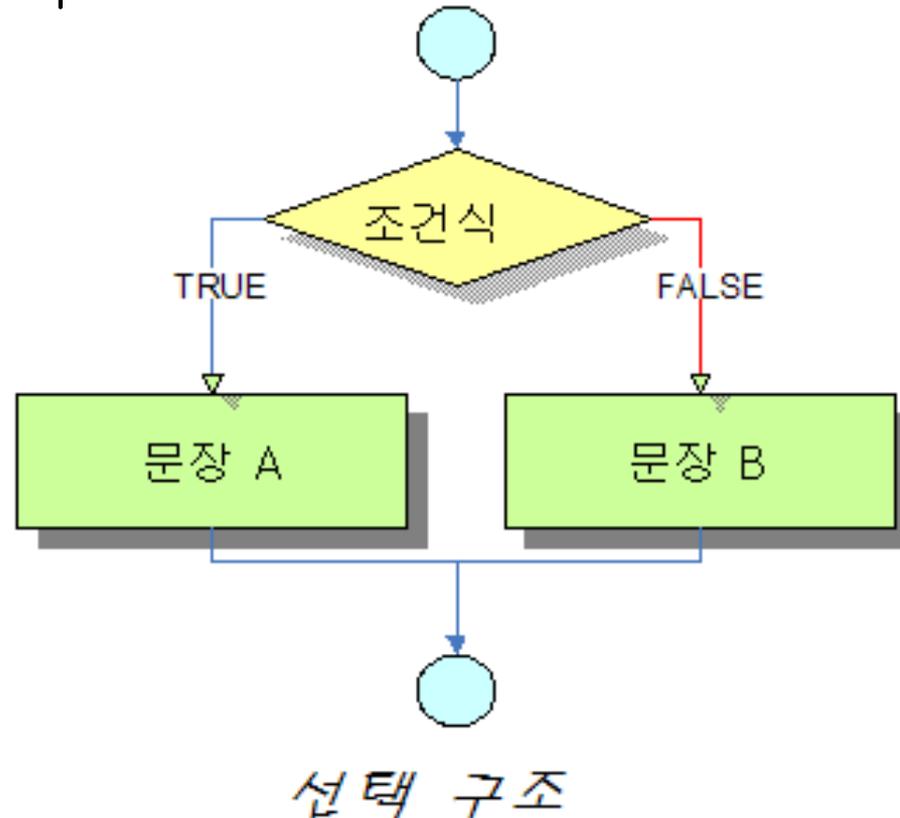
조건문은
프로그램의
흐름을
변경합니다.





