

# 제3장 – 제어문

## Outline

- 3.1 if문(1)
- 3.2 if문(2)
- 3.3 for문
- 3.4 while문
- 3.5 루프문의 중단
- 3.6 switch문



## 3.1 if문(1)

- If문이란?

예 >>

```
class Even {  
    public static void main (String[] args) {  
        int a = 5 ;  
  
        if (a % 2 == 0 )  
            System.out.println(a + "는 우수입니다." ) ;  
        else  
            System.out.println(a + "는 기수입니다." ) ;  
    }  
}
```

실행결과 >>

5는 기수입니다.



## 3.1 if문(1)

### • If문이란?

```
class Score {
    public static void main (String [] args ) {
        int s = 65 ;

        System.out.println("당신의 점수는 "+s+"점입니다.") ;

        if( s<70 )
        {
            System.out.println("평균까지 앞으로 "+(70-s) +"점");
            System.out.println("힘내세요!");
        }
        else
        {
            System.out.println("잘 하셨습니다!") ;
        }
    }
}
```

<< 예

블록

블록은 필요 없지만,  
있어도 상관없습니다.

당신의 점수는 65점입니다.  
평균까지 앞으로 5점  
힘내세요 !

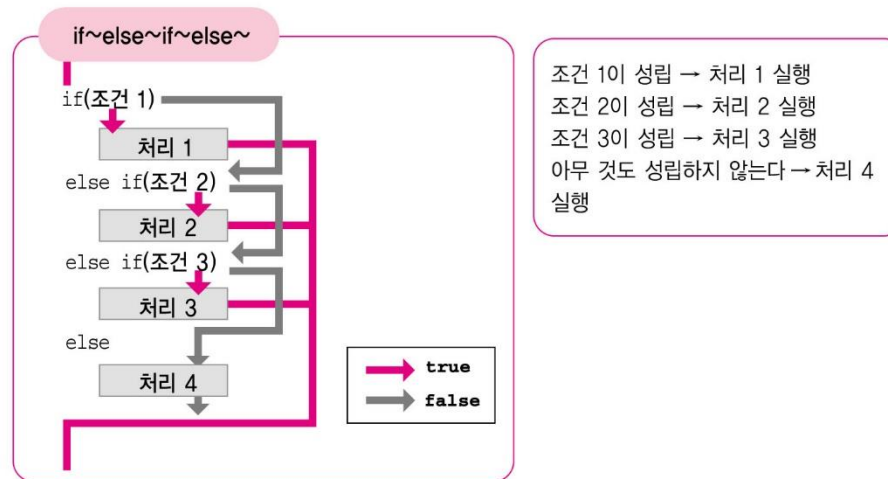


<< 실행결과



## 3.2 if문(2)

- 연속된 if문



## 3.2 if문(2)

- 연속된 if문

예 >>

```
class Rank {  
    public static void main(String [] args) {  
        int num = 1000 ;  
        System.out.print(num+"은");  
  
        if( 0 <= num && num <= 9 )  
            System.out.println("1자리 수입니다." ) ;  
        else if ( 10 <= num && num <=99 )  
            System.out.println("2자리 수입니다." ) ;  
        else if ( 100 <= num && num <=999 )  
            System.out.println("3자리 수입니다." ) ;  
        else  
            System.out.println("4자리 수입니다." ) ;  
    }  
}
```

