

[Área personal](#)[Mis cursos](#)[DWECC](#)[Glosario UD1](#)

## Glosario UD1

Navegue por el glosario usando este índice.

[Especial](#) | [A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [Ñ](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#) | [TODAS](#)

Página: [1](#) [2](#) [3](#) [4](#) (Siguiente)

[TODAS](#)

### A

#### Adobe Flash

(Última edición: miércoles, 27 de septiembre de 2023, 19:39)

Adobe Flash es una herramienta de desarrollo para la creación de contenido multimedia e interactivo.

Dado que Adobe ya no admite Flash Player después del 31 de diciembre de 2020, y que bloqueó la ejecución del contenido Flash en Flash Player a partir del 12 de enero de 2021, Adobe recomienda encarecidamente a todos los usuarios que desinstalen Flash Player inmediatamente para ayudar a proteger sus sistemas.

Pero dignifiquemos a Flash: supuso la entrada de vídeos, "webcams" en vivo, realidad aumentada e incluso dicen un JavaScript mejor: Actionscript 3. Todo esto al comienzo de la Web cuando todo era más estático y no existían las mejoras del HTML5 actual.

#### El principio del fin de Flash

En el año 2010 Steve Jobs publicó una carta en la cual explicaba que Apple dejaba la tecnología de Flash fuera de sus sistemas. Según Apple Flash no estaba preparada para el nuevo mundo de la tecnología móvil y el desarrollo táctil que empezaba a triunfar.

La mayoría de gigantes siguieron sus pasos. En 2015 Google y Firefox anunciaron que comenzarían a bloquear el contenido en Flash Player en sus navegadores. Microsoft se sumó al cambio algo más tarde.

Palabra(s) clave:



#### AJAX

(Última edición: miércoles, 27 de septiembre de 2023, 19:39)

AJAX (JavaScript Asíncrono y XML) es un término para describir dos capacidades de los navegadores que han estado presentes por años, pero que habían sido ignoradas por muchos desarrolladores Web, hasta que surgieron aplicaciones como Gmail, Google suggest y Google Maps.

Dichas capacidades son:

- La posibilidad de hacer peticiones al servidor sin tener que volver a cargar la página.
- La posibilidad de analizar y trabajar con documentos XML.



#### APIs de navegador

(Última edición: miércoles, 27 de septiembre de 2023, 19:38)

Construcciones integradas en el navegador creadas con el lenguaje JavaScript y que permiten implementar funcionalidad mucho más fácilmente.

Ejemplos de APIs del navegador:

- HTML DOM API proporciona acceso a varias funciones del navegador, como pestañas y ventanas, estilos CSS, hojas de estilo, historial del navegador...
- La API de geolocalización proporciona acceso a la información de ubicación geográfica asociada con el dispositivo.
- La API de almacenamiento web proporciona los mecanismos mediante los cuales el navegador puede almacenar información de tipo clave/valor, de una forma mucho más intuitiva que utilizando cookies. Los dos mecanismos en el almacenamiento web son sessionStorage y localStorage
- APIs que obtienen datos del servidor, comúnmente usadas para actualizar pequeñas secciones de una página web. Las APIs hacen esto posible gracias a que incluyen XMLHttpRequest y la Fetch API. También puede encontrar el término Ajax que describe esta técnica.
- Las APIs para dibujar y manipular gráficos ya son soportadas por la mayoría de navegadores. Las más populares son Canvas y WebGL, que permiten crear escenas 2D y 3D.
- APIs de audio y vídeo como HTMLMediaElement, la Web Audio API, y WebRTC



---

## APIs de terceros

(Última edición: miércoles, 27 de septiembre de 2023, 19:42)

Construcciones integradas en plataformas de terceros (por ejemplo Twitter, Facebook) que permiten usar algunas de las funcionalidades de esa plataforma en tus páginas web (como por ejemplo mostrar tus últimos Tweets en tu página web).

Ejemplos de APIs de terceros:

- YouTube API - Permite visualizar vídeos en tu sitio web
- Twitter API - Permite visualizar Tweets en tu sitio web
- Google Maps API - Permite hacer todo tipo de cosas con mapas en tus páginas web (incluso hace funcionar Google Maps).
- Conjunto de APIs de Facebook - Permiten visualizar información de Facebook en tu sitio web
- Twilio API - provee de un framework para crear la funcionalidad de llamadas y videollamadas en tus aplicaciones, enviar SMS o MMS y más.



---

## APIs Web

(Última edición: miércoles, 27 de septiembre de 2023, 19:37)

Las Interfaces de Programación de Aplicaciones (APIs por sus siglas en inglés) son construcciones disponibles en los lenguajes de programación que permiten a los desarrolladores crear funcionalidades complejas de una manera simple. Estas abstraen el código más complejo para proveer una sintaxis más fácil de uso.

