

Tema 1. Selección de arquitecturas y herramientas de programación

Contenido

Tema 1. Selección de arquitecturas y herramientas de programación	1
La web	2
Las fases de Internet.....	3
Elementos básicos	4
Arquitectura cliente-servidor: el navegador	5
Diferentes motores de rénder	6
Intérpretes de JavaScript.....	7
Lenguajes en entorno cliente	8
JavaScript	9
Ejecución de JavaScript.....	10
Integración de JavaScript en un documento	11
Tres formas de incluir código JavaScript en un documento HTML.....	12
Incluir alternativa con <noscript>.....	13
Uso de la consola del navegador	15

La web

Lenguajes de marcado

GML. (1969-70). Charles F. Goldfarb
IBM

SGML. SGM estandarizado como
norma ISO. 1980

HTML. 1990 Tim Berners-Lee

XML. 1998 W3C

XHTML. 2000 W3C

HTML5. 2014. W3C / WHATWG

Navegadores Web

WorldWideWeb (1990) - <https://worldwideweb.cern.ch/>

Lynx (1992)

Mosaic (1993) – Imágenes

Netscape (1994) / Internet Explorer (1995) -GUERRA NAVEGADORES

Opera (1995)

Mozilla Firefox (2002)

Safari (2003). Mac

Google Chrome (2008)

Microsoft Edge (2015)

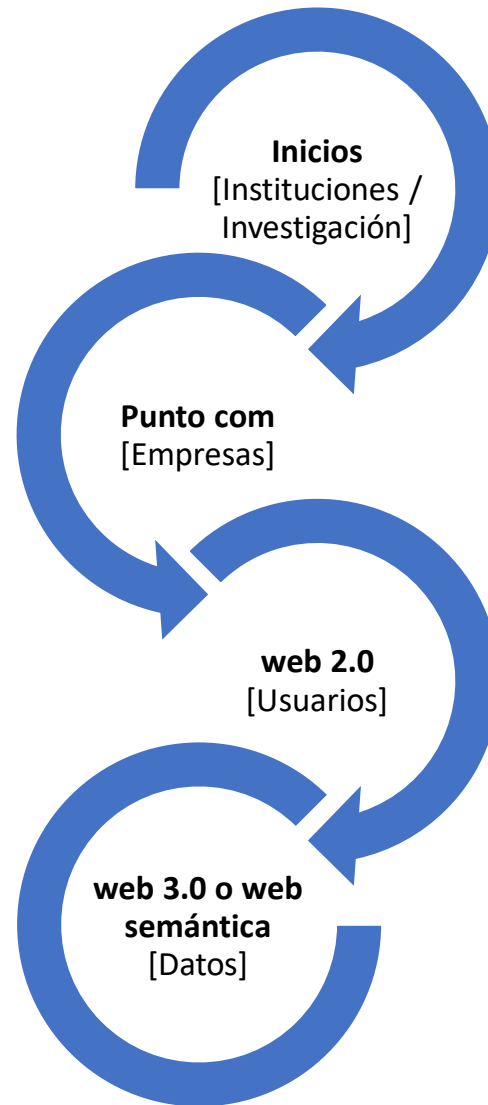
Organismos estandarizadores

W3C (World Wide Web Consortium): estándares para la web, libres y no limitados por derechos de autor.

