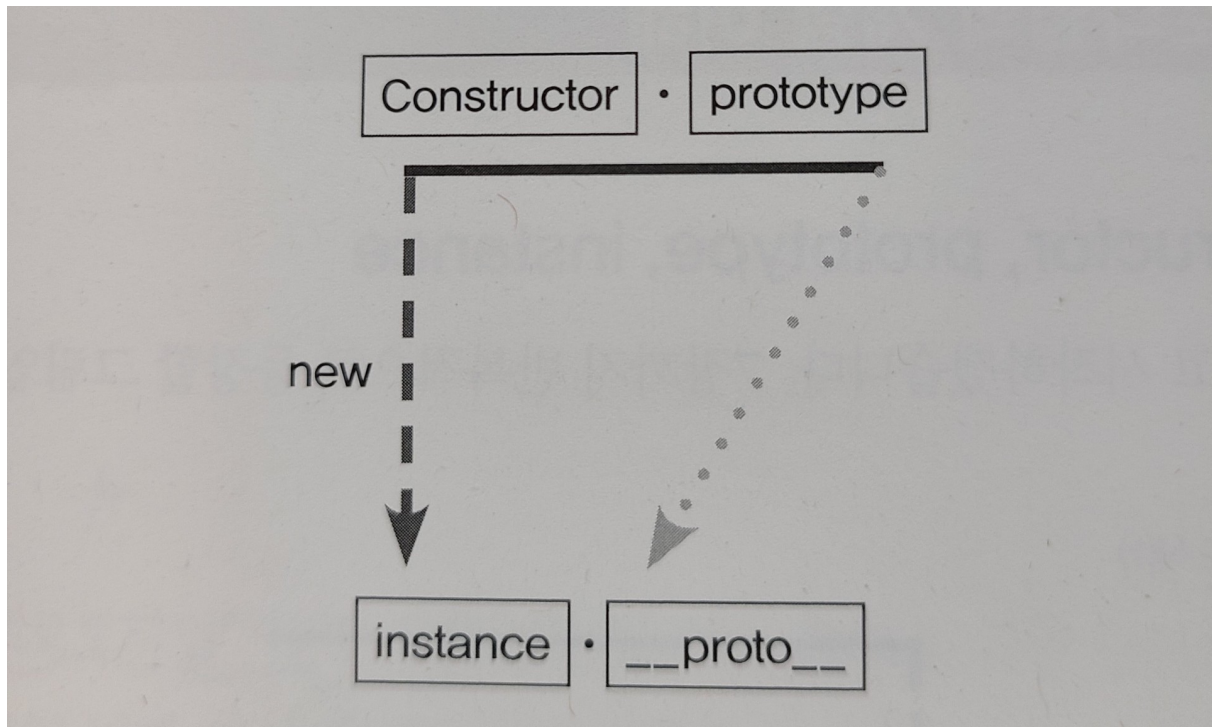


# VI. 프로토타입



프로토타입 도식

1. 자바스크립트는 프로토타입 기반 언어이다. 어떤 객체를 원형으로 삼고 이를 복제(참조)함으로써 상속과 비슷한 효과를 얻는다.
2. `var instance = new Constructor()`
  - a. 어떤 생성자 함수(Constructor)를 new 연산자와 함께 호출하면
  - b. Constructor에서 정의된 내용을 바탕으로 새로운 인스턴스(instance)가 생성된다.
  - c. 이때 instance에는 `__proto__` 라는 프로퍼티가 자동으로 부여되는데,
  - d. 이 프로퍼티는 Constructor의 prototype이라는 프로퍼티를 참조한다.
3. `__proto__` 는 생략 가능한 프로퍼티이다. 구조를 설계한 브랜든 아이크의 아이디어로 이 해의 영역은 아니다.
4. `__proto__` 를 생략하지 않으면 `this` 는 `suzi.__proto__` 를 가리키지만, 이를 생략하면 `suzi` 를 가리킨다. `suzi.__proto__` 에 있는 메서드인 `getName`을 실행하지만 `this` 는 `suzi` 를 바라보게 할 수 있게 된 것이다.

- a. Constructor.prototype의 메서드를 마치 자신의 메서드인 것처럼 호출 할 수 있다.

```
suzi.__proto__.getName  
-> suzi(.suzi.__proto__ ).getName  
-> suzi.getName
```

5. 다시 한번 정리하자면 new 연산자로 Constuctor를 호출하면 instance가 만들어지는데, 이 instance의 생략 가능한 프로퍼티인 `__proto__` 는 Constuctor의 prototype을 참조한다!
6. 프로토타입 체이닝: 자바스크립트 엔진은 데이터를 자신의 프로퍼티들을 검색해서 원하는 메서드가 있으면 그 메서드를 실행하고, 없으면 `__proto__` 를 검색해서 있으면 그 메서드를 실행하고, 없으면 다시 `__proto__` 를 검색해서 실행하는 식으로 진행한다.
7. Constructor.prototype에는 constuctor라는 프로퍼티가 있는데, 이는 다시 생성자 함수 자신을 가리킨다. 이 프로퍼티는 인스턴스가 자신의 생성자 함수가 무엇인지를 알고자 할 때 필요한 수단이다.
8. 최종적으로는 Object.prototype에 도달하게되며, 이런식으로 `__proto__` 안에 `__proto__` 를 찾아가는 과정을 프로토타입 체이닝이라고 한다.
9. 프로토타입 체이닝을 통해 각 프로토타입 메서드를 자신의 것처럼 호출할 수 있다. 이때 접근 방식은 자신으로부터 가장 가까운 대상부터 점차 먼 대상으로 나아가며, 원하는 값을 찾으면 검색을 중단한다.
10. Object.prototype에서는 모든 데이터 타입에서 사용할 수 있는 범용적인 메서드만이 존재하며, 객체 전용 메서드는 어느 데이터 타입과 달리 Object 생성자 함수에 스택하게 담겨있다.
11. 프로토타입 체인은 반드시 2단계로만 이뤄지는 것이 아니라 무한대의 단계를 생성할 수도 있다.