

Prototype

Prototype vs Class

- 자바스크립트는 프로토타입 기반의 객체지향 언어이다.
- ES6에서 Class 문법이 추가되었지만, 자바스크립트가 클래스 기반으로 바뀐 것은 아니다.
- Class는 **Syntax Sugaring**일 뿐이다.

Prototype, 언제 쓰는 것인가?

```
function Person() {  
    this.eyes = 2;  
    this.nose = 1;  
}  
  
var kim = new Person();  
var park = new Person();  
  
console.log(kim.eyes); // 2  
console.log(kim.nose); // 1  
  
console.log(park.eyes); // 2  
console.log(park.nose); // 1
```

kim 과 park 는 eyes 와 nose 를 공통적으로 갖고 있는데, 메모리에는 각각 eyes 와 nose 가 두 개씩 총 4개가 할당된다. 이러한 문제를 프로토타입으로 해결해보자.

```
function Person() {}  
  
Person.prototype.eyes = 2;  
Person.prototype.nose = 1;  
  
var kim = new Person();  
var park = new Person();  
  
console.log(kim.eyes); // 2
```

간략히 줄이자면 `Person.prototype` 이라는 빈 Object가 어딘가에 존재하며, `Person` 함수로부터 생성된 객체(예시에서는 `kim`, `park`)들은 해당 Object에 들어 있는 값을 모두 쓸 수 있다.

Prototype Link와 Prototype Object

Prototype Link와 Prototype Object라는 것이 있으며, 이 둘을 통틀어 Prototype이라고 한다.

Prototype Object

객체는 언제나 함수(Function)로 생성된다.

```
function Person() {}  
var personObject = new Person();
```

- 여기서 `personObject` 객체는 `Person` 이라는 함수로 생성된 객체이다.
- 언제나 객체는 함수에서 시작된다.
- 자주 쓰이는 일반적인 객체 생성도 마찬가지이다.


```
var obj = {};
```

위 코드는 다음 코드와 같다.

```
var obj = new Object();
```

- 위 코드에서 `Object` 가 자바스크립트에서 기본적으로 제공하는 함수이다.
- `Object` 와 마찬가지로 `Function`, `Array` 도 모두 함수로 정의되어 있다.

그렇다면 이것이 Prototype Object와 무슨 상관이 있는가?

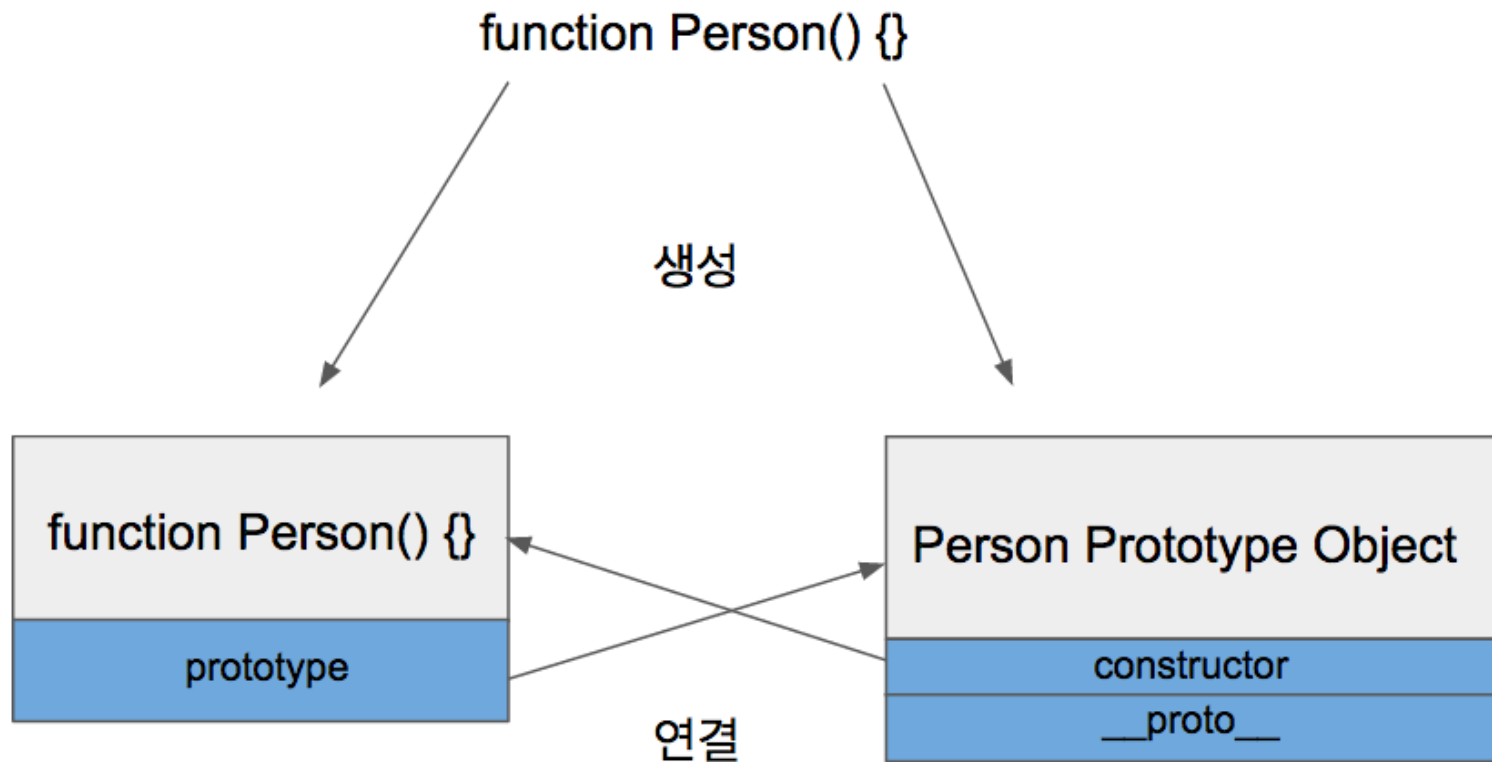
함수가 정의될 때는 2가지 일이 동시에 이루어진다.

