

Fullscreen API

Quick Guides for Masterminds

J.D Gauchat

www.jdgauchat.com

Cover Illustration by **Patrice Garden**

www.smartcreativz.com

Quick Guides for Masterminds
Copyright © 2018 by John D Gauchat
All Rights Reserved

No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system without the prior written permission of the copyright owner.

Companies, services, or product names used in this eBook are for identification purposes only. All trademarks and registered trademarks are the property of their respective owners.

The content of this guide is a collection of excerpts from the book HTML5 for Masterminds. For more information, visit www.formasterminds.com.

The information in this eBook is distributed on an "as is" basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author nor the publisher shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this eBook is available at www.formasterminds.com

Copyright Registration Number: 1140725

1st Edition 2018

What is Inside

This guide will teach you how to use the Fullscreen API to take your website or web application full screen. After reading this guide, you will know how to take an element full screen, how to detect if an element is full screen, and how to style the elements when they are in full screen mode.

About this Guide

This guide is a collection of excerpts from the book [HTML5 for Masterminds](#). The information included in this guide will help you understand a particular aspect of web development, but it will not teach you everything you need to know to develop a website or a web application. If you need a complete course on web development, read our book [HTML5 for Masterminds](#). For more information, visit our website at www.formasterminds.com.

What Do You Need

This guide assumes that you have a basic knowledge of HTML, CSS and JavaScript, and you know how to create files and upload them to a server. If you don't know how to program in HTML, CSS or JavaScript, you can download our guides [Introduction to HTML](#), [Introduction to CSS](#), and [Introduction to JavaScript](#). For a complete course on web development, read our book [HTML5 for Masterminds](#).

IMPORTANT: We recommend you to execute the examples in this guide with the latest versions of Google Chrome and Mozilla Firefox (www.google.com/chrome and www.mozilla.com). You can also check the state of the current implementations at www.canIuse.com. To find examples, resources, links and videos, visit our website at www.formasterminds.com.

The Basics: To create the files for the examples of this guide, you can use a text editor (Notepad or Text Edit), or a professional editor like Atom (www.atom.io). If you do not have a server to test the files, you can install a server in your computer with a package like MAMP (www.mamp.info). For more information, read our free guide [Web Development](#).

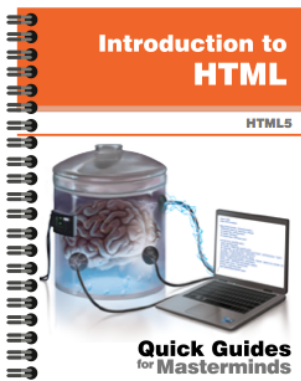
Recommendations



Introduction to JavaScript Quick Guides for Masterminds

This guide will teach you how to program with JavaScript. After reading this guide, you will know how to create a program in JavaScript, how to define functions and objects, and how to read and modify an HTML document dynamically.

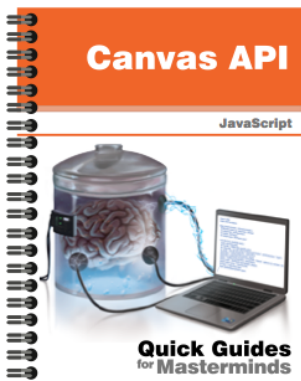
[More Information](#)



Introduction to HTML Quick Guides for Masterminds

This guide will teach you how to create your website's documents with HTML. After reading this guide, you will know how to work with HTML elements, how to define a document's structure, and how to organize its content.

[More Information](#)



Canvas API Quick Guides for Masterminds

This guide will teach you how to create graphic applications for your website with the Canvas API. After reading this guide, you will know how to create graphics for the web, how to manipulate images, and how to generate animations and small video games.

[More Information](#)

More Guides Available at www.formasterminds.com

Table of Contents

FULLSCREEN API

Modern Applications

Full Screen

Full Screen Styles

QUICK REFERENCE

Properties

Methods

Events

Fullscreen API

Modern Applications

The Web is a multipurpose and multimedia platform where everything is possible. With so many interesting new applications, the word *navigation* has almost lost significance. The browser's interface is not only unnecessary but sometimes may get in the way. For this reason, browsers include the Fullscreen API.

Full Screen

The Fullscreen API lets us expand any element in the document to occupy the entire screen. As a result, the browser's interface is hidden in the background, allowing the user to focus his or her attention on our video, picture, application or video game.

The API provides a few properties, methods and events to take an element to full screen mode, exit the full screen mode, and retrieve some information from the element and the document that are participating in the process.

fullscreenElement—This property returns a reference to the element that is in full screen mode. If no element is full screen, the property returns **null**.

fullscreenEnabled—This Boolean property returns **true** when the document is able to go full screen or **false** otherwise.

requestFullscreen()—This method is available for every element in the document. It activates full screen mode for the element.

exitFullscreen()—This method is available for the document. If an element is in full screen mode, the method cancels the mode and returns the focus back to the browser window.

