

# 03강 제어문

SCSC java SIG

## 조건문

The if-else statement  
The Switch statement

## 반복문

The For statement  
The For-each statement  
The While statement

# The **if-else** statement

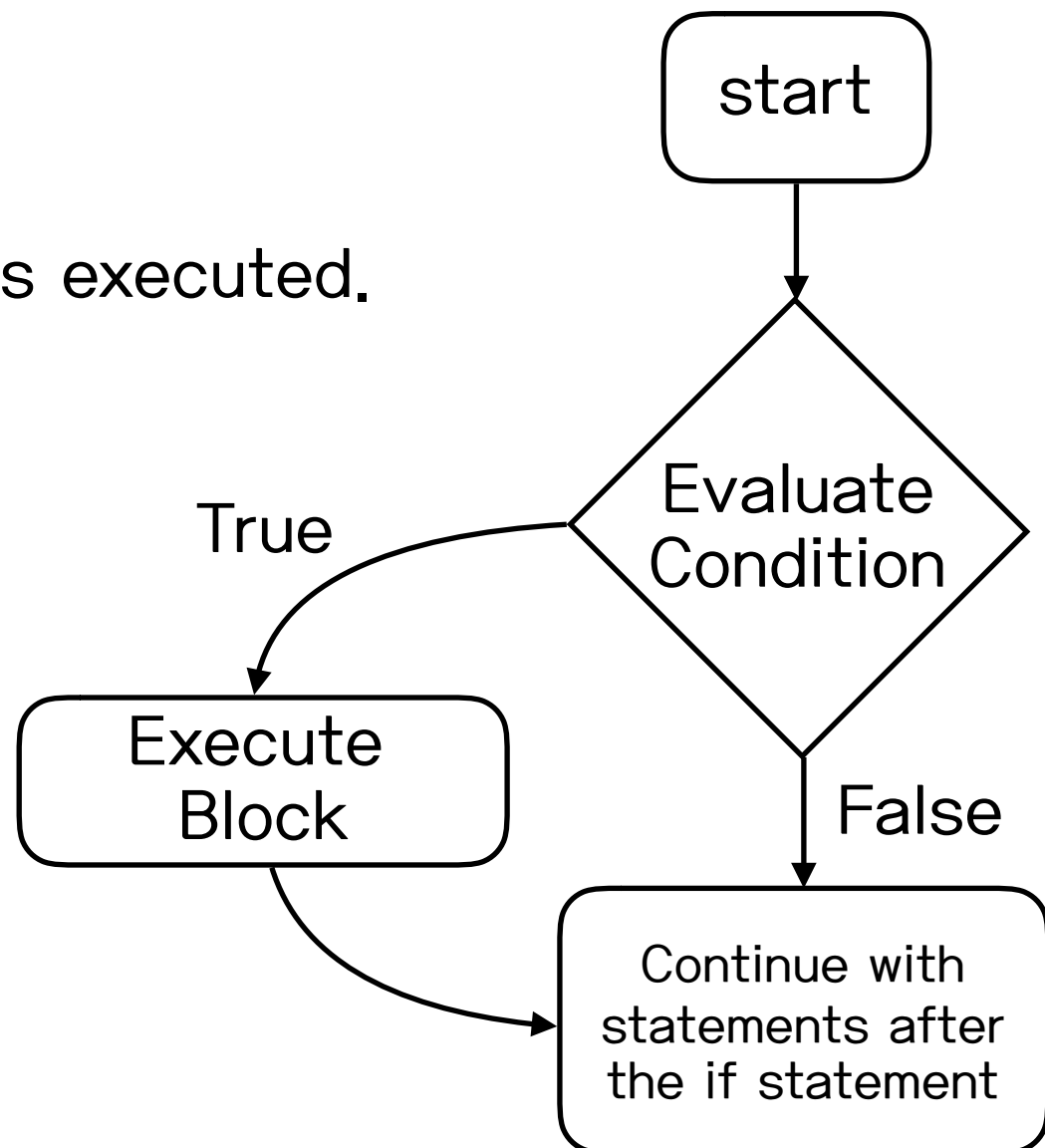
조건의 참/거짓에 따라 서로 다른 명령을 수행하게 만드는 구문

condition : a boolean expression

block : a list of instructions

if the condition is true, then block is executed.

```
...  
if ( condition1 ) {  
    block1  
} else if ( condition2 ) {  
    block2  
} else {  
    default block  
}  
...
```



# The **if-else** statement : Exercise

# 학점 계산 프로그램

if-else문을 이용하여 입력된 성적에 대해 학점을 부여하는 프로그램을 작성합니다.