1. 해당 함수에 **Constructor(생성자)** 자격 부여

- Constructor 자격이 부여되면 `new` 를 통해 객체를 만들어 낼 수 있게 된다.
- 이것이 함수만 `new` 키워드를 사용할 수 있는 이유이다.

2. 해당 함수의 **Prototype Object** 생성 및 연결

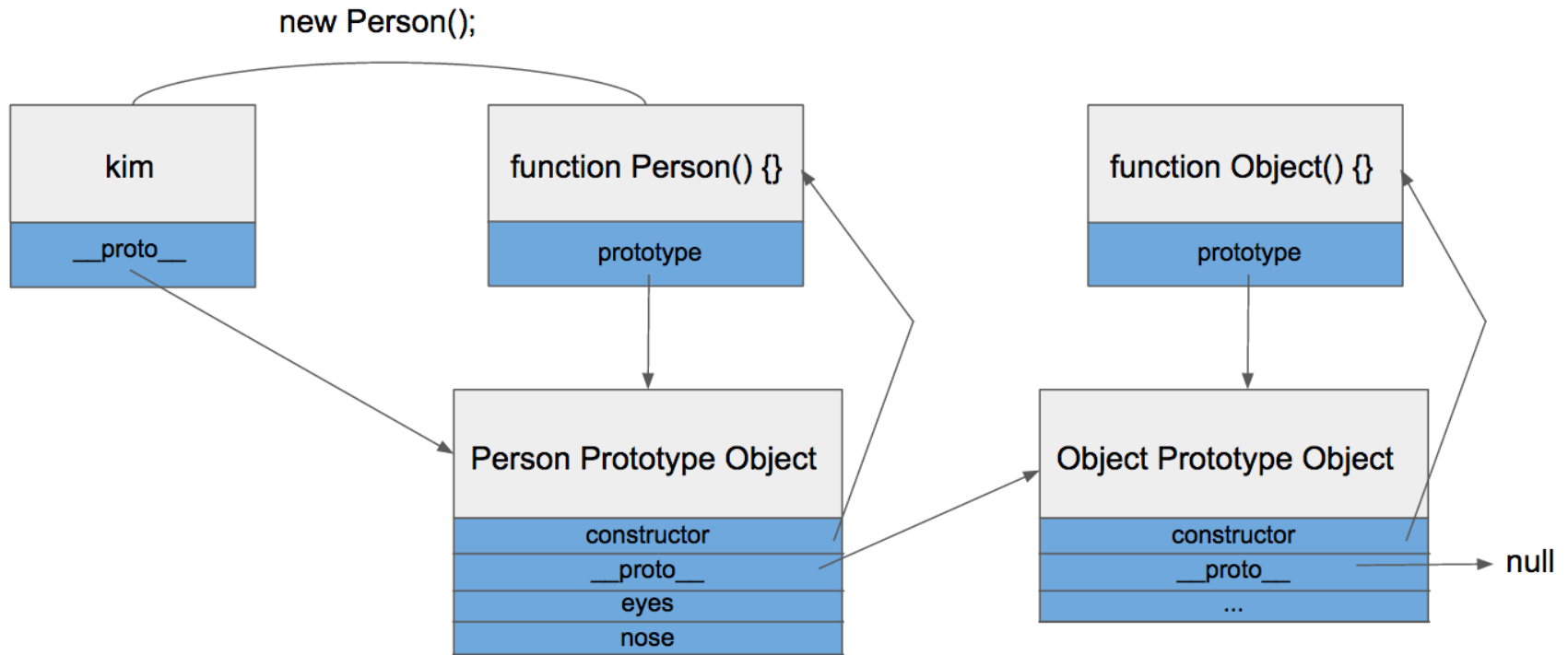
- 함수를 정의하면 함수 생성뿐 아니라, **Prototype Object** 도 같이 생성 된다.



- 그리고 생성된 함수는 `prototype` 이라는 속성을 통해 **Prototype Object**에 접근할 수 있다.
- Prototype Object는 일반적인 객체와 같으며 기본 속성으로 `constructor` 과 `__proto__` 를 갖고 있다.
- `constructor` 은 Prototype Object와 같이 생성되었던 함수를 가리키고 있다.
- `__proto__` 가 바로 **Prototype Link**이다.

Prototype Link

- `prototype` 속성은 함수만 갖고 있는 반면, `__proto__` 속성은 모든 객체가 갖고 있다.
- `__proto__` 는 객체가 생성될 때의 조상이었던 함수의 Prototype Object를 가리킨다.
- 연속적으로 `__proto__` 를 통해 연결된 형태를 프로토타입 체인이라고 한다.



- 프로토타입 체인 구조 덕분에 모든 객체는 `Object`의 자식이라고 할 수 있으며, `Object`의 모든 속성을 사용할 수 있다.

Reference

[Javascript] 프로토타입 이해하기