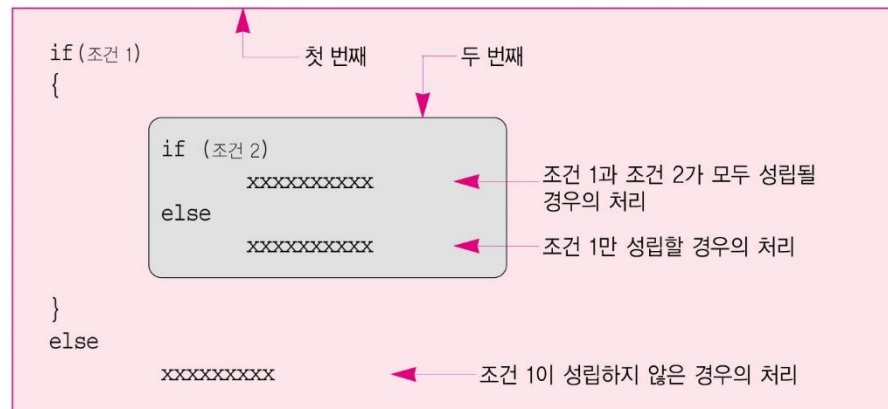
1000은 4자리 수입니다. ◀ 어떤 조건에도 맞지 않기 때문에 else 이하를 실행합니다.

<< 실행결과



## 3.2 if문(2)

- 중첩된 if문



## 3.2 if문(2)

- 중첩된 if문

```
class Score {  
    public static void main(String[] args) {  
        int a = 90 ;  
  
        if(a > 80) {  
            if(a == 100 )  
                System.out.println("만점입니다.") ;  
            else  
                System.out.println("조금 더 노력하세요.") ;  
        }  
        else  
            System.out.println("힘내세요!") ;  
    }  
}
```

◀ 예

조금만 더 노력하세요.

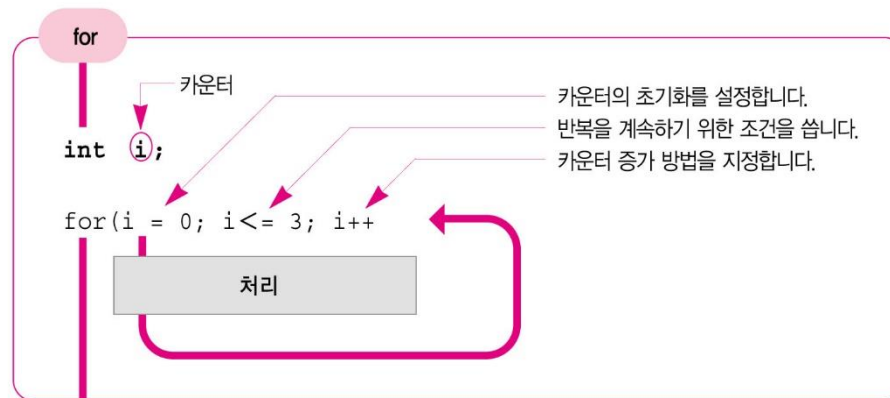
■

◀ 실행결과



## 3.3 for문

- For문이란?



i의 초기값을 0으로 하고, 하나씩 값을 증가해 가면서 3 이하일 때 처리를 반복하여 실행합니다.



## 3.3 for문

### • For문이란?

예 >>

```
class Counter {
    public static void main (String[] args) {
        int i ;
        for(i = 1 ; i < 4 ; i++ )
            System.out.println("안녕하세요."+i) ;
    }
}
```

변수 i에 1을 대입  
 "안녕하세요. 1"을 표시  
 i++을 실행(i = 2)  
 i < 4 이므로 반복  
 "안녕하세요. 2"를 표시  
 i++을 실행 (i = 3)  
 i < 4 이므로 반복  
 "안녕하세요. 3"을 표시  
 i++을 실행(i = 4)  
 i < 4 가 아니므로 루프를 종료

