

제어문(반복문)

# 반복문이란?

- 반복문

지정한 횟수 또는 조건에 따라 반복하여 실행하며 개발자에 따라 코드의 가독성이 좋아지거나, 코드 중복을 줄일 수 있음

# for문

- for문

초기식, 조건식, 증감식으로 구성되며 반복할 횟수를 알고 있는 경우에 사용함.  
대부분의 경우 for문으로 사용됨

- for문

```
for(초기식; 조건식; 증감식) {  
    반복 실행할 코드  
}
```

- 특정 조건이 만족할 때 까지 코드를 반복 수행

- 예시

```
for(int i=0; i<10; i++) {  
    System.out.println("현재 숫자는 "+i+" 입니다.");  
}
```

# for문 활용하기 - 1

## • 설명

스캐너를 통해 입력 받은 값 만큼 반복

```
Scanner sc = new Scanner(System.in);  
System.out.println("반복할 횟수를 입력하세요 : ");  
int num = sc.nextInt();  
  
for(int i=0; i<num; i++) {  
    System.out.println("현재 반복 횟수는 "+i+" 입니다.");  
}
```

반복할 횟수를 입력하세요 :

6

현재 반복 횟수는 0 입니다.

현재 반복 횟수는 1 입니다.

현재 반복 횟수는 2 입니다.

현재 반복 횟수는 3 입니다.

현재 반복 횟수는 4 입니다.

현재 반복 횟수는 5 입니다.

# for문 활용하기 - 2

## • 설명

문자열의 길이만큼 반복 할 때는 입력 받은 변수.length()를 사용하여 문자열 길이만큼 반복 횟수를 지정함

```
Scanner sc = new Scanner(System.in);
System.out.println("문자열의 길이만큼 반복할때는? : ");
String str = sc.nextLine();

for(int i=0; i<str.length(); i++) {
    System.out.println("현재 반복 횟수는 "+i+" 입니다.");
}
```

문자열의 길이만큼 반복할때는? :  
abcd  
현재 반복 횟수는 0 입니다.  
현재 반복 횟수는 1 입니다.  
현재 반복 횟수는 2 입니다.  
현재 반복 횟수는 3 입니다.

## for문 활용하기 - 3

- Q1. 숫자 1부터, 사용자가 입력한 값만큼 숫자를 출력하는 프로그램을 작성하시오.

1. 스캐너로 사용자로부터 숫자를 입력 받으세요.
2. 만약, 사용자가 숫자 10을 입력 한다면, 1부터 10까지의 숫자를 출력하면 됩니다.

## for문 활용하기 - 4

- Q2. 짝수를 출력하는 프로그램을 작성하시오.

1. 범위는 1부터 20까지 입니다.
2. 스캐너로 입력 받지 않아도 되며, 1~20 사이의 짝수를 출력하면 됩니다.

## for문 활용하기 - 4

- Q3. 로또 번호를 생성하는 프로그램을 작성 하시오.

1. 6개의 숫자를 출력 하세요.
2. 아래의 코드를 참고하여 메시지를 출력하시면 됩니다.

// 최상단에 작성

```
import java.util.Random;
```

```
Random random = new Random();
```

// for문안에 작성, 1부터 45까지의 범위에서 랜덤한 숫자를 생성해주는 부분

```
int lottoNumber = random.nextInt(45) + 1;
```



# 중첩 for문

## • 설명

1. 첫번째 for문이 실행 :  $i = 0$
2. 안에 있는 for문이 실행 :  $j = 0$ 부터 9까지 반복
3. 첫번째 for문이 실행 :  $i = 1$
4. 안에 있는 for문이 실행 :  $j = 0$ 부터 9까지 반복
5. 그 외 같은 방식으로 반복

```
for(int i=0; i<3; i++) {  
    System.out.println("첫번째 for문 i : "+i);  
    for(int j=0; j<10; j++) {  
        System.out.println("j : "+j);  
    }  
}
```

첫번째 for문 i : 0

j : 0  
j : 1  
j : 2  
j : 3  
j : 4  
j : 5  
j : 6  
j : 7  
j : 8  
j : 9

첫번째 for문 i : 1

j : 0  
j : 1  
j : 2  
j : 3  
j : 4  
j : 5  
j : 6  
j : 7  
j : 8  
j : 9

첫번째 for문 i : 2

j : 0  
j : 1  
j : 2  
j : 3  
j : 4  
j : 5  
j : 6  
j : 7  
j : 8  
j : 9

## for문 활용하기 - 5

- Q4. 중첩 for문을 사용하여 구구단 프로그램을 작성하시오.

1. 중첩 for문을 사용하여 2단부터 9단까지의 구구단 내용을 출력하세요.

# while문

- while문

조건식으로 구성되며 반복할 횟수를 알고 있는 경우에 사용하며 증감식 또는 분기문은 생략 가능하지만, 무한루프가 발생할 가능성이 매우 높으므로 사용에 주의가 필요함

- while문

```
while(조건식) {  
    반복 실행할 코드  
    증감식 또는 분기문  
}
```

- 조건식이 거짓(false)이 될 때 까지 무한 반복

- 예시

```
int cnt = 0;  
while(cnt < 10) {  
    System.out.println("현재 반복 횟수 : "+cnt);  
    cnt++;  
}
```

# while문 활용하기

## • 설명

입력 받은 숫자가 0이 될 때 까지 반복하여 입력 받음

ex)

1 → 조건에 만족하지 않으니 반복 실행

2 → 조건에 만족하지 않으니 반복 실행

0 → 조건에 만족하여 종료

```
Scanner scanner = new Scanner(System.in);  
int input = scanner.nextInt();  
  
while (input != 0) {  
    // 입력값을 처리하는 코드  
    input = scanner.nextInt();  
}
```



1  
2  
0

# do-while문

- do-while문

do 안의 내용은 조건식과 상관 없이 최초 1번은 무조건 실행됨.  
즉, 참(true), 거짓(false)을 판단하기 전에 무조건 1번 이상 실행 해야할 코드가 있을 경우에 사용함.

- while문

```
do {  
    반복 실행할 코드  
    증감식 또는 분기문  
} while(조건식);
```

- 최초 1번은 무조건 실행
- 그 이후 조건식이 거짓(false)이 될 때 까지 무한 반복

- 예시

```
do {  
    System.out.println("조건식과 상관없이 무조건 1번 이상 실행됩니다.");  
} while(false);
```

# do-while문 활용하기

## • 설명

최초 1회는 조건식을 만족하지 않으나, do로 인해 실행되며 그 이후는 조건식에 만족할 때 까지 반복

ex)

12 → 조건에 만족하지 않으니 반복 실행

13 → 조건에 만족하지 않으니 반복 실행

6 → 조건에 만족하여 종료

```
Scanner scanner = new Scanner(System.in);  
int input;  
  
do {  
    System.out.print("1부터 10까지의 숫자를 입력하세요: ");  
    input = scanner.nextInt();  
} while (input < 1 || input > 10);  
  
System.out.println("입력한 숫자는 유효합니다: " + input);
```

1부터 10까지의 숫자를 입력하세요: 12  
1부터 10까지의 숫자를 입력하세요: 13  
1부터 10까지의 숫자를 입력하세요: 6  
입력한 숫자는 유효합니다: 6