

A decorative graphic featuring two diagonal wooden sticks, one on the left and one on the right, meeting at a central horizontal box. The sticks are light brown with visible grain and are secured to the box with small yellow circular fasteners. The central box is white with a thin black border and rounded corners.

반복문

반복문

- while

```
class WhileBasic {  
    public static void main(String[] args) {  
        int num = 0;  
        while(num < 5) {  
            System.out.println("I like Java " + num);  
            num++;  
        }  
    }  
}
```

- while 문 안에 위치한 두 문장이 총 5회 실행
- 조건 검사 후 true이면 while 문 안의 코드 실행

- while의 기본 문법

```
while(반복조건) {  
    반복 조건이 true 시 실행되는 영역  
}
```

반복문

- for

```
class ForBasic {  
    public static void main(String[] args) {  
        for(int i = 0; i < 5 ; i++) {  
            System.out.println("I love Java " + i);  
        }  
    }  
}
```

- for 문 안에 위치한 두 문장이 총 5회 실행
- 조건 검사 후 true이면 for 문 안의 코드 실행

- for의 기본 문법

```
for(반복기준; 반복조건; 반복기준 증가/감소) {  
    반복 조건이 true 시 실행되는 영역  
}
```

반복문

- break : 가장 근접한 반복문을 찾아서 해당 반복문을 종료하고 빠져나감

```
class BreakBasic {  
    public static void main(String[] args) {  
        int num = 1;  
        boolean search = false;  
        while(num < 100) {  
            if(num % 5 == 0 && num % 7 == 0) {  
                search = true;  
                break;  
            }  
            num++;  
        }  
        if(search) {  
            System.out.println("찾는 정수 : " + num);  
        } else {  
            System.out.println("5의 배수이자 7의 배수인 수를 찾지 못하였습니다.");  
        }  
    }  
}
```

반복문

- continue : continue가 실행되는 시점에서 아래에 있는 코드를 실행하지 않고 조건 검사 부분으로 이동

```
class ContinueBasic {  
    public static void main(String[] args) {  
        int num = 0;  
        int count = 0;  
  
        while(num < 100) {  
            if (num % 5 != 0) {  
                num++;  
                continue;  
            }  
            System.out.println(num);  
            count++;  
            num++;  
        }  
        System.out.println("count : " + count);  
    }  
}
```

반복문

- 중첩 반복문

```
class ForInFor {  
    public static void main(String[] args) {  
        for(int i = 0; i < 3; i++) {  
            System.out.println("-----");  
            for (int j = 0; j < 3; j++) {  
                System.out.print "[" + i + ", " + j + " ] ");  
            }  
        }  
    }  
}
```

반복문

1. 1부터 99까지의 합을 구하는 프로그램을 작성하되 while문을 이용하여 작성하시오.
2. 1부터 99까지의 합을 구하는 프로그램을 작성하되 for문을 이용하여 작성하시오.
3. while문을 이용하여 1부터 100까지 출력하고 이어서 거꾸로 100에서 1까지 출력하는 프로그램을 작성하시오.
4. for문을 이용하여 1부터 100까지 출력하고 이어서 거꾸로 100에서 1까지 출력하는 프로그램을 작성하시오.
5. 1000 이하 자연수 중에서 2의 배수이고 7의 배수인 수를 출력하고, 그 수들의 합을 구해서 출력하는 프로그램을 while문을 이용하여 작성하시오.
6. 1000 이하 자연수 중에서 2의 배수이고 7의 배수인 수를 출력하고, 그 수들의 합을 구해서 출력하는 프로그램을 for문을 이용하여 작성하시오.
7. 1부터 10까지의 곱의 결과를 출력하는 프로그램을 while문을 이용해서 작성하시오.
8. 1부터 10까지의 곱의 결과를 출력하는 프로그램을 for문을 이용해서 작성하시오.
9. 구구단 중 5단을 출력하는 프로그램을 while문을 이용하여 작성하시오.
10. 구구단 중 5단을 출력하는 프로그램을 for문을 이용하여 작성하시오.
11. 자연수 1부터 시작하여 모든 홀수를 더해 나간다. 그 합이 자연수 몇에서 1000을 초과하는지, 그리고 1000을 초과한 순간의 값은 얼마가 되는지 계산하여 출력하는 프로그램을 작성하시오.

반복문

12. 구구단을 출력하시오.
13. 구구단의 짝수 단인 2, 4, 6, 8단만 출력하는 프로그램을 작성하되 2단은 2*2까지, 4단은 4*4까지, 6단은 6*6까지, 8단은 8*8까지 출력하도록 작성하시오.
14. 임의의 정수를 입력 받아 입력 받은 정수의 모든 약수를 출력하는 프로그램을 작성하시오.
15. 임의의 정수 3개를 기억하고 정수의 순서와 숫자를 맞추는 야구 게임을 만드시오. 자리와 숫자가 같으면 스트라이크, 정답 숫자만 있다면 볼입니다. 단, 정답 입력 기회는 10회로 설정하시오. (예 : 3, 5, 7이 정답이라면 5, 3, 7은 1스트라이크 2볼이며 3, 5, 7입력 시 정답)
16. 두 수를 입력 받아 두 수의 최대 공약수, 최소 공배수를 구하는 프로그램을 작성하시오. 단 유클리드 호제법을 이용하시오.
17. 정수 하나를 입력 받아 피보나치 수열에서 입력 받은 숫자에 해당하는 순서의 숫자를 출력하는 프로그램을 작성하시오. (피보나치 수열 : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,)