



SEGUNDA EDICIÓN

Carlos Azaustre

# Aprendiendo JavaScript

Desde cero hasta ECMAScript 6+



# Aprendiendo JavaScript

Aprende las bases del lenguaje web más demandado.  
Desde cero hasta ECMAScript 6+.

Carlos Azaustre

Este libro está a la venta en <http://leanpub.com/aprendiendo-javascript>

Esta versión se publicó en 2021-01-26



Éste es un libro de [Leanpub](#). Leanpub anima a los autores y publicadoras con el proceso de publicación. [Lean Publishing](#) es el acto de publicar un libro en progreso usando herramientas sencillas y muchas iteraciones para obtener retroalimentación del lector hasta conseguir el libro adecuado.

© 2015 - 2021 Carlos Azaustre

# ¡Tuitea sobre el libro!

Por favor ayuda a Carlos Azaustre hablando sobre el libro en [Twitter](#)!

El tuit sugerido para este libro es:

[Acabo de leer el ebook "Aprendiendo JavaScript" de @carlosazaustre](#)

El hashtag sugerido para este libro es [#aprendiendojavascript](#).

Descubre lo que otra gente dice sobre el libro haciendo clic en este enlace para buscar el hashtag en Twitter:

[#aprendiendojavascript](#)

*A Paola, Aroa y Elon.*

# Índice general

<b>Introducción</b> . . . . .	<b>1</b>
<b>Breve historia de JavaScript</b> . . . . .	<b>2</b>
Orígenes . . . . .	2
Node.js . . . . .	4
<b>Tipos de variables</b> . . . . .	<b>7</b>
Definición . . . . .	7
Tipos . . . . .	8

# Introducción

JavaScript es el lenguaje de la web. Si necesitas programar en un navegador web, necesitas JavaScript. Bien es cierto que puedes utilizar otros lenguajes, como Dart, pero el estándar es JavaScript.

Gracias a él tenemos aplicaciones como Gmail, o Twitter, que son fuertemente dinámicas y hacen que la experiencia de uso sea mucho mejor que antaño, cuando las páginas web tenían que recargarse cada vez que realizábamos una acción.

Es un lenguaje muy demandado en la industria hoy en día, ya que además de utilizarse en el navegador, también puede usarse en el lado del servidor (Node.js). Con la multitud de frameworks que existen pueden crearse *Single Page Applications* que emulan la experiencia de una aplicación móvil en el navegador. También pueden crearse aplicaciones híbridas con herramientas como Ionic y Cordova. ¿Has oído hablar del desarrollo basado en componentes? Te sonarán entonces Polymer y/o React. Con React Native puedes crear aplicaciones nativas para iOS y Android con únicamente JavaScript. ¿Aplicaciones Isomórficas? Hoy en día todo es posible con JavaScript.

Si es la primera vez que te incursas en el mundo web, te puede resultar abrumadora la cantidad de herramientas, preprocesadores, frameworks, etc.. Pero siempre que empezamos, cometemos el mismo error. Aprendemos la herramienta antes que el lenguaje.

Por eso me he animado a escribir este ebook que estás leyendo ahora mismo. Para enseñarte desde las bases hasta las más recientes novedades y patrones de diseño utilizando JavaScript puro (también llamado *Vanilla JS*). Una vez conoces las bases del lenguaje, ya puedes adentrarte en cualquier herramienta del mundo web.

A partir de que fue aprobada la versión 6 de ECMAScript, se reemplazó la anterior (v5.1) que habíamos usado varios años. Esta nueva versión trajo muchas novedades y cambios al lenguaje. En este ebook no he querido dejarlo de lado y hablo de cual es el equivalente en código entre la versión antigua (5.1) y la presente.

Espero que disfrutes del ebook tanto como yo lo he hecho escribiéndolo y actualizándolo para ti y te sirva para tu carrera profesional. Puedes encontrarme en las siguientes redes:

- [En Youtube](#)<sup>1</sup>
- [En Twitch](#)<sup>2</sup>
- [En Instagram](#)<sup>3</sup>
- [Y en mi Web/Blog](#)<sup>4</sup>

Sin más, te dejo con el ebook. ¡Disfruta y aprende!

---

<sup>1</sup>[https://youtube.com/carlosazaustre?sub\\_confirmation=1](https://youtube.com/carlosazaustre?sub_confirmation=1)

<sup>2</sup><https://twitch.tv/carlosazaustre>

<sup>3</sup><https://instagram.com/carlosazaustre>

<sup>4</sup><https://carlosazaustre.es>

# Breve historia de JavaScript

Antes de empezar con las particularidades del lenguaje, es conveniente conocer un poco de historia. De dónde viene JavaScript y cómo ha crecido su popularidad en los últimos años. ¡Prometo ser rápido y pasar cuanto antes al código!