처리  
순서

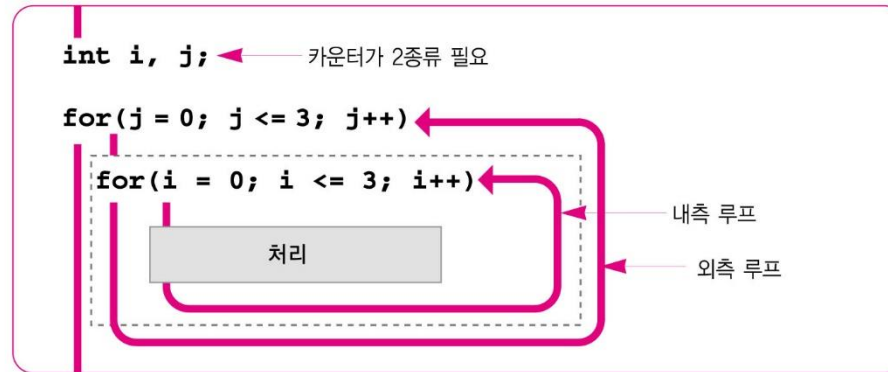
실행결과 >>

안녕하세요.1  
 안녕하세요.2  
 안녕하세요.3  
 ■



## 3.3 for문

- for문이란?



2중 루프에서는 값의 변화가 다음과 같습니다.

j	i	i-j	처리 순서
1	1	0	↓
	2	1	
	3	2	
2	1	-1	
	2	0	
	3	1	

## 3.3 for문

- for문이란?

```
class Multiply {  
    public static void main (String [] args) {  
        int i, j ;  
  
        for(j = 1; j <= 9; j++)  
            for(i = 1; i <= 9; i++)  
                System.out.println(j + "x" + i + " = " + j * i ) ;  
    }  
}
```

&lt;&lt; 예

```
1 x 1 = 1  
1 x 2 = 2  
:  
9 x 8 = 72  
9 x 9 = 81  
■
```

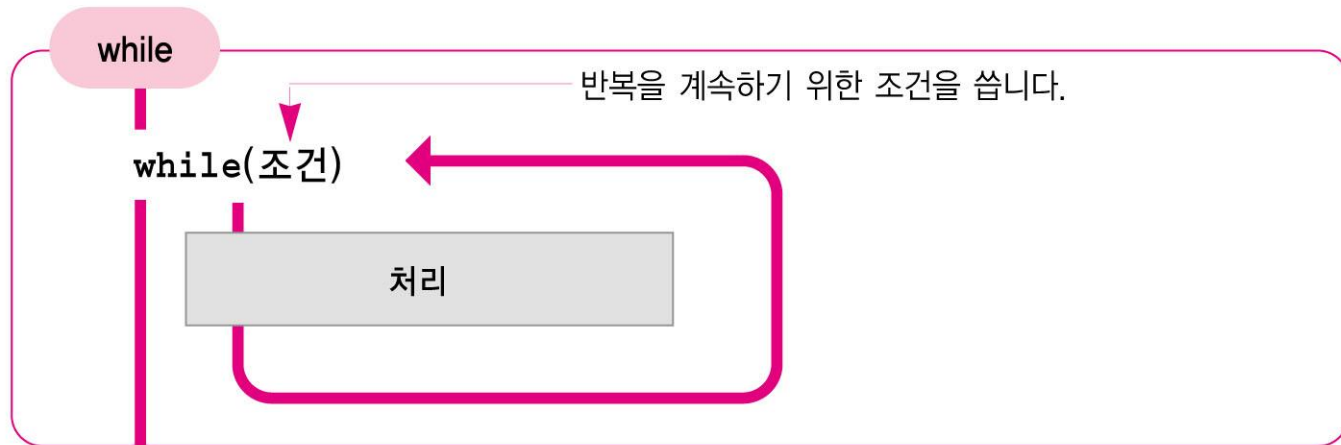
← 구구단을 모두 표시합니다.

&lt;&lt; 실행결과



## 3.4 while문

- while문이란?



조건이 성립하는 한 처리를 반복합니다.

## 3.4 while문

- do ~ while문



조건이 성립하는 한, 처리를 계속 반복합니다(반드시 한번은 실행합니다).

