



UD1: Arquitecturas en la web e Introducción al lenguaje JavaScript (I)

Desarrollo Web en Entorno Cliente

María Rodríguez Fernández mariarfer@educastur.org

Al final de la clase...

- Sabrás qué **arquitectura** tienen las aplicaciones web
- Recordarás “qué pinta tiene” **JavaScript**
- Conocerás los principales **navegadores** Web y podrás compararlos
- Realizarás tu **primer script** en JavaScript en segundo curso



Arquitectura de aplicaciones Web

Nacimiento de la Web

- Tim Berners-Lee (**CERN**) (1989)
 - Presenta el World Wide Web (www)
- Primer documento formal: 1991
 - Internet existía desde los 60
- Estandarización: **Consortio W3C**
 - Organización que se responsabiliza de desarrollar y mantener los estándares Web
 - Entre sus [miembros](#) están organismos oficiales, empresas, universidades..

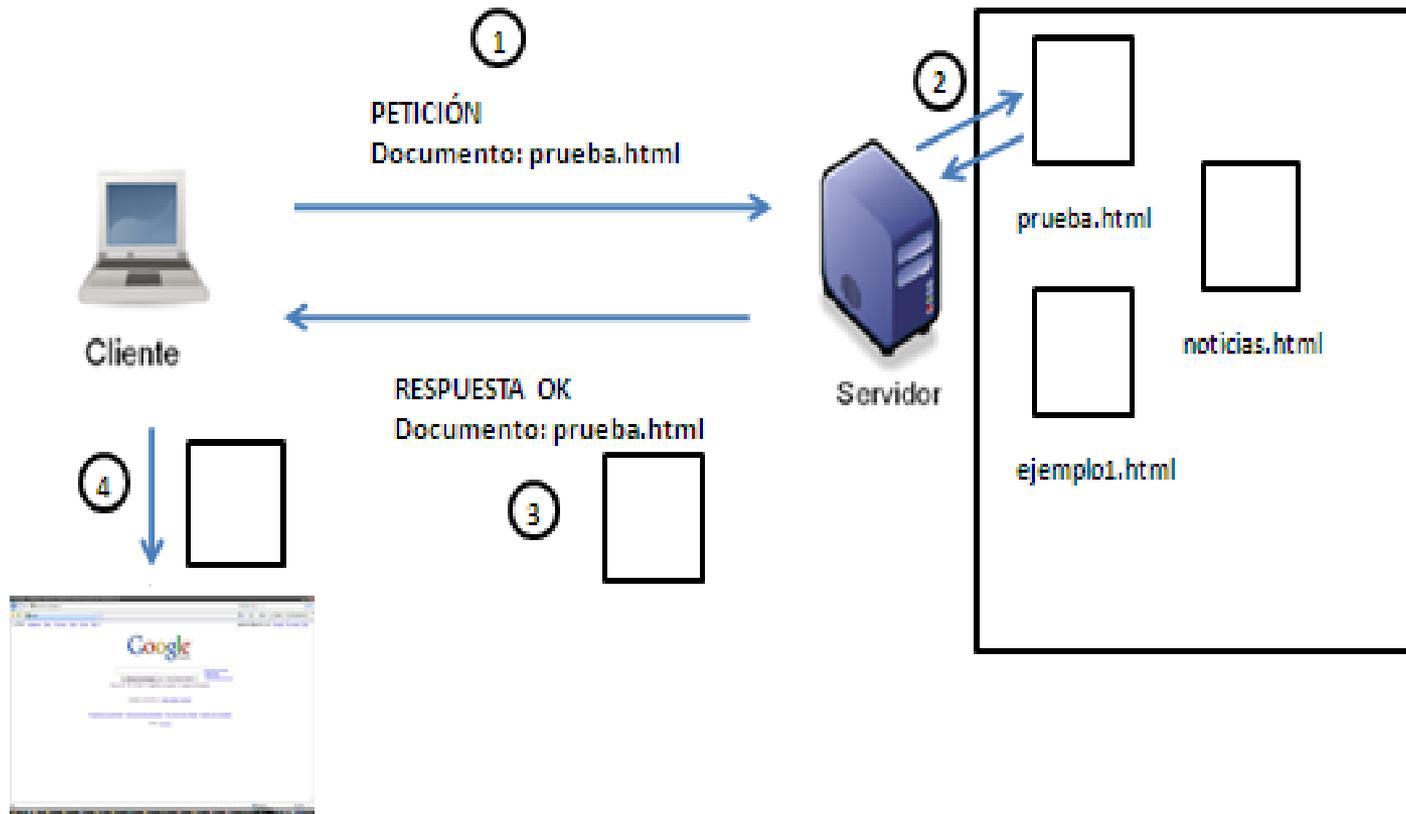


La Web



- Funciona siguiendo el modelo cliente-servidor
 - **Servidor**: Entidad que presta el servicio
 - **Ciente**: Entidad que consume el servicio
- Es un sistema que facilita el acceso a distintos recursos que ofrece Internet
 - Los recursos están situados en **servidores**
 - Los programas que nos permiten acceder a los recursos mediante una interfaz agradable son los **navegadores Web**
 - **Recurso** = Página Web, fotografía, vídeo, etc..

