

어서와 *Java*는 처음이지!

제3장 선택과 반복

자바에서는 조건을 따져서
문장을 수행시키나봐요.

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

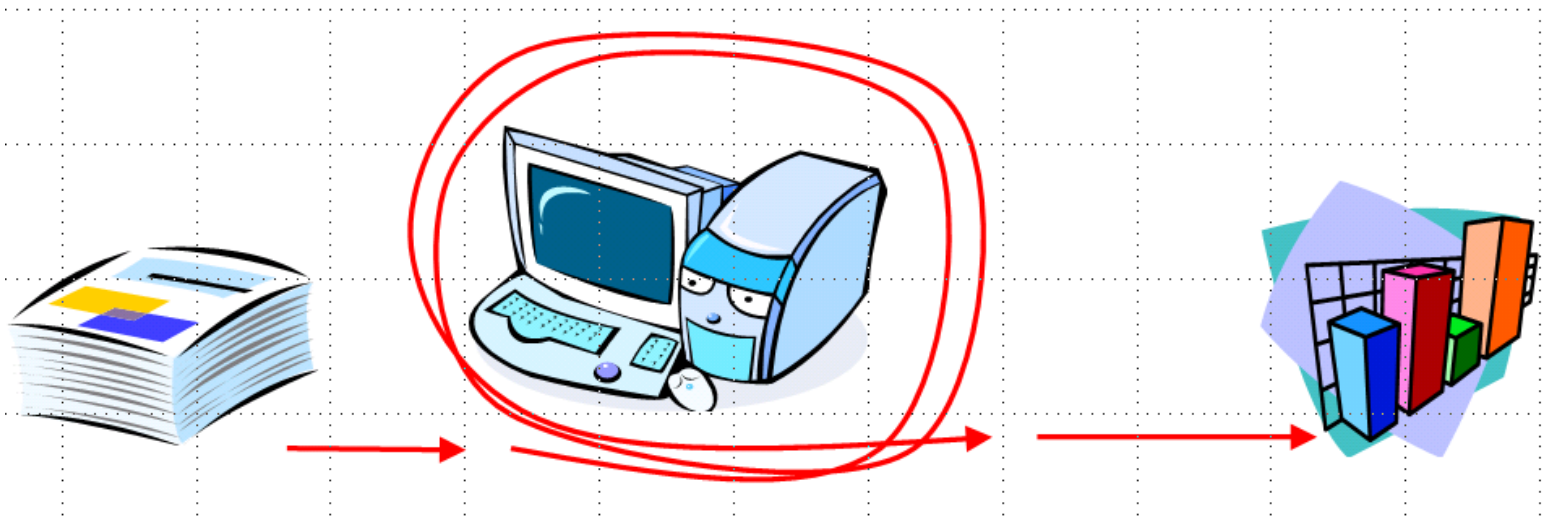




반복문

Q) 반복 구조는 왜 필요한가?

- A) 같은 처리 과정을 되풀이하는 것이 필요하기 때문이다.
- B) 학생 30명의 평균 성적을 구하려면 같은 과정을 30번 반복하여야 한다.