90점 이상은 A,  
80점 이상 90점 미만은 B,  
80점 미만은 C

# The **if-else** statement : Exercise

## # 학점 계산 프로그램

```
public static void main(String[] args){  
    double score = 85;  
    char grade;  
  
    if(score>=90){  
        grade = 'A';  
    } if else(score>=80){  
        grade = 'B';  
    } else {  
        grade = 'C';  
    }  
    System.out.println("your grade is " + grade);  
}
```

콘솔창

your grade is B

# The **switch** statement

The switch statement is a multi-way branch that makes a decision based on a result of expression

```
...  
switch ( Expression ) {  
    case Label1 :  
        Statements1;  
        break;  
    case Label2 :  
        Statements2;  
        break;  
    default :  
        Statements3;  
}  
...
```

# The **switch** statement

Each case consists of the keyword case followed by

- A constant called the case label
- A colon
- A list of statements

The result of expression and the case label has to be constant

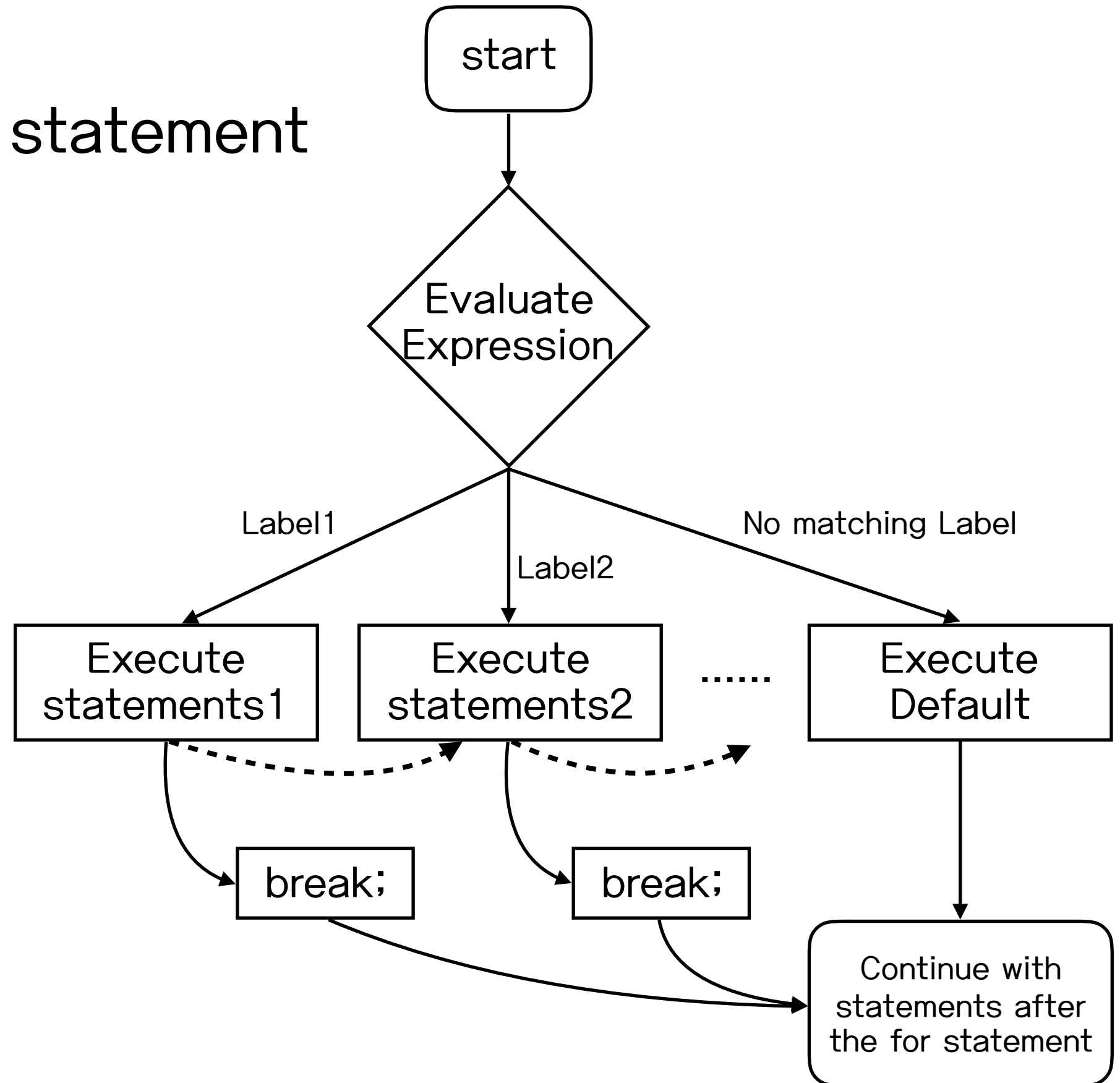
- Integer (int)
- Character (char)
- List of characters (String) (since JDK 1.7)

