☰   🏠   HTML   CSS   JAVASCRIPT   MORE ▾   ◑   🌐   🔍

w3schools.com

THE WORLD'S LARGEST WEB DEVELOPER SITE

JavaScript Function Call

⟨ Previous   Next ⟩

Method Reuse

With the `call()` method, you can write a method that can be used on different objects.

All Functions are Methods

In JavaScript all functions are object methods.

If a function is not a method of a JavaScript object, it is a function of the global object (see previous chapter).

The example below creates an object with 3 properties, firstName, lastName, fullName.

Example

```
var person = {
  firstName:"John",
  lastName: "Doe",
  fullName: function () {
    return this.firstName + " " + this.lastName;
  }
}
person.fullName();   // Will return "John Doe"
```

Try it Yourself »

The **this** Keyword

In a function definition, `this` refers to the "owner" of the function.

In the example above, `this` is the **person object** that "owns" the **fullName** function.

In other words, **this.firstName** means the **firstName** property of **this object**.

Read more about the `this` keyword at JS this Keyword.

The JavaScript call() Method

The `call()` method is a predefined JavaScript method.

It can be used to invoke (call) a method with an owner object as an argument (parameter).

With `call()` , an object can use a method belonging to another object.

This example calls the **fullName** method of person, using it on **person1**:

Example

```
var person = {
  fullName: function() {
    return this.firstName + " " + this.lastName;
  }
}
var person1 = {
  firstName:"John",
  lastName: "Doe"
}
var person2 = {
  firstName:"Mary",
  lastName: "Doe"
}
person.fullName.call(person1);  // Will return "John Doe"
```

☰   ⌂   HTML   CSS   JAVASCRIPT   MORE ▾   ◑   🌐   🔍

This example calls the **fullName** method of person, using it on **person2**:

Example

```
var person = {
  fullName: function() {
    return this.firstName + " " + this.lastName;
  }
}
var person1 = {
  firstName:"John",
  lastName: "Doe"
}
var person2 = {
  firstName:"Mary",
  lastName: "Doe"
}
person.fullName.call(person2);  // Will return "Mary Doe"
```

Try it Yourself »

The call() Method with Arguments

The `call()` method can accept arguments:

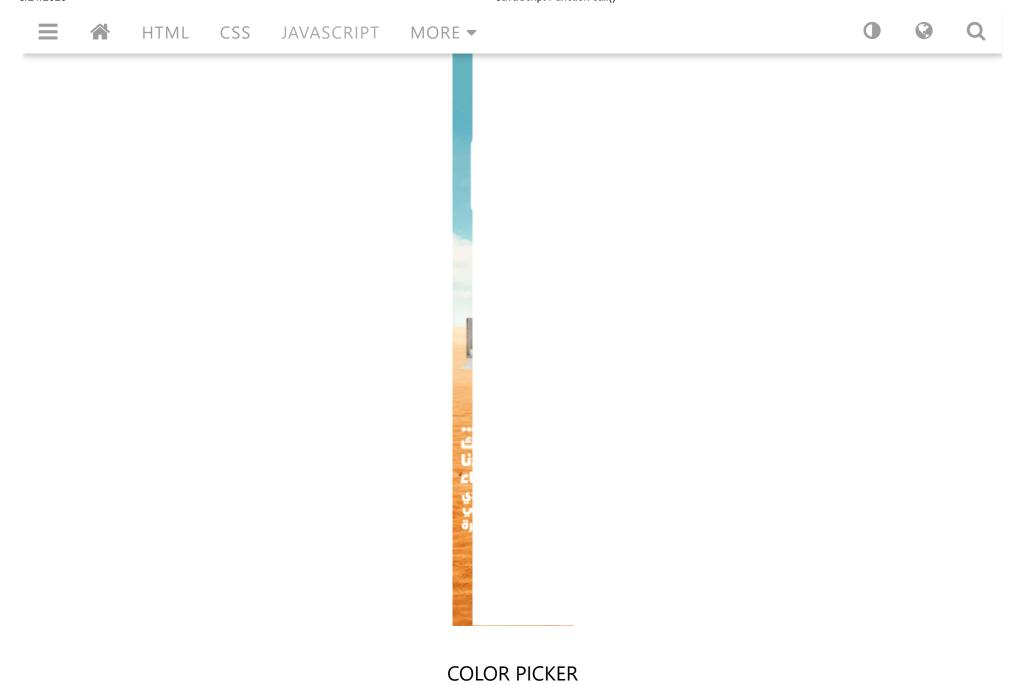```
tion(city, country) {
.firstName + " " + this.lastName + "," + city + "," + country;

nn",
e"