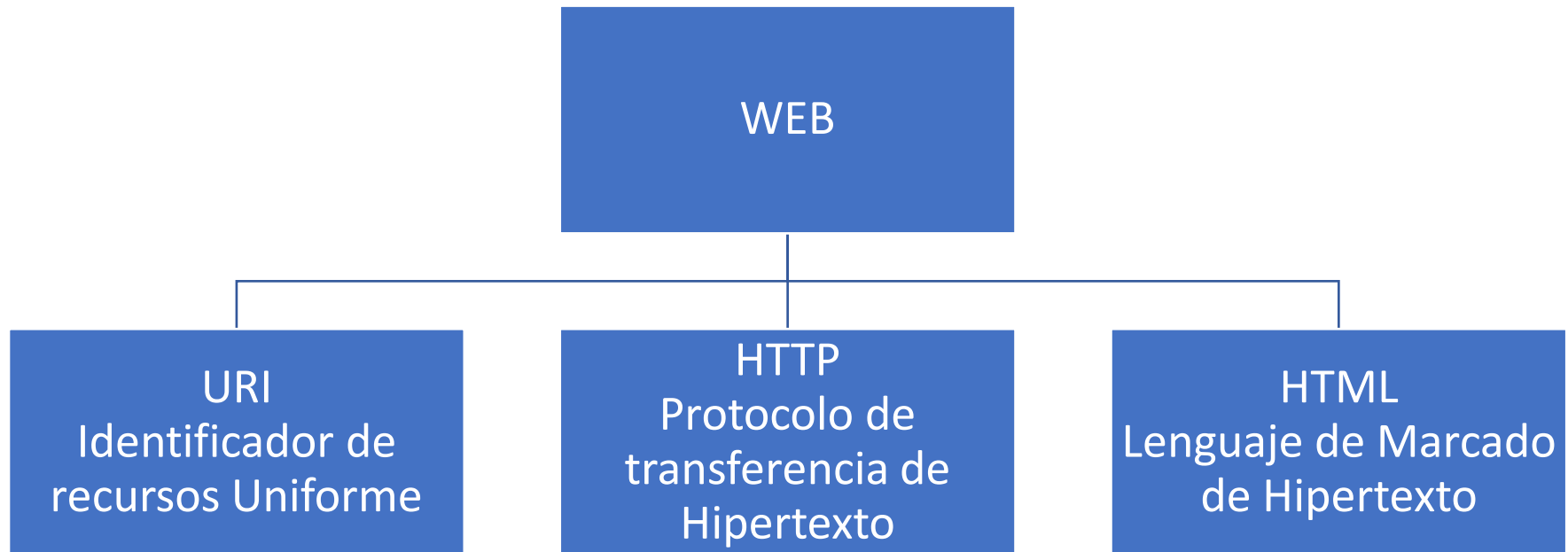
WHATWG (Web Hypertext Application Technology Working Group)

