

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

# JavaScript Object Properties

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

Properties are the most important part of any JavaScript object.

## JavaScript Properties

Properties are the values associated with a JavaScript object.

A JavaScript object is a collection of unordered properties.

Properties can usually be changed, added, and deleted, but some are read only.

## Accessing JavaScript Properties

The syntax for accessing the property of an object is:

```
objectName.property      // person.age
```

or

```
objectName["property"]   // person["age"]
```

or

```
objectName[expression]   // x = "age"; person[x]
```

The expression must evaluate to a property name.

## Example 1

```
person.firstname + " is " + person.age + " years old.";
```

Try it Yourself »

## Example 2

```
person["firstname"] + " is " + person["age"] + " years old.";
```

Try it Yourself »

# JavaScript for...in Loop

The JavaScript `for...in` statement loops through the properties of an object.

## Syntax

```
for (variable in object) {  
    // code to be executed  
}
```

The block of code inside of the `for...in` loop will be executed once for each property.

Looping through the properties of an object:

## Example

```
var person = {fname:"John", lname:"Doe", age:25};  
  
for (x in person) {  
    txt += person[x];  
}
```

[Try it Yourself »](#)

## Adding New Properties

You can add new properties to an existing object by simply giving it a value.

Assume that the person object already exists - you can then give it new properties:

### Example

```
person.nationality = "English";
```

[Try it Yourself »](#)

## Deleting Properties

The `delete` keyword deletes a property from an object:

### Example

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};  
delete person.age;    // or delete person["age"];
```

Try it Yourself »

The **delete** keyword deletes both the value of the property and the property itself.

After deletion, the property cannot be used before it is added back again.

The **delete** operator is designed to be used on object properties. It has no effect on variables or functions.

The **delete** operator should not be used on predefined JavaScript object properties. It can crash your application.

---

## Property Attributes

All properties have a name. In addition they also have a value.

The value is one of the property's attributes.

Other attributes are: enumerable, configurable, and writable.

These attributes define how the property can be accessed (is it readable?, is it writable?)

In JavaScript, all attributes can be read, but only the value attribute can be changed (and only if the property is writable).

( ECMAScript 5 has methods for both getting and setting all property attributes)

---

## Prototype Properties

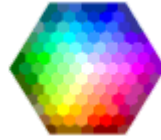
JavaScript objects inherit the properties of their prototype.

The **delete** keyword does not delete inherited properties, but if you delete a prototype property, it will affect all objects inherited from the prototype.

◀ Previous

Next ▶

## 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



## CERTIFICATES

HTML

CSS

JavaScript

SQL

Python

PHP

jQuery

Bootstrap

XML

[Read More »](#)







---

[REPORT ERROR](#)

[PRINT PAGE](#)

[FORUM](#)

[ABOUT](#)

---

[Top Tutorials](#)

[Top References](#)

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

## Top Examples

- HTML Examples
- CSS Examples
- JavaScript Examples
- How To Examples
- SQL Examples
- Python Examples
- W3.CSS Examples
- Bootstrap Examples
- PHP Examples
- jQuery Examples
- Java Examples
- XML Examples

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

## Web Certificates

- HTML Certificate
- CSS Certificate
- JavaScript Certificate
- SQL Certificate
- Python Certificate
- jQuery Certificate
- PHP Certificate
- Bootstrap Certificate
- XML Certificate

[Get Certified »](#)

---

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our terms of use, cookie and privacy policy. Copyright 1999-2019 by Refsnes Data. All Rights Reserved.

Powered by W3.CSS.