Petición de una página Web

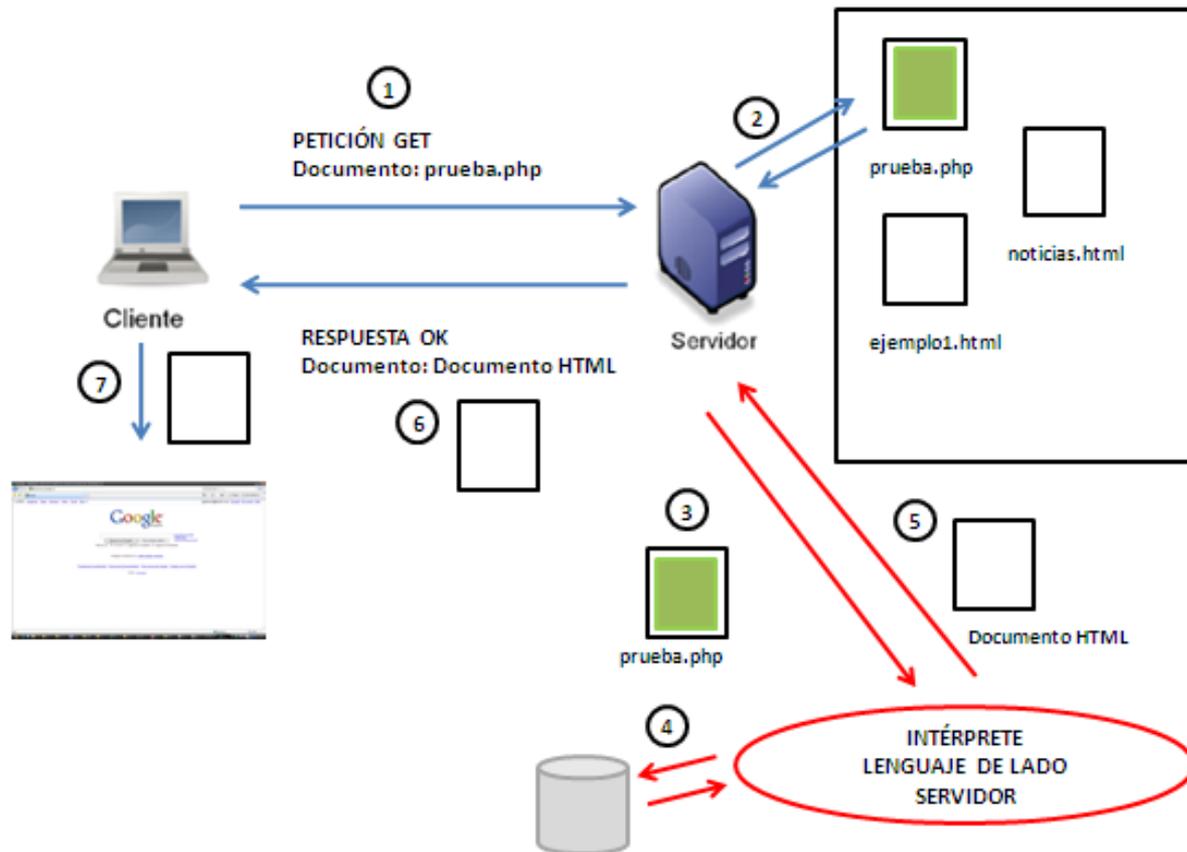


El lado servidor

- Servidor Web (hardware y software)
- Tecnologías y lenguajes de lado servidor
 - **PHP**, ASP, JSP..
 - Bases de datos



