



APRENDERAPROGRAMAR.COM

DEPURAR JAVASCRIPT.  
CONSOLE.LOG. DEBUG DE  
ERRORES CON FIREFOX  
(FIREBUG), CHROME,  
INTERNET EXPLORER.  
IDES. (CU01151E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

**Resumen:** Entrega nº51 del Tutorial básico "JavaScript desde cero".

Autor: César Krall

## DEPURAR JAVASCRIPT

Si vienes siguiendo el curso de JavaScript y has ido realizando los ejercicios con seguridad te habrás enfrentado a problemas a la hora de ejecutar los scripts relacionados con errores en el código. Estos errores pueden dar lugar a que no se ejecute lo esperado, o simplemente que no se ejecute nada. La dificultad radica en que no tenemos ningún mensaje de aviso del error.



Por tratarse de un lenguaje interpretado y embebido JavaScript resulta un lenguaje relativamente difícil de depurar. No hay un método único o preferido por la comunidad de programadores, por ello hablaremos de alternativas o posibilidades para depurar JavaScript.

Agruparemos en primer lugar las alternativas en tres grandes grupos que vamos a citar superficialmente para luego centrarnos en una de estas alternativas:

a) **Herramientas de validación** del código (validadores) on-line o instaladas en un pc. Estas herramientas, en líneas generales, reciben un código JavaScript como entrada y devuelven una lista de los errores o problemas detectados en el código. Estos problemas pueden ser de sintaxis o de otro tipo. Como herramientas on-line podemos citar JSLint (<http://www.jshint.com/>) y JSHint (<http://www.jshint.com/>). Entre las herramientas para instalar en el computador de desarrollo tenemos Google Closure Linter (<https://developers.google.com/closure/utilities/>) y JavaScript Lint (<http://www.javascriptlint.com/>).

b) **IDEs para desarrollo web**. En general se trata de entornos de desarrollo que proveen algunas herramientas o extensiones capaces de detectar errores en el código ó al menos facilitar ayuda que debería dar lugar a que existan menos errores en el código. Son en general herramientas que proveen muchas funcionalidades y por tanto más pesadas que un simple editor de texto. Podemos citar dentro de este grupo Eclipse, IDE que permite la instalación de plugins entre los que cabe destacar JSDT (JavaScript Developer Tools), Komodo IDE, NetBeans, Microsoft Visual Studio con Visual Web Developer, WebStorm y Aptana Studio.

c) **Herramientas nativas** o que se integran en los navegadores web y que permiten el análisis del código y depuración de errores. Mozilla Firefox, es uno de los navegadores que ha sido tradicionalmente preferido por los programadores por las herramientas de análisis y depuración que provee y por atenerse a los estándares. De cara a la depuración de código JavaScript dispone de herramientas nativas (Developer tools, propia del explorador) como la consola y de diferentes plugins para desarrolladores entre los que destaca Firebug. El navegador Chrome de Google provee herramientas nativas en lo que se denomina Chrome Developer Tools. Internet Explorer de Microsoft ofrece también sus Developer Tools. En la mayor parte de los navegadores se accede a las developer tools pulsando F12.

## ¿QUÉ ALTERNATIVA ELEGIR?

Dado que existen numerosas alternativas la pregunta que nos planteamos es: ¿qué alternativa elegir? Para el desarrollo web profesional posiblemente debiéramos decantarnos por un IDE y conocer sus herramientas de depuración nativas o en forma de extensiones. Pero en este curso, donde nos centramos en comprender JavaScript y no en la productividad como programadores, vamos a recomendar que utilices las herramientas de desarrollador (developer tools) de tu navegador web, principalmente con el fin de tener ayuda a la hora de localizar errores en nuestro código JavaScript. Esta es una vía relativamente cómoda, rápida y sin necesidad de instalación de software adicional, para localizar errores en nuestro código.

¿Qué navegador ofrece mejores posibilidades de cara a la depuración? Preferimos no opinar al respecto. Firefox ha tenido siempre muy buena prensa entre los programadores e Internet Explorer al revés. Pero el mundo del desarrollo de aplicaciones web y apps para smartphones se mueve a gran velocidad y las opiniones que eran válidas hace unos años pueden no serlo en la actualidad.

Te recomendamos que optes por las herramientas de depuración de tu propio navegador web (el que vengas utilizando), y si consideras que no te responde adecuadamente, simplemente prueba otro.

## UN CÓDIGO DE EJEMPLO

A modo de ejemplo consideraremos que hemos desarrollado un código JavaScript pero al tratar de ejecutarlo no hay respuesta alguna. Entendemos (suponemos) que hay un error y queremos que una herramienta de depuración nos facilite su localización.

