

VII. 클래스

1. **클래스**는 어떤 사물의 공통 속성을 모아 정의한 **추상적인 개념**이고, **인스턴스**는 클래스의 속성을 지니는 **구체적인 사례**입니다. 상위 클래스(super)의 조건을 충족하면서 더욱 구체적인 조건이 추가된 것을 하위 클래스(subclass)라고 합니다.
 - a. 인스턴스는 하나의 클래스만을 바탕으로 만들어집니다.
 - b. 어떤 인스턴스가 다양한 클래스에 속할 수는 있지만 이 클래스들은 모두 인스턴스 입장에서는 '직계존속'입니다.
 - c. 인스턴스를 생성할 때 호출할 수 있는 클래스는 오직 하나뿐일 수 밖에 없습니다.
2. 클래스가 먼저 정의되어야만 그로부터 공통적인 요소를 지니는 개체들을 생성할 수 있습니다. 프로그래밍 언어에서의 클래스는 사용하기에 따라 추상적인 대상일 수도 있고 구체적인 개체가 될 수도 있습니다.
 - a. 추상적인 개념의 클래스: 구체적인 인스턴스가 사용할 메서드를 정의한 '틀'의 역할을 담당하는 목적을 가질 때 → 일반적인 사용 방식
 - b. 개체로서의 클래스: 클래스 자체를 this로 해서 직접 접근해야만 하는 스택틱 메서드를 호출할 때
3. 클래스의 prototype 내부에 정의된 메서드를 프로토타입 메서드라고 하며, 이들은 인스턴스가 마치 자신의 것처럼 호출할 수 있습니다.
4. 한편, 클래스(생성자 함수)에 직접 정의한 메서드를 스택틱 메서드라고 하며, 이들은 인스턴스가 직접 호출할 수 없고 클래스(생성자 함수)에 의해서만 호출할 수 있습니다.
5. 클래스 상속을 흉내 내기 위한 세가지 방법
 - a. 바로 SubClass.prototype에 SuperClass의 인스턴스를 할당한 다음 프로퍼티를 모두 삭제하는 방법
 - b. 빈 함수(Bridge)를 활용하는 방법
 - c. Object.create를 이용하는 방법