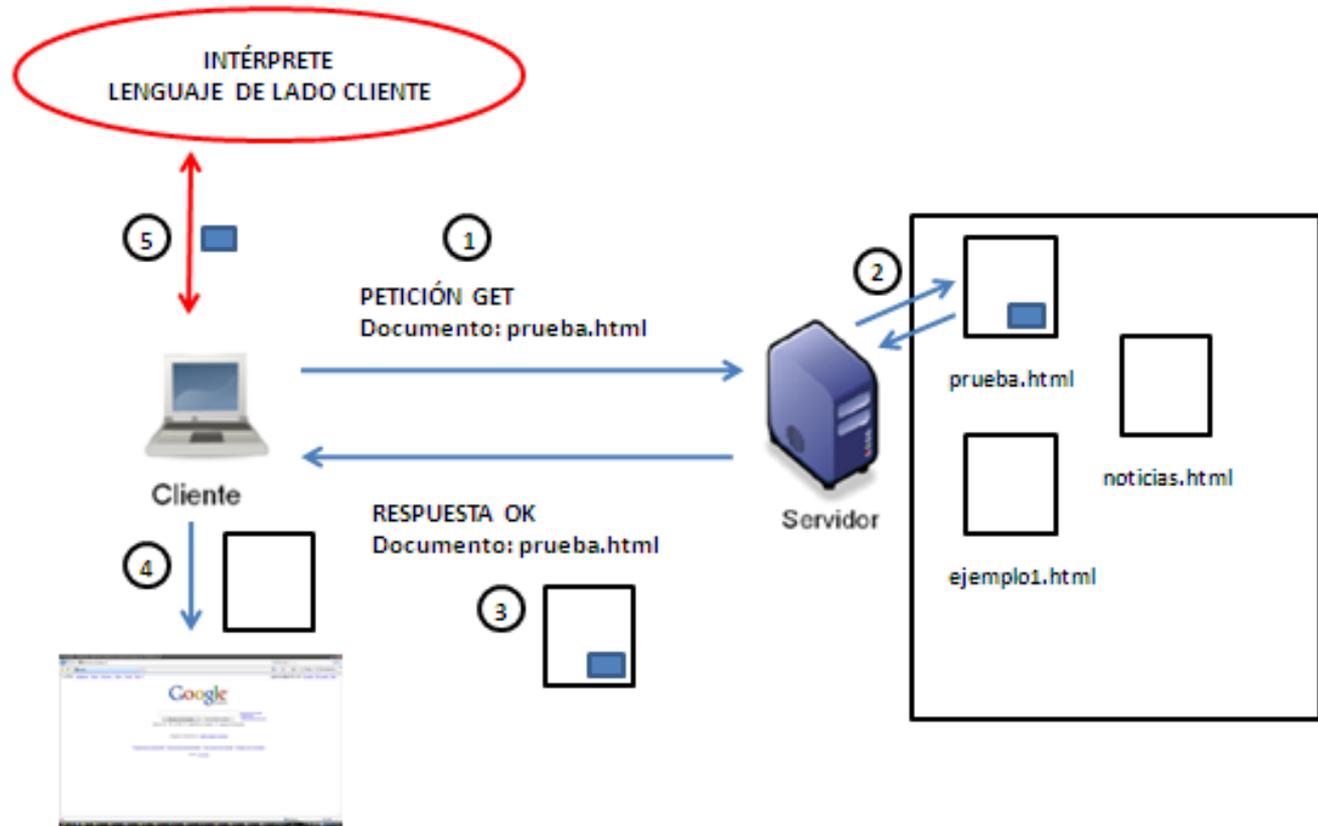
Lenguajes de lado servidor



Tecnologías Web de lado cliente

- Navegador Web
- Lenguajes entendidos por el navegador:
 - Estructura y contenido: **HTML**
 - Presentación: **CSS**
 - Comportamiento: **JavaScript**
