

JS-중급 2

☰ 태그	
📅 날짜	@2023년 5월 30일

생성자 함수

앞서 객체를 만드는 법은 배웠다.

```
let user = {  
  name : 'Mike',  
  age : 30,  
}
```

위 처럼 만드는 것을 객체 리터럴이라고 한다.

개발을 하다보면 이렇게 비슷한 객체를 여러개를 만들어야 하는
상황이 생긴다. 그럴때 사용하는 것이 생성자 함수이다.

생성자 함수는 보통 첫 글자를 대문자로 해서 함수를 만들어 준다.

User라는 함수로 만들었고, 이름과 나이를 인자로 받아 this에
넣어주고 있다.

그리고 new 연산자를 사용해서 함수를 호출한다.

각각 다른 변수명으로 세번 연달아서 호출했다.

전달하는 값도 각각 다르게 전달했다.

순식간에 비슷한 객체 3개를 만들었다.

생성자 함수는 봉어빵 틀이나 와플펜이라고 생각하면 된다.

필요한 재료들을 넣어주고 찍어내는 것이다.

지금 필요한 재료는 이름과 나이였다.

생성되는 객체는 와플이다.

어떻게 동작하는지 살펴보자.

```
function User(name, age){
  // 1. 빈 객체를 만들고 this를 할당한다.
  // this = {}

  this.name = name;
  this.age = age;
  // 2. 함수 본문을 실행하면서 this에 프로퍼티를 추가

  // 3. this를 반환한다.
  // return this;

  // 실제로 1.과 3. 은 코드에 없는것.
}
```

new를 붙혀 실행하는 순간 위의 방식으로 알고리즘이 동작한다.
이렇게 객체를 만들면 일일이 객체 리터럴을 사용하는 것보다 훨씬 빠르고 일관성있게 객체를 만들 수 있다.
스테이지 변경되어도 저 생성자 함수만 고치면 된다.

어떤 함수라도 new를 붙혀서 사용하면 저 알고리즘으로 동일하게 실행된다.
생성자 함수는 첫글자를 대문자로 하는게 관례이다.

// 메소드를 추가해보자.

```
function User(name, age){
  this.name = name;
  this.age = age;
  this.sayName = function(){
    console.log(this.name);
  }
}
```

모든 유저에게는 sayName이라는 method가 있어서 자신의 이름을 말하는 기능이 있다.
this로 할당된 객체에 sayName메소드를 추가하고 유저를 하나 더 만들었다.

```
function User(name, age){
  this.name = name;
  this.age = age;
  this.sayName = function(){
    console.log(this.name);
  }
}

let user5 = new User('Han', 40);
```

```
user5.sayName(); // 'Han'
```

예제

```
function Item(title, price){
  // this = {};
  this.title = title;
  this.price = price;
  this.showPrice = function(){
    console.log(`가격은 ${price}원 입니다.`);
  }
  //return this;
}

const item1 = new Item('인형', 3000);
const item2 = new Item('지갑', 4000);
const item3 = Item('가방', 5000); // new 없음

console.log(item1, item2, item3);

item3.showPrice();

// [object Object]
{
  "title": "인형",
  "price": 3000,
  "showPrice": "function () {\n    console.log(`가격은 ${price}원 입니다.`);\n  }"
} // [object Object]
{
  "title": "지갑",
  "price": 4000,
  "showPrice": "function () {\n    console.log(`가격은 ${price}원 입니다.`);\n  }"
}

undefined
```