조건문

- 조건에 따라서 여러 개의 실행 경로 가운데 하나를 선택



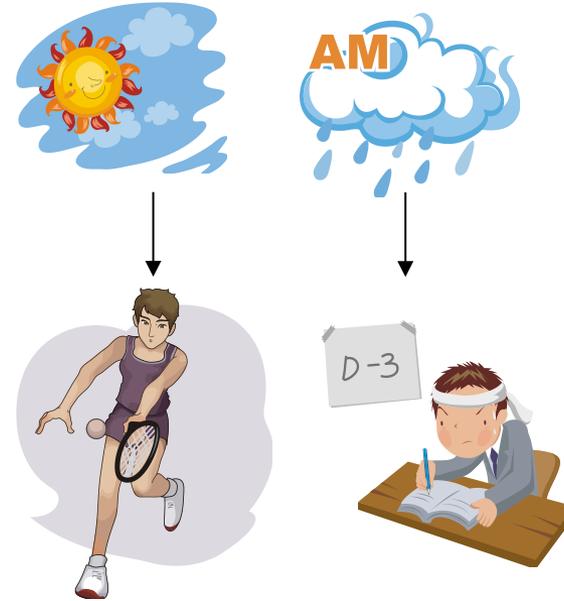
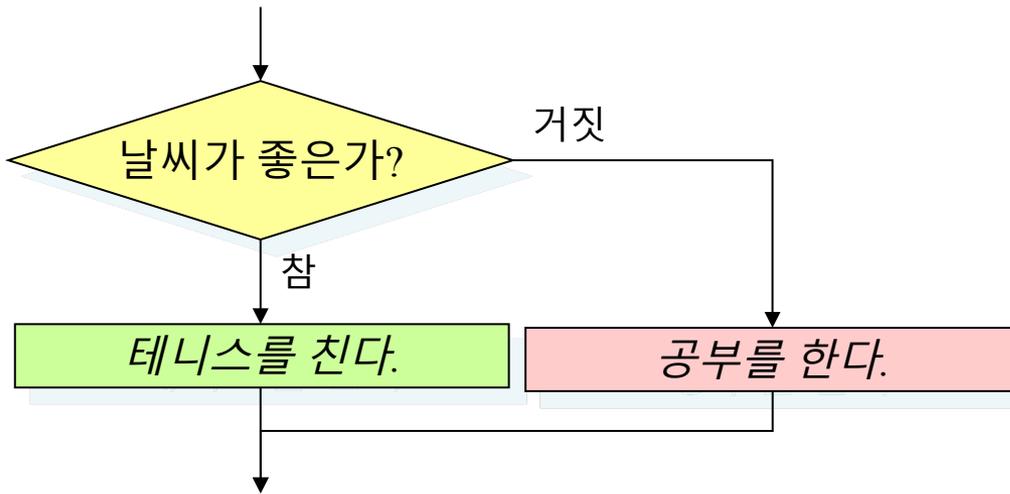


일상 생활에서의 조건문의 예

- 만약 비가 오지 않으면 테니스를 친다.
- 만약 결석이 1/30이상이면 F학점을 받는다.
- 만약 시간이 없는 경우에는 택시를 탄다.
- 만약 날씨가 좋고 공휴일이면 공원에 산책을 간다.
- 점수가 60점 이상이면 합격이고 그렇지 않으면 불합격이다.



if-else 문



if(조건식)

문장 1;

else

문장 2;

조건식이 참이면 실행된다.

조건식이 거짓이면 실행된다.



if-else 선택 구조

형식

조건식

```
if( number > 0 )  
{  
    System.out.println("양수입니다.");  
}  
else  
{  
    System.out.println("0이거나 음수입니다.");  
}
```

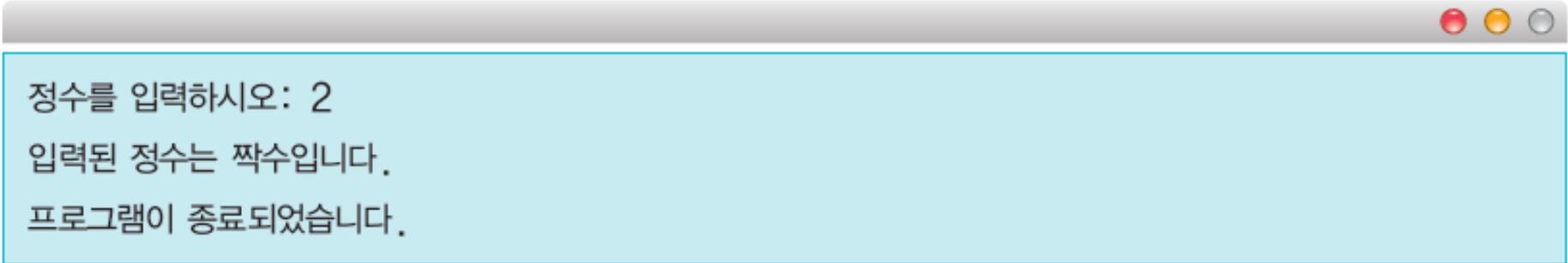
조건식이 참일 때 실행되는
문장, then 절이라고 한다.

조건식이 참이 아닐 때 실행되는
문장, else 절이라고 한다.