## Orígenes

En 1995, Brendan Eich (ex-CEO de Mozilla) desarrolló lo que sería la primera versión de JavaScript para el navegador Netscape Navigator. En aquel momento se llamó Mocha y después fue renombrado a LiveScript. El nombre de JavaScript se le dió debido a que Netscape añadió compatibilidad con Java en su navegador y era una tecnología muy popular en aquel momento. Además Netscape fue adquirida por Sun Microsystems, propietaria de la marca Java. Esto supone que hoy en día haya una pequeña confusión y mucha gente confunda Java con JavaScript o lo considere una extensión del lenguaje, pero no es cierto, hay que aclarar que Java y JavaScript no tienen nada que ver.



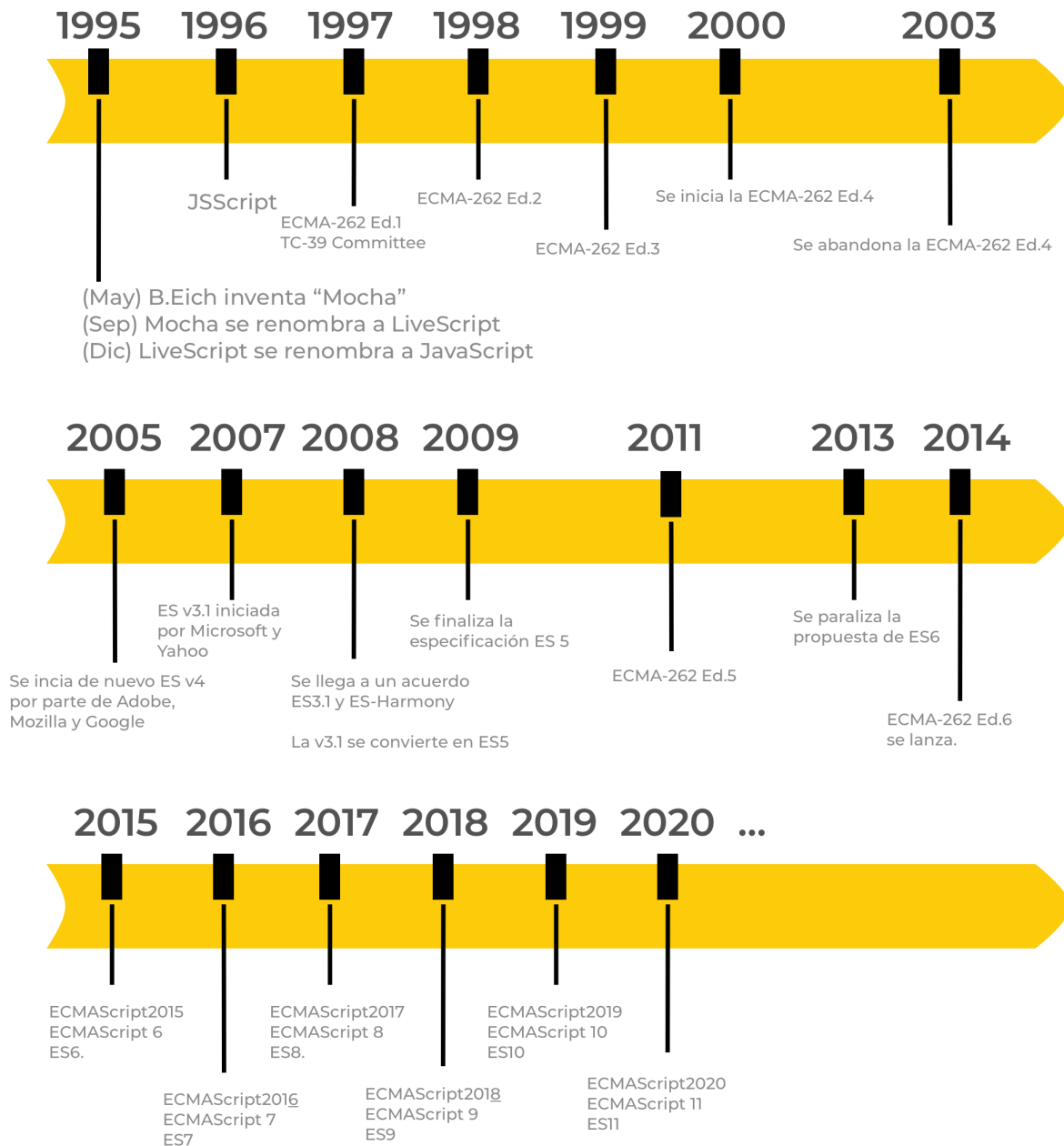
TC-39

En 1997 se crea un comité (llamado [TC39](https://tc39.es/)<sup>5</sup>) para crear un estándar de JavaScript por la *European Computer Manufacturers Association*, *ECMA*. En ese comité se diseña el estándar del DOM, Document Object Model para, de esta manera, evitar incompatibilidades entre los navegadores. Es a partir de entonces cuando los estándares de JavaScript se rigen por [ECMAScript](https://tc39.es/ecma262/)<sup>6</sup>.