- En el pasado, con plugin adicional: **Flash** (Adobe), **Silverlight** (Microsoft)...

Tecnologías Web del lado cliente

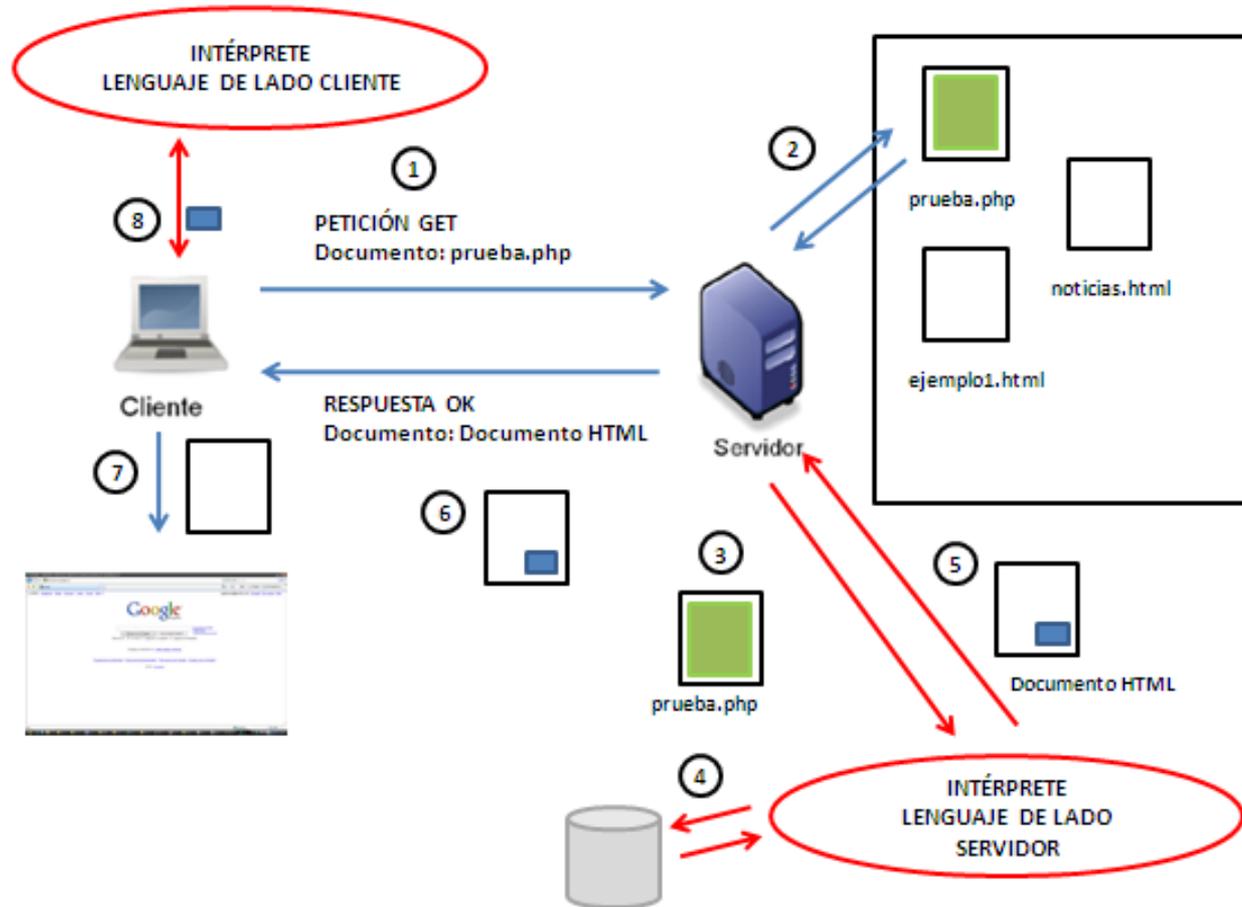


- NOTA: El cuadradito azul representa el código JavaScript

¿Cliente o Servidor?



