

Javascript Objects

- description : Javascript Object Constructors
- author :
- email : shlim@repia.com
- lastupdate : 2021-05-03

The source of this article

Javascript Object Constructors

```
// Constructor function for Person Objects
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
}

// Create a Person object
let myFather = new Person("Anthony", "Young", 50, "green");

// Display age
document.getElementById("demo").innerHTML =
  "My father is " + myFather.age + ".";      // My father is 50.
```

(constructor function)

Object Types (Blueprints) (Classes)

(JS Object Constructors)

"type"

" "

" "

(object constructor function)

function Person()

new

```
// Constructor function for Person objects
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
}

// Create two person objects
let myFather = new Person("John", "Doe", 50, "blue");
let myMother = new Person("Sally", "Rally", 48, "green");

// Display age
document.getElementById("demo").innerHTML =
  "My father is " + myFather.age + ". My mother is " + myMother.age + ".";
// My father is 50. My mother is 48.
```

The this Keyword

JavaScript this

“ ”

this

, this

this

가

가

'this'

가

. 'this'

Adding a Property to an Object

가

Example

```
// Constructor function for Person objects
function Person(first, last, age, eye) {
  this.firstName = first;
```

```

    this.lastName = last;
    this.age = age;
    this.eyeColor = eye;
}

// Create 2 Person objects
let myFather = new Person("John", "Doe", 50, "blue");
let myMother = new Person("Sally", "Rally", 48, "green");

// Add nationality to first object
myFather.nationality = "English";

// Display nationality
document.getElementById("demo").innerHTML =
    "My father is " + myFather.nationality;    // My father is English

```

myMother가 (person 가) myFather 가 .

Adding a Method to an Object

가 :

Example

```

// Constructor function for Person Objects
function Person(first, last, age, eye) {
    this.firstName = first;
    this.lastName = last;
    this.age = age;
    this.eyeColor = eye;
}

// Create 2 Person objects
let myFather = new Person("John", "Doe", 50, "blue");
let myMother = new Person("Sally", "Field", 48, "green");

// Add a name method to first object
myFather.name = function () {
    return this.firstName + " " + this.lastName;
};

// Display full Name
document.getElementById("demo").innerHTML =
    "My father is " + myFather.name();    // My father is John Doe

```

Adding a Property to a Constructor

가

가

Example

```
// Constructor function for Person objects
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
}

// You can NOT add a new property to a constructor function
Person.nationality = "English";

// Create 2 Person objects
let myFather = new Person("John", "Doe", 50, "blue");
let myMother = new Person("Sally", "Rally", 48, "green");

// Display nationality
document.getElementById("demo").innerHTML =
  "The nationality of my Father is " + myFather.nationality;
  // The nationality of my Father is undefined
```

가

,

가

.

Example

```
// Constructor function for Person objects
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
  this.nationality = "English";
}

// Create 2 Person objects
let myFather = new Person("John", "Doe", 50, "blue");
let myMother = new Person("Sally", "Rally", 48, "green");

// Display nationality
```

```
document.getElementById("demo").innerHTML =
  "My father is " + myFather.nationality + ". My mother is " +
  myMother.nationality;
  // My father is English. My mother is English
```

(default values) 가 .

Adding a Method to a Constructor

Example

```
// Constructor function for Person Objects
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
  this.name = function () {
    return this.firstName + " " + this.lastName
  };
};

// Create a Person object
let myFather = new Person("John", "Doe", 50, "blue");

// Display full name
document.getElementById("demo").innerHTML =
  "My father is " + myFather.name(); // My father is John Doe
```

가

가

가

Example

```
function Person(firstName, lastName, age, eyeColor) {
  this.firstName = firstName;
  this.lastName = lastName;
  this.age = age;
  this.eyeColor = eyeColor;
  this.changeName = function (name) {
    this.lastName = name;
  };
}
```

```
};  
}
```

changeName() person lastName .

Now You Can Try:

```
// Constructor function for Person objects  
function Person(firstName, lastName, age, eyeColor) {  
    this.firstName = firstName;  
    this.lastName = lastName;  
    this.age = age;  
    this.eyeColor = eyeColor;  
    this.changeName = function (name) {  
        this.lastName = name;  
    };  
}  
  
// Create a Person object  
let myMother = new Person("Sally", "Rally", 48, "green");  
  
// Change last name  
myMother.changeName("Doe");  
  
// Display last name  
document.getElementById("demo").innerHTML =  
"My mother's last name is " + myMother.lastName; // My mother's last name  
is Doe
```

JavaScript **this** myMother “ ” person .

Built-in JavaScript Constructors

JavaScript (built-in constructor)가 .

Example

```
let x1 = new Object(); // A new Object object  
let x2 = new String(); // A new String object  
let x3 = new Number(); // A new Number object  
let x4 = new Boolean(); // A new Boolean object
```

```
let x5 = new Array();           // A new Array object
let x6 = new RegExp();          // A new RegExp object
let x7 = new Function();        // A new Function object
let x8 = new Date();            // A new Date object

// Display the type of all objects
document.getElementById("demo").innerHTML =
  "x1: " + typeof x1 + "<br>" + // x1: object
  "x2: " + typeof x2 + "<br>" + // x2: object
  "x3: " + typeof x3 + "<br>" + // x3: object
  "x4: " + typeof x4 + "<br>" + // x4: object
  "x5: " + typeof x5 + "<br>" + // x5: object
  "x6: " + typeof x6 + "<br>" + // x6: object
  "x7: " + typeof x7 + "<br>" + // x7: function
  "x8: " + typeof x8 + "<br>";  // x8: object
```

```
Math()      . Math()      . new      Math
```

Did You Know?

	JavaScript	(primitive data types)
String, Number	Boolean	.
	가	(primitive values)
:		
new Object()	{}	.
new String()	""	.
new Number()	12345	.
new Boolean()	true / false	.
new Array()	[]	.
new RegExp()	/()/	.
new Function()	() {}	.

Example

```
let x1 = {};           // object
let x2 = "";           // string
let x3 = 0;            // number;
let x4 = false;        // boolean
```

```
let x5 = [];           // object (not array)
let x6 = /()/;         // object
let x7 = function () { }; // function

// Display the type of all
document.getElementById("demo").innerHTML =
  "x1: " + typeof x1 + "<br>" + // x1:object
  "x2: " + typeof x2 + "<br>" + // x2:string
  "x3: " + typeof x3 + "<br>" + // x3:number
  "x4: " + typeof x4 + "<br>" + // x4:boolean
  "x5: " + typeof x5 + "<br>" + // x5:object
  "x6: " + typeof x6 + "<br>" + // x6:object
  "x7: " + typeof x7 + "<br>"; // x7: function
```

String Objects

```
(var firstName = "John")
new (var firstName = new String ( "John"))
```

[JS Strings](#)

Number Objects

```
(var x = 123)
new (var x = new Number (123))
```

[JS Numbers](#) 가

Boolean Objects

```
(var x = false)
new (var x = new Boolean (false))
```

[JS Booleans](#)

, [Javascript](#), [Objects](#), [Constructors](#)

From:
<http://rwiki.repia.com/> -

. - 2023.12

Permanent link:
http://rwiki.repia.com/doku.php?id=wiki:javascript:javascript_note:js_object_constructors&rev=1620021870



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