**프로토 타입**

자바스크립트는 프로토타입기반 객체지향언어라고 표현 한다. 이 프로토타입이라는 것은 무엇을 의미하는 것인가 ?

let objectLiteral = {}; *// === function newObj(){return {}}*

let objectConstructor = new Object(); *// 2*

첫 번째 리터럴 방법은 객체를 만들어 내는 숏컷이고, 두번째는 생성자를 이용한 방법이다. 이 두방법 모두 Object생성자의 프로토 타입을 상속 받아 생성되었다.

Es6이전에는 클래스라는 개념이 없었다. 그래서 새로운 인스턴스를 만들기 위해서 기존의 객체를 복사하는 방법을 택했다. 객체의 원형이라는 뜻의 프로토타입을 이용해서 말이다. 그렇다면 이 프로토 타입은 어떻게 사용되는 것인가 .

function Person(){

*this*.hand = 2;

*this*.body = 1;

*this*.nose = 1;

}

let kim = new Person();

let lee = new Person();

console.log(kim.hand); *// 2*

console.log(lee.hand); *// 2*

처음 보는 형태의 객체 선언 방식 일 것이다. 자바스크립트는 생성자 consturctor가 클래스에 소속하지 않는다. 그래서 생성자는 그냥 함수에 불과하다. 그 함수에 new 를 붙이게 되면 그냥 객체를 만든다. 함수에 new를 붙이면 리턴 값은 객체가 된다.

function Person(){}// 평범한 함수

**var p = new Person(); // new 붙이고 함수 호출하면 ---> 함수가 아니라 생성자라 한다.(객체의 생성자)**--> 비어있는 객체를 만들고 p에 반환

p.name = 'nimi';

p.introduce = function(){// 프로퍼티에 익명함수를 넣게 되면 이 프로퍼티는 메소드라고 불리게 된다.

    return 'My name is '+this.name;

}

document.write(p.introduce());

오래된 객체를 만드는 방법이다.

function Person(){

*this*.hand = 2;

*this*.body = 1;

*this*.nose = 1;

}

let kim = new Person();

let lee = new Person();

console.log(kim.hand); *// 2*

console.log(lee.hand); *// 2*

자 다시 이 코드를 보자 . 우리는 Person이라는 함수를 통해서 kim과 lee 객체를 생성했다. 이들은 손과, 발, 코라는 프로퍼티를 가지고 있다. 메모리에 입장으로는 지금 6개의 영역에 같은 값이 저장되었을 것이다.

다음으로는 프로토타입을 사용하여 객체를 생성하는 코드를 만들어보자 .

function Person(){}

Person.prototype.hand = 2;

Person.prototype.body = 1;

Person.prototype.nose = 1;

let kim = new Person();

let lee = new Person();