Repeated case labels are not allowed

If no match is found, the case labeled default is executed

- The default case is optional, but recommended, even if it simply prints a message

# The **switch** statement





# The **switch** statement : Exercise

## # 학점 계산 프로그램 2

switch문을 활용하여 각 학점에 대한 평가를 출력하는 프로그램을 작성합니다.

학점이 A,B 인 학생에게는 “참 잘했습니다.”

학점이 C,D 인 학생에게는 “좀 더 노력하세요.”

학점이 F인 학생에게는 “다음 학기에 다시 수강하세요.”

# The **switch** statement : Exercise

## # 학점 계산 프로그램 2

```
public static void main(String[] args){  
    char grade = 'B';  
  
    switch (grade) {  
        case A :  
        case B :  
            System.out.println("참 잘했습니다.");  
            break;  
        case C:  
        case D:  
            System.out.println("좀 더 노력하세요.");  
            break;  
        case F:  
            System.out.println("다음 학기에 다시 수강하세요.");  
            break;  
        default :  
            System.out.println("잘못 입력하셨습니다.");  
    }  
}
```

콘솔창

좀 더 노력하세요.

# The **while** statement

A while statement repeats its block while a condition remains true

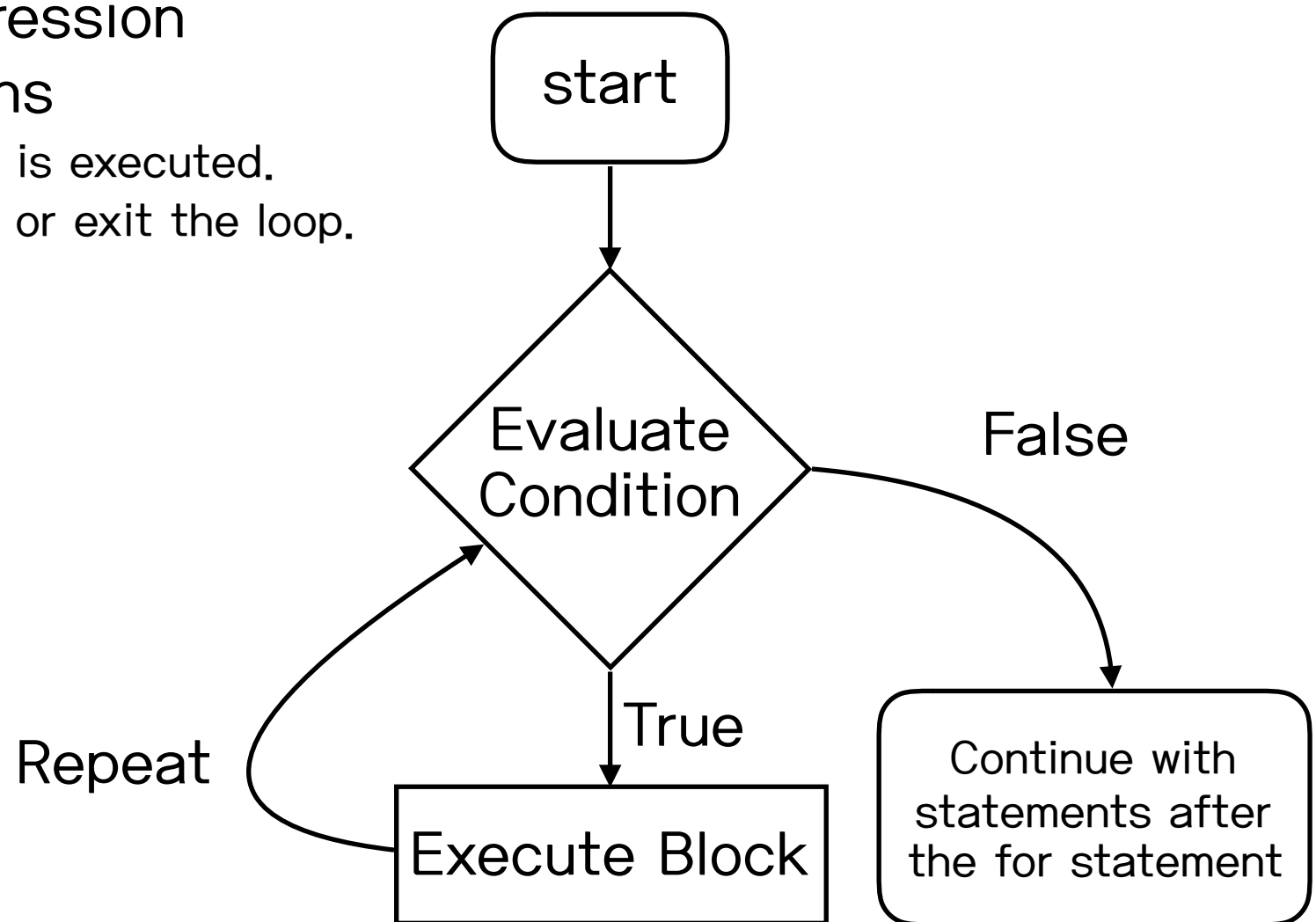