---

<sup>5</sup><https://tc39.es/>

<sup>6</sup><https://tc39.es/ecma262/>



#### cronologia de javascript

En 1999 se estandariza la versión 3 de JavaScript que se mantuvo vigente hasta hace relativamente poco. Hubo algunos intentos de lanzar una versión 4, pero la que finalmente se estandarizó y sigue hasta el momento es la versión 5 de ECMAScript, aprobada en 2011.



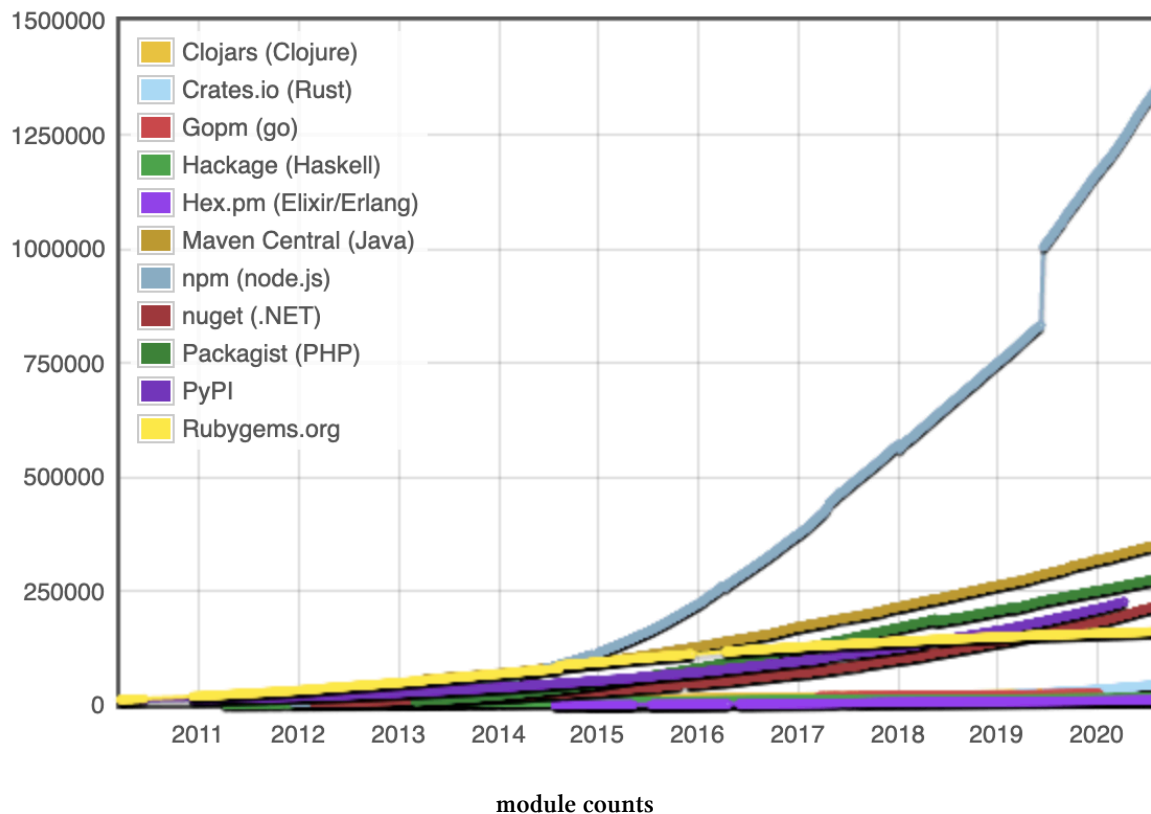
En Junio de 2013 el borrador de la versión 6 se quedó parado, pero en diciembre de 2014 finalmente fue aprobado y se estandarizó en julio de 2015.

