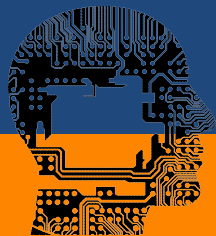




한국IT진흥부설

정보보호교육학원

아이섹



# 자바 프로그래밍 기초

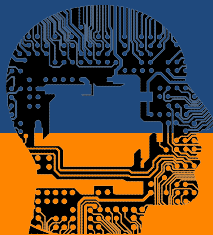
## 메서드

By SoonGu Hong(Kokono)



한국IT진흥부설

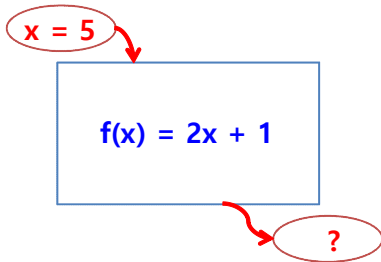
정보보호교육원 아이섹



# 자바 프로그래밍 기초

## 1. 메서드 선언

## 1-1-1. 메서드란?



- 메서드란 **객체의 기능**을 표현하는 코드 블록을 말합니다.
- 수학의 함수처럼 어떤 데이터를 메서드에 전달하면 특정 값을 반환하는 형태로 만들어져 있습니다.

## 1-2-1. 메서드 선언

메서드가 리턴하는 결  
과의 데이터 타입

메서드가 실행할 때 필요한  
데이터를 받기 위한 변수

```
리턴타입 메서드이름 ( [매개변수 선언, ...] ) {  
    실행할 코드를 작성하는 곳  
}
```

## 1-2-2. 메서드 선언 예시

```
static int calcTotal(int x) {  
    int total = 0;  
    for (int n = 1; n <= x; n++) {  
        total += n;  
    }  
    return total;  
} //end calcTotal method
```

- 메서드는 코드들을 기능단위로 사용하기 위한 목적으로 선언합니다. 위 코드는 1부터 x까지의 총합을 구하는 코드를 구성할 목적으로 선언한 예시입니다.
- 메서드는 메서드 안에서는 선언할 수 없습니다.
- 가장 앞에 붙어있는 static은 지금 단계에선 무시하세요!

## 1-3. 메서드 호출

```
public static void main(String[] args) {  
    System.out.println("메서드 테스트 시작!");  
  
    int a = calcTotal(10);  
    System.out.println(a);  
  
    int b = calcTotal(50);  
    System.out.println(b);  
  
    System.out.println(calcTotal(100));  
}  
//end main method
```



```
메서드 테스트 시작!  
55  
1275  
5050
```

- 메서드를 선언했다면 선언한 자체로는 사용이 불가능합니다.
- 메서드는 반드시 **호출(call)**을 통해 기능을 동작시켜야 합니다.
- 호출 시에는 메서드의 이름과 함께 메서드에게 전달할 값을 소괄호 안에 넣어줍니다.

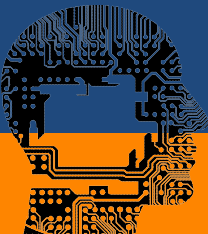


한국IT진흥부설

정보보호교육학원 아이섹

# 자바 프로그래밍 기초

## 2. 매개 변수



## 2-1. 매개변수란?

```
static int calcRangeTotal(int begin, int end) {  
    int total = 0;  
    for(int n=begin; n<=end; n++)  
        total += n;  
    return total;  
}
```

- 매개변수란 메서드 실행을 위해 필요한 데이터를 메서드 내부로 전달할 목적으로 사용하는 변수입니다.

- 모든 메서드가 매개변수가 필요한 것은 아닙니다. 이를테면 사람은 일기를 쓰기 위해 펜이라는 매개체가 필요하지만 일기를 읽기 위해서는 다른 도구가 없어도 되는 것처럼 말이죠



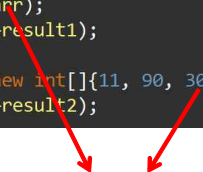
## 2-2-1. 매개변수의 수를 모를 경우1

```
static int calcNumberTotal(int[] numbers) {  
    int total = 0;  
    for (int n : numbers) {  
        total += n;  
    }  
    return total;  
}
```

- 메서드의 매개변수는 메서드를 선언할 때 미리 필요한 만큼의 매개변수를 나열하여 선언합니다.
- 예를 들면 두 수의 합을 구하는 메서드는 정수 2개가 매개 변수로 필요하겠죠?
- 하지만 n개의 합을 구하는 메서드와 같이 매개 변수의 수를 사전에 모를 경우 **배열**을 이용합니다.