Cliente y Servidor

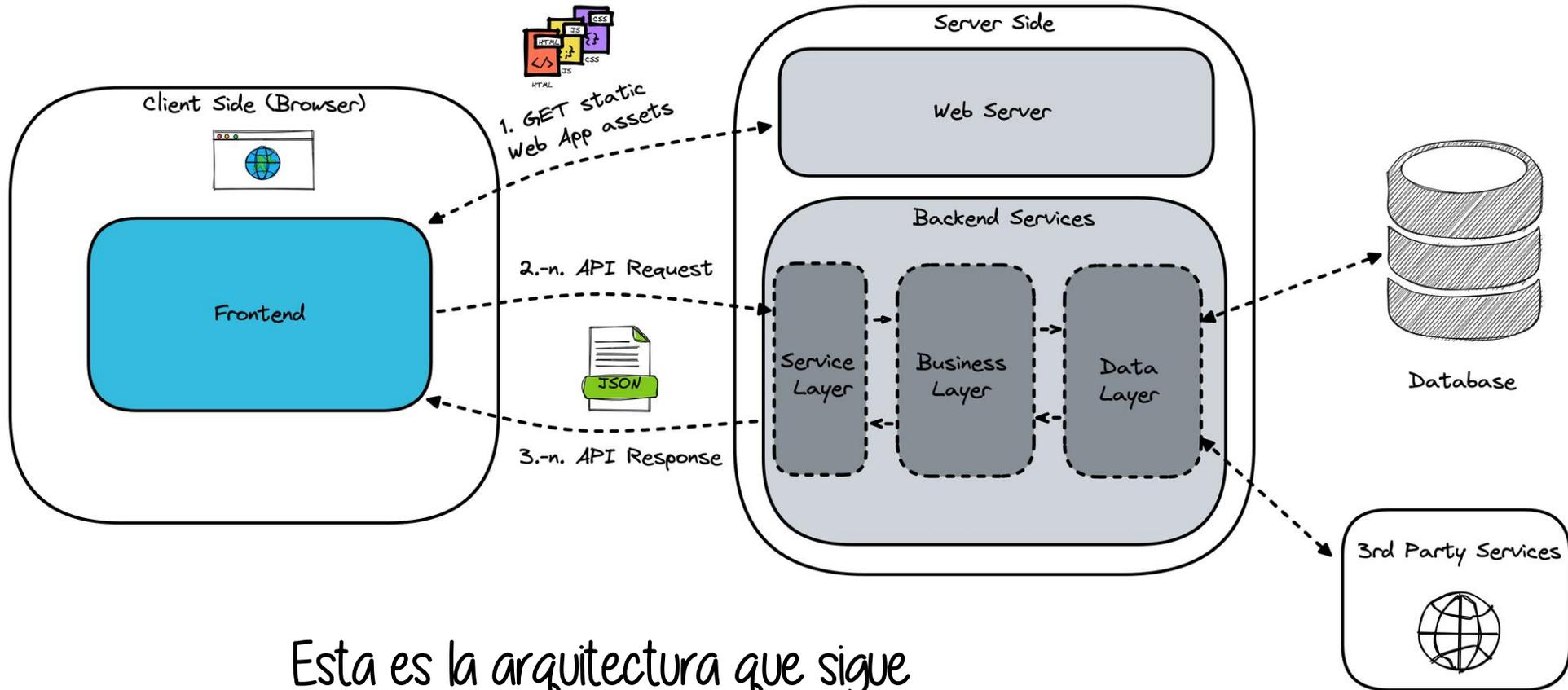


AJAX



- Es una tecnología a medio camino entre cliente y servidor
- Evita tener que hacer nuevas peticiones de la página completa para cada recurso que se pide
 - Sólo se pide la parte que ha cambiado
 - No es necesario refrescar la página completa, permitiendo aplicaciones mucho más complejas

Servicios web



Esta es la arquitectura que sigue
React, Vue, Angular, etc.



