





학습목표

- 1. 메소드의 개념과 필요성을 설명할 수 있다.
- 2. 메소드 구조를 만들고 활용할 수 있다.
- 3. 메소드 오버로딩을 이해한다.







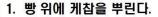














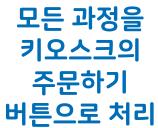




5. 피클, 치즈, 양상추를 얹는다.

6. 빵을 올리면 완성!

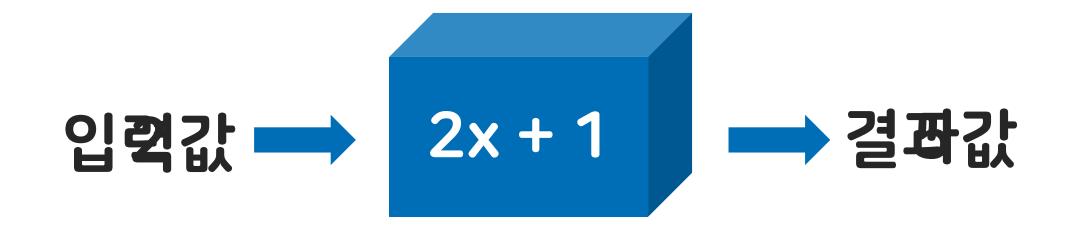
















아이디, 비밀번호

입력값 — 알고리즘

로그인 로직

로그인 성공유무

→ 결과값





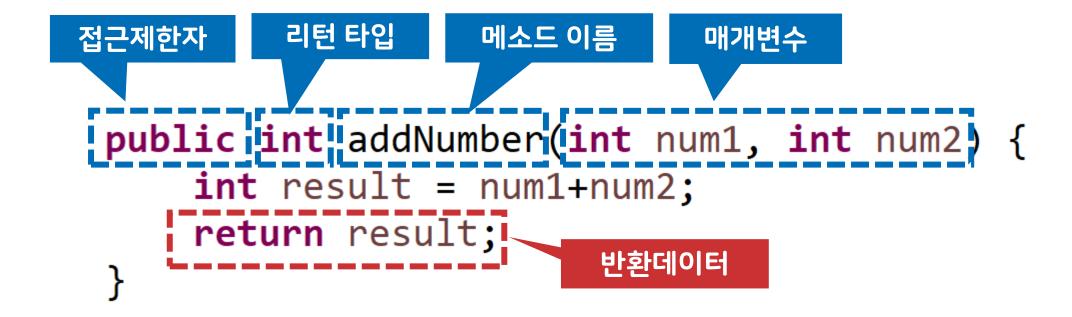
어떤 작업을 수행하기 위한 명령문의 집합 여러 줄의 코드를 하나로 묶어서 표현한 형태

```
System.out.print("아이디 입력 : ");
String id = sc.next();
System.out.print("비밀번호 입력 : ");
String pw = sc.next();
if(id.equals("smhrd")&&pw.equals("12345")) {
    System.out.println("로그인성공!");
}else {
    System.out.println("로그인실패!");
}
```

반복되는 코드 최소화! 유지보수 편함









1. 메소드 예제 - 사칙연산

- 1. 덧셈, 뺄셈, 나눗셈, 곱셈이 가능한 메소드 4개를 생성하세요.
- 2. 아래와 같이 메소드를 사용하고 결과값을 출력하세요.

```
      public static void main

      add(3, 5);
      Tip!

      sub(3, 5);
      메소드 사용하는 코드를 통해 메소드 명, 리턴 타입, 매개변수 유추하기!

      div(3, 5);
      매개변수 유추하기!
```

```
Markers Properticated Name A

8

-2

15
```





2. 메소드 예제 - 계산기

- 1. 정수형 num1과 num2를 입력 받고, 문자형 op를 선언해 원하는 연산자를 넣으세요.
- 2. num1과 num2를 op에 맞게 연산하여 최종 값을 반환해주는 cal메소드를 만드세요.

단, 빼기를 수행할 때는 더 큰 수에서 작은 수를 빼세요!