IDPF (International Digital Publishing Forum). Integrado en W3C

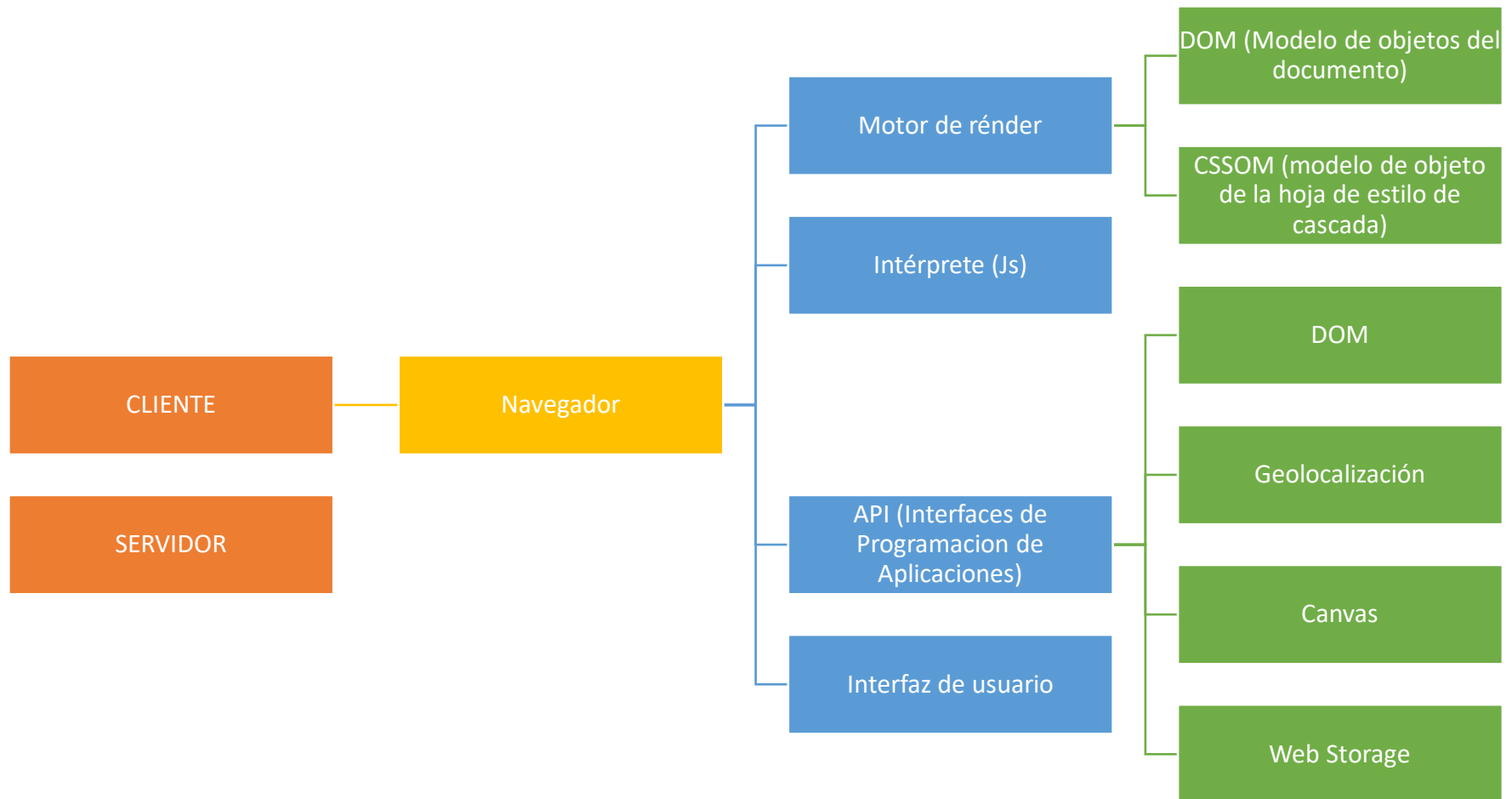
Las fases de Internet



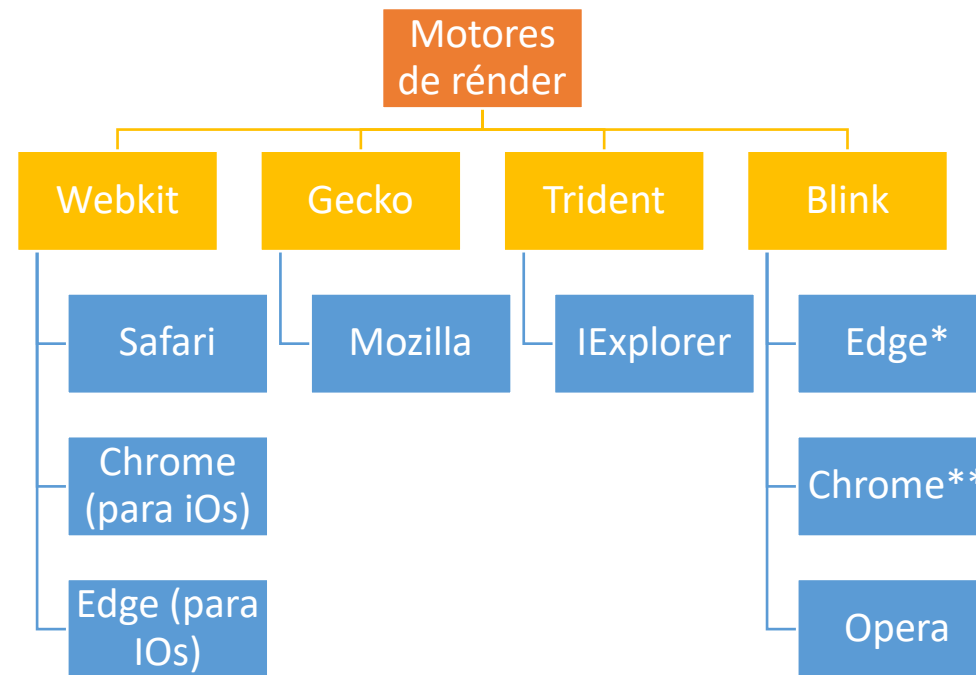
Elementos básicos



Arquitectura cliente-servidor: el navegador