JavaScript fue diseñado para añadir efectos y animaciones a los sitios web , pero ha ido evolucionando mucho a lo largo de los años, convirtiéndose en un lenguaje multipropósito. Es a partir de 2005, con la llegada de GMail y su uso de la tecnología AJAX, *Asynchronous JavaScript And XML* (gracias al objeto *XMLHttpRequest* creado por Microsoft para Internet Explorer 5.0), lo que lanzó su popularidad.

## Node.js

En 2009, Ryan Dahl creó Node.js. Node es un entorno de ejecución para JavaScript en el servidor a partir del motor V8 de renderizado de JavaScript que utiliza el navegador Chrome de Google. Node facilita la creación de aplicaciones de servidor altamente escalables. Hoy en día es muy popular para el desarrollo de Microservicios, APIs, aplicaciones web Full-stack, isomórficas, etc... Su comunidad es muy grande, y su sistema de paquetes y librerías NPM, *Node Package Manager*, (Aunque hoy en día ya no solo engloba paquetes de Node, también para JavaScript del lado cliente) está cerca del millón y medio de módulos, convirtiéndolo en el más grande de todos por delante de Java, Ruby, PHP, etc...

## Module Counts



Fuente: [Module Counts](https://www.modulecounts.com/)<sup>7</sup>

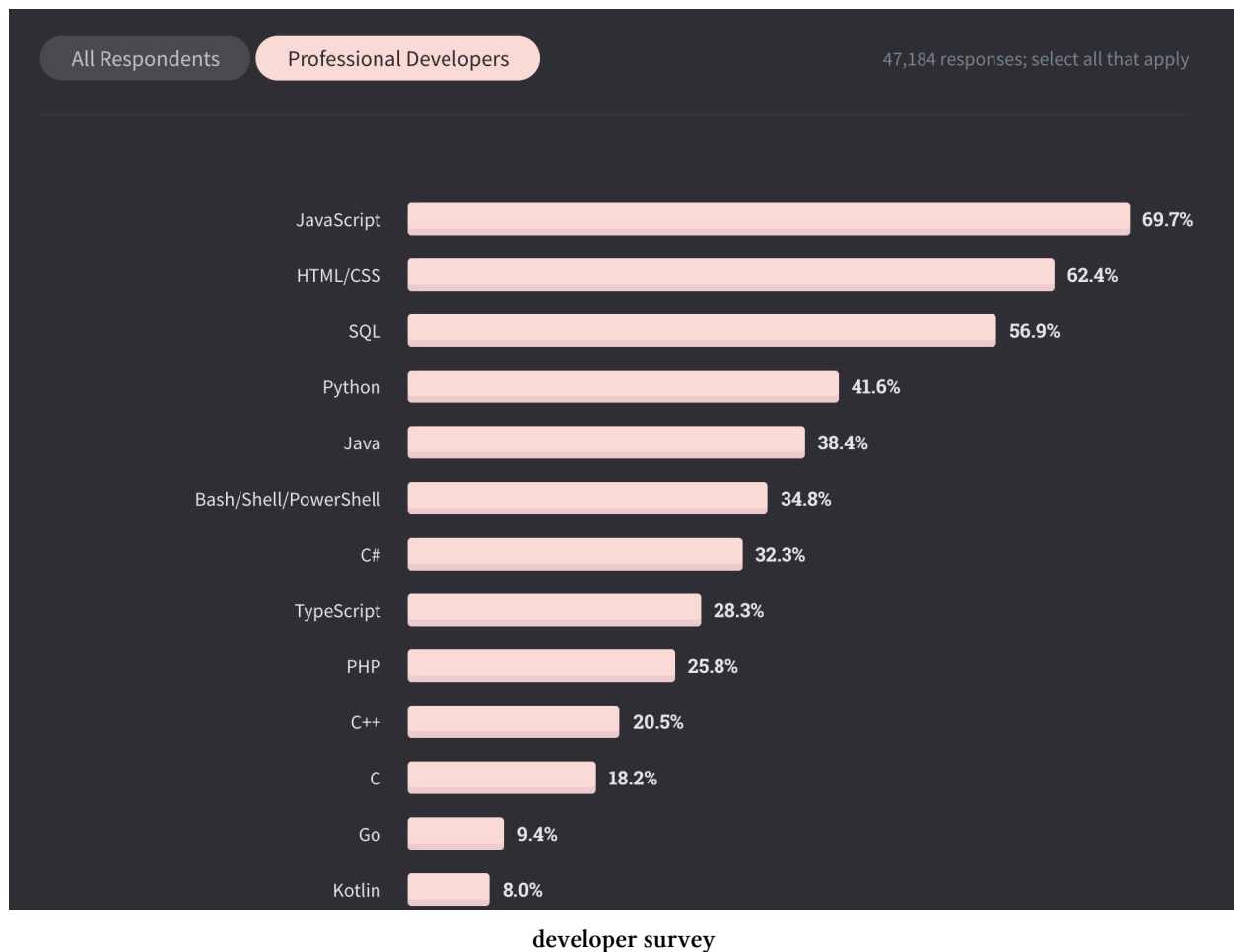
Hoy en día JavaScript se utiliza en muchos sitios, *Frontend*, *Backend*, *Server Side Rendering*, micro-controladores, *Internet of Things*, *wearables*, etc... Convirtiéndole en el lenguaje de programación del presente.