JavaScript del lado cliente, particularmente, tiene muchas APIs disponibles. Estas no son parte del lenguaje en sí, sino que están construidas sobre el núcleo de este lenguaje de programación. Por lo general, se dividen en dos categorías:

- Las APIs de navegador están integradas en el navegador web y permiten manejar datos del navegador y de su entorno informático. Ejemplos de estas APIs son:
  - API de Geolocalización proporciona algunas construcciones simples de JavaScript para obtener datos de ubicación con los que, por ejemplo, trazar tu ubicación en un mapa de Google. Realmente, el navegador está haciendo uso de códigos de bajo nivel complejos en segundo plano (por ejemplo, C++) para comunicarse con el hardware GPS del dispositivo (o lo que esté disponible para determinar los datos de posición), recuperar datos de posición y devolverlos al entorno del navegador para su uso en tu código. Pero una vez más, la API se encarga de abstraer esta complejidad.
  - La API de Audio Web proporciona un sistema poderoso y versátil para controlar audio en el navegador, permitiendo a los desarrolladores escoger fuentes de audio, agregar efectos al audio, crear visualizaciones de audios, aplicar efectos espaciales...
- Las APIs de terceros no están incluidas por defecto en el navegador, y por lo general es necesario obtener el código e información desde algún lugar de la Web. Por ejemplo, la API de Twitter permite hacer cosas como mostrar tus últimos tweets en un sitio web. Proporciona un conjunto especial de construcciones que puedes usar para consultar el servicio de Twitter y devolver información específica.

Fuente:

- [https://developer.mozilla.org/es/docs/Learn/JavaScript/Client-side\\_web\\_APIs/Introduction](https://developer.mozilla.org/es/docs/Learn/JavaScript/Client-side_web_APIs/Introduction)
- <https://mdn.github.io/voice-change-o-matic/>



---

## Aplicación web

(Última edición: miércoles, 27 de septiembre de 2023, 19:51)

Una aplicación web es un programa similar a una aplicación de escritorio, pero que es utilizada dentro del navegador, y tiene dos ventajas importantes:

- Es ubicuo: Una aplicación web está disponible en casi cualquier equipo que tenga un navegador web incorporado. Debido a que no existe necesidad de instalar una aplicación, la información del usuario está disponible sin importar el equipo desde donde se acceda a la aplicación.
- Es auto-actualizable: Una aplicación web no reside en el equipo, sino en un servidor web. Esto tiene como ventaja que puede ser actualizada sin necesidad de la interacción del usuario.

Como desventajas:

- Su disponibilidad depende de otros factores: De una conexión a Internet, del servidor de la aplicación (tanto para aplicaciones de Internet como intranets), y en situaciones menos comunes, del navegador usado.
- Está limitado al navegador: El navegador por definición está limitado en cuanto a lo que puede acceder del equipo, lo que en términos técnicos se conoce como *sandboxing*. Este tipo de limitaciones, por consiguiente, limitan a las aplicaciones web que se ejecutan dentro de él.

Las aplicaciones web no son de ahora y no implican utilizar solo JavaScript. Existían aplicaciones web antes del llamado *Web 2.0*, que utilizan JavaScript, Java, Flash, Flex, Silverlight, e incluso algunas sólo utilizan HTML y CSS. Sin embargo, el avance que ha tenido JavaScript, desde la *Web 2.0* hasta el HTML5, ha logrado superar en alguna forma las desventajas que tenían las aplicaciones web.

Ahora se pueden realizar peticiones al servidor sin recargar toda la página, tener una comunicación interactiva con el servidor, realizar algunas operaciones sin necesidad de tener una conexión, leer y escribir archivos en el equipo, entre otros.



## Applet

(Última edición: miércoles, 27 de septiembre de 2023, 19:42)

Un applet es un componente Java que se incluye en una página web y se ejecuta en el navegador. Se programan en Java y, por tanto, se benefician de su potencia y flexibilidad. Son pequeños programas que se transfieren con las páginas web y que el navegador ejecuta en el espacio de la página (a través de un módulo o extensión concretos). Los applets se programan en Java y se envían al cliente precompilados, es por ello que la manera de trabajar de estos varía un poco con respecto a los lenguajes de script como JavaScript. Una de las principales ventajas de utilizar applets es que son mucho menos dependientes del navegador que los scripts en JavaScript, incluso independientes del sistema operativo del ordenador donde se ejecutan.

Como desventajas notables frente al uso de JavaScript, los applets son más lentos de procesar y que tienen un espacio delimitado en la página donde se ejecutan, es decir, no se mezclan con todos los componentes de la página ni tienen acceso a ellos.

Un applet es una pequeña aplicación Java que se descarga y ejecuta en el navegador.

Pero actualmente la tecnología necesaria para los applets Java ya no está soportada por los navegadores más usados. Los applets fueron muy populares pero la seguridad prevalece en el mundo de los navegadores



---

## async. Atributo de la etiqueta script

(Última edición: miércoles, 27 de septiembre de 2023, 19:41)

Atributo booleano que indica al navegador, si es posible, ejecutar el código JS nada más cargarse. Esto no afecta a los scripts escritos dentro de la etiqueta (es decir a aquellos que no tienen el atributo src).

Los scripts cargados con el atributo `async` descargarán el script en un hilo aparte, sin bloquear el renderizado de la página y lo ejecutará tan pronto como el script se termine de descargar. No hay garantía de que los scripts se ejecuten en un orden específico, solo que no detendrán la visualización del resto de la página. Es mejor usar `async` cuando los scripts de la página se ejecutan de forma independiente y no dependen de ningún otro script de la página.

Por ejemplo, si tienes los siguientes elementos script:

```
<script async src="js/vendor/jquery.js"></script>
<script async src="js/script2.js"></script>
<script async src="js/script3.js"></script>
```

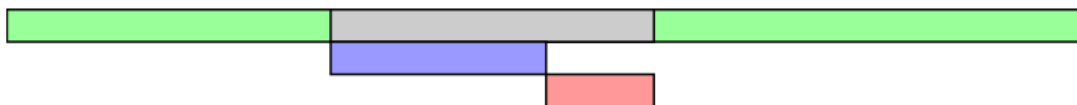
No puedes confiar en el orden en que se cargarán los scripts. `jquery.js` se puede cargar antes o después de `script2.js` y `script3.js` y si este es el caso, cualquier función en esos scripts dependiendo de `jquery` producirá un error porque `jquery` no se definirá en el momento en que se ejecute el script.

