JavaScipt Classes

- description : JavaScipt Classes
- author : 오션
- email : shlim@repia.com
- lastupdate : 2021-04-12

Source of the article

- JavaScipt Classes

ES6라고도 하는 ECMAScript 2015는 JavaScript 클래스를 도입했습니다.

JavaScript 클래스는 JavaScript 오브젝트를위한 템플릿(templates)입니다.

=====JavaScript Class Syntax `class` 키워드를 사용하여 클래스를 만듭니다.

항상 `constructor()`라는 메서드를 추가합니다.

```
class ClassName {
  constructor() { ... }
}
```

example

```
class Car {
  constructor(name, year) {
    this.name = name;
    this.year = year;
  {
}
```

위의 예에서는 “Car”라는 클래스를 만듭니다.

클래스에는 “ 이름(name)“과 “ 연도(year)“라는 두 가지 초기 속성(initial properties)이 있습니다.

JavaScript 클래스는 오브젝트가 아닙니다.

JavaScript 오브젝트를 위한 템플릿(template)입니다.

Using a Class

클래스가 있으면, 클래스를 사용하여 오브젝트를 만들 수 있습니다.

```
<!DOCTYPE html>
<html>
<body>
    <h2>JavaScript Class</h2>
    <p>How to use a JavaScript Class</p>
    <p id="demo"></p>
    <script>
        class Car {
            constructor(name, year) {
                this.name = name;
                this.year = year;
            }
        }
        myCar = new Car("Ford", 2014);
        document.getElementById("demo").innerHTML = "My car is " +
myCar.name + " " + myCar.year;
    </script>
</body>
</html>
```

위의 예제는 **Car class**를 사용하여 두 개의 **Car objects**를 만듭니다.

constructor 메서드는 새로운 오브젝트가가 생성될 때 자동으로 호출됩니다.

The Constructor Method

constructor 메서드는 특별한 메서드입니다.

- “constructor”라는 정확한 이름이 있어야 합니다.
- 새로운 오브젝트가 생성되면 자동으로 실행됩니다.
- 오브젝트 속성(object properties)을 초기화하는 데 사용됩니다.

constructor 메서드를 정의하지 않으면, JavaScript는 빈(empty) constructor 메서드를 추가합니다.

Class Methods

클래스 메소드는 오브젝트 메소드와 동일한 구문으로 생성됩니다.

`class` 키워드를 사용하여 클래스를 만듭니다.

항상 `constructor()` 메서드를 추가하십시오.

그런 다음 원하는 수의 메서드를 추가합니다.

Syntax

```
class ClassName {
  constructor() { ... }
  method_1() { ... }
  method_2() { ... }
  method_3() { ... }
}
```

Car age를 반환하는 “age”라는 클래스 메서드를 만듭니다.

예제

```
<!DOCTYPE html>
<html>
<body>
    <h2>JavaScript Class Method</h2>
    <p>How to define and use a Class method.</p>
    <p id="demo"></p>
    <script>
        class Car {
            constructor(name, year) {
                this.name = name;
                this.year = year;
            }
            age() {
                let date = new Date();
                return date.getFullYear() - this.year;
            }
        }
        let myCar = new Car("Ssangyong", 2016);
        document.getElementById("demo").innerHTML = "My car is " +
myCar.age() + " years old.";
    </script>
</body>
</html>
```

예제

클래스 메소드에 매개 변수(parameters)를 보낼 수 있습니다.

```
<!DOCTYPE html>
<html>
<body>
```

```
    <h2>JavaScript Class Method</h2>
    <p>Pass a parameter into the "age()" method.</p>
    <p id="demo"></p>
    <script>
        class Car {
            constructor(name, year) {
                this.name = name;
                this.year = year;
            }
            age(x) {
                return x - this.year;
            }
        }
        let date = new Date();
        let year = date.getFullYear();
        let myCar = new Car("Ssangyong", 2016);
        document.getElementById("demo").innerHTML = "My car is " +
myCar.age(year) + " years old.";
    </script>
</body>
</html>
```

"use strict"

클래스의 구문은 ”strict mode“로 작성되어야 합니다.

“strict mode” 규칙을 따르지 않으면 오류가 발생합니다.

예제

“stict mode”에서 선언하지 않고 변수를 사용하면, 오류가 발생합니다.

```
<!DOCTYPE html>
<html>
<body>
    <h2>JavaScript Clasess are written in "strict mode"</h2>
    <p>In a JavaScript Class you cannot use variable without declaring
it.</p>
    <p id="demo"></p>
    <script>
        class Car {
            constructor(name, year) {
                this.name = name;
                this.year = year;
            }
            age() {
                // date = new Date();  // This will not work
```

```
                let date = new Date(); // This will wrok
                return date.getFullYear() - this.year;
            }
        }
        myCar = new Car("Volvo", 2020);
        document.getElementById("demo").innerHTML = "My Car is " +
myCar.age() + " years old.";
    </script>
</body>
</html>
```

오션,, Javascript, class, intro

From:
http://rwiki.repia.com/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://rwiki.repia.com/doku.php?id=wiki:javascript:javascript_note:js_classes_-_class_intro&rev=1618191593

Last update: **2022/03/10 19:52**