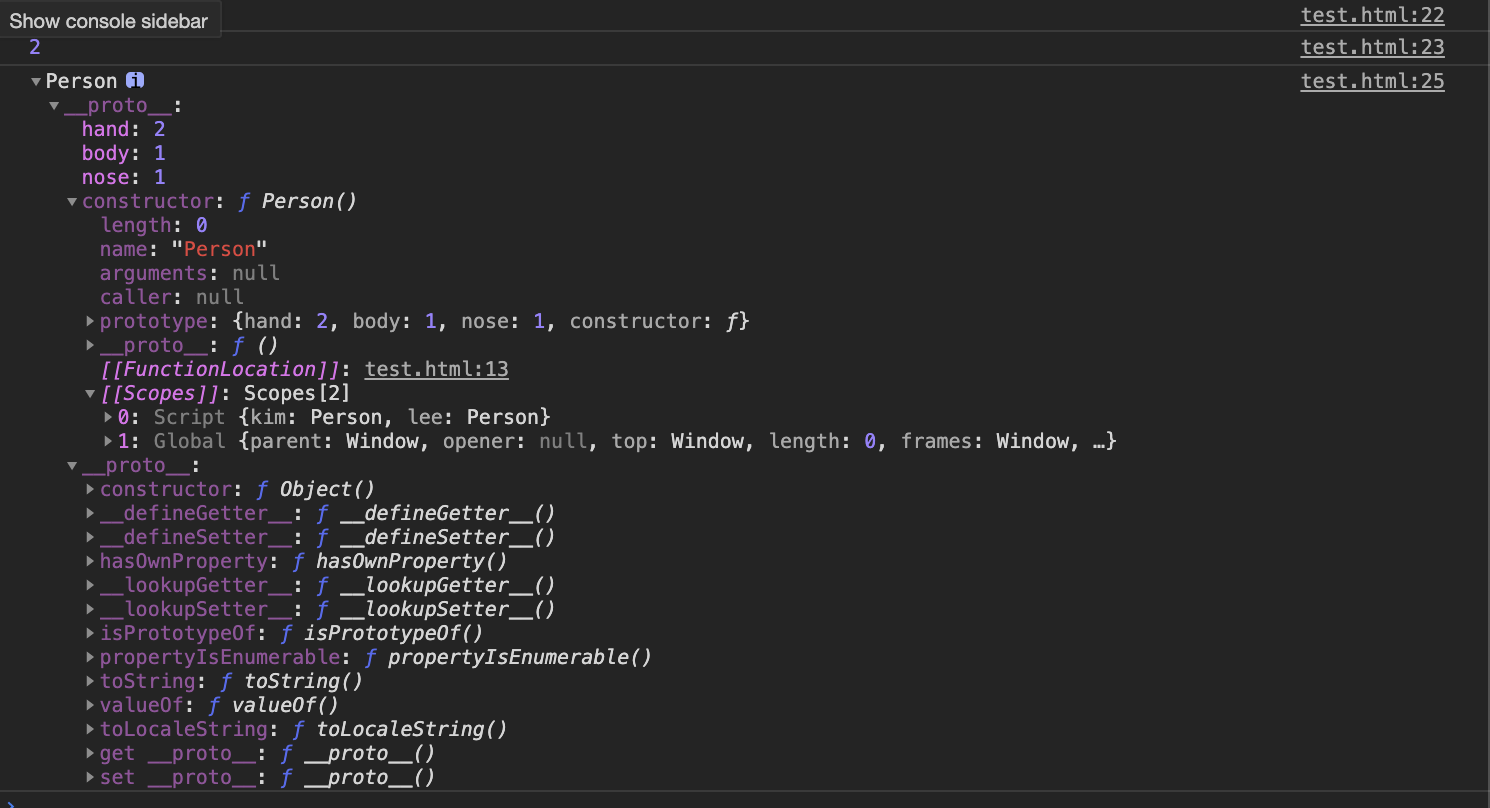
console.log(kim.hand) *// 2*

console.log(lee.hand) *// 2*

위에 것과 별 차이가 없어 보이는데, 큰 차이가 있다. 메모리 관점에서 다시 보자, kim과 lee는 자신이 손,몸,코를 가지고 있는게 아니라 현재 prototype이라는 어떤 object의 손몸코를 공유하고 있다. 그렇다는 말은 위의 코드 처럼 생성된 객체의 수만큼 변수가 저장되는 것이 아니라 같은 생성자를 통해서 만들어진 객체들은 하나의 프로토 타입이라는 오브젝트를 공유하고 있다는 것이다.

자바스크립트에서 함수가 정의 될 때, 함수는 자동으로 constructor가 부여 된다. 생성자가 부여된 함수는 new를 통해서 객체를 생성 할 수 있게 된다. 또한 프로토 타입 객체가 생성되고 이 객체와 함수가 가지고 있는 프로토 타입 속성은 연결된다.

그런데 신기한건 kim이라는 객체는 손,몸,코 프로퍼티를 가지고 있지 않음에도 어디선가 값을 가져와서 2를 찍어낸다. 이때 나타나는 개념이 프로토타입 링크이다.

