

04

04-2. 생성자 함수

JavaScript Object Oriented Programming

생성자 함수

객체 모형을 이용한 객체 생성

- 동일한 멤버로 구성되는 객체를 여러 개 만드는 경우에는 객체를 만드는 모형이 하나 선언하고 그 모형을 이용해 여러 개의 객체를 쉽게 만드는 방법이 있으면 쉬울 것 같습니다.
- 객체를 만드는 모형을 선언하고 그 모형을 이용해 객체를 생성하는 방법은 2가지가 있습니다.
- 생성자 함수를 이용해 객체 생성
- 클래스를 이용해 객체 생성



생성자

- 객체지향 프로그래밍에서 중요한 용어입니다.
- 한마디로 정의하면 어떤 객체의 모형이 있고 그 모형으로 객체를 만들어주는 역할을 생성자라고 합니다.
- 소프트웨어 언어마다 객체의 모형을 정의하고 이 모형을 통해 객체를 만드는 방법에는 차이가 있지만 모든 언어에서 이 객체를 만드는 작업을 "객체를 생성한다" 라는 표현을 하고 그 객체 생성 역할을 하는 것을 "생성자"라고 합니다.

생성자 함수

생성자 함수란?

- 생성자 함수는 객체를 만드는 역할을 하는 함수라는 의미입니다.
- 어떤 업무를 위해 필요한 코드를 담고 그 함수가 호출되었을 때 함수내의 코드가 실행되는 일반 함수와 문법적인 차이는 없습니다.
- 단지 일반 함수 선언하듯이 선언되었는데 그 함수는 어떤 업무의 로직을 실행시키기 위한 목적이 아니라 객체를 생성하는 목적으로 선언된 함수라는 의미에서 생성자 함수라고 부릅니다.
- 생성자 함수는 관습상 일반 함수와 구분하기 위해서 첫 글자를 대문자로 선언합니다.

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

생성자 함수

new 연산자를 이용해 객체 생성

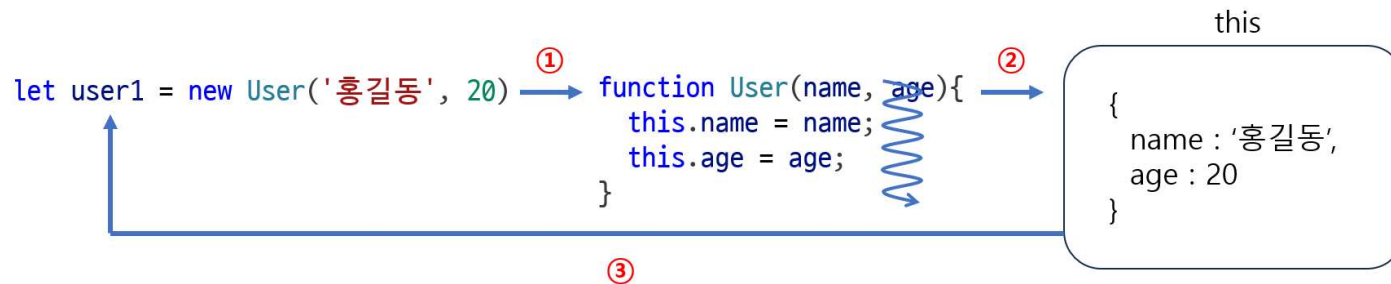
- 생성자 함수도 함수입니다.
- function 예약어로 선언되고 매개변수가 있을 수 있으며 외부에서 함수를 호출해서 사용합니다.
- 그런데 생성자 함수는 객체를 생성하기 위한 함수이며 이 생성자 함수를 호출해서 객체를 생성할 때는 new 연산자를 이용해야 합니다.
- new 는 예약어이며 흔히 객체생성 연산자라고 부릅니다.

```
//new 로 호출  
let user1 = new User('홍길동', 20)
```

생성자 함수

new 연산자를 이용해 객체 생성

- new 라는 의미 자체가 객체를 새로 생성하겠다는 것임으로 함수 호출과 동시에 객체가 자동으로 만들어지며 그 객체에 this로 멤버들을 등록하며 이렇게 만들어진 객체가 최종 생성한 곳에 자동으로 반환되게 됩니다.