## Legend

- HTML parsing
- HTML parsing paused
- Script download
- Script execution

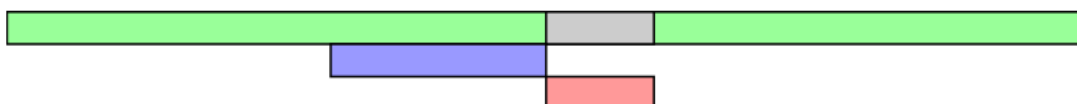
## <script>

Let's start by defining what `<script>` without any attributes does. The HTML file will be parsed until the script file is hit, at that point parsing will stop and a request will be made to fetch the file (if it's external). The script will then be executed before parsing is resumed.



## <script async>

`async` downloads the file during HTML parsing and will pause the HTML parser to execute it when it has finished downloading.



## <script defer>

`defer` downloads the file during HTML parsing and will only execute it after the parser has completed. `defer` scripts are also guaranteed to execute in the order that they appear in the document.



Palabra(s) clave:



C

## Carga de scripts

(Última edición: miércoles, 27 de septiembre de 2023, 19:54)

Un problema común es que todo el HTML de una página se carga en el orden en que aparece y el script suele bloquear el análisis del HTML. Si estás utilizando JavaScript para manipular elementos en la página (DOM), tu código no funcionará si el JavaScript se carga y procesa antes que el elemento en cuestión.

Para solucionarlo se puede usar el evento `DOMContentLoaded`, que permite hacer algo nada más se carguen los elementos del árbol DOM

```
document.addEventListener("DOMContentLoaded", function() {  
    ...  
});
```

Otra solución es usar en la etiqueta `script` el atributo `defer`, que le dice al navegador que el script se carga simultáneamente mientras se sigue analizando el HTML, pero que el script se ejecutará al finalizar de cargar el árbol DOM. Se descarga el script en un hilo separado, mientras que el resto de la página (el DOM, etc.) se descarga, por lo que los scripts no bloquean la carga de la página.

Los atributos `async` y `defer` en la etiqueta `script` indican al navegador que descargue los scripts en un hilo separado, mientras que el resto de la página (el DOM, etc.) se descarga, por lo que los scripts no bloquean la carga de la página.



---

## Chromium

(Última edición: miércoles, 27 de septiembre de 2023, 19:41)

Chromium es una base de código abierto para desarrollar un navegador web, mantenida por diversas compañías que posteriormente usan el código fuente para crear su propia versión de navegador con características adicionales.

Todo empezó en el 2008, cuando Google lanzó Chrome 1.0 y tres meses después, la empresa liberó su código fuente junto a un proyecto al que llamaron Chromium. La idea era la de obtener la ayuda de la comunidad para mejorar entre todos el motor del navegador, y que tanto Google como todos los que quisiesen utilizarlo para desarrollar otros navegadores pudieran beneficiarse del esfuerzo colectivo.

Actualmente Google Chrome, Opera, Vivaldi, Yandex y Microsoft Edge son hermanos gracias a que comparten la misma base común: Chromium. La diferencia entre unos y otros navegadores son:

- plugins añadidos
- mantenimiento dedicado
- forma de trabajar con los procesos
- otras opciones diseñadas por los desarrolladores para hacer único a cada navegador.



---

## Cliente-servidor

(Última edición: miércoles, 27 de septiembre de 2023, 19:41)

Los sitios web siguen un modelo basado en la programación cliente-servidor con tres elementos comunes:

- El lado del servidor (server-side): incluye el hardware y software del servidor Web así como diferentes elementos de programación y tecnologías incrustadas. Las tecnologías pueden abarcar un rango amplio, desde programas CGI escritos en PERL hasta aplicaciones multihilo basadas en Java, incluyendo tecnologías de servidor de bases de datos que soporten múltiples sitios web.
- El lado del cliente (client-side): este elemento hace referencia a los navegadores web y está soportado por tecnologías como HTML, CSS y JavaScript, los cuales se utilizan para crear la presentación de la página o proporcionar características interactivas.
- La red: describe los diferentes elementos de conectividad utilizados para mostrar el sitio web al usuario.



---

## D

### defer. Atributo de la etiqueta script

(Última edición: miércoles, 27 de septiembre de 2023, 19:53)

Atributo booleano que indica al navegador que el script debe ser ejecutado después de que el documento entero sea analizado.

Los scripts cargados con el atributo defer se ejecutarán en el orden en que aparecen en la página y los ejecutará tan pronto como se descarguen el script y el contenido

```
<script defer src="js/vendor/jquery.js"></script>
<script defer src="js/script2.js"></script>
<script defer src="js/script3.js"></script>
```

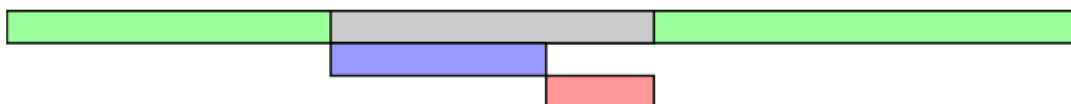
Todos se cargarán en el orden indicado y no se ejecutarán hasta que el contenido del DOM esté cargado.