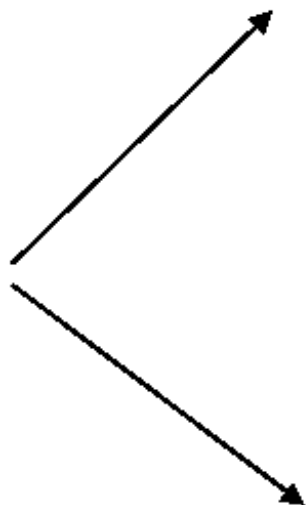




반복문의 종류



반복문



while



for

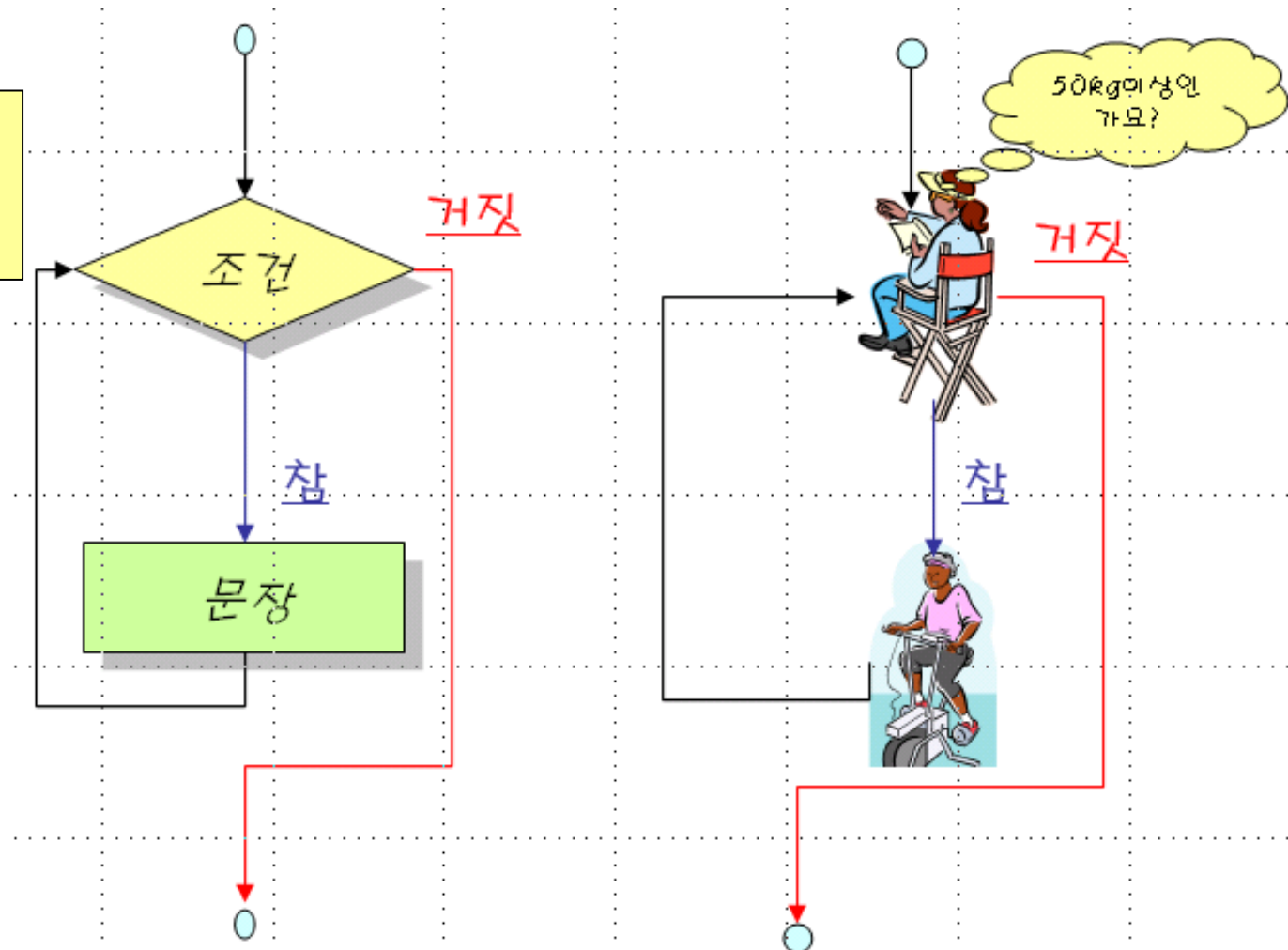




while 문

- 주어진 조건이 만족되는 동안 문장들을 반복 실행한다

while(조건식)
문장;





while 문의 구조

형식

`while` (조건식)

{

문장1;

문장2;

...

}

반복을 하는 조건이다. 조건이 참이면 반복을 계속한다.

반복되는 문장이다.

조건이 참이면
반복하는 구조입니다.





