

JAVA 3주차!

복습 & 실습

지금까지 만들었던 프로그램들을 함수로 나눠보자!

객체지향 - 클래스

자바를 구성하는 가장 기본단위

클래스 (class)?

객체를 정의해 논 것 -> 객체를 찍어낼 수 있는 틀

객체?

프로그램상으로 무언가 역할이 있는 집단 - 코드의 덩어리
(기능의 집단)

Dog.java

```
public class Dog {  
    String name;  
    int age;  
  
    public Dog() {  
  
    }  
  
    public void say() {  
        System.out.println("왈왈");  
    }  
}
```

클래스와 인스턴스

클래스를 통해 만들어낸 객체가 인스턴스

Main.java

```
public class Main() {  
    public static void main(String args[]) {  
        Dog dog = new Dog();  
        dog.say();  
    }  
}
```

이 때 dog(소문자)가 인스턴스이다.

실습

고양이, 인간 객체를 만들어보자

메소드 오버로딩

같은 이름의 메소드를 여러개 만드는 것

```
public void say() {  
    System.out.println("왈왈");  
}  
  
public void say(String name) {  
    System.out.println("My name is " + name);  
}
```

다른점은 괄호 안의 매개변수(parameter) 이다.

Main.java

```
public class Main() {  
    public static void main(String args[]) {  
        Dog dog = new Dog();  
        dog.say();  
        dog.say("간장!");  
    }  
}
```

결과 :

```
왈왈  
My name is 간장!
```

같은 이름의 다른 동작을 하는 여러개의 메서드

생성자

객체를 생성할 때 사용하는 문법

```
public class Dog {  
  
    String name;  
    int age;  
  
    public Dog() {  
  
    }  
  
    public Dog(String a, int b) {  
        this.name = a;  
        this.age = b;  
    }  
}
```

```
public void say() {  
    System.out.println("왈왈");  
}
```

```
public void say(String name) {  
    System.out.println("My name is " + name);  
}
```

```
public void introduce() {  
    System.out.println(  
        "저의 이름은 " + this.name +  
        "이고, 나이는 " + this.age + "살 입니다."  
    );  
}
```

```
}
```

Main.java

```
public class Main() {  
    public static void main(String args[]) {  
        Dog dog = new Dog("진저", 2);  
        dog.introduce();  
    }  
}
```

결과 :

저의 이름은 진저이고, 나이는 2살 입니다.

실습

계산기를 객체로 구현해보자