## 2-2-2. 매개변수의 수를 모를 경우1 - 호출 예시

```
int[] arr = {10, 20, 30, 10};  
int result1 = calcNumberTotal(arr);  
System.out.println("result1: "+result1);  
  
int result2 = calcNumberTotal(new int[]{11, 90, 30, 30});  
System.out.println("result2: "+result2);
```



```
static int calcNumberTotal(int[] numbers) {  
    int total = 0;  
    for (int n : numbers) {  
        total += n;  
    }  
    return total;  
}
```

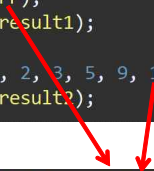
## 2-3-1. 매개변수의 수를 모를 경우2

```
static int calcNumberTotal(int... numbers) {  
    int total = 0;  
    for (int n : numbers) {  
        total += n;  
    }  
    return total;  
}
```

- 또 다른 방법으로는 ... 을 사용한 가변 파라미터 방식을 적용할 수 있습니다.
- int...은 콤마로 나열되어 들어오는 여러 매개값들을 배열로 묶어서 사용할 수 있게 해줍니다.
- 값을 콤마로 나열하여 전달할 수도 있고 배열로 전달할 수도 있습니다.

## 2-3-2. 매개변수의 수를 모를 경우2 - 호출 예시

```
int[] arr = {10, 20, 30, 10};  
int result1 = calcNumberTotal(arr);  
System.out.println("result1: "+result1);  
  
int result2 = calcNumberTotal(1, 2, 3, 5, 9, 11, 19, 99);  
System.out.println("result2: "+result2);
```



```
static int calcNumberTotal(int... numbers) {  
    int total = 0;  
    for (int n : numbers) {  
        total += n;  
    }  
    return total;  
}
```

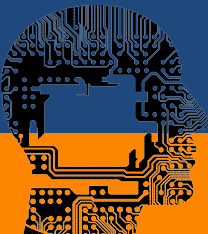


한국IT진흥부설

정보보호교육학원 아이섹

# 자바 프로그래밍 기초

## 3. 리턴문



### 3-1. return

```
static int add(int n1, int n2) {  
    return n1 + n2;  
}
```

- 메서드 선언부에 리턴 타입이 있는 메서드는 반드시 메서드 안에 return이라는 키워드가 있어야 합니다.
- return문의 리턴값은 리턴타입이거나 리턴 타입으로 변환될 수 있어야 합니다.
- 예를들어 리턴타입이 int라면 리턴값은 byte, short, int가 리턴되어도 됩니다. 자동 형변환이 일어나기 때문입니다.

## 3-2. return은 메서드의 탈출문!

```
static int add(int n1, int n2) {  
    return n1 + n2;  
    System.out.println(n1 * n2);  
}
```

Unreachable code  
1 quick fix available:

- 메서드는 return을 만나는 순간 즉시 종료됩니다.
- 따라서 위와 같이 return이후에 코드를 적게되면 Unreachable code라는 메시지와 함께 컴파일 오류가 납니다.

### 3-3. return이 없는 void메서드

```
static void multiply(int n1, int n2) {  
    System.out.printf("%d x %d = %d\n"  
        , n1, n2, n1 * n2);  
}
```

- 메서드에 항상 return문이 필요한 건 아닙니다.
- 위와 같은 코드는 2개의 정수의 곱셈을 출력하고 싶을 뿐이지 곱한 값을 가져오고 프로그램에서 가져오고 싶지는 않기 때문에 리턴을 사용하지 않습니다.
- 리턴값이 존재하지 않는다면 리턴할 데이터도 없으므로 리턴타입이 존재하지 않습니다. 이럴 때는 리턴타입을 비워두면 안되고 **void**라고 명시해야 합니다.



### 3-4. Quiz1

```
public static void main(String[] args) {  
  
    System.out.println("10의 약수의 총합: " + calcDivisor(10));  
    System.out.println("-----");  
    System.out.println("30의 약수의 총합: " + calcDivisor(30));  
}
```

```
10의 약수: 1 2 5 10  
10의 약수의 총합: 18  
-----  
30의 약수: 1 2 3 5 6 10 15 30  
30의 약수의 총합: 72
```

문제) 위와 같은 결과가 나오는 프로그램을 코딩하세요.

- 요구사항

1. 정수 1개를 전달할 시 해당 정수의 모든 약수가 출력되고 약수의 총합을 리턴하는 메서드 `calcDivisor`를 선언하세요.

### 3-5. Quiz2

```
int[] numbers = {57, 89, 78, 91, 93, 47};  
double[] arrTotAvg = calcArrayTotal(numbers);  
System.out.printf("총점: %d점, 평균: %.2f점\n"  
    , (int)arrTotAvg[0], arrTotAvg[1]);
```

총점: 455점, 평균: 75.83점

문제) 위와 같은 결과가 나오는 프로그램을 코딩하세요.

- 요구사항

1. 정수 배열이나 정수 값 목록을 전달할 시 해당 정수의 총합과 평균을 리턴하는 메서드 `calcArrayTotal`을 선언하세요.

**감사합니다**  
**THANK YOU**