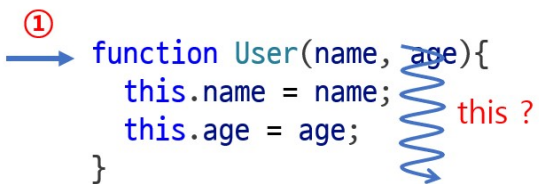


생성자 함수

new 연산자를 이용해 객체 생성

- new 를 이용하지 않고 함수를 호출하게 되면 명시적으로 객체를 생성하겠다고 선언한 것이 아님으로 객체가 자동으로 만들어지지 않습니다.

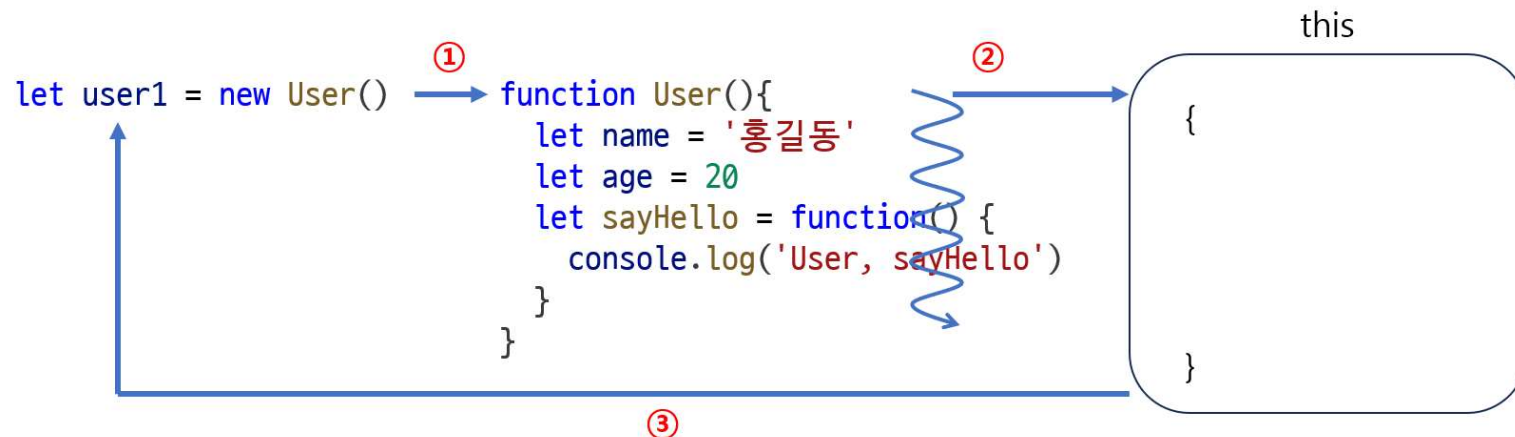
```
let user1 = User('홍길동', 20) → function User(name, age){  
    this.name = name;  
    this.age = age;  
}
```

 ① this ?