Condition : a boolean expression

Block : a list of instructions

if the condition is true, then block is executed.

if the condition is false, then stop or exit the loop.

```
...  
while ( Condition ) {  
    Block  
}  
...
```



# The **while** statement : Example

0 1 2 3 4 5 6 7 8 9 를 출력하려면?

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

```
    int i=0;
```

```
    while(i<10){
```

```
        System.out.print(i+" ");
```

```
        i++;
```

```
    }
```

```
}
```

# The **for** statement

A for statement executes the body of a loop a fixed number of times

Initialization : The Initialization initializes the loop.

It's executed once, as the loop begins.

Condition : When the Condition evaluates to be false, the loop terminates.

Update : The Update is invoked after each iteration through the loop

It is acceptable for this expression to increase or decrease a value.

Block : a list of instructions.

...

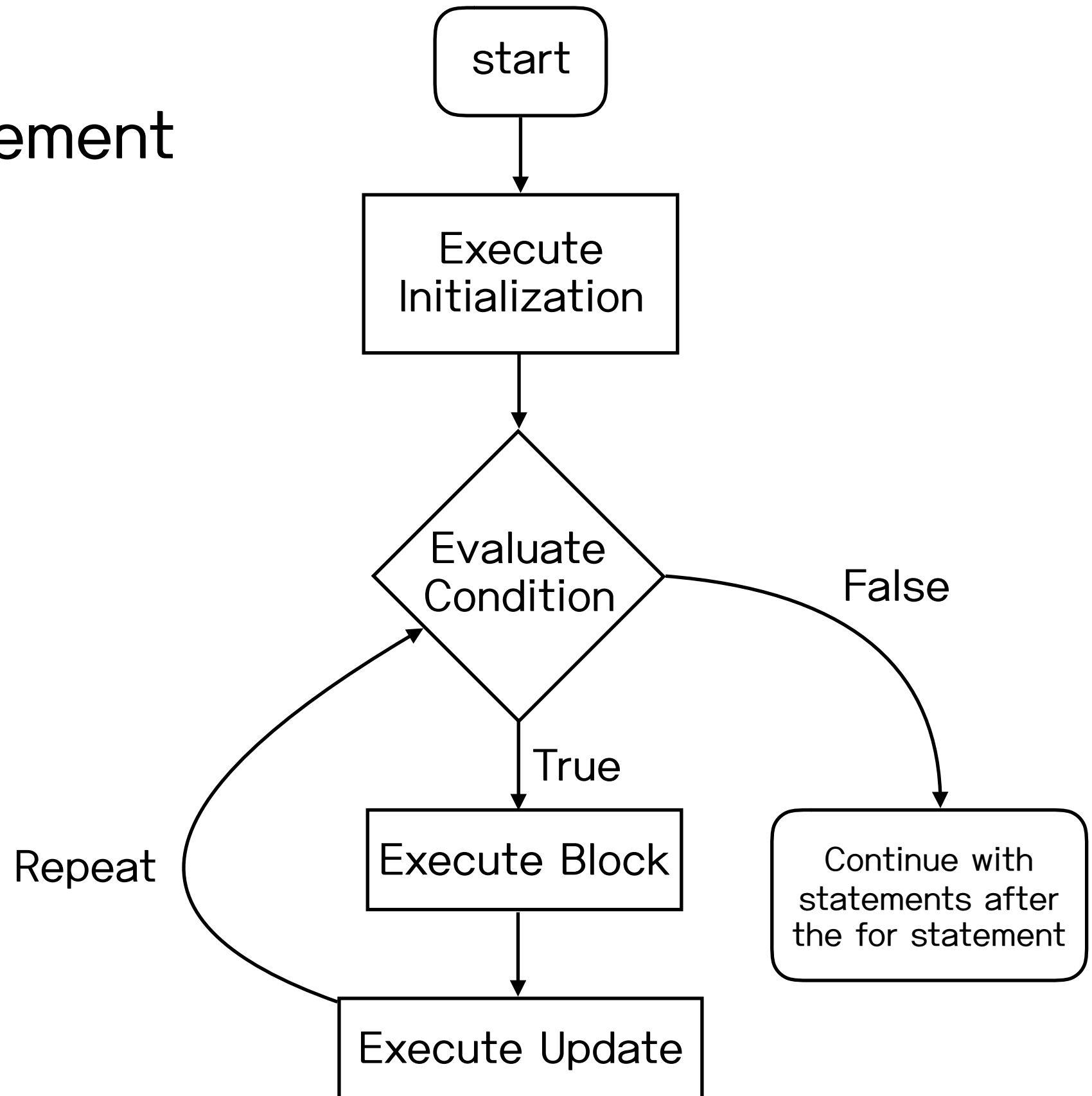
```
for ( Initialization; Condition; Update ){
```

```
    Block
```

```
}
```