## Legend

- HTML parsing
- HTML parsing paused
- Script download
- Script execution

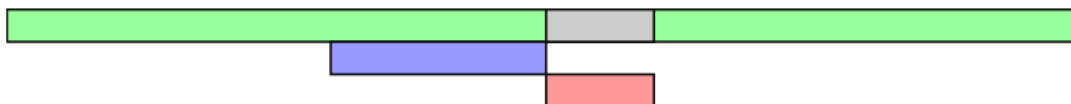
## <script>

Let's start by defining what **<script>** without any attributes does. The HTML file will be parsed until the script file is hit, at that point parsing will stop and a request will be made to fetch the file (if it's external). The script will then be executed before parsing is resumed.



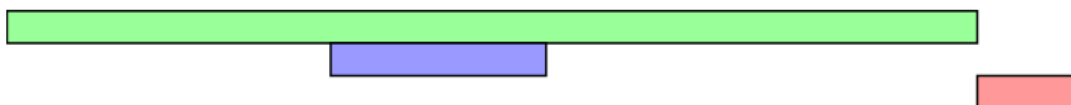
## <script async>

**async** downloads the file during HTML parsing and will pause the HTML parser to execute it when it has finished downloading.



## <script defer>

**defer** downloads the file during HTML parsing and will only execute it after the parser has completed. **defer** scripts are also guaranteed to execute in the order that they appear in the document.



Palabra(s) clave:



## E

### Ejecución de código en el navegador

(Última edición: miércoles, 27 de septiembre de 2023, 19:41)

Cuando cargas una página web en tu navegador, estás ejecutando tu código (HTML, CSS y JavaScript) dentro de un entorno de ejecución (la pestaña del navegador). Esto es como una fábrica que toma materias primas (el código) y genera un producto (la página web).

Un uso muy común de JavaScript es modificar dinámicamente HTML y CSS para actualizar una interfaz de usuario, a través de la API del modelo de objetos del documento (DOM). El código de tus documentos web generalmente se cargan y ejecutan en el orden en que aparece en la página. Si JavaScript se carga e intenta ejecutarse antes de que se hayan cargado el HTML y el CSS al que afecta, pueden producirse errores. En esos casos se debe ejecutar el código JS cuando los elementos del DOM estén cargados.

Esto se soluciona:

- Desde el código JS usando el evento **DOMContentLoaded**
- Desde la etiqueta script indicando con el atributo **defer** que se ejecute al final de la carga HTML y CSS.



## F

### Frameworks JavaScript

(Última edición: miércoles, 27 de septiembre de 2023, 19:40)

El siguiente paso a las librerías, los frameworks JavaScript (como Angular y Ember) suelen ser paquetes de HTML, CSS, JavaScript y otras tecnologías que se instalan y luego se usan para escribir una aplicación web completa desde cero. La diferencia clave entre una librería y un framework es la "Inversión del control". Cuando se llama a un método desde una librería, el desarrollador tiene el control. Con un framework el control se invierte: el framework llama al código del desarrollador.



## G

### Google Chrome

(Última edición: miércoles, 27 de septiembre de 2023, 19:43)

Google Chrome es un navegador web de código cerrado desarrollado por Google, aunque derivado de proyectos de código abierto (como el motor de renderizado Blink).

Está disponible gratuitamente. Se considera desde hace tiempo el navegador más usado de la Web. Utiliza el motor de renderizado Blink y el motor JavaScript V8, (teóricamente dos veces más rápido que Firefox 3 y que la versión beta de Safari 4.)

### Crítica desde Firefox

*En 2008, Google introdujo Chrome, y su impacto innovador en la tecnología del navegador fue inmediato. Era más rápido para cargar sitios web, ocupaba un espacio mínimo en la pantalla y ofrecía una interfaz de usuario innegablemente simple.*

*Avanzando hasta hoy, el panorama competitivo para los navegadores ha cambiado con muchas personas que comienzan a cuestionarse qué está sucediendo con sus datos en línea tales como el historial de navegación, las contraseñas y otra información confidencial. Mucho ha cambiado desde 2008 cuando Chrome entró en escena. En Firefox, hemos estado trabajando duro para rediseñar nuestra interfaz y proporcionar a los usuarios un número cada vez mayor de mejoras de privacidad y rendimiento que están activas de forma predeterminada, así como muchas herramientas útiles para el navegador.*

Fuente

<https://www.mozilla.org/es-ES/firefox/browsers/compare/chrome/>

Palabra(s) clave:



## I

### Integración de JavaScript con etiquetas HTML

(Última edición: miércoles, 27 de septiembre de 2023, 19:47)



JavaScript se integra a su página HTML de manera similar a CSS. Mientras que CSS usa etiquetas `<link>` para aplicar hojas de estilo externas y elementos `<style>` para aplicar hojas de estilo internas a HTML, JavaScript sólo necesita la etiqueta `<script>`. Podemos integrarlo con tres filosofías distintas, aunque sólo una es la recomendada.