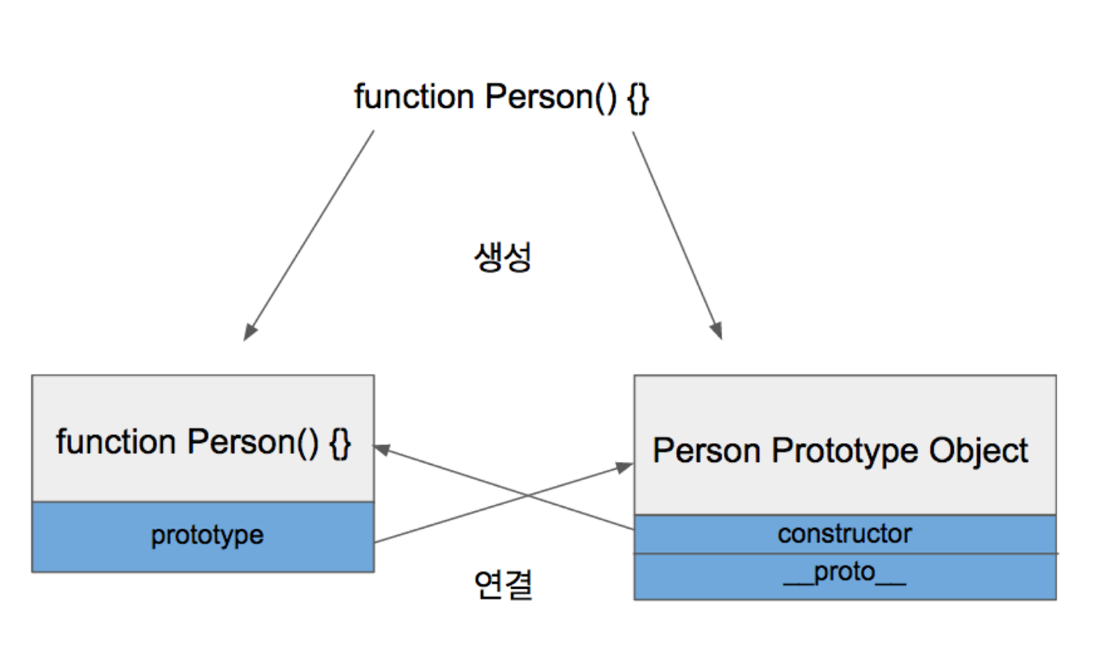


브라우저에서 kim을 찍어보면 node에서 보이지 않는 속성들이 나타난다.

\_\_proto\_\_라는 것을 가지고 있는데, 이는 자신을 생성했던 함수가 가지고 있는 속성들을 가르킨다. 자신의 부모의 속성들을 물려받은 것이다. 마치 dna처럼 말이다. 때문에 kim.hand라는 의미는 부모인 Person.prototype이 가르키고 있는 prototype Object의 속성 중 hand라는 속성을 가르킨다.

이렇게 데이터들은 연결에 연결을 하고 있으며 이는 자바스크립트의 핵심개념이다. 괜히 프로토타입의 기반언어라고 하는 것이 아니다.

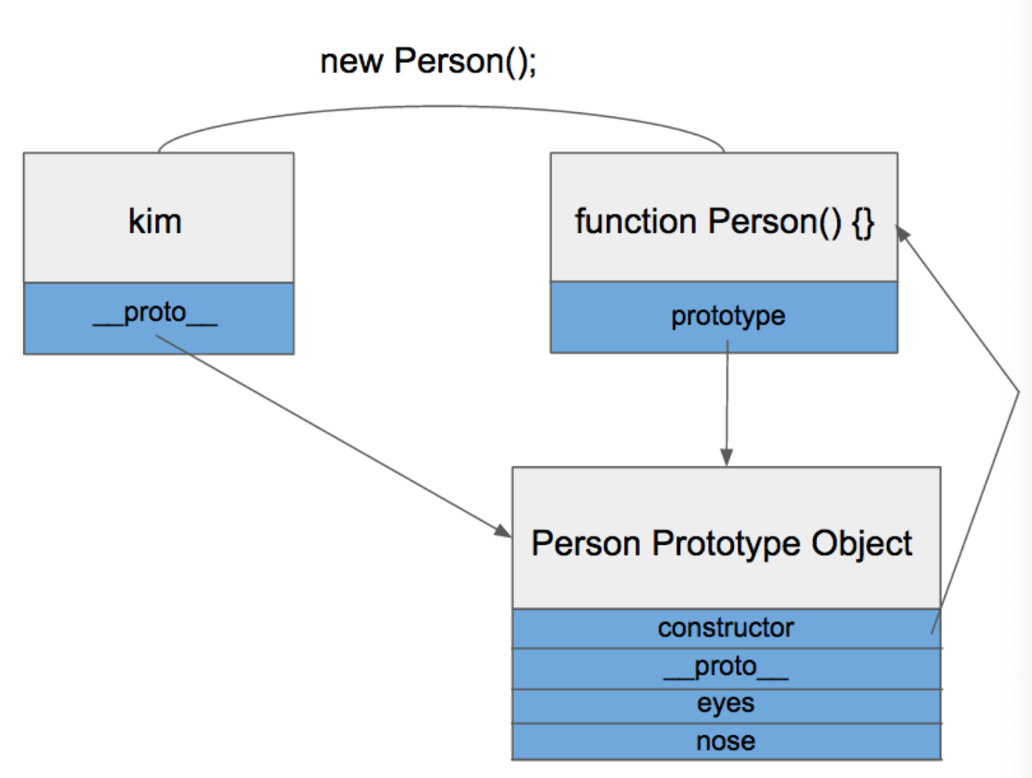
정리



함수를 정의하면 함수만 생성되는 것이 아니라 함수 객체의 속성인 프로토타입이 생성된다.

이 프로토 타입이라는 속성을 통해 프로토타입 객체에 접근 할 수 있다. 왜냐면 함수도 일종의 객체이기 때문이다. 그래서 이 프로토타입 객체는 일반적인 객체와 같으나 기본적인 속성 consturctor와 \_proto\_를 가지고 있다.





그래서 이 kim객체가 eyes를 직접 가지고 있지 않기 때문에 eyes속성을 찾을 때 까지 상위 프로토 타입을 탐색한다. 최상위인 Object의 프로토 타입 Object까지 도달 했는데도 못 찾았을 경우 언디파인드를 리턴한다. 이렇게 \_proto\_속성을 통해 상위 프로토 타입과 연결되어 있는 형태를 프로토 타입 체인이라고 한다.