


JavaScript Objects

[< Previous](#)[Next >](#)

Real Life Objects, Properties, and Methods

In real life, a car is an **object**.

A car has **properties** like weight and color, and **methods** like start and stop:

Object	Properties	Methods
	<p>car.name = Fiat</p> <p>car.model = 500</p> <p>car.weight = 850kg</p> <p>car.color = white</p>	<p>car.start()</p> <p>car.drive()</p> <p>car.brake()</p> <p>car.stop()</p>

All cars have the same **properties**, but the property values differ from car to car.

All cars have the same **methods**, but the methods are performed at different times.

JavaScript Objects

You have already learned that JavaScript variables are containers for data values.

This code assigns a **simple value** (Fiat) to a **variable** named car:

```
var car = "Fiat";
```

[Try it Yourself »](#)



HTML

CSS

JAVASCRIPT

MORE ▾

REFERENCES ▾



This code assigns **many values** (Fiat, 500, white) to a **variable** named car:

```
var car = {type:"Fiat", model:"500", color:"white"};
```

Try it Yourself »

The values are written as **name:value** pairs (name and value separated by a colon).

JavaScript objects are containers for **named values**.

Object Properties

The name:values pairs (in JavaScript objects) are called **properties**.

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

Property	Property Value
firstName	John
lastName	Doe
age	50
eyeColor	blue

Object Methods

Methods are **actions** that can be performed on objects.

Methods are stored in properties as **function definitions**.

Property	Property Value
firstName	John
lastName	Doe
age	50



HTML

CSS

JAVASCRIPT

MORE ▾

REFERENCES ▾



```
fullName      function() {return this.firstName + " " + this.lastName;}
```

JavaScript objects are containers for named values called properties or methods.

Object Definition

You define (and create) a JavaScript object with an object literal:

Example

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

Try it Yourself »

Spaces and line breaks are not important. An object definition can span multiple lines:

Example

```
var person = {  
  firstName:"John",  
  lastName:"Doe",  
  age:50,  
  eyeColor:"blue"  
};
```

Try it Yourself »

Accessing Object Properties

You can access object properties in two ways:

```
objectName.propertyName
```

or

```
objectName["propertyName"]
```

Example1

[Try it Yourself »](#)

Example2

```
person["lastName"];
```

[Try it Yourself »](#)

Accessing Object Methods

You access an object method with the following syntax:

```
objectName.methodName()
```

Example

```
name = person.fullName();
```

[Try it Yourself »](#)

If you access a method **without ()**, it will return the **function definition**:

Example

```
name = person.fullName;
```

[Try it Yourself »](#)

A method is actually a function definition stored as a property value.

Do Not Declare Strings, Numbers, and Booleans as Objects!

When a JavaScript variable is declared with the keyword "new", the variable is created as an object:

```
var x = new String();           // Declares x as a String object
var y = new Number();           // Declares y as a Number object
```

[HTML](#)[CSS](#)[JAVASCRIPT](#)[MORE ▼](#)[REFERENCES ▼](#)

Avoid String, Number, and Boolean objects. They complicate your code and slow down execution speed.

You will learn more about objects later in this tutorial.

Test Yourself with Exercises!

[Exercise 1 »](#)[Exercise 2 »](#)[Exercise 3 »](#)[◀ Previous](#)[Next ▶](#)

COLOR PICKER



HOW TO