

For 문의 구성 및 개요, 설명

```
public class ForTest {
    public static void main(String[] args) {

        // for문의 구성
        // for(초기화 코드; 조건식 코드; 증감식 코드)로 구성됩니다.
        // 초기화 코드는 for문에 최초 진입시 한 번만 실행합니다.
        // 조건식 코드는 for문 내부(중괄호 파트)로 진입하기 위한 조건입니다.
        // 증감식 코드는 for문 내부의 동작이 완료된 이후 동작할 코드입니다.
        // 이후 다시 조건식 코드로 가서 조건을 비교하고 충족한다면 위 루틴을 반복하며
        // 만약 조건식 코드에서 조건이 위배된다면 for문이 종료됩니다.

        // for문을 작성하는 방법
        // 1. for를 적고 소괄호를 열고 닫고 중괄호를 열고 닫습니다.
        // 2. 초기화 코드 위치에 초기화가 필요하다면 초기화 코드를 작성한다.
        // 3. 조건식 코드 위치에 어떤 조건에서 동작시킬지 동작 조건을 작성한다.
        // 4. 증감식 코드 위치에 증가시키거나 감소시킬 값에 대한 코드를 작성한다.
        // 5. 중괄호 내부에 조건이 만족되었을 경우 동작시킬 코드를 작성한다.

        // 작거나 같다(<=)에서 주의점: 작아도 참, 같아도 참, 크면 거짓
        // 작다(<)에서 주의점: 작으면 참, 같으면 거짓, 커도 거짓
        for(int i = 1; i <= 10; i++) { //1씩 증가
            for(int i = 1; i <= 10; i+=2) { //2씩 증가
                System.out.println("i = " + i);
            }
        }
    }
}
```

```
public class InfinityLoopWithForTest {
    public static void main(String[] args) {

        // while(true)와 다르게 for의 경우엔 무조건이라는 식을 만들 수 있다.
        // 조건부에 아무런 조건이 없다는 것은 ??? 무조건

        /*
        for(;;) {
            System.out.println("나도 무한 루프야");
        }
        */
    }
}
```

For문을 활용한
예제 (무한루프)

```
// 아래 케이스는 단순히 i에 대해 무한히 1씩 더하는 케이스다.
// for(int i = 1; ; i++) {

// 이 코드는 무한하게 i에 대해서는 1씩 더하고 j에 대해서는 1씩 뺀다.
```

- For문을 이용한 평균값 구하기

```
public class AverageWithForTest {  
    public static void main(String[] args) {  
        int sum = 0;  
  
        for (int i = 1; i <= 10; i++) {  
            sum += i; // sum = sum + i  
            System.out.println("sum = " + sum);  
        }  
        System.out.println("최종 합산 값 = " + sum);  
  
        // 주식과 관련된 작업을 한다면 이 부분을 상당히 조심해야 한다.  
        float average = sum / 10;  
        System.out.println("평균 = " + average);  
  
        // 차이가 나는 이유가 뭘까?  
        // 위의 케이스는 sum이 int형, 숫자 10도 디폴트로 int형입니다.  
        // (기본적으로 정수는 int, 소수점은 double을 채택함)  
  
        // sum은 int형이지만 나누는 숫자가 소수점(float)이기 때문에  
        // 강제로 소수점 연산이 수행되어 5.5라는 결과를 얻게 됨  
        average = sum / 10.0f;  
        System.out.println("평균 = " + average);  
    }  
}
```

- 여기서 중요한점은 int 타입은 정수만을 사용하기 때문에 뒤에 붙는 소수점은 제거를 해버림.
- float 타입은 소수점을 다루기 때문에 뒤에 붙는 나머지 소수점들을 보여줌.

