

클래스는 속성을 나타내는 변수와, 동작을 나타내는 메소드로 이루어져 있습니다. 저번에 변수들을 정의했으니까, 이번에는 메소드를 써봅시다.

## 개요

**메소드(Method)**는 클래스에 있는 함수입니다. 자바에서 모든 함수는 클래스에 있기 때문에 사실 자바의 모든 함수는 메소드입니다.

## 기본 문법 (Syntax)

먼저 메소드의 기본 문법부터 봅시다.

```
int sum(int x, int y) {  
    return x + y;  
}
```

**sum**이라는 메소드를 썼는데요. 이 함수는 일단 파라미터로 정수형 **x**와 정수형 **y**를 받습니다. 그리고 이 메소드는 **꼭** 정수형을 리턴시켜줘야 합니다. 그렇지 않으면 오류가 나옵니다! **return x + y;**는 정수형을 리턴시켜주니까 문제 없겠죠?

만약 리턴값이 없으면 자료형 대신 **void**라고 써주면 됩니다. 예를 들어서 예전에 썼던 **main** 함수도 리턴값이 없기 때문에 **void**를 썼었습니다.

```
public static void main(String[] args) {  
    System.out.println("Hello world!");  
}
```

## 예제

'은행 계좌' 예제로 돌아가봅시다. '은행 계좌'에는 '입금' 기능, '출금' 기능, 그리고 '이체' 기능이 필요하겠죠? 각각 틀만 써보겠습니다.

```

public class BankAccount {
    int balance;
    Person owner;

    // 파라미터: 입금할 액수
    // 리턴: 성공 여부 (불린)
    boolean deposit(int amount) {

    }

    // 파라미터: 출금할 액수
    // 리턴: 성공 여부 (불린)
    boolean withdraw(int amount) {

    }

    // 첫 번째 파라미터: 받는 사람 (Person)
    // 두 번째 파라미터: 이체할 금액 (정수)
    // 리턴: 성공 여부 (불린)
    boolean transfer(Person to, int amount) {

    }
}

```

'입금' 기능은 **deposit** 메소드로 썼는데요. 이 메소드는 파라미터로 입금할 액수 **amount** 를 받고, 입금 성공 여부를 나타내는 불린 값을 리턴해줍니다.

'출금' 기능은 **withdraw** 메소드로 썼습니다. **withdraw** 는 파라미터로 출금할 액수 **amount** 를 받고, 성공 여부를 나타내는 불린 값을 리턴해줍니다.

마지막으로 '계좌 이체' 기능은 **transfer** 메소드로 썼습니다. **transfer** 은 파라미터를 두 개 받습니다:

1. 첫 번째는 받을 사람을 나타내는 **Person** 타입의 **to** 입니다.
2. 두 번째는 이체할 금액 **amount** 입니다.

다른 메소드들과 마찬가지로 리턴 값은 성공 여부를 나타내는 불린 값입니다.

지금은 메소드에 아무 내용이 없고, 따라서 리턴 타입을 지키지 못했기 때문에 오류가 나오는데요. 실제로 내용을 채워넣는 것은 인스턴스 강의 후에 하겠습니다.



수업을 완료하셨으면 체크해주세요.



수강생 Q&A 보기



Questions?  
질문하기

```
assignment_id=329&sort_by=popular)
(/questions/new?
```

[이전 강의 메소드 \(/assignments/256\)](/assignments/256)

assignment\_id=329&op1=%E%A%B0%9D%EC%B2%B4+%EC%A7