El navegador Web

¿Qué navegador es mejor?

- La elección debería hacerse en base a criterios como:
 - Rapidez
 - Rendimiento general
 - Plugins y añadidos
 - Compatibilidad
- Algunas comparativas 2023
 - [TechRadar](#)
 - [DigitalTrends](#)



Ronda rápida

- ¿Cuál es tu navegador de internet favorito?

• NAVEGADORES MODERNOS •

¿QUÉ SOMOS?



¡NAVEGADORES!



¡NAVEGADORES! ¡NAVEGADORES!

¿QUÉ QUEREMOS?



¡MAYOR VELOCIDAD!

¡MAYOR VELOCIDAD!

¡MAYOR VELOCIDAD!

¿Y CUÁNDO LO QUEREMOS?



¡AHORA MISMO!

¡AHORA MISMO!

¡AHORA MISMO!

¡NAVEGADORES!

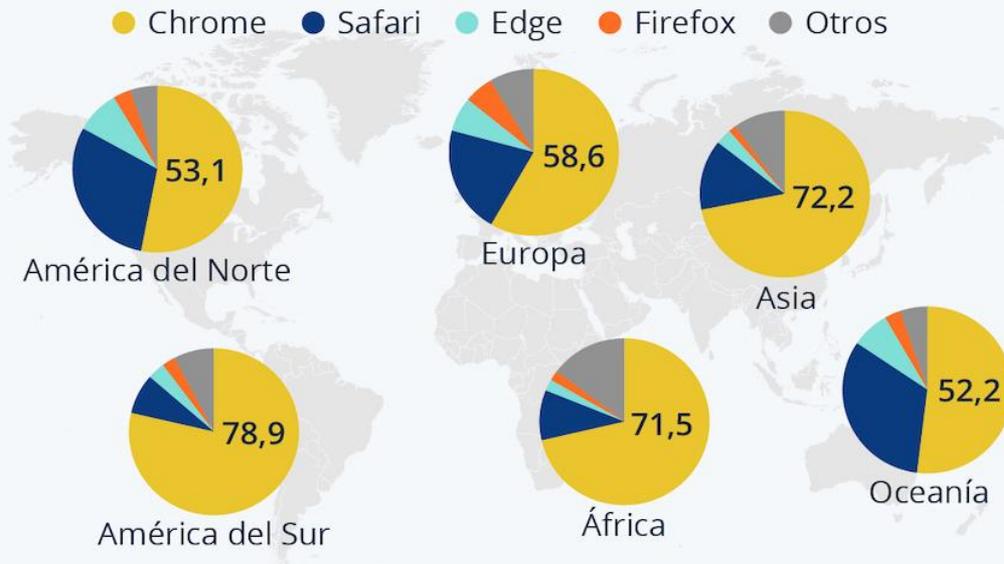


PINTZAP.COM

¿Qué navegador es el más usado?

Google Chrome se ha apoderado del mundo

Cuota de mercado* de navegadores seleccionados en agosto de 2023, por región (en porcentaje)



* En todas las plataformas, incluidos ordenadores, tablets y móviles.

Fuente: StatCounter



statista

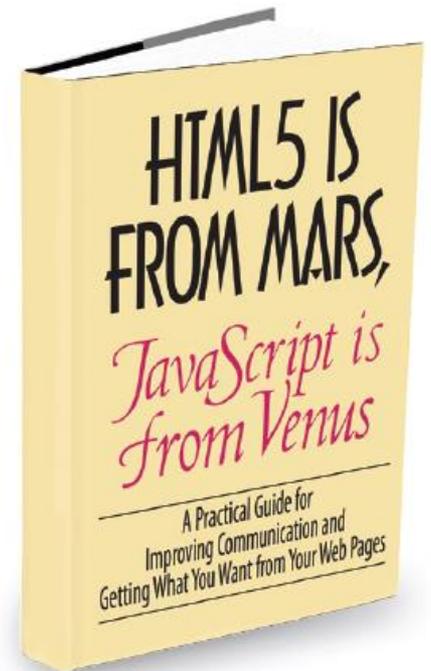


Introducción a Javascript

¿Por qué JavaScript?