예제: 0부터 9까지 출력하기

0 1 2 3 4 5 6 7 8 9

LoopExmaple1.java

```
01 public class LoopExample1 {
02     public static void main(String[] args) {
03         int i = 0;
04         while (i < 10) {
05             System.out.print(i+" ");
06             i++;
07         }
08     }
09 }
```



예제: 구구단 출력

구구단 중에서 출력하고 싶은 단을 입력하시오: 9

$$9 * 1 = 9$$

$$9 * 2 = 18$$

$$9 * 3 = 27$$

...

LoopExample2.java

```
01 // while 문을 이용한 구구단 출력 프로그램
02 import java.util.*;
03
04 public class LoopExample2 {
05     public static void main(String[] args) {
06         int n;
07         int i = 1;
08         System.out.print("구구단 중에서 출력하고 싶은 단을 입력하시오: ");
09         Scanner scan = new Scanner(System.in);
10         n = scan.nextInt();
11         while (i <= 9) {
12             System.out.println(n + "*" + i + "=" + n * i);
13             i++;
14         }
15     }
16 }
```

여기서는 먼저 사용자로부터 출력하고 싶은 구구단의 단수를 받아서 변수 n에 저장한다. 여기서의 루프 제어 변수는 i이다. i의 초기값이 0이 아니고 1인 것에 유의하라. 구구단은 1부터 곱해야 하기 때문에 0이 아니고 1로 초기화를 하였다. 그리고 반복 루프도 9보다 작거나 같을 때까지 반복하도록 하였다.



예제: $(1+2+3+\dots+9+10)$ 계산하기



합계=55



CalSum.java

```
01 public class CalSum {  
02     public static void main(String[] args) {  
03  
04         int i = 1;  
05         int sum = 0;  
06         while (i <= 10) {  
07             sum = sum + i;  
08             i++;  
09         }  
10         System.out.println("합계=" + sum);  
11  
12     }  
13 }
```

sum은 0으로 초기화되어야 한다.

i가 10이하이면 i의 값을
sum에 더한다.

i는 1부터 시작한다.



do-while 문



반복 조건을 가장
아래에서 검사합니다.





do-while 문의 예

DoWhile1 .java

```
01 public class DoWhile1 {  
02     public static void main(String[] args){  
03         int i = 10;  
04         do {  
05             System.out.println("i의 값: " + i);  
06             i++;  
07         } while (i < 3);  
08     }  
09 }
```