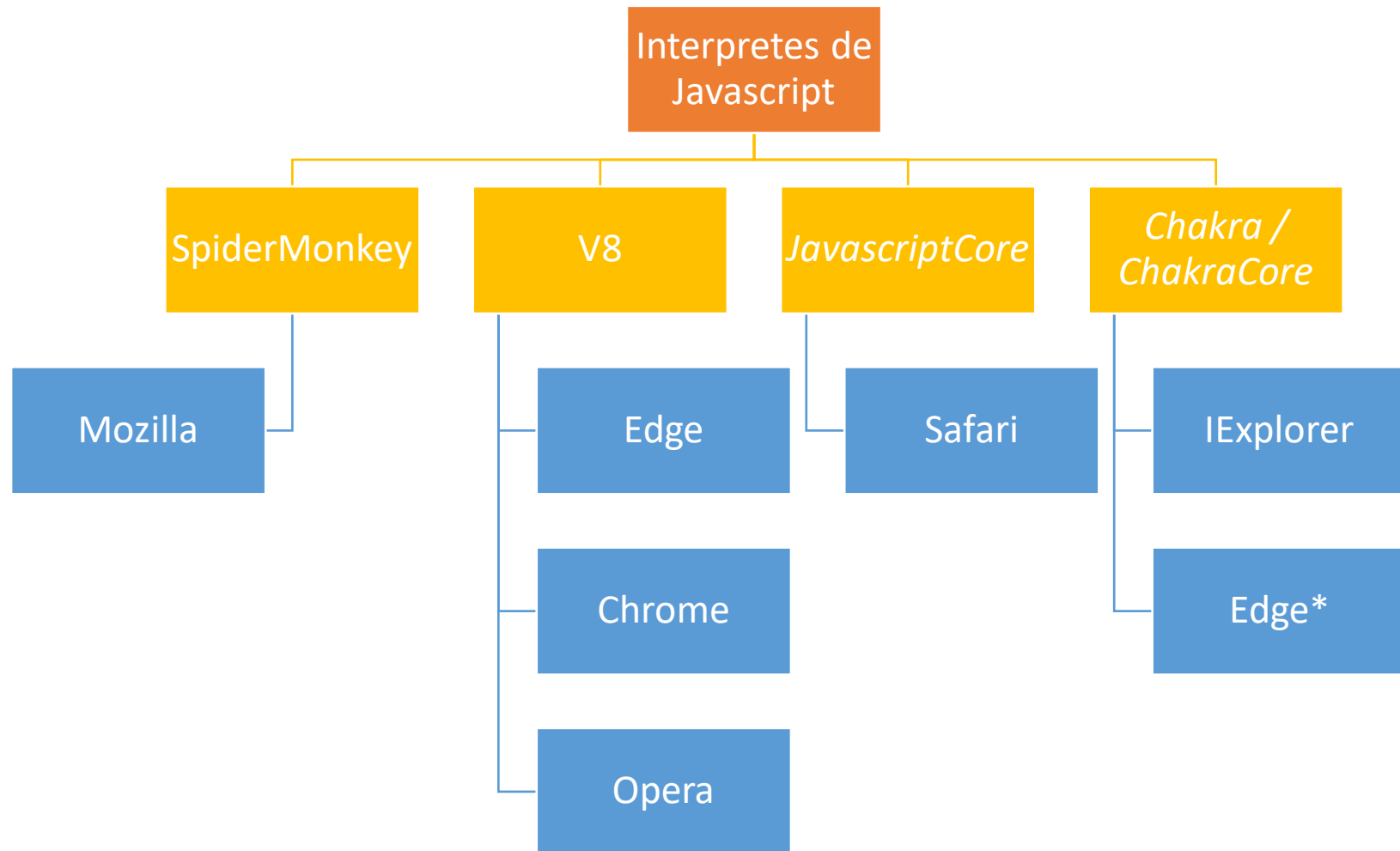
Diferentes motores de rénder



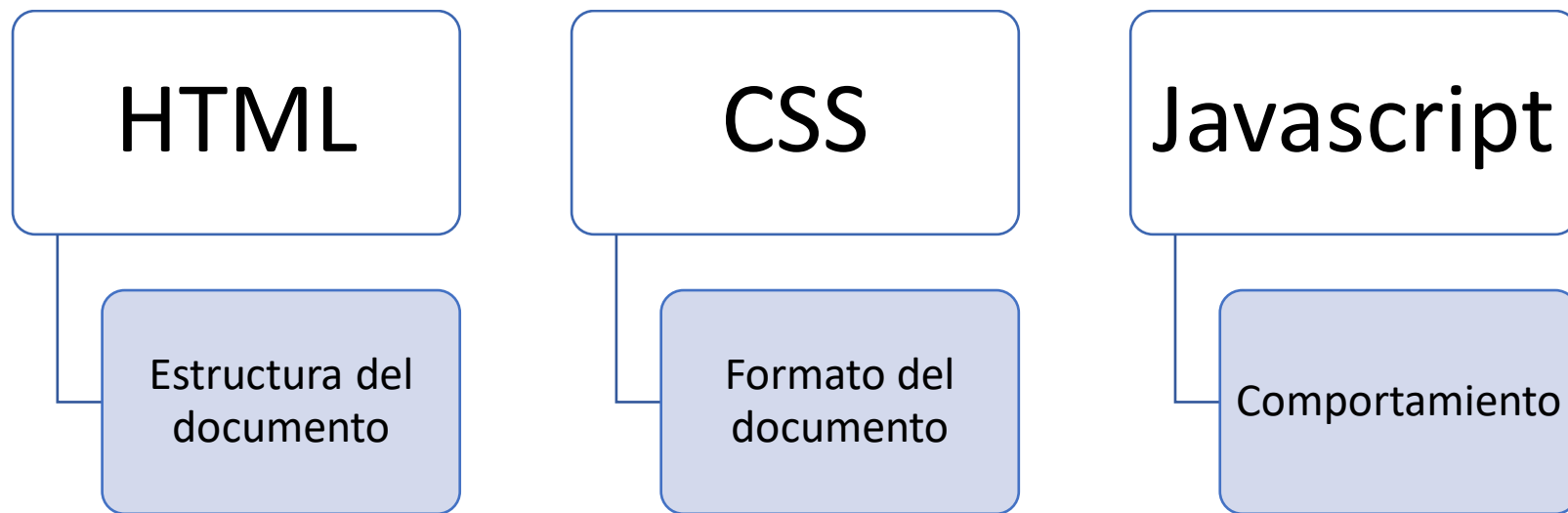
*Desde 2020, antes usaba EdgeHTML

**Google recientemente ha abandonado webkit por Blink

Intérpretes de JavaScript



Lenguajes en entorno cliente



HTML. Lenguaje de marcado de hipertexto, encargado de definir la estructura del documento y su presentación básica

CSS, siglas en inglés de *Cascading Style Sheets*, hojas de estilo en cascada, lenguaje creado en 1996 para poder dar formato adecuadamente al contenido web.

