

% <- 나머지 구하는 연산자

num % 2 = 0 -> 나머지가 없어서 나누어 떨어진다

num%7 = 0 -> 7로 나누어서 나누어 떨어짐 -> num은 7의 배수

배수

-> 약수

-> 최대공약수

% <- sqrt( )<- 제곱근

Math <- Math.random

ceil(2.6) floor(2.6)

3항 연산자

조건식의 결과가 들어갈 변수를 선언

그 변수 의 조건이 맞는가 ? 참이면 요거 : 거짓이면 요거

1 2 3 <- 리턴값이 작으면 혹은 문법이 어렵지 않으면

```

class SwitchBreak {
    public static void main(String[] args) {
        int n = 3;

        switch(n) {
            case 1:
                System.out.println("Simple Java");
                break;
            case 2:
                System.out.println("Funny Java");
                break;
            case 3:
                System.out.println("Fantastic Java");
                break;
            default:
                System.out.println("The best programming language");
        }

        System.out.println("Do you like Java?");
        for(String aa : args){
            System.out.println(aa);
        }
    }
}

```

클래스 만들고  
main() 자동화 붙는거지

String배열

ctrl+f11 실행 -> main()

```
int k = 1;
while (k <= 100 ) {
    System.out.println(+1);
    k++;
}
int k1 = 100;
while (k1 >=0 ) {
    System.out.println(+1);
    k1--;
}
```

!((i >= 100) && (j < 1))

**사용자가 그만! 이라고 입력할 때 까지 1을 더  
한 것을 출력해라**

for(int i = 0; i < 100 ; i++)

```
int l = 100 ;
```

```
while(i < 100){
```

```
System.out.println("로그인 되었다 ");
```

```
}
```

```
while(rs.next( )){
```

```
}
```

```
(key : value) pair 짝
```

$$m \% 2 == 0 \ \&\& \ m \% 7 == 0$$

2의 배수 **이고** 7의 배수

2와 7의 최소공배수의 배수

**AND** -> **이고** \*

**OR** -> **이거나** +

**조건** **&&** **조건**

0 / 1 2진법

0 -> false    1 -> true

true && true -> true

true || true -> true

false && true -> false

0 \* 1 = 0

false || true -> true

0 + 1 = 1

false && false -> false

false || false -> false

패리티 비트 검사

0000000001 0000000000 < 이 자료가  
진짜지 변조되었는지

```
int [] a = new int[ 10 ];
```

```
for (int i = 0 ; i <= 11 ; i++){
```

```
    print( a[ i ] );
```

```
}
```

0 ~ 11

**a[ 0 ],**

**a[ 11 ] a[ 10 ]?**



```

class Search99 {
    public static void main(String[] args) {
        for(int i = 0; i < 10; i++) {
            for(int j = 0; j < 10; j++) {
                if((i * 10 + j) + (j * 10 + i) == 99)
                    System.out.println(i + ", " + j);
            }
        }
    }
}

```

**if((i \* 10 + j) + (j \* 10 + i) == 99)**

39 <- 일의 자리 수인 9만 출력 하고 싶다.

-a \* 10 a???

사용자의 나이에서 뒷자리 숫자가 8인 사람에게 선물

$$(i * 10 + j) \% 10 = j$$

$$23 \% 10 = 3$$

사용자의 나이에서 뒷자리 숫자가 8인 사람  
에게 선물

```
int age = ???
```

```
if(age % 10 == 8){  
    print("축하합니다.");  
}
```

```
System.out.println("09. 문제의 의도가 무엇인가?");  
System.out.println("1) 두 자리 ab ba 를 더해서 99가 나오는가?");
```

```
List<int[]> permu_1 = new ArrayList<>(); // 순열 -> 순서 상관 O
```

```
String abToString = "";  
String baToString = "";  
int ab = 0;  
int ba = 0;
```

```
// 순열 뽑기
```

```
for(int tempA = 1; tempA < 10; tempA++) {  
    for(int tempB = 1 ; tempB < 10; tempB++) {  
        abToString = Integer.toString(tempA) + Integer.toString(tempB);  
        baToString = Integer.toString(tempB) + Integer.toString(tempA);
```

```
        ab = Integer.parseInt(abToString);  
        ba = Integer.parseInt(baToString);
```

```
        if(ab + ba == 99) {  
            permu_1.add(new int[]{tempA, tempB});  
        }  
    }  
}
```

// 조합으로 만들기

```
List<int[]> combi_1 = new ArrayList<>(permu_1); // 조합 -> 순서 상관 X
for(int i = 0; i<combi_1.size(); i++) {
    int[] temp = combi_1.get(i);
    for(int j = i+1; j<(combi_1.size()); j++) {
        int[] compare = combi_1.get(j);
        if(temp[0] == compare[1]) {
            combi_1.remove(j);
        }
    }
}
```

```
System.out.println("순열");
printList(permu_1);
System.out.println("조합");
printList(combi_1);
```

```
System.out.println();
System.out.println("2) a+b=9 이면 되는건가");
List<int[]> permu_2 = new ArrayList<>(); // 순열 -> 순서 상관 O
```

// 순열 뽑기

```
for(int tempA = 1; tempA < 10; tempA++) {
```