The API also offers the following events.

fullscreenchange—This event is fired by the document when an element enters or exits the full screen mode.

fullscreenerror—This event is fired by the element in case of failure (the full screen mode is not available for that element or for the document).

The **requestFullscreen()** method and the **fullscreenerror** event are associated with the elements in the document, but the rest of the properties, methods, and events are part of the **Document** object, and therefore they are accessible from the **document** property.

Listing 1: Taking a `<video>` element full screen

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Full Screen</title>
  <script>
    var video;
    function initiate() {
      video = document.getElementById("media");
      video.addEventListener("click", gofullscreen);
    }
    function gofullscreen() {
      if (!document.webkitFullscreenElement) {
        video.webkitRequestFullscreen();
        video.play();
      }
    }
    window.addEventListener("load", initiate);
  </script>
</head>
<body>
  <section>
    <video id="media" width="720" height="400"
poster="poster.jpg">
      <source src="trailer.mp4">
      <source src="trailer.ogg">
    </video>
  </section>
</body>
</html>
```

In the code in Listing 1, we add a listener for the **load** event to the window to execute a function called **initiate()** as soon as the document is loaded. When the event is fired, this function adds a listener for the **click** event to the **<video>** element. In consequence, the **gofullscreen()** function, set as the listener of this event, is executed every time the user clicks on the video. In this function, we use the **fullscreenElement** property to detect whether an element is already in full screen mode or not, and if not, the video is made full screen by the **requestFullscreen()** method. At the same time, the video is played with the **play()** method.

IMPORTANT: This is an experimental API. The properties, methods, and events have been prefixed by browsers until the final specification is implemented. In the

case of Google Chrome, instead of the original names we have to use **webkitRequestFullscreen()**, **webkitExitFullscreen()**, **webkitfullscreenchange**, **webkitfullscreenerror** and **webkitFullscreenElement**. For Mozilla Firefox, the names are **mozRequestFullScreen()**, **mozCancelFullScreen()**, **mozfullscreenchange**, **mozfullscreenerror** and **mozFullscreenElement**.

Full Screen Styles

Browsers adjust the dimensions of the **<video>** element to the size of the screen automatically, but for any other element, the original dimensions are preserved, and the free space on the screen is filled with a black background. For this reason, CSS includes a pseudo-class called **:full-screen** to modify the styles of the element when it expands to full screen.

IMPORTANT: The last specification declares the name of this pseudo-class without the hyphen (**:fullscreen**), but, at the time of writing, browsers like Mozilla Firefox and Google Chrome have only implemented the previous specification and work with the name mentioned in this guide (**:full-screen**). It also has to be prefixed to work properly (**:-webkit-full-screen**, **:-moz-full-screen**, etc.).

Listing 2: Taking any element full screen

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Full Screen</title>
  <style>
    #player:-webkit-full-screen, #player:-webkit-full-
screen #media{
      width: 100%;
      height: 100%;
    }
  </style>
<script>
  var video, player;
  function initiate() {
    video = document.getElementById("media");
    player = document.getElementById("player");
    player.addEventListener("click", gofullscreen);
  }
  function gofullscreen() {
    if (!document.webkitFullscreenElement) {
      player.webkitRequestFullscreen();
    }
  }
</script>
```



```

        video.play();
    } else {
        document.webkitExitFullscreen();
        video.pause();
    }
}
window.addEventListener("load", initiate);
</script>
</head>
<body>
    <section id="player">
        <video id="media" width="720" height="400"
poster="poster.jpg">
            <source src="trailer.mp4">
            <source src="trailer.ogg">
        </video>
    </section>
</body>
</html>

```

In Listing 2, we take the **<section>** element and its content full screen. The **gofullscreen()** function is modified to activate and deactivate the full screen mode for this element. As in the previous example, the video is played when it is in full screen mode, but now it will be paused when the mode is canceled.

These improvements are insufficient to make our player full screen. In full screen mode, the new container for the element is the screen, but the original dimensions and styles for the **<section>** and **<video>** elements are preserved intact. Using the **:full-screen** pseudo-class, we change the values of the **width** and **height** properties of these elements to 100%, matching the dimensions of the container. Now the elements occupy the whole screen and truly reflect the full screen mode.

Do It Yourself: Create a new HTML file to test the examples in this guide. Once the document is opened in the browser, click on the video to activate the full screen mode and click again to deactivate it.

Quick Reference

Properties

fullscreenElement—This property returns a reference to the element that is in full screen mode. If no element is full screen, the property returns **null**.

fullscreenEnabled—This Boolean property returns **true** when the document is able to go full screen or **false** otherwise.

Methods

requestFullscreen()—This method is available for every element in the document. It activates full screen mode for the element.

exitFullscreen()—This method is available for the document. If an element is in full screen mode, the method cancels the mode and returns the focus back to the browser window.

Events

fullscreenchange—This event is fired by the document when an element enters or exits the full screen mode.

fullscreenerror—This event is fired by the element in case of failure (the full screen mode is not available for that element or for the document).

For Masterminds

Book Series

for more Books and Quick Guides visit

www.formasterminds.com