JavaScript se incorpora a la web para facilitar la interactividad y la creación de aplicaciones más complejas a partir de 1995, en el seno del navegador Netscape. Hoy estandarizado como **EcmaScript** y es también por tanto un estándar en el desarrollo web.

JavaScript

El lenguaje javascript fue creado en 1995 para integrarse como lenguaje interpretado en el navegador Netscape para posibilitar la interactividad en las páginas web. Ya en 1997 se propuso crear una estandarización del lenguaje, lo que dio lugar a ECMAScript, que actualmente es también una norma ISO y es la referencia para diferentes lenguajes de guion, incluidas las sucesivas versiones de javascript.

Aunque inicialmente Javascript es un lenguaje que tiene su entorno de ejecución en el navegador web, su uso se ha ido extendiendo también a desarrollos en servidor y actualmente Javascript se utiliza con cada vez más frecuencia en estos entornos. Por ejemplo, está muy extendido Node.js, basado en el desarrollo del motor V8 de Google.

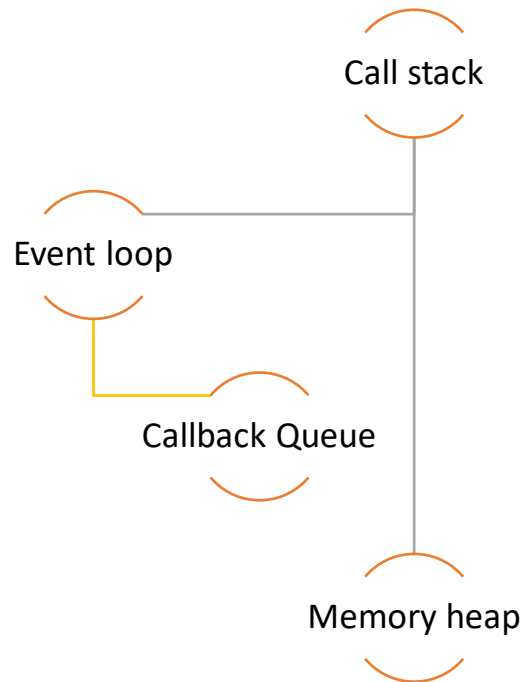
Igualmente, otros cambios se han producido desde su evolución, por ejemplo, inicialmente Javascript no era un lenguaje orientado a objetos, pero desde la versión ECMAScript 6 de 2015 se han incorporado importantes novedades entre las que se encuentra el uso de clases para la programación orientada a objetos lo que dota a este lenguaje de nuevas capacidades. Actualmente se considera por tanto un lenguaje orientado a objetos, inicialmente basado en prototipos y entre sus características que lo definen frente a otros lenguajes está el de ser un lenguaje débilmente tipado (no necesita definir el tipo de las variables y este puede variar) y dinámico (permite cambios en tiempo de ejecución).

Aunque es un **lenguaje interpretado**, en la actualidad los navegadores cuentan con un sistema **JIT (Just In Time compiler)** que “compila” para que pueda ser interpretado más rápidamente

Ejecución de JavaScript

JavaScript es en principio¹ un lenguaje **monohilo**, es decir, solo hay un hilo de ejecución: las tareas se ejecutan secuencialmente. Hay una **única pila de llamadas (Call Stack)** y tareas que esperan cola (**Callback Queue**) para pasar a la pila de ejecución cuando está vacía. De ello se encarga el bucle de eventos (**Event Loop**).

Las variables, objetos, funciones se almacenan en el almacén de memoria (**Memory Heap**).



¹ Es posible trabajar en forma multihilo en JavaScript, derivando hilos secundarios mediante la API Web Worker.

Integración de JavaScript en un documento

Cuando Javascript tiene su entorno de desarrollo en el navegador web, el código JavaScript se integra en el código de un documento HTML.

```
<html>
  <head>
    <title>Título obligatorio del
documento</title>
  </head>
  <body>
    <-- contenido del documento-->
  </body>
</html>
```

Habitualmente se introduce en el prólogo del documento HTML, es decir, dentro del bloque HEAD, aunque es posible introducirlo en cualquier parte del documento.