El código que usaremos es este:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">
<script type="text/javascript">
function saludar() { alert ('Hola, soy ' + this.nombre); }
function ejemploObjetos() {
var unObjeto = new Array();
unObjeto.saludar();
}
</script>
</head>
<body><div id="cabecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos JavaScript</h3></div>
<div style="color:blue;" id="pulsador" onclick="ejemploObjetos()"> Probar </div>
</body>
</html>
```

## CACHÉ DE NAVEGADOR

Un problema que aparece en ocasiones es que el navegador guarda información en caché de modo que no se ejecuta el código que hemos modificado, sino código guardado en la caché del navegador. Normalmente este problema se soluciona haciendo una recarga completa pulsando CTRL + F5 (teclas ctrl y F5 al mismo tiempo).

## DESPLEGAR LA CONSOLA DEL NAVEGADOR

La mayoría de los navegadores permiten desplegar una consola o ventana auxiliar en la parte inferior del navegador, que sirve para fines de desarrollo de código y mostrar mensajes de error. Para activarla en tu navegador pulsa F12 (esto funciona en la mayor parte de los navegadores modernos, pero puede no estar disponible en algunos).

Una vez desplegada la consola, carga el archivo html con el código que contiene el error y pulsa en "Probar".



Por consola se muestra un mensaje como: `TypeError: unObjeto.saludar is not a function ejemplo1.html:7`

Este mensaje nos indica el tipo de error (se trata de invocar una función que no existe) y la línea donde se ha detectado el error dentro del archivo (línea 7).

El mensaje de error puede diferir según el navegador que estemos utilizando, por ejemplo otro navegador puede mostrar `<<Uncaught TypeError: undefined is not a function ejemplo1.html:7>>`

Este sencillo método para detectar y localizar errores te será muy útil para detectar y solucionar errores cuando estés creando código JavaScript.

Si tu navegador no te permite desplegar consola te recomendamos que lo actualices (puede que sea demasiado antiguo) o que te cambies de navegador.

## CONSOLE.LOG

La consola tiene otra utilidad clave: permite mostrar mensajes informativos introducidos en el código pero no visibles como contenido de la página web.

Para mostrar mensajes a través de la consola usaremos la sintaxis: `console.log(argumento)`, donde argumento es aquello que debe ser mostrado por consola (puede ser un texto, una variable, un objeto, etc.).

Por ejemplo:

```
var unObjeto = new Array();  
console.log(unObjeto);
```

Nos devuelve a través de la consola (que tendremos que haber activado previamente pulsando F12) Array [ ] y la línea en que se genera el mensaje.

Console.log tiene un funcionamiento análogo a alert pero una ventaja importante respecto a esta función: por un lado, si olvidas en el código un console.log las repercusiones serán menores puesto que no será visible a menos que se active la consola, y los usuarios no navegan con la consola activada. Por otro lado, permite mostrar mensajes sin detener la ejecución del script, lo cual nos ahorra tiempo y facilita el trabajo.

## FIREBUG: EXTENSIÓN DE FIREFOX

Debido a que es una herramienta con una larga tradición entre los desarrolladores web, merece la pena dedicarle unas líneas a la extensión FireBug de Mozilla Firefox.

Firebug es una extensión del navegador Firefox que nos permite editar webs y hojas de estilo, monitorizar tiempos de carga, depurar javascript y ver los errores y procesos de carga en la página además de explorar el DOM.

Firebug dispone de su propia consola donde igualmente podemos visualizar los mensajes de error. Si utilizas Firefox y estás interesado en conocer e instalar esta herramienta te recomendamos que leas el artículo [Firebug, una extensión gratuita de Firefox para programadores y diseñadores web](#) en la sección de Herramientas Informáticas de aprenderaprogramar.com.



## RESUMEN

Existen múltiples vías para depurar código JavaScript y puedes optar por aquellas que mejor se adapten a tí. Como alternativa rápida y sencilla, te recomendamos que cuando estés probando código trabajes con la consola de tu navegador desplegada y que te valgas de console.log para mostrar mensajes que no detengan la ejecución de los scripts.

## EJERCICIO

Ejecuta este script con la consola de tu navegador activada. ¿Qué mensajes aparecen en la consola antes y después de pulsar <<Probar>>? Corrige el error que contiene el código. ¿Qué mensajes aparecen ahora en la consola? ¿Qué interpretación podemos darle a los resultados que hemos obtenido?

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">
<script type="text/javascript">
function ejemploObjetos() {
var unObjeto = new Array();
console.log('unObjeto es' + unObjeto);
console.log(unObjeto);
console.log('this es ' + this);
alert('Hello');
}
</script>
</head>
<body><div id="cabecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos JavaScript</h3></div>
<div style="color:blue;" id="pulsador" onclick="ejemploObjetos()"> Probar </div>
</body>
</html>
```

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros [aprenderaprogramar.com](http://aprenderaprogramar.com).

**Próxima entrega:** CU01152E

**Acceso al curso completo** en [aprenderaprogramar.com](http://aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:  
[http://aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=78&Itemid=206](http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=78&Itemid=206)