생성자 함수

this 예약어로 객체 멤버 등록

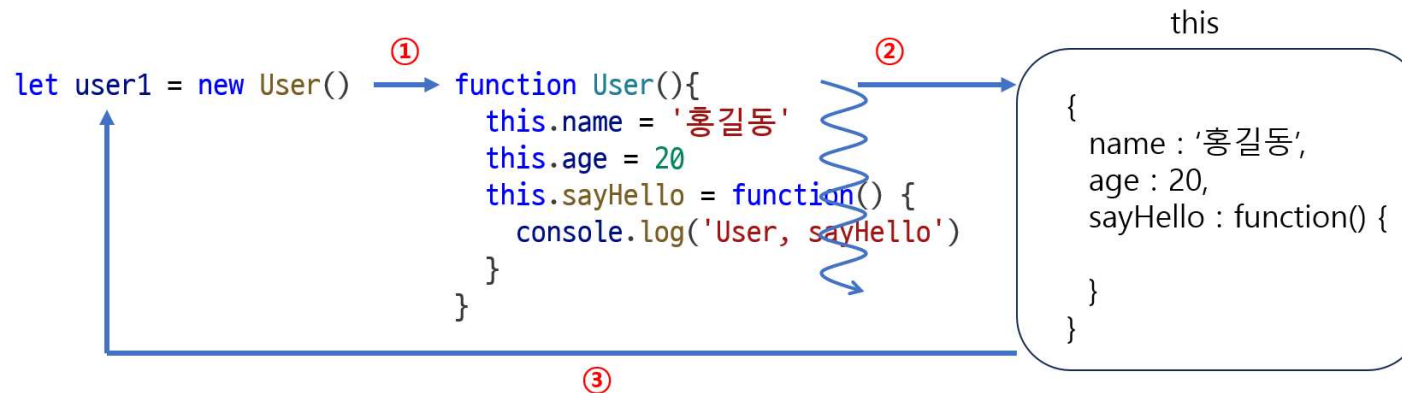
- 함수내에는 변수와 다른 함수가 선언될 수 있습니다.
- 그런데 함수가 생성자 함수로 이용된다면 함수내에 선언된 변수와 함수는 객체에 포함되어야 합니다.
- 객체에 포함될 변수, 함수는 this 예약어로 선언되어야 하며 this 예약어를 사용하지 않은 변수, 함수는 생성되는 객체와 관련이 없습니다.



생성자 함수

this 예약어로 객체 멤버 등록

- 생성자 함수에 의해 생성되는 객체에 등록되어야 하는 변수, 함수는 this 예약어로 등록해 주어야 합니다.



생성자 함수

생성자 함수의 리턴 값

- 일반 함수는 함수가 실행되고 결과 값을 함수 호출한 곳에 반환하기 위해서는 return 예약어를 이용합니다.
- 그런데 생성자 함수는 객체 생성이 주 목적임으로 new 연산자로 함수를 호출하게 되면 자동으로 만들어지는 객체가 return 예약어를 사용하지 않아도 자동으로 반환되게 됩니다.
- 만약 생성자 함수에서 return 구문을 사용하게 되면 어떻게 될까?
- return 구문을 생성자 함수에서 사용한다고 문제되지는 않습니다.
- 단지 return 에 의해 반환되는 값이 기초 데이터인지 객체인지에 따라 차이가 있습니다.

생성자 함수

생성자 함수의 리턴 값

- new 로 호출되는 생성자 함수에서 이처럼 기초 데이터를 return 시키면 이 return 은 무시되며 자동으로 만들어진 객체가 반환되게 됩니다.

생성자 함수에서 기초 데이터 반환

```
1 function User(name, age){  
2   this.name = name  
3   this.age = age  
4   return 10
```

- 생성자 함수에서 명시적으로 return 을 이용해 특정 객체를 반환한다면 자동으로 생성된 this 객체는 무시됩니다.

```
function User(name, age){  
  this.name = name  
  this.age = age  
  return {  
    prodId: 1,  
    prodName: '노트북'  
  }  
}
```

생성자 함수

외부에서 멤버 추가

- 생성자 함수로 객체를 생성한 후에 원한다면 그 객체에 변수, 함수등의 멤버를 추가할 수 있습니다.
- 생성자함수로 생성한 객체에 원한다면 멤버를 추가할 수 있고, 멤버를 추가하게 되면 동일한 생성자 함수로 선언된 객체끼리 상이한 멤버를 가지게 할 수 있게 됩니다.

```
function User(name, age){  
  this.name = name  
  this.age = age  
}  
  
let user1 = new User('홍길동', 20)  
let user2 = new User('김길동', 30)  
  
console.log(user1)//User {name: '홍길동', age: 20}  
console.log(user2)//User {name: '김길동', age: 30}  
  
user1.address = 'seoul'  
user2.phone = '010-1111-1111'
```



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어
William Shakespeare