## JavaScript interno

```
<script>
  // JavaScript va aquí
</script>
```

La etiqueta debe ir antes del `</head>`, y hay que tener en cuenta que el árbol DOM no estará cargado (uso del evento `DOMContentLoaded`)

```
document.addEventListener("DOMContentLoaded", function() {
  ...
});
```

## JavaScript externo

Se localiza el código JS en un documento externo, y es la forma recomendada para aislar el diseño del comportamiento, y así facilitar la depuración en el desarrollo

```
<script src="script.js" defer></script>
```

## JavaScript en línea

En este caso habrá fragmentos de código JavaScript dentro de HTML.

```
<button onclick="alert()">¡Haz click!</button>
```

Es una mala práctica mezclar HTML con JavaScript, y es ineficiente: tendría que incluir el atributo `onclick` en cada botón al que desee que se aplique JavaScript.

Referencia:

[how\\_do\\_you\\_add\\_javascript\\_to\\_your\\_page](#)

Palabra(s) clave:



## integrity. Atributo de etiqueta script

(Última edición: miércoles, 27 de septiembre de 2023, 19:46)

Contiene información de metadatos que es usada por el navegador para verificar que el fichero fue entregado libre de manipulación inesperada.

La integridad de los recursos secundarios (SRI) permite que el navegador verifique que los archivos que se entregan no se hayan modificado. Actualmente, los navegadores están implementando esta especificación. Agregar el nuevo atributo `integrity` garantizará que su aplicación obtenga esta mejora de seguridad, ya que los navegadores la admiten.

jQuery CDN es compatible con SRI. A continuación un ejemplo para incluir la librería jQuery con el atributo `integrity` incluido:

```
<script
  src="https://code.jquery.com/jquery-3.6.0.js"
  integrity="sha256-H+K7U5CnX11h5ywQfKtSj8PCmoN9aaq30gDh27Xc0jk="
  crossorigin="anonymous"></script>
```



## J

## JavaScript

(Última edición: miércoles, 27 de septiembre de 2023, 19:46)

JavaScript (o JS) es un lenguaje de scripts de alto nivel incorporado en los navegadores que permite implementar interactividad en páginas web/apps. JavaScript también está disponible en otros entornos de programación, como Node.

JavaScript se utiliza principalmente en el navegador, lo que permite a los desarrolladores manipular el contenido de la página web a través del DOM, manipular datos con AJAX y IndexedDB, dibujar gráficos con canvas, interactuar con el dispositivo que ejecuta el navegador a través de varias APIs y más. JavaScript es uno de los lenguajes más utilizados en el mundo, debido al reciente crecimiento y mejora en el rendimiento de las APIs disponibles en los navegadores.

Dentro de las tecnologías Web, JavaScript es la tercera capa junto con HTML y CSS.

- HTML es el lenguaje de marcas que usamos para estructurar y dar significado a nuestro contenido web, por ejemplo, definiendo párrafos, encabezados y tablas de datos, o incrustando imágenes y vídeos en la página.
- CSS es un lenguaje de reglas de estilo que usamos para aplicar estilo a nuestro contenido HTML, por ejemplo, establecer colores de fondo y fuentes, y distribuir nuestro contenido en varias columnas.
- JavaScript es un lenguaje de secuencias de comandos que le permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes y prácticamente todo lo demás.



---

## JavaScript es código interpretado y no compilado

(Última edición: miércoles, 27 de septiembre de 2023, 19:46)

En los lenguajes interpretados, el código se ejecuta de arriba a abajo y el resultado de ejecutar el código se devuelve inmediatamente. No tienes que transformar el código en una forma diferente antes de que el navegador lo ejecute. El código se recibe en su forma de texto amigable para el programador y se procesa directamente desde allí.

Los lenguajes compilados, por otro lado, se transforman (compilan) a código máquina antes de que sean ejecutados por la computadora. Por ejemplo, C/C++ se compila a código máquina que luego ejecuta la computadora. El programa se ejecuta desde un formato binario, que se generó a partir del código fuente del programa original.

JavaScript es un lenguaje de programación interpretado ligero. El navegador web recibe el código JavaScript en su forma de texto original y ejecuta el script a partir de ahí. Desde un punto de vista técnico, la mayoría de los intérpretes de JavaScript modernos utilizan una técnica llamada compilación en tiempo real (JIT, just-in-time) para mejorar el rendimiento; el código fuente de JavaScript se compila en un formato binario más rápido mientras se usa el script, de modo que se pueda ejecutar lo más rápido posible. Sin embargo, JavaScript todavía se considera un lenguaje interpretado, ya que la compilación se maneja en el entorno de ejecución, en lugar de antes.



---

## JavaScript. Orígenes e Historia

(Última edición: miércoles, 27 de septiembre de 2023, 19:48)

Concebido como un lenguaje de lado del servidor por Brendan Eich (entonces empleado por Netscape Corporation), JavaScript pronto llegó a Netscape Navigator 2.0 en septiembre de 1995. JavaScript disfrutó de un éxito inmediato y Internet Explorer 3.0 introdujo la compatibilidad con JavaScript con el nombre de JScript en agosto de 1996.

En noviembre de 1996, Netscape comenzó a trabajar con ECMA International para hacer de JavaScript un estándar de la industria. Desde entonces, el JavaScript estandarizado se llama ECMAScript y se especifica en ECMA-262. Desde la llegada de ECMA-262, 6th edition, en Junio de 2015 (la especificación de scripting en la que se basa JavaScript), ECMA International se ha dedicado a actualizar JavaScript anualmente.