Tres formas de incluir código JavaScript en un documento HTML

Usando un fichero independiente con la extensión .js enlazado en el código HTML mediante la etiqueta SCRIPT

- `<script src="miCarpeta/mifichero.js" type="text/javascript"></script>`

Escribiendo directamente el código javascript en el interior del propio código HTML dentro de una etiqueta SCRIPT:

- `<script type="text/javascript">
 var a=1;
</script>`

Ejecutándolo directamente a través de un evento HTML o enlace usando javascript:

- `Lanzar mensaje`
- `<p onclick="miFuncion()">Ejecutar función</p>`

Usando el «protocolo» javascript: es posible también ejecutar JavaScript directamente en la barra de direcciones del navegador (salvo en navegadores como Firefox que lo han restringido por defecto).

Incluir alternativa con <noscript>

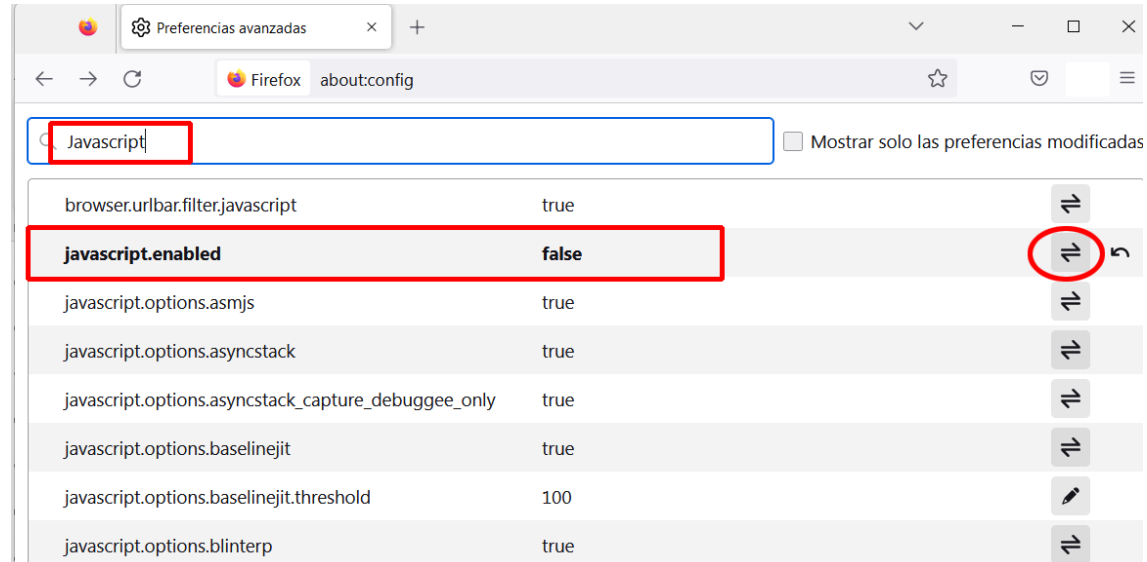
```
<noscript>
```

```
Hola, mundo
```

```
</noscript>
```

SE MOSTRARÁ CUANDO NO HAY SOPORTE DE JAVASCRIPT O ESTÁ DESHABILITADO:

Firefox: *about:config* > *JavaScript.enable* (True a False)



Chrome: Configuración > Privacidad y Seguridad > Configuración de sitios > Javascript

← JavaScript

🔍 Buscar

Los sitios suelen usar JavaScript para mostrar contenido interactivo, como videojuegos o formularios web

Comportamiento predeterminado

Los sitios usan esta opción automáticamente cuando los visitas.

