

JavaScript Object Constructors

It is considered good practice to name constructor functions with an upper-case first letter.

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

<h2>JavaScript Object Constructors</h2>

<p id="demo"></p>

<script>
// 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
var myFather = new Person("John", "Doe", 50, "blue");

// Display age
document.getElementById("demo").innerHTML =
"My father is " + myFather.age + ".";
</script>

</body>
</html>
```

Output:

JavaScript Object Constructors

My father is 50.

Object Types (Blueprints) (Classes)

The examples from the previous chapters are limited. They only create single objects.

Sometimes we need a "**blueprint**" for creating many objects of the same "type".

The way to create an "object type", is to use an **object constructor function**.

In the example above, `function Person()` is an object constructor function.

Objects of the same type are created by calling the constructor function with the `new` keyword:

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

<h2>JavaScript Object Constructors</h2>

<p id="demo"></p>

<script>

// 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
var myFather = new Person("John", "Doe", 50, "blue");
var myMother = new Person("Sally", "Rally", 48, "green");

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

</script>

</body>
</html>
```

Output:

JavaScript Object Constructors

My father is 50. My mother is 48.

Adding a Method to a Constructor

Your constructor function can also define methods:

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

<h2>JavaScript Object Constructors</h2>

<p id="demo"></p>

<script>

// 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
var myFather = new Person("John", "Doe", 50, "blue");

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

</script>

</body>
</html>
```

Output:

JavaScript Object Constructors

My father is John Doe