Recientemente, la popularidad de JavaScript se ha expandido aún más gracias a la exitosa plataforma Node.js, el entorno de ejecución de JavaScript multiplataforma más popular fuera de el navegador. Node.js, creado con el motor de JavaScript V8 de Chrome: permite a los desarrolladores utilizar JavaScript como lenguaje de programación para automatizar las tareas en una computadora y crear servidores HTTP y Web Sockets completamente funcionales.



## JavaScript. Ventajas e inconvenientes

(Última edición: miércoles, 27 de septiembre de 2023, 19:50)

### Ventajas de JavaScript

- **Velocidad:** JavaScript tiende a ser muy rápido porque a menudo se ejecuta inmediatamente dentro del navegador del cliente. Siempre que no requiera recursos externos, JavaScript no se ralentiza por llamadas a un servidor backend. Además, los principales navegadores admiten la compilación JIT (justo a tiempo) para JavaScript, lo que significa que no es necesario compilar el código antes de ejecutarlo.
- **Simplicidad:** la sintaxis de JavaScript se inspiró en Java y es relativamente fácil de aprender en comparación con otros lenguajes populares como C++.
- **Popularidad:** JavaScript está en todas partes de la web y, con la llegada de Node.js, se utiliza cada vez más en el backend. Hay innumerables recursos para aprender JavaScript. Tanto StackOverflow como GitHub muestran una cantidad cada vez mayor de proyectos que usan JavaScript.
- **Interoperabilidad:** a diferencia de PHP u otros lenguajes de programación, JavaScript se puede insertar en cualquier página web. JavaScript se puede utilizar en muchos tipos diferentes de aplicaciones debido al soporte en otros lenguajes como Pearl y PHP.
- **Carga del servidor:** JavaScript es del lado del cliente, por lo que reduce la demanda en los servidores en general, y es posible que las aplicaciones simples no necesiten un servidor en absoluto.
- **Interfaces enriquecidas:** JavaScript se puede utilizar para crear funciones como arrastrar y soltar y componentes como controles deslizantes, todo lo cual mejora enormemente la interfaz de usuario y la experiencia de un sitio.
- **Funcionalidad extendida:** los desarrolladores pueden extender la funcionalidad de las páginas web escribiendo fragmentos de JavaScript para complementos de terceros como Greasemonkey.
- **Versatilidad:** hay muchas formas de usar JavaScript a través de los servidores Node.js. Si tuviera que iniciar Node.js con Express, usar una base de datos de documentos como MongoDB y usar JavaScript en la interfaz para los clientes, es posible desarrollar una aplicación JavaScript completa de adelante hacia atrás usando solo JavaScript.
- **Actualizaciones:** Desde la llegada de ECMA-262, 6th edition, en Junio de 2015 (la especificación de scripting en la que se basa JavaScript), ECMA International se ha dedicado a actualizar JavaScript anualmente.

### Desventajas de JavaScript

- **Seguridad del lado del cliente:** dado que el código JavaScript se ejecuta en el lado del cliente, los errores y descuidos a veces pueden explotarse con fines maliciosos. Debido a esto, algunas personas optan por deshabilitar JavaScript por completo.
- **Compatibilidad con el navegador:** si bien los scripts del lado del servidor siempre producen el mismo resultado, los diferentes navegadores a veces interpretan el código JavaScript de manera diferente. Actualmente las diferencias son mínimas, aunque se recomienda que se pruebe el script en todos los navegadores principales.

Fuente:

<https://www.freecodecamp.org/news/the-advantages-and-disadvantages-of-javascript/>



## L

### Librerías JavaScript

(Última edición: miércoles, 27 de septiembre de 2023, 19:44)

Por lo general es uno o más archivos JavaScript que contienen funciones personalizadas que puedes añadir a tu página web para acelerar o habilitar la escritura de funcionalidades comunes. Por ejemplo jQuery, Mootools y React.



## M

### MDN Web Docs

(Última edición: miércoles, 27 de septiembre de 2023, 19:44)

MDN Web Docs (Mozilla Developer Network Web Docs) es el sitio web oficial de Mozilla para la documentación de estándares web y de los proyectos de Mozilla.

MDN Web Docs es un recurso para desarrolladores, mantenido por la comunidad de desarrolladores voluntarios, que aloja muchos documentos sobre diferentes tecnologías web, como HTML5, JavaScript, CSS, Web APIs, Django, Node.js, WebExtensions y MathML.



## Microsoft Edge

(Última edición: miércoles, 27 de septiembre de 2023, 19:44)

Microsoft Edge es el navegador web desarrollado por Microsoft para reemplazar a Internet Explorer (IE). Se lanzó en 2015 y reemplazó definitivamente a IE en junio de 2022. Aunque Internet Explorer 11 continuará teniendo soporte hasta el 13 de enero de 2032

Deja atrás un motor de renderizado propio, EdgeHTML, en favor de Blink, el motor de Chrome, o de su motor JavaScript, Chakra, para sustituirlo por el V8 de Chromium. Sin embargo también es cierto que este proyecto Open Source, base de este navegador y otros como Vivaldi, Opera o Brave, es cada vez "menos Google" gracias a la implicación de todos estos actores.

### Un poco de historia

Con Windows 10, Microsoft introdujo su navegador Edge para competir con Firefox y Chrome, haciéndolo el navegador preinstalado y predeterminado en millones de PCs vendidos. Incluso así, los usuarios fueron lentos en su adopción y Microsoft finalmente anunció planes para relanzar Edge como un navegador basado en Chromium (Chromium es el proyecto de navegador de código abierto de Google). Desde enero de 2020, el navegador Edge basado en Chromium de Microsoft ha reemplazado las versiones previas de Edge. Pese a que Edge está construido en la base de Chromium de Google, un número de funcionalidades únicas lo diferencian del navegador Chrome de Google.

