

객체 지향 프로그래밍

건국대학교 멋쟁이사자처럼 #S4

객체 지향 프로그래밍

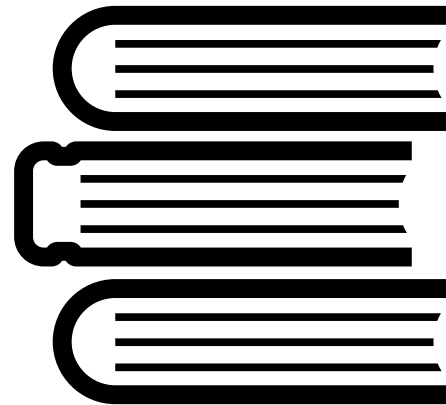
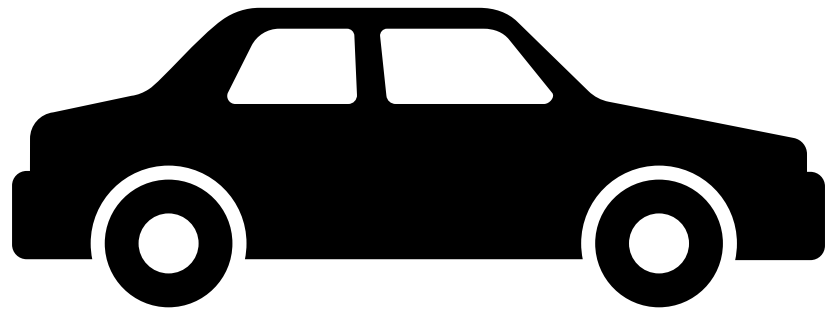
OOP(Object Oriented Programming)

부품에 해당하는 객체들을 먼저 만들고,

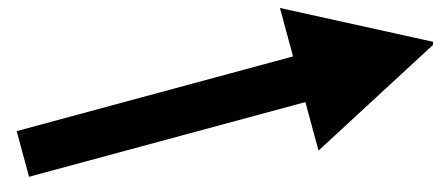
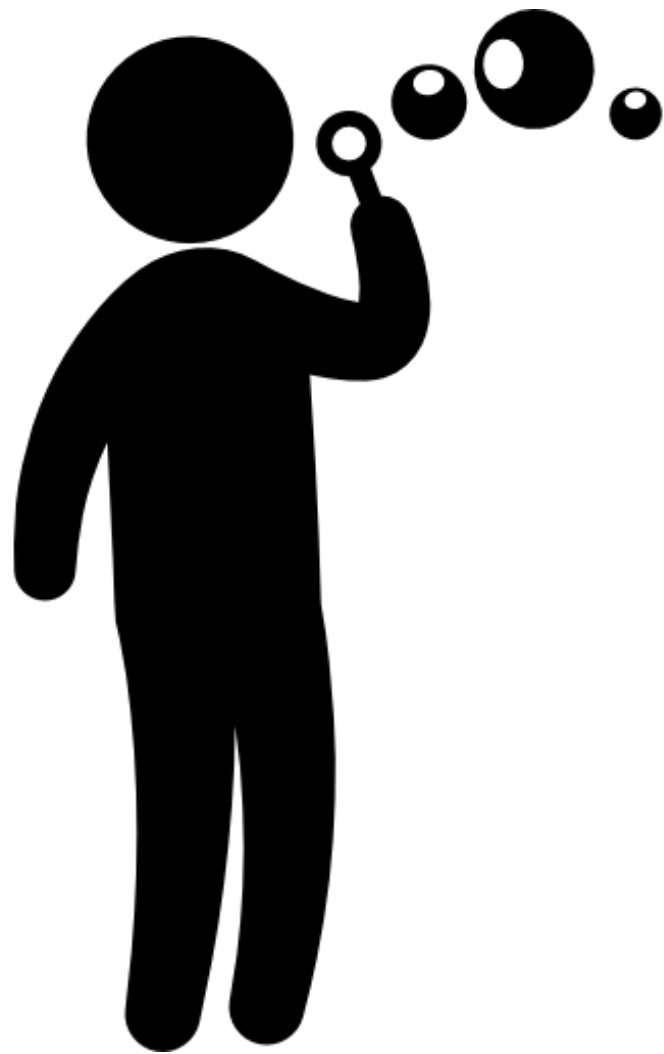
이것들을 하나씩 조립해서 완성된 프로그램을 만드는 기법