☒ < > Los sitios pueden usar JavaScript

☐ No permitir que los sitios usen JavaScript

Comportamientos personalizados

Los sitios indicados a continuación tienen una configuración personalizada en lugar de la predeterminada

No pueden usar JavaScript

Añadir

No se ha añadido ningún sitio web

Pueden usar JavaScript

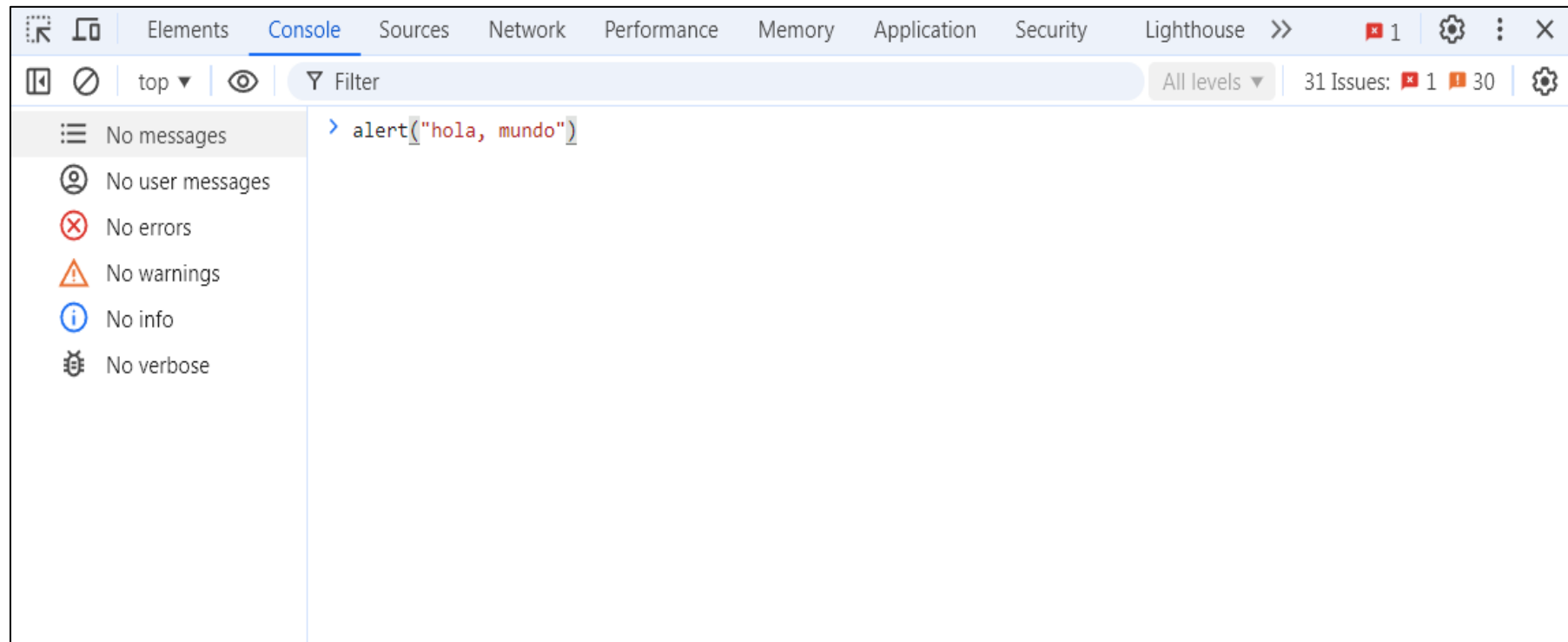
Añadir

No se ha añadido ningún sitio web

Uso de la consola del navegador

También podemos ejecutar directamente Javascript en el navegador usando la consola.

F12 en Firefox y Chrome, Ctrl+Shift+J en Edge



La consola será imprescindible tanto para ver los mensajes de error en la ejecución de nuestro código como para enviarnos información, por ejemplo, valores de variables, directamente desde nuestra aplicación: uso de `console.log`.