.call(person1, "Oslo", "Norway");
```

Next ›

HTML    CSS    JAVASCRIPT    MORE ▾

COLOR PICKER

## HOW TO

Tabs
Dropdowns
Accordions
Side Navigation
Top Navigation
Modal Boxes
Progress Bars
Parallax
Login Form
HTML Includes
Google Maps
Range Sliders
Tooltips
Slideshow
Filter List
Sort List

## SHARE

HTML
CSS
JavaScript
SQL
Python
PHP
jQuery
Bootstrap
XML

Read More »
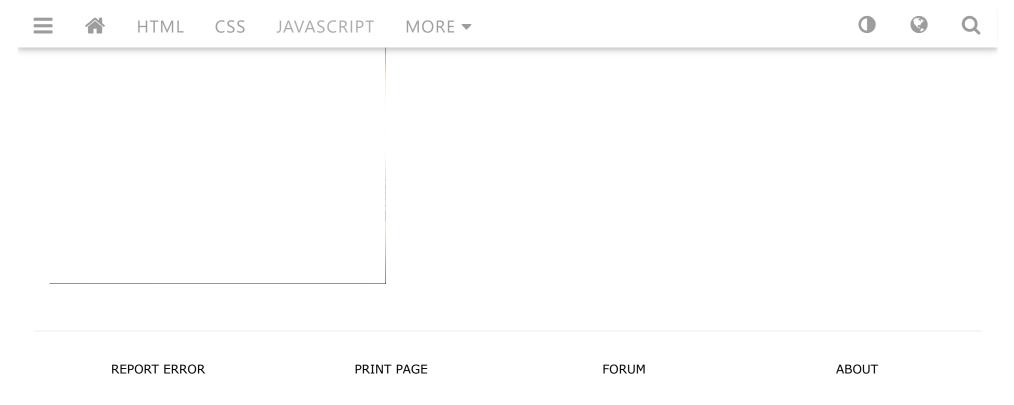
☰    🏠    HTML    CSS    JAVASCRIPT    MORE ▾                              ◐    🌐    🔍

REPORT ERROR                PRINT PAGE                FORUM                ABOUT
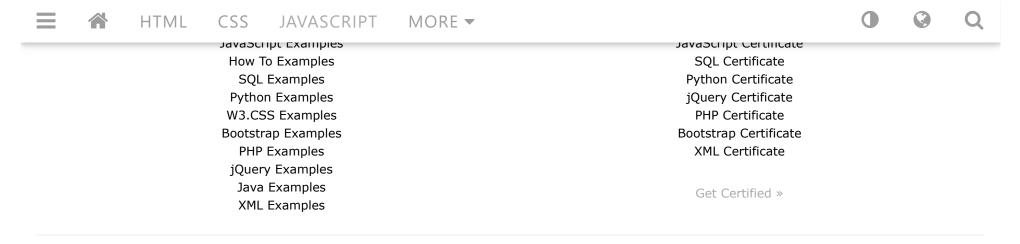
## Top Tutorials

HTML Tutorial
CSS Tutorial
JavaScript Tutorial
How To Tutorial
SQL Tutorial
Python Tutorial
W3.CSS Tutorial
Bootstrap Tutorial
PHP Tutorial
jQuery Tutorial
Java Tutorial
C++ Tutorial

## Top Examples

## Top References

HTML Reference
CSS Reference
JavaScript Reference
SQL Reference
Python Reference
W3.CSS Reference
Bootstrap Reference
PHP Reference
HTML Colors
jQuery Reference
Java Reference
Angular Reference

## Web Certificates

JavaScript Examples                                               JavaScript Certificate
How To Examples                                                    SQL Certificate
SQL Examples                                                      Python Certificate
Python Examples                                                    jQuery Certificate
W3.CSS Examples                                                    PHP Certificate
Bootstrap Examples                                                Bootstrap Certificate
PHP Examples                                                       XML Certificate
jQuery Examples
Java Examples                                                     Get Certified »
XML Examples

w3schools.com