

삼육대학교 SW융합교육원 김진호

자바스크립트 기초교육

07. Object constructor function (객체 생성자 함수)

7-1. Object constructor function

7-2. constructor function

❖ 7. Object constructor function

7-1. Object constructor function

7-1-1. Object constructor function

```
const student = new Object(); // 빈 객체 생성
student.name = '유관순';      // 프로퍼티 추가
student.age = 16;
```

- new 연산자와 함께 Object 생성자 함수를 호출하면 빈 객체를 생성하여 반환한다.
빈 객체 생성 이후 프로퍼티 또는 메서드를 추가하여 객체를 완성할 수 있다.
- 반드시 Object 생성자 함수를 사용해 객체를 생성할 필요는 없다.
객체 리터럴을 사용하는 것이 더 간편하다.
- 자바스크립트에는 Object 생성자 함수 이외에도 String, Number, Boolean, Function, Array, Date, RegExp 등의 빌트인 생성자 함수를 제공한다.

❖ 7. Object constructor function

7-2. constructor function (생성자 함수)

7-2-1. constructor function

```
function Student(name, age) {  
    this.name = name;  
    this.age = age;  
    this.getInfo = function() {  
        return `${this.name}(은)는 ${this.age}세입니다.`;  
    }  
}  
  
const student = new Student('장보고', 30); // 인스턴스의 생성  
console.log(student.getInfo());           // 메서드 호출
```

- 객체 리터럴을 이용한 객체 생성 방식은 직관적이고 간편하지만, 단 하나의 객체만 생성한다. 따라서, 동일 프로퍼티를 갖는 객체를 여러 개 생성해야 하는 경우 매번 같은 프로퍼티를 기술하기에 비효율적이다.
- 객체를 생성하기 위한 템플릿처럼 생성자 함수를 사용하여 프로퍼티 구조가 동일한 객체 여러 개를 간편하게 생성할 수 있다.

❖ 7. Object constructor function

7-2. constructor function (생성자 함수)

7-2-2. instance creation process (인스턴스 생성 과정)

```
function Student(name, age) {  
  // 1. 암묵적으로 인스턴스가 생성되고 this에 바인딩 되는 과정이 런타임 이전에 실행된다.  
  console.log(this);  
  // 2. this에 바인딩 되어 있는 인스턴스를 초기화 한다.  
  this.name = name;  
  this.age = age;  
  this.getInfo = function() {  
    return `${this.name}(은)는 ${this.age}세입니다.`;  
  }  
  // 3. 완성된 인스턴스가 바인딩 된 this가 암묵적으로 반환된다.  
  // 만약 명시적으로 객체를 반환하면 암묵적인 this 반환이 무시된다.  
  // return {};  
  // 하지만 명시적으로 원시 값을 반환하면 원시 값 반환은 무시되고 암묵적으로 this가 반환된다.  
  // return 1;  
  // 생성자 내부에서 return은 생략하는 것이 기본이다.  
}  
  
// 인스턴스 생성  
const student = new Student('홍길동', 20);  
console.log(student);
```

❖ 7. Object constructor function

7-2. constructor function (생성자 함수)

7-2-3. differences from regular function (일반 함수와 차이점)

- 일반 함수와 생성자 함수의 특별한 형식적 차이는 없다.
(첫 문자를 대문자로 기술하여 구별하려 노력한다.)
- new 연산자와 함께 호출하면 생성자 함수로 동작하는 것이다.
만약 new 연산자와 함께 호출하지 않으면 일반 함수로 동작한다.
- 생성자 함수가 new 연산자 없이 호출 되는 것을 방지하기 위해
ES6에서는 new.target을 지원한다.
- 대부분의 빌트인 생성자 함수(Object, String, Number, Boolean, Date, RegExp, ...)는
new 연산자와 함께 호출 되었는지를 확인한 후 적절한 값을 반환한다.
 - String, Number, Boolean의 경우 new 연산자 없이 호출하면 객체 값이 아닌 문자열,
숫자, 불리언 값을 반환한다. 이를 통해 데이터 타입을 변환하는데 사용할 수 있다.