i의 값: 10



예제: 정확한 입력받기

올바른 월을 입력하시오 [1-12]: 13

올바른 월을 입력하시오 [1-12]: 14

올바른 월을 입력하시오 [1-12]: 0

올바른 월을 입력하시오 [1-12]: 1

사용자가 입력한 월은 1



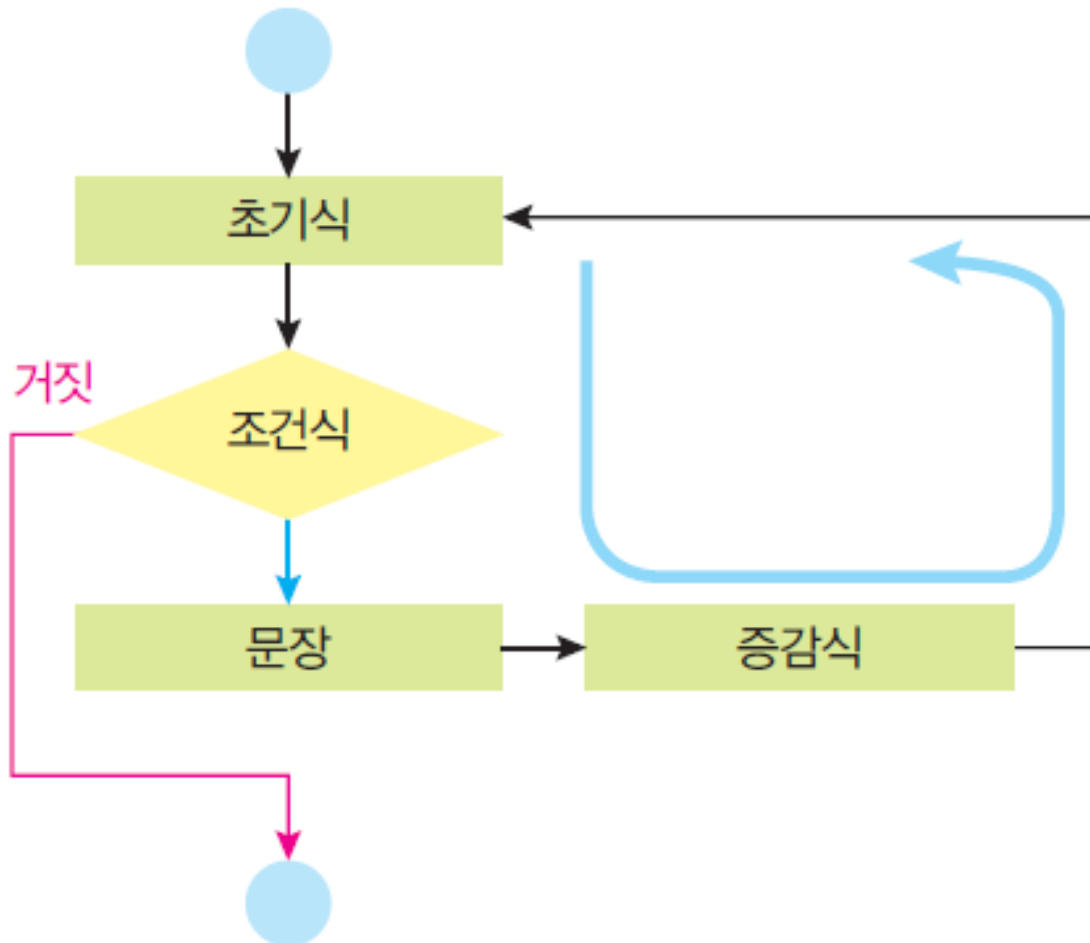
CheckInput.java

```
01 import java.util.Scanner;
02
03 public class CheckInput {
04     public static void main(String args[]) {
05         Scanner input = new Scanner(System.in);
06         int month;
07         do {
08             System.out.print("올바른 월을 입력하시오 [1-12]: ");
09             month = input.nextInt();
10
11         } while (month < 1 || month > 12);
12
13     System.out.println("사용자가 입력한 월은 " + month);
14 }
```



for 루프

- 정해진 횟수만큼 반복하는 구조





for 문의 구조

형식

```
for(초기식; 조건식; 증감식) {  
    반복문장;  
}
```

형식

```
for(i=0; i<5; i++) {  
    System.out.print("Hello");  
}
```



예제: 0부터 4까지 출력하기

ForExample1

```
01 public class ForExample1 {  
02     public static void main(String[] args) {  
03         for (int i = 0; i < 5; i++) {  
04             System.out.println("i의 값은: " + i);  
05         }  
06     }  
07 }
```

for 문 안에서 제어 변수를 선언할 수 있다.

```
i의 값은: 0  
i의 값은: 1  
i의 값은: 2  
i의 값은: 3  
i의 값은: 4
```



예제: 정수의 합 계산하기

1부터 10까지의 정수의 합 = 55

Sum.java

```
01 public class Sum {  
02     public static void main(String[] args) {  
03         int sum = 0;  
04  
05         for (int i = 1; i <= 10; i++)  
06             sum += i;  
07  
08         System.out.printf("1부터 10까지의 정수의 합 = %d", sum);  
09  
10     }  
11 }
```

for 문 안의 초기식에서
제어 변수를 선언할 수 있다.