Fuente

<https://www.mozilla.org/es-ES/firefox/browsers/compare/edge/>

Palabra(s) clave:



## Motores JS

(Última edición: miércoles, 27 de septiembre de 2023, 19:44)

Los **intérpretes** de JavaScript o motores JS están integrados en múltiples aplicaciones de uso cotidiano. Estas aplicaciones proporcionan su propio modelo de acceso y gestión de los módulos que componen la aplicación y para ello comparten el lenguaje JavaScript en cada aplicación.

Los motores JS que actualmente están integrados en los navegadores:

- V8: usado en Google Chrome, así como en Chromium, Microsoft Edge, Brave y Opera. También es usado del lado del servidor en Couchbase, MongoDB y Node.js.
- SpiderMonkey: usado en Mozilla Firefox
- JavaScriptCore: usado en Safari



## Mozilla Firefox

(Última edición: miércoles, 27 de septiembre de 2023, 19:49)

Mozilla Firefox es un navegador web libre y de código abierto desarrollado para distintas plataformas, está coordinado por la Corporación Mozilla y la Fundación Mozilla. Usa el motor Gecko para renderizar páginas web y SpiderMonkey como su motor JavaScript.

Aunque hubo unos tiempos en los que Firefox podía presentar batalla frente a Chrome, actualmente no hay opción: Chrome a ganado la guerra. Pero Mozilla defiende su navegador frente a Google Chrome en estos términos:

*Creemos que es justo decir que Firefox y Chrome están pisándose los talones en términos de **portabilidad y utilidad**, con Chrome teniendo una ligera ventaja en la utilidad debido a su gran biblioteca de extensiones y características adicionales. Pero en términos de **privacidad**, Firefox gana con nuestro compromiso de preservar los datos en línea de nuestros usuarios y proporcionar servicios gratuitos integrados como administradores de contraseñas que también te alertan si ocurre una filtración de datos que involucre tus credenciales.*

*Para fines prácticos, obviamente no hay nada que te impida usar ambos navegadores (Firefox para esos momentos de la vida en los que la privacidad realmente importa y Chrome si todavía estás interesado en el ecosistema de Google). Sin embargo, con el creciente número de **incursiones en nuestros datos personales**, Firefox puede ser la opción correcta a largo plazo para aquellos de nosotros que valoramos proteger nuestra privacidad personal en línea.*

Referencia:

<https://www.mozilla.org/es-ES/firefox/browsers/compare/chrome/>



---

## N

### Navegador o cliente Web

(Última edición: miércoles, 27 de septiembre de 2023, 19:49)

Un navegador web, o explorador web (browser), es una aplicación, distribuida habitualmente como software libre, que permite a un usuario acceder (y, normalmente, visualizar) a un recurso publicado por un servidor web a través de Internet y descrito mediante una dirección URL (Universal Resource Locator).

Se utilizan "navegar" por recursos de tipo hipertexto, comúnmente descritos en HTML, ofrecidos por servidores web de todo el mundo a través de Internet. Desde la creación de la Web a principios de los años 90, los navegadores web han evolucionado desde meros visualizadores de texto que, aunque no ofrecían capacidades multimedia (visualización de imágenes), cumplían su propósito.

La guerra de navegadores (en inglés: Browser Wars) es el nombre popular dado a la competencia entre los navegadores web por el dominio del mercado. Fue una lucha que hubo entre Microsoft, con su navegador Internet Explorer, y Netscape, con Netscape Navigator. Ambos querían dominar el mercado de navegadores web (particularmente en la plataforma Windows) durante finales de los años 1990.

Inicialmente Netscape Navigator fue el estándar de facto frente a Mosaic y Lynx (desarrollados en campus universitarios). Pero Microsoft compró licencias de Mosaic para crear la primera versión de Internet Explorer.

Actualmente Internet Explorer está en declive y Google Chrome apenas tiene competencia frente a sus adversarios Safari, Firefox y Edge.



---

## P

### Petición Web

(Última edición: miércoles, 27 de septiembre de 2023, 19:49)

Los navegadores web se comunican con los servidores web usando el **Protocolo de Transferencia de HyperTexto** (HyperTextTransfer Protocol HTTP). Cuando pinchas en un enlace sobre una página web, envías un formulario o ejecutas una búsqueda, el explorador envía una petición (Request) HTTP al servidor.

Esta **petición** incluye:

- Una URL que identifica el servidor de destino y un recurso (ej. un fichero HTML, un punto de datos particular en el servidor, o una herramienta a ejecutar).
- Un método que define la acción requerida (por ejemplo, obtener un fichero o salvar o actualizar algunos datos). Algunos de estos métodos son: GET, POST, HEAD, PUT, DELETE...
- Se puede codificar información adicional con la petición (por ejemplo, datos de un formulario HTML). La información puede ser codificada como:
  - Parámetros URL
  - Datos POST. Las peticiones POST añaden nuevos recursos, cuyos datos están codificados dentro del cuerpo de la petición.
  - Cookies de lado cliente. Los Cookies contienen datos de sesión acerca del cliente, incluyendo las claves que el servidor puede usar para determinar su estado de inicio de sesión y los permisos/accesos a los recursos.