예제: 짝수, 홀수 구별하기

키보드에서 입력받은 정수가 홀수인지 짝수인지를 말해주는 프로그램을 작성하여 보자.



```
정수를 입력하시오: 2  
입력된 정수는 짝수입니다.  
프로그램이 종료되었습니다.
```

직접 입력
하여 확인



EvenOdd.java

```
01 import java.util.Scanner;
02
03 public class EvenOdd {
04     public static void main(String[] args) {
05         // if 문을 사용하여 홀수와 짝수를 구별하는 프로그램
06         int number;
07         Scanner input = new Scanner(System.in);
08         System.out.print("정수를 입력하십시오: ");
09         number = input.nextInt();
```

```
10
11         if (number % 2 == 0) {
12             System.out.println("입력된 정수는 짝수입니다.");
13         } else {
14             System.out.println("입력된 정수는 홀수입니다.");
15         }
16
17         System.out.println("프로그램이 종료되었습니다. ");
18     }
19 }
```

number를 2로 나누어서
0이면 짝수이다.



LAB: 두수 중 큰 수 출력하기

- 사용자로부터 두 개의 정수를 입력 받아서 둘 중에 큰 수를 출력하는 프로그램을 작성하여 보자.

```
첫번째 정수: 10  
두번째 정수: 20  
큰 수는 20
```

직접 입력
하여 확인

Larger.java

```
01 import java.util.Scanner;
02
03 public class Larger {
04     public static void main(String[] args) {
05         int x, y, max;
06
07         Scanner input = new Scanner(System.in);
08         System.out.print("첫번째 정수: ");
09         x = input.nextInt();
10
11         System.out.print("두번째 정수: ");
12         y = input.nextInt();
13
14         if (x > y)
15             max = x;
16         else
17             max = y;
18
19         System.out.println("큰 수는 " + max);
20     }
21 }
```

조건에 따라 실행하는
문장이 하나이면 중괄호를
생략할 수 있다.



예제: 성과급 계산

- 하나의 예로 사원이 실적 목표 1000만원을 달성하였을 경우에는 실적 목표를 초과한 금액의 10%를 성과급으로 받는 프로그램을 작성하여 보자.

실적을 입력하시오(단위: 만원): 1200

실적 달성

보너스: 20



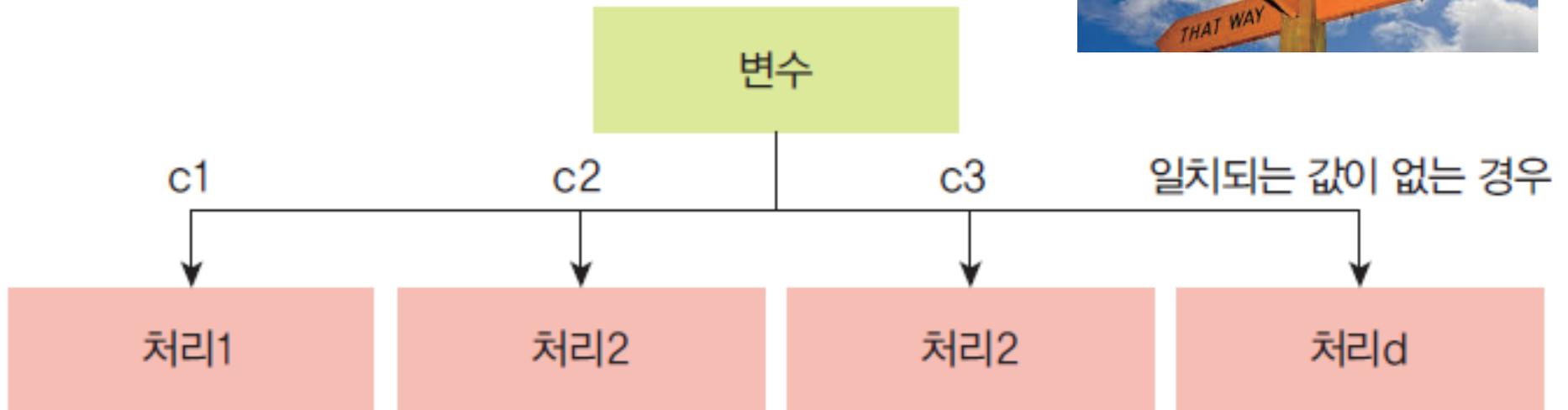
```
01 import java.util.Scanner;
02
03 public class Bonus {
04     public static void main(String args[]) {
05         final int targetSales = 1000;
06         int mySales;
07         int bonus;
08         String result;
09
10         Scanner input = new Scanner(System.in);
11         System.out.print("실적을 입력하시오(단위: 만원): ");
12         mySales = input.nextInt();
13
14         if (mySales >= targetSales) {
15             result = "실적 달성";
16             bonus = (mySales - targetSales) / 10;
17         } else {
18             result = "실적 달성 못함";
19             bonus = 0;
20         }
21
22         System.out.println(result + "\n" + "보너스: " + bonus);
23     }
24 }
```