중첩 반복문

형식

```
for(i=0; i<5; i++) {  
    for(k=0; k<5; k++) {  
        반복문장;  
    }  
}
```

외부 반복문

내부 반복문



예제 : 사각형 모양 출력하기

```
*****  
*****  
*****  
*****  
*****
```



NestedLoop.java

```
01 import java.util.*;
02
03 public class NestedLoop {
04     public static void main(String[] args) {
05
```

```
06         for (int y = 0; y < 5; y++) {
07             for (int x = 0; x < 10; x++)
08                 System.out.print("*");
09
10             System.out.println("");
11         }
12
13     }
14 }
```

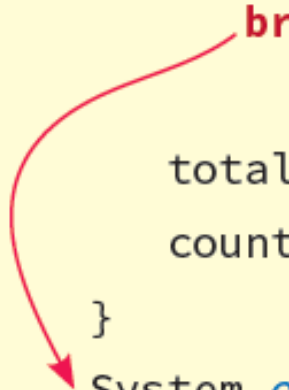
위의 프로그램을 실행하면 50개의 *가 화면에 5X10의 정사각형 모양으로 출력된다. *를 출력하는 문장의 외부에는 두개의 for 루프가 중첩되어 있다. 외부의 for 루프는 변수 y를 0에서 4까지 증가시키면서 내부의 for 루프를 실행시킨다. 내부의 for 루프는 변수 x를 0에서 9까지 증가시키면서 print() 메소드를 호출한다. 내부 for 루프가 한번 실행될 때마다 화면에는 한 줄의 *가 그려진다. 내부 for 루프가 한 번씩 종료될 때마다 줄바꿈 문자가 화면에 출력되어 다음 줄로 넘어가게 된다.



Break 문

- break;
- 하나의 반복문을 빠져나온다.

```
01 import java.util.*;
02
03 public class Averager {
04     public static void main(String[] args) {
05         int total = 0;
06         int count = 0;
07         Scanner scan = new Scanner(System.in);
08         while (true) {
09             System.out.print("점수를 입력하시오: ");
10             int grade = scan.nextInt();
11             if (grade < 0)
12                 break;
13
14             total += grade;
15             count++;
16         }
17         System.out.println("평균은 " + total / count);
18     }
19 }
```



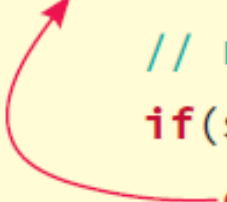


continue 문

- continue;
- 반복문에서 해당 문장 아래의 문장을 체크하지 않고 반복문 조건식 체크로 되돌아간다.

ContinueTest.java

```
01 public class ContinueTest {
02     public static void main(String[] args) {
03
04         String s = "no news is good news";
05         int n = 0;
06
07         for (int i = 0; i < s.length(); i++) {
08             // n 이 나오는 회수를 센다.
09             if(s.charAt(i) != 'n')
10                 continue;
11
12             // n의 개수를 하나 증가한다.
13             n++;
14         }
15         System.out.println("문장에서 발견된 n의 개수 " + n);
16     }
17 }
```





LAB: 숫자 추측 게임



Guess the same number that the computer has guessed.
The number will range from 1 to 100.

정답을 추측하여 보시오: 10

제시한 정수가 낮습니다.

정답을 추측하여 보시오: 30

제시한 정수가 낮습니다.

정답을 추측하여 보시오: 60

제시한 정수가 높습니다.

정답을 추측하여 보시오: 59

축하합니다. 시도횟수=4



NumberGame.java

```

01 import java.util.Scanner;
02
03 public class NumberGame {
04     public static void main(String args[]) {
05         int answer = 59;    // 정답
06         int guess;
07         Scanner sc=new Scanner(System.in);
08         int tries = 0;
09         // 반복 구조
10         do {
11             System.out.print("정답을 추측하여 보시오: ");
12             guess = sc.nextInt();
13             tries++;
14
15             if (guess > answer) // 사용자가 입력한 정수가 정답보다 높으면
16                 System.out.println("제시한 정수가 높습니다.");
17             if (guess < answer) // 사용자가 입력한 정수가 정답보다 낮으면
18                 System.out.println("제시한 정수가 낮습니다.");
19         } while (guess != answer);
20
21         System.out.println("축하합니다. 시도횟수=" + tries);
22     }
23 }

```

if 문을 사용하여 guess가 answer보다 작은지 큰지를 검사하여 적당한 메시지를 출력한다. do...while 루프의 마지막 조건 검사 부분에서 guess가 answer와 같은지를 검사한다. 만약 guess가 answer와 같으면 반복을 중단한다.



Q & A