객체(Object)

자동차, 책, 사람처럼 개념상 물리적으로 존재하는 것들



객체 = 속성(변수) + 동작(메소드)



생년월일

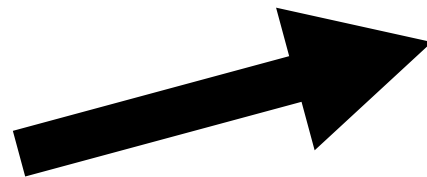
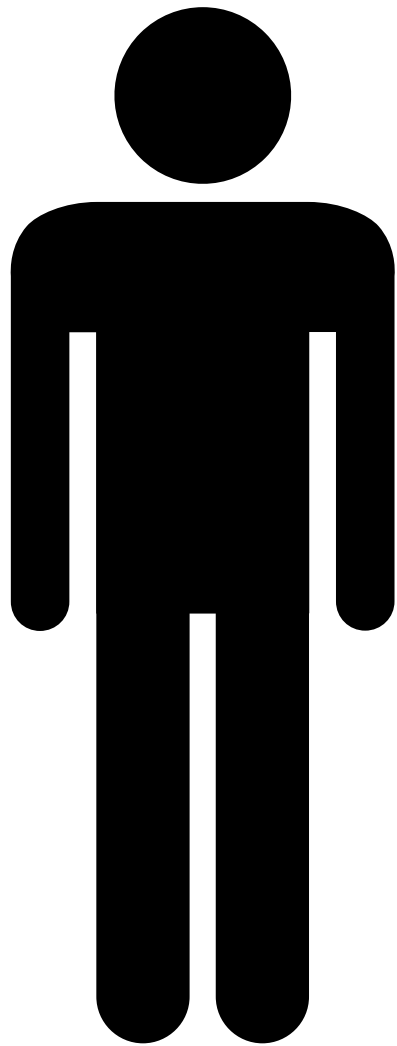