...

# The **for** statement



# The **for** statement : Example

0 1 2 3 4 5 6 7 8 9 를 출력하려면?

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

```
    for(int i=0;i<10;i++){  
        System.out.print(i+" ");  
    }  
}
```

# The **for-each** statement

배열과 함께 등장할 예정입니다.



# The **break** statement in Loops

“지금 진행중인 반복문을 즉시 종료하라.”

A break statement can be used to end a loop immediately  
-The break statement ends only the innermost loop or switch statement that contains the break statement

## Example

```
int counter = 0;
while(counter < 10){
    counter++;
    System.out.println("counter : " + counter);
    if(counter > 5){
        break;
    }
}
System.out.println("final value : " + counter);
```

# The **continue** statement in Loops

“지금 진행 중인 Block을 종료하고, 다음 반복을 진행하라.”

- Ends current loop iteration
- Begins the next one

```
...  
while ( Condition ) {  
    .....  
    continue;  
    .....  
}  
...
```

```
...  
for ( Initialization; Condition; Update ){  
    .....  
    continue;  
    .....  
}  
...
```

## Example

```
for (int counter = 0; counter < 10; counter++) {  
    if (counter % 2 == 0) {  
        continue;  
    }  
    System.out.println("counter : " + counter);  
}
```

# DIY

## # 윤년 판정 프로그램

특정 연도를 입력받아 윤년인지 아닌지를 판별하는 함수를 작성하시오.

연수가 4로 나뉘 떨어지는 해는 우선 윤년

그 중에서 100으로 나뉘 떨어지는 해는 평년

다만 400으로 나뉘 떨어지면 다시 윤년

ex) 2000년 == 윤년, 2016년 == 윤년, 2100년 == 평년

//skeleton code

```
public static void main(String[] args){
    System.out.println(isLeapYear(2016));
    System.out.println(isLeapYear(2000));
}

public static boolean isLeapYear(int year){
    boolean answer;
    //method block
    return answer;
}
```

콘솔창

true  
true

# DIY

## # 좌표 판별 method

한 점을 구성하는 (x, y) 좌표를 입력받고,  
이 점이 (50, 40), (50, 80), (100, 40), (100, 80)을 꼭지점으로 갖는 사각형 안에  
있는지 판별하는 메소드를 작성하시오.

//skeleton code

```
public static void main(String[] args){
    System.out.println(isInSquare(60,60));
    System.out.println(isInSquare(40,80));
}

public static boolean isInSquare(int x, int y){
    boolean answer;
    //...
    //method block
    //...
    return answer;
}
```

콘솔창

true  
false