La última encuesta realizada por [StackOverflow](https://insights.stackoverflow.com/survey/2020) dirigida a personas dedicadas a la programación<sup>8</sup> refleja que JavaScript sigue siendo el lenguaje más utilizado con diferencia<sup>9</sup>.

<sup>7</sup><http://www.modulecounts.com/>

<sup>8</sup><https://insights.stackoverflow.com/survey/2020>

<sup>9</sup><https://insights.stackoverflow.com/survey/2020#technology-programming-scripting-and-markup-languages>



Fuente: [Stackoverflow 2020 Developer Survey](https://insights.stackoverflow.com/survey/2020)<sup>10</sup>

Toda la documentación y referencia sobre JavaScript se puede encontrar en el [sitio web de desarrolladores de Mozilla](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia)<sup>11</sup>, muy recomendable de visitar cuando se tienen dudas sobre cómo se usa o implementa una función u objeto determinado.

<sup>10</sup><https://insights.stackoverflow.com/survey/2020>

<sup>11</sup><https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia>

# Tipos de variables

JavaScript es un lenguaje débilmente tipado. Esto quiere decir que no indicamos de qué tipo es cada variable que declaramos. Todas las variables admiten todos los tipos, y pueden ser reescritas. Es una de las cosas buenas y malas que tiene JavaScript.

## Definición

Las variables son espacios de memoria donde almacenamos temporalmente datos desde los que podemos acceder en cualquier momento de la ejecución de nuestros programas. Tienen varios tipos y clases que veremos a continuación.

Para definir una variable en JavaScript, utilizamos la palabra reservada `let` y le damos un nombre, por ejemplo:

```
1 let miDato;
```

También le podemos asignar un valor en la misma línea que la declaramos, por ejemplo, a continuación a la variable `dato` le asignamos el valor 5 :

```
1 let dato = 5;
```

O podemos primero declarar la variable y más adelante, en otra línea, asignarle un valor:

```
1 let dato;  
2 dato = 5;
```

Debemos intentar que los nombres de las variables sean lo más descriptivos posibles, de manera que con solo leerlos sepamos que contienen y así nuestros desarrollos serán más ágiles.

Los nombres de las variables siempre han de comenzar por una letra, el símbolo `$` o `_`, nunca pueden comenzar por números u otros caracteres especiales. JavaScript también distingue entre mayúsculas o minúsculas, por tanto no es lo mismo `miDato` que `MiDato` o `miDato`, para JavaScript son nombres diferentes y las tratará de manera diferente.

# Tipos

JavaScript tiene 4 tipos primitivos de datos para almacenar en variables. Estos son:

- number
- boolean
- string
- undefined

## number

Sirve para almacenar valores numéricos. Son utilizados para contar, hacer cálculos y comparaciones. Estos son algunos ejemplos:

```
1 let miEntero = 1;  
2 let miDecimal = 1.33;
```

## boolean

Este tipo de dato almacena un bit que indica `true` o `false`. Los valores booleanos se utilizan para indicar estados. Por ejemplo, asignamos a una variable el estado `false` al inicio de una operación, y al finalizarla lo cambiamos a `true`. Después realizamos la comprobación necesaria.

```
1 let si = true;  
2 let no = false;
```

## string

Las variables de tipo `string` almacenan caracteres o palabras. Se delimitan entre comillas simples o dobles. Ejemplo:

```
1 let dato = "Esto es un string";  
2 let otroDato = "Esto es otro string";
```

## undefined

Este tipo se utiliza cuando el valor de una variable no ha sido definido aún o no existe. Por ejemplo

```
1  var dato; // su valor es undefined
2  var dato = undefined;
```

Otro tipo de almacenamiento de datos que tiene JavaScript son los Objetos. En JavaScript todo es un objeto, hasta las funciones. Todo hereda de la clase `Object`. Se pueden definir como una estructura donde se agregan valores. Dentro de las clases que heredan de `Object` tenemos `Array`, `Date`, etc... que veremos más adelante.