3. Math.random()

```
public class RandomTest {  
    public static void main(String[] args) {  
  
        for (;;) {  
            // Math.random()은 [0 ~ 1)에 해당하는 소수점 데이터를 출력합니다.  
            // 0.0 이상 1.0 미만 사이의 소수점  
            // 여기에 10을 곱하기 때문에  
            // 0.0 이상 10.0 미만 (9.9xxxxx 이하)  
            // 소수점이지만 값을 int로 처리하면 소수점은 버려짐  
            // 0 ~ 9 까지의 정수가 출력됨  
            System.out.println((int)(Math.random() * 10));  
        }  
    }  
}
```

- Math.random()은 위와같이 랜덤으로 0.0 ~ 1.0 미만 사이의 소수점을 출력한다.
- 여기서 int타입 변수와 10을 곱하게 되면 정수로 출력이 됨

4. For문 연습문제

```
System.out.println("11번 문제 입니다.");
// 11번
// 1 ~ 10까지 출력
for (int i = 1; i < 11; i++) {
    // %3d는 먼저 %d가 정수형 숫자를 출력하는 역할을 수행함을 상기한다.
    // 여기서 3의 의미는 3칸을 확보하라는 뜻이다.
    // 숫자 10의 경우 2칸을 차지하고
    // 붙어 있으면 10인지 구별이 안되니까 공백 1칸을 추가로 확보한다.
    // 그러므로 910으로 표현되지 않고 9 10 으로 깔끔하게 표현된다.
    // --- 3칸 확보
    System.out.printf("%3d", i);

    // i 값이 5로 나뉘서 떨어지면 엔터 적용
    if (i % 5 == 0) {
        // 아무것도 출력 안하고 엔터만 적용함(println())
        System.out.println();
    }
}

// 12번
// 1 ~ 20에서 3의 배수를 출력
for (int i = 3; i < 21; i += 3) {
    System.out.println("i = " + i);
}

System.out.println("13번 문제 입니다.");

for (int i = 1; i <= 30; i++) {
    if(i % 2 == 0){
        System.out.println(i + " = 짝수 입니다.");
    }else if(i % 2 == 1){
        System.out.println(i + " = 홀수 입니다");
    }
}
```

5. While 타입과 Math.random()을 사용한 주사위 예제 feat. (Thread.sleep)

```
public class DiceTest {  
    public static void main(String[] args) throws InterruptedException {  
        System.out.println("주사위 눈금을 굴려봅시다!");  
  
        while (true){  
            // [0.0 ~ 1.0)  
            // 0.0 이상 1.0 미만  
            // 0.0 이상 6.0 미만 (5.9xxx 이하)  
            // 1.0 이상 7.0 미만 (6.9xxx 이하)  
            // 1 ~ 6 까지 주사위 눈금 숫자만큼  
  
            // 랜덤에도 종류가 있습니다.  
            // 가우시안 랜덤(정규 분포), 푸아송 분포, 이항 분포, 기하 분포 등등  
            // Math.random() 이라는 랜덤은 Uniform Random(모든 항목이 동일한 확률을 가짐)  
            System.out.println((int)(Math.random() * 6 + 1));  
            // 1초 = 1000 밀리세컨드 = 1000000 마이크로세컨드 = 1000000000 나노세컨드  
            // 1초 = 10^3 ms = 10^6 us = 10^9 ns  
            // 결국 0.5초 대기하라는 의미  
            // 그래서 결과가 0.5 초마다 출력되는 것임  
            Thread.sleep( millis: 500);  
        }  
    }  
}
```

- 여기서 중요한 포인트는 Math.random()의 문법을 입력 시켰을 경우 0.0 ~ 0.1 미만의 수를 랜덤으로 출력하게 하고 그 범위를 6을 곱하고 1을 더해서 0.1 부터 7.0미만의 숫자를 뽑게 한다.
- 그리고 여기서 Thread.sleep이란 문법 또한 들어가게 되는데 이것은 (괄호)안에 들어간 숫자 만큼 잠깐 sleep 쉬라고 하는것 같다. 위의 예시와 같이 500 밀리 세컨드 다시 말해 0.5초 동안 텀을 주게 된다.