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("정수1 입력 : ");
    int num1 = sc.nextInt();
    System.out.print("정수2 입력 : ");
    int num2 = sc.nextInt();
    char op = '-';
    System.out.println(cal(num1, num2, op));
}
```

```
Markers ☐ Properties 육 Servers 월 Data Source E <br/>
<terminated> test [Java Application] C:₩Users₩cloud₩.p2₩
정수1 입력 : 50
정수2 입력 : 15
35
```





3. 메소드 예제 - 10에 더 가까운 수 구하기

- 1. 정수형 num1과 num2를 입력 받으세요.
- 2. num1과 num2중 10에 더 가까운 수를 반환하는 close10메소드를 생성하세요.

단, 두 숫자 모두 10과의 차이가 같다면 0을 반환하세요!

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("정수 입력 : ");
    int num1 = sc.nextInt();
    System.out.print("정수 입력 : ");
    int num2 = sc.nextInt();
    int result = close10(num1, num2);
    System.out.println("10에 가까운 수 : " + result);
}
```





4. 메소드 예제 - 완전 수 구하기(level up!)

1. num2가 num1의 약수인지 확인하여 약수라면 true, 아니라면 false를 반환하는 isDivisor 메소드를 만들어주세요.

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("num1 입력 : ");
    int num1 = sc.nextInt();
    System.out.print("num2 입력 : ");
    int num2 = sc.nextInt();
    boolean divisor = isDivisor(num1, num2);
    System.out.println(divisor);
}
```

```
Markers ☐ Properties ા Servers ☐ Data Souterminated > test [Java Application] C:₩Users₩cloud₩ num1 입력 : 10 num2 입력 : 2 true

Markers ☐ Properties 의 Servers ☐ Data Souterminated > test [Java Application] C:₩Users₩cloud₩ num1 입력 : 9 num2 입력 : 2 false
```





4. 메소드 예제 - 완전 수 구하기(level up!)

2. 자신을 제외한 약수의 총합을 구하는 getSum 메소드를 작성하세요.

```
public static void main(String[] args) {
    System.out.println(getSum(7));
}

public static void main(String[] args) {
    System.out.println(getSum(44));
}

**Markers ** Properties **
**Lerminated> test [Java Apple **
**Lerminated>
```





4. 메소드 예제 - 완전 수 구하기(level up!)

3. 입력받은 매개변수가 완전수라면 true, 아니라면 false를 반환하는 isPerfect 메소드를 생성하세요.

```
public static void main(String[] args) {
    System.out.println(isPerfect(7));
}
```

```
public static void main(String[] args) {
    System.out.println(isPerfect(6));
}
```

```
Markers Properties
<terminated> test [Java Appletics]
```

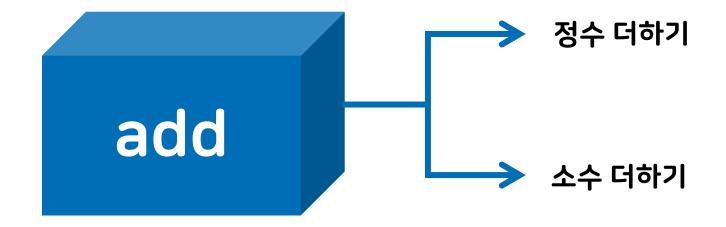




메소드 오버로딩이란?

메소드의 이름은 같지만 매개변수를 다르게 함으로써

서로 다른 메소드를 만드는 기법



매개변수의 개수 및 매개변수의 데이터 타입으로 구분함

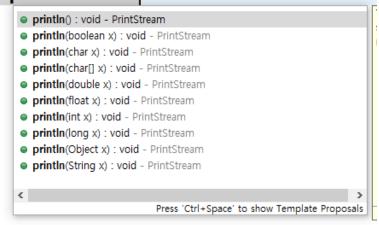




메소드 오버로딩의 조건

- 1. 메소드 이름이 같아야 한다.
- 2. 매개변수의 개수 혹은 타입(자료형)이 달라야 한다.
- 3. 매개변수와 메소드 이름이 동일하고 리턴 타입만 다른 경우는 오버로딩이 아니다.

System.out.println



메소드 이름의 낭비 방지!

개발자가 편해짐!



다음시간에 배울 내용

객체지향프로그래밍(OOP)

