

[Constructor 란?]

많은 수의 똑같은 객체를 만드는데 우리 손으로 하나하나 타이핑을 하기엔 귀찮다..

귀찮은 일을 나 대신 컴퓨터가 해준다면 ?

내가 편히 누워 자고 있는 동안

컴퓨터가 열심히 객체를 만들도록 하나의 틀을 통해,

똑같은 형식의 객체를 찍어내는 객체공장을 만들면 편할 것이다.

객체 봉어빵을 찍어내는 공장이 있다면,

그 안에 들어갈 소가 팔인지 슈크림인지 야채인지만 정해주면 되는 것이다.

그 공장이 바로 **Constructor, 즉 생성자**다.

[Prototype 이란?]

객체공장 안에는 객체 봉어빵을 찍어내는 기계가 있을 것이다.

그 기계에게 객체 봉어빵을 만들 때 참고하라고 설계도를 주면

노동 조건이 개선되어 좋아할 것이다.

Prototype 은 **생성자** 함수로 찍어낸 객체 안에 들어갈 메서드의 기본 제작 청사진이라고 할 수 있다.

객체를 찍어내는 기계에게 별도로 새로운 형태의 메서드를 주문 제작하지 않는 이상

기계는 Prototype 대로 메서드를 찍어낸다.

생성자 함수로 생성된 객체가 공통으로 가지는 공간.

[Prototype 을 쓰지 않으면?]

Prototype 없이 **생성자 함수**를 호출할 때마다 새로 메서드를 만들면 메모리를 낭비하게 된다.

참고할 기본 형태가 없어서 객체 찍는 기계가 힘든 것이다.

new 키워드

```
<script>
function Constructor(value) {
  this.value = value;
}
var constructor = new Constructor('Hello');
alert(constructor.value);
</script>

<script>
function Constructor(value) {
  this.value = value; // this.value는 window.value
}
var constructor = Constructor('Hello'); //일반 함수가 되고
alert(constructor);
</script>
```

new 키워드로 함수를 호출하면 객체를 위한 공간을 만들고 **this** 키워드가 해당 공간을 의미한다. 그렇지 않는 경우, 일반적으로 **this** 키워드를 사용하면 **window** 객체를 나타낸다. **window** 객체에 속성을 추가한 것이 된다.