실적 목표가 상수로
정의되었다.



switch 문

- 만약 가능한 실행 경로가 여러 개인 경우에는 switch 문을 사용





switch 선택 구조

전체적인 구조



형식

```
switch(변수)
{
    case c1:
        처리문장1;
        break;
    case c2:
        처리문장2;
        break;
    ...
    default:
        처리문장d;
        break;
}
```

변수와 일치하는 값을 가진 case 절이 실행됩니다.





예제: 정수-> 한글

- 사용자가 값을 입력하면 화면에 “하나”, “둘” 과 같이 출력하는 코드를 작성하여 보자.

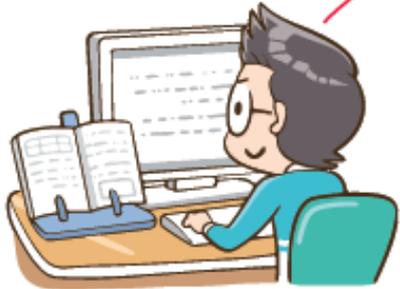
숫자를 입력하시오: 1

하나



예제: 정수-> 한글

사용자가 number에
1을 입력한다.



```
switch(number)
{
    case 0:
        System.out.println("없음");
        break;
    case 1:
        System.out.println("하나");
        break;
    case 2:
        System.out.println("둘");
        break;
    default:
        System.out.println("많음");
        break;
}
```

값이 0이면
이쪽으로 오세요!

값이 1이면
이쪽으로 오세요!

값이 2이면
이쪽으로 오세요!

일치하는 값이 없으면
이쪽으로 오세요!



만약 break 문이 없다면



```
switch(number)
{
    case 0:
        System.out.println("없음");
        break;
    case 1:
        System.out.println("하나");
    case 2:
        System.out.println("둘");
        break;
    default:
        System.out.println("많음");
        break;
}
```

case 1에 break문이 없으므로 case 2의 문장들이 이어서 실행됩니다.





LAB: 월의 일수 출력하기

Mars						
sun	mon	tue	wed	thu	fri	sat
			1	2	3	4
5	6	7	8	9	10	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

일수를 알고 싶은 월을 입력하시오: 12

월의 날수는 31

일수를 알고 싶은 월을 입력하시오: 6

월의 날수는 30



```
01 import java.util.*;
02
03 public class DaysInMonth {
04     public static void main(String[] args) {
05         int month;
06         int days = 0;
07
08         System.out.print("일수를 알고 싶은 월을 입력하시오:");
09         Scanner scan = new Scanner(System.in);
10
11         month = scan.nextInt();
12         switch (month) {
13             case 4:
14             case 6:
15             case 9:
16             case 11:
17                 days = 30;
18                 break;
19             case 2:
20                 days = 28;
21                 break;
22             default:
23                 days = 31;
24                 break;
25         }
26         System.out.println("월의 날수는 " + days);
27     }
28 }
```

고의적으로 break문을 생략
하여서 여러 가지 경우를 동일한
문장으로 처리하고 있다.



Q & A