이름



성별

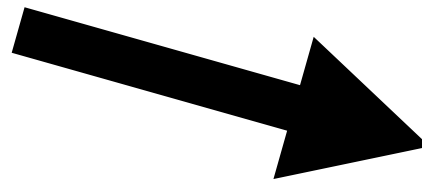
객체 = 속성(변수) + **동작(메소드)**



달리기



씻기

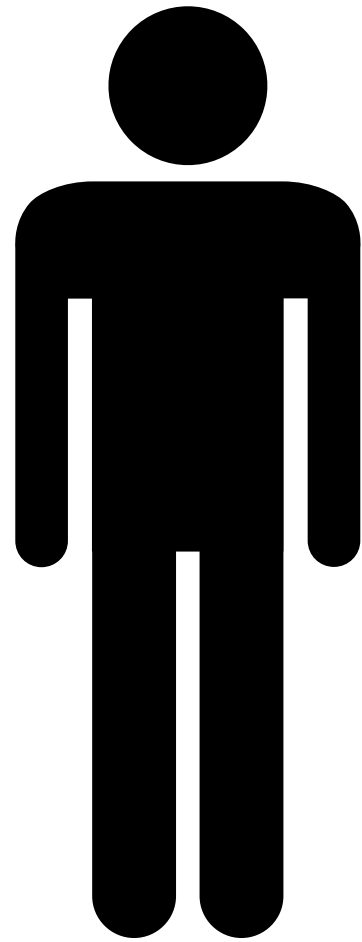


자전거타기

class = 객체의 설계도

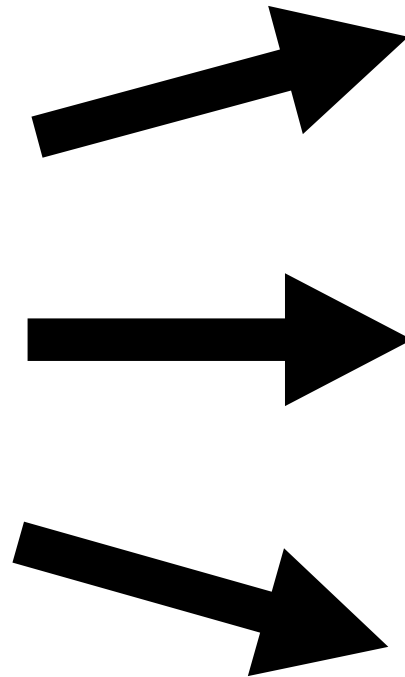
class = 객체의 변수 + 메소드

인스턴스(instance)
= 클래스로부터 만들어진 객체



사람 클래스

- 달리기
- 걷기
- 씻기



인스턴스



이름 : 박현진

나이 : 23

- 달리기
- 걷기
- 씻기



이름 : 이현경

나이 : 23

- 달리기
- 걷기
- 씻기

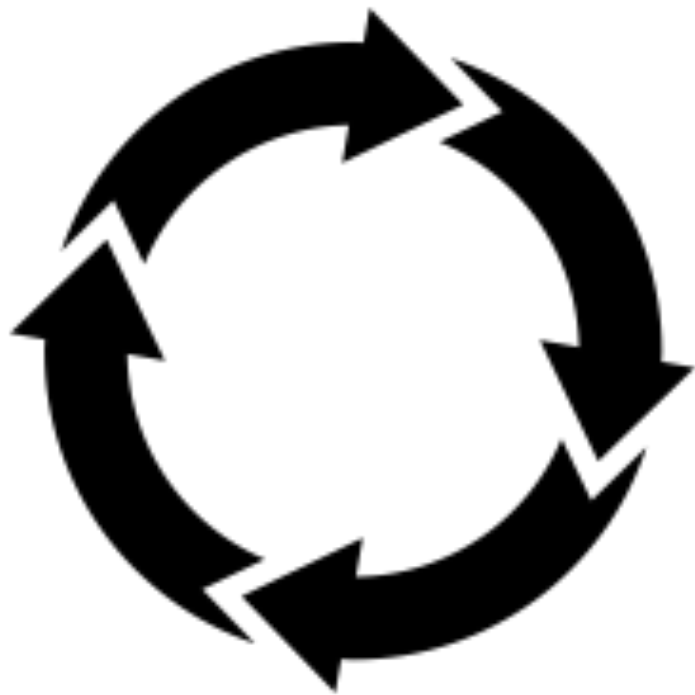


이름 : 정다혜

나이 : 24

- 달리기
- 걷기
- 씻기

코드의 재사용 + 중복제거



Person 클래스를 만들어보자!

- 클래스 초기화(oop0.rb)
- 지역,전역 변수 (oop1.rb)
- 인스턴스 변수 (oop2.rb)
- 클래스 변수 (oop3.rb)
- 상속 (oop4.rb)
- 오버라이딩 (다형성) (oop5.rb)
- public / private (캡슐화, 은닉화) (oop6.rb)
- attr_accessor (oop7.rb / oop8.rb)

Ruby = 모든 것이 객체

```

[dahyeui-MacBook-Pro:Ruby dh0023$ irb
[irb(main):001:0> p 1.to_s
"1"
=> "1"
[irb(main):002:0> p true.to_s
"true"
=> "true"
[irb(main):003:0> p 'hello'.class
String
=> String
[irb(main):004:0> p String.ancestors
[String, Comparable, Object, Kernel, BasicObject]
=> [String, Comparable, Object, Kernel, BasicObject]
[irb(main):005:0> p 10.class
Integer
=> Integer
[irb(main):006:0> Integer.ancestors
=> [Integer, Numeric, Comparable, Object, Kernel, BasicObject]
[irb(main):007:0> p true.class
TrueClass
=> TrueClass
[irb(main):008:0> TrueClass.ancestors
=> [TrueClass, Object, Kernel, BasicObject]
[irb(main):009:0>

```

문자열, 수치 뿐만 아니라 true/false nil같은 객체에도 클래스가 존재한다.
Object 클래스의 자식 관계로 구성