## 3.4 while문

- do ~ while문

```
class Sum {  
    public static void main (String[] args) {  
        int s = 0, i = 1 ;  
  
        do {  
            i++;  
            s = s + i ;  
        } while( i < 10) ;  
        System.out.println("1에서 "+ i +"까지의 합은 "+ s);  
    }  
}
```

&lt;&lt; 예

1에서 10까지의 합은 55



&lt;&lt; 실행결과



## 3.4 while문

- 무한 루프에 빠지지 않도록



```
int a = 0;
```

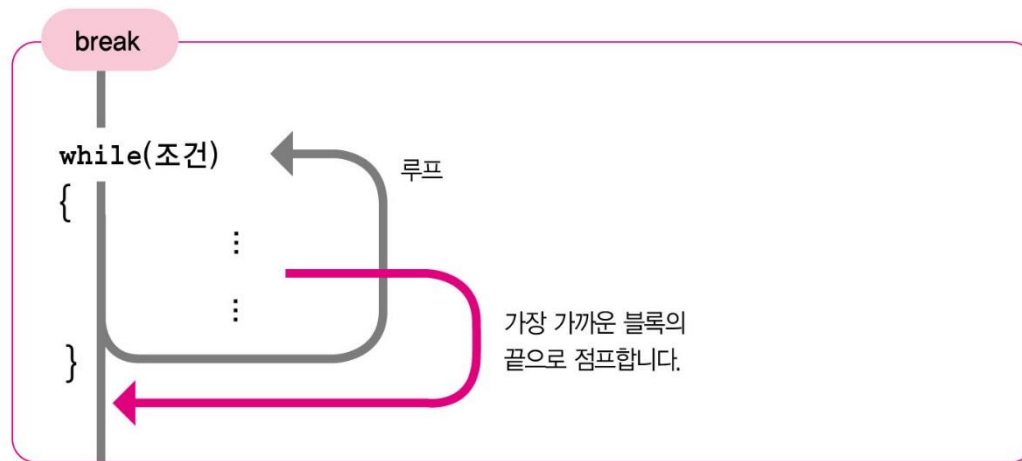
```
while(a < 5) {  
    System.out.println(a) ;  
}
```



← a = a + 1; 같이 a를 증가시키는 부분이 없습니다. 이렇게 하면 a값이 변하지 않기 때문에 무한 루프가 되어 버립니다.

## 3.5 루프의 중단

- 반복을 중단하려면 **break**로





## 3.5 루프의 중단

- 반복을 중단하려면 **break**로

예 >>

```
class Break {  
    public static void main(String[] args) {  
        int a, b = 2 ;  
  
        for(a = 0 ; a < 5 ; a++) {  
            if(b - a <= 0 )  
                break ;  
            System.out.println(b + "-" + a + "=" + (b-a)) ;  
        }  
    }  
}
```

b-a의 값이 0이하가 되면  
루프를 종료합니다.

실행결과 >>

0+1=1

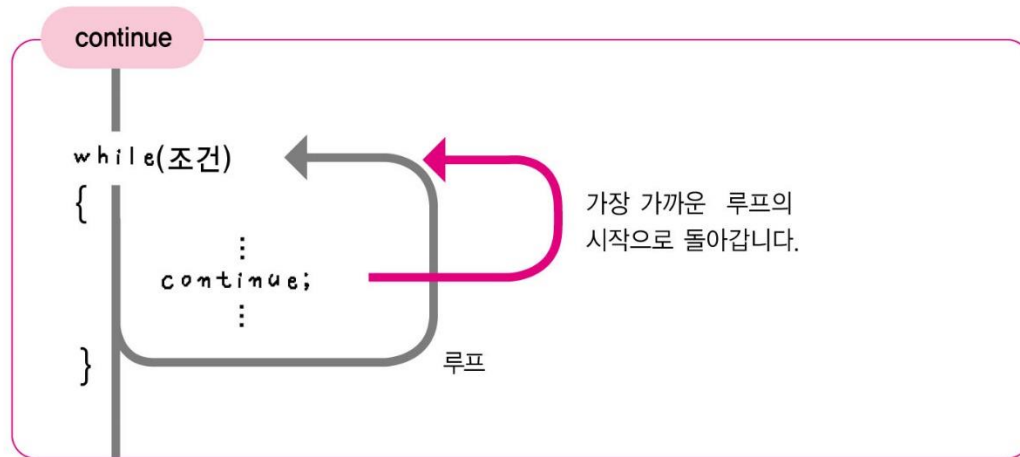
1+1=2

■ 2-2는 0이 되기 때문에 루프를 종료합니다.