Los servidores web esperan los mensajes de petición de los clientes, los procesan cuando llegan y responden al explorador web con un mensaje de respuesta HTTP. La respuesta contiene un código de estado de respuesta HTTP que indica si la petición ha tenido éxito o no (ej. "200 OK" para indicar éxito, "404 Not Found" si el recurso no ha podido ser encontrado, "403 Forbidden" si el usuario no está autorizado a acceder al recurso, etc). El cuerpo de la respuesta de éxito a una petición GET contendría el recurso solicitado.

Cuando se devuelve una página HTML es renderizada por el explorador web. Como parte del procesamiento el explorador puede descubrir enlaces a otros recursos (ej. una página HTML normalmente referencia las páginas JavaScript y CSS), y enviará peticiones HTTP separadas para descargar estos ficheros.

Fuente: [https://developer.mozilla.org/es/docs/Learn/Server-side/First\\_steps/Client-Server\\_overview](https://developer.mozilla.org/es/docs/Learn/Server-side/First_steps/Client-Server_overview)

Palabra(s) clave:



## S

### Safari

(Última edición: miércoles, 27 de septiembre de 2023, 19:49)

Safari es un navegador web de código cerrado desarrollado por Apple Inc. Está disponible para macOS, iPadOS e iOS, y de 2007 a 2012, estaba disponible para Windows.

Está escrito sobre el framework WebKit, que incluye a WebCore, el motor de renderizado, y JavaScriptCore, el intérprete de JavaScript. Por su parte, WebKit (el motor de renderizado del navegador) está basado en el motor KHTML, creado por el proyecto KDE para su navegador Konqueror. Como resultado de esto, el motor interno de Safari es software libre y es liberado bajo los términos de la licencia LGPL. Las mejoras al código de HTML por parte de Apple son incorporadas al código de KDE

Los usuarios de Mac o iPhone, están familiarizados con el navegador web Safari porque está preinstalado como el navegador predeterminado para los usuarios de productos Apple. Esto le da una ventaja inicial significativa frente a otros navegadores

Palabra(s) clave:



### Seguridad del navegador

(Última edición: miércoles, 27 de septiembre de 2023, 19:49)

Cada pestaña del navegador tiene su propio depósito separado para ejecutar código (estos depósitos se denominan "entornos de ejecución" en términos técnicos). Esto significa que, en la mayoría de los casos, el código de cada pestaña se ejecuta de forma completamente independiente y el código de una pestaña no puede afectar el código en otra pestaña, o en otro sitio web. Esta es una buena medida de seguridad para que los piratas no escriban código para robar información de otros sitios web, entre otras cosas.

Pero además existe la política del mismo origen: los scripts de un sitio web no tienen acceso a la información enviada a otro sitio web (de otro dominio) como pudiera ser nombres de usuario, contraseñas o cookies

<https://es.wikipedia.org/wiki/JavaScript#Seguridad>



## src. Atributo de la etiqueta script

(Última edición: miércoles, 27 de septiembre de 2023, 19:48)

Este atributo especifica la URI del script externo; este puede ser usado como alternativa a scripts embebidos directamente en el documento. Si el script tiene el atributo src, no debería tener código dentro de la etiqueta.



## SRI

(Última edición: miércoles, 27 de septiembre de 2023, 19:48)

SRI (Subresource Integrity) es una nueva especificación del W3C que permite a los desarrolladores web asegurarse de que los recursos alojados en servidores de terceros no hayan sido manipulados. Se recomienda el uso de SRI siempre que las librerías se carguen desde una fuente de terceros.

### Por qué se necesita incluir crossorigin="anonymous"

Cuando la solicitud no está en el mismo origen, el atributo crossorigin debe estar presente para verificar la integridad del archivo.

Si un atributo crossorigin el navegador elegirá 'abrir por error', lo que significa que cargará el recurso como si el atributo integrity no estuviera configurado, perdiendo efectivamente toda la seguridad que aporta SRI en primer lugar.

Fuente:

<https://www.srihash.org/>



## T

## type. Atributo de la etiqueta script

(Última edición: miércoles, 27 de septiembre de 2023, 19:48)

type

Este atributo identifica el lenguaje de scripting en que está escrito el código embebido dentro de la etiqueta script, o referenciada utilizando el atributo src. Los valores posibles están especificados como un MIME type (tipo MIME).

Algunos ejemplos de tipos MIME que pueden ser utilizados son: text/javascript, text/ecmascript, application/javascript, y application/ecmascript. Este atributo es opcional porque actualmente el valor por defecto es un script JavaScript.

```
<!-- HTML4 y (x)HTML -->
<script type="text/javascript" src="javascript.js"></script>
<!-- HTML5 -->
<script src="javascript.js"></script>
```

Si el tipo MIME especificado no es un tipo JavaScript, el contenido embebido dentro de la etiqueta script es tratado como un bloque de datos que no será procesado por el navegador.

Si el tipo especificado es module, el código es tratado como un módulo JavaScript.

```
<script src="index.js" type="module"></script>
```



---

Página: [1](#) [2](#) [3](#) [4](#) (Siguiente)  
TODAS



**Junta de Andalucía**  
Consejería de Desarrollo Educativo  
y Formación Profesional

Junta de Andalucía Consejería de Desarrollo Educativo y Formación Profesional

Calle Juan Antonio de Vizarrón s/n · Edf. Torretriana

Isla de la Cartuja · 41092 Sevilla

[Accesibilidad](#) | [Portal Séneca](#) | [Plataformas de Aprendizaje](#)