## 3.5 루프의 중단

- 반복을 다음 회로 넘기는 `continue`



## 3.5 루프의 중단

- 반복을 다음 회로 넘기는 `continue`

```
class Continue {
    public static void main(String [] args) {
        int a, b = 1 ;
        for(a = 0 ; a < 4 ; a++) {
            if(a + b == 2)
                continue ;
            System.out.println(a + "+" + b + "=" + (a+b)) ;
        }
    }
}
```

a+b=2일 때는 루프의 시작으로 돌아갑니다.

<< 예

```
0+1=1
2+1=3
3+1=4
■
```

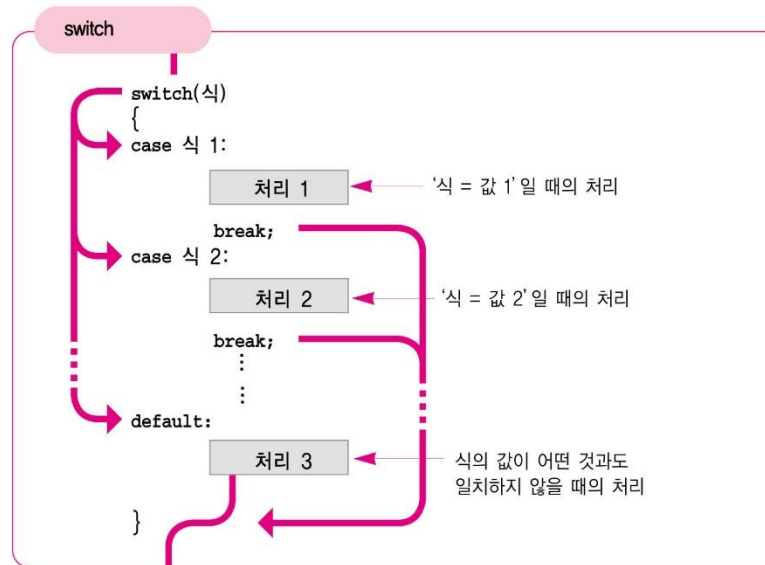
1+1은 2가 되기 때문에 표시하지 않고 다음으로 넘어갑니다.

<< 실행결과



## 3.6 switch문


- 여러 개의 처리를 선택



식의 값에 따라 다른 처리를 선택하여 실행합니다.

## 3.6 switch문


- 여러 개의 처리를 선택



```
String a = "Hello" ;

switch(a) {
    case "Hello" :
        System.out.println(a) ;
        break ;
    :
}

```



```
String a = "Hello" ;

if(a.equals("Hello")) {
    System.out.println(a) ;
}
else if(...
    :

```

**equals( ) 메소드**  
문자열을 비교하기 위한  
메소드입니다.

## 3.6 switch문

- 여러 개의 처리를 선택

```
class Select {  
    public static void main(String [] args) {  
        for(int a = 5; a <= 8; a++) {  
            System.out.println(a + " ÷ 3 = " + a/3);  
            switch(a%3) {  
                case 1:  
                    System.out.println(" :나머지는 1입니다.");  
                    break ;  
                case 2:  
                    System.out.println(" :나머지는 2입니다.");  
                    break ;  
                default :  
                    System.out.println(" :나머지는 0입니다.");  
                    break ;  
            }  
        }  
    }  
}
```

&lt;&lt; 예

5 ÷ 3 = 1 :나머지는 2입니다.  
6 ÷ 3 = 2 :나머지는 0입니다.  
7 ÷ 3 = 2 :나머지는 1입니다.  
8 ÷ 3 = 2 :나머지는 2입니다.

■

&lt;&lt; 실행결과