- HTML5 incorpora nuevas características que ayudan a superar las limitaciones de HTML
 - Gráficos, almacenamiento de datos, etc..
- Gran parte de las funcionalidades de HTML5 se definen como bibliotecas de funciones JavaScript
 - Van de la mano, aunque sean muy distintos.



JavaScript: Algo de historia

- Originalmente creado para el navegador NetScape (1995)
 - Nombres: Mocha → LiveScript → JavaScript
 - Desarrollado en [10 días!](#)
 - Comparte el nombre con Java por razones comerciales
- JavaScript es estandarizado por ECMA desde 1997
 - Su versión “oficial” se llama ECMAScript
 - Versión más extendida es ES6 o ECMAScript 2015
 - Última versión ES2023 o ECMAScript 2023
 - Revisiones anuales
 - Es aceptado por todos los navegadores, aunque puede haber diferencias o problemas de compatibilidad (cada vez menos)



Principales características

- **Lenguaje interpretado**
 - Se ejecuta en el navegador línea a línea.
 - No necesita ser compilado.
- **Tipado débil**
 - No tenemos que indicar el tipo de las variables.
- **Tipado dinámico**
 - Podemos cambiar el tipo de las variables en tiempo de ejecución.

Ventajas

- Estándar.
- Rápido (cada vez más).
- Versátil (cada vez más).
- Muy compatible con dispositivos de todo tipo.





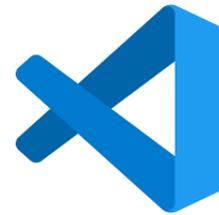
¡Manos a la obra!

Nuestro entorno de trabajo

- Navegador Google Chrome



- VS Code 1.82: Extensiones para empezar...
 - ESLint – JavaScript
 - LiveServer (opcional)
 - Spanish pack (opcional)



- Github



Javascript y HTML

- Un archivo JS siempre irá asociado a un archivo HTML que contiene la estructura de la página.
- Podemos incluir el código JS:
 - En el propio archivo HTML
 - **En un archivo externo**

JavaScript en el propio archivo

- Metemos el código dentro de la etiqueta `<script>`
 - Habitualmente en `<head>` pero puede ir también en `<body>`
 - El código se cargará en la página en el momento en que añada la etiqueta
 - No se puede acceder a los elementos de la página que no se han cargado

```
<!DOCTYPE html>
<html>
<head>
  <script> /* Lo que va aquí es código Javascript */
  alert("Hola");
</script>
</head>
<body>
Aquí iría la página
</body>
</html>
```



Archivos externos

- Es la práctica más utilizada y recomendable.
- Definimos todo el código JavaScript en un archivo externo (con extensión js)
 - Habitualmente introducimos los archivos en una carpeta diferenciada, por ejemplo js o scripts
- Enlazamos con el archivo JavaScript usando una etiqueta `<script>` e indicando la ruta del archivo
 - En HTML5 no es necesario poner el tipo MIME



```
alert("Hola");
```

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

iTu turno!

```
<!DOCTYPE html>
<html>
  <head>
    <title>DWEC - Primer script</title>
    <script src="js/holamundo.js"></script>
  </head>
  <body>
    <div>DWEC - Probando</div>
  </body>
</html>
```

ejemplo.html

```
alert("Hola mundo");
```

js/codigo.js

Herramientas de depuración

- Los principales navegadores incluyen herramientas de depuración de JavaScript que permiten realizar tareas en tiempo de ejecución:
 - Visualizar errores y avisos en tiempo
 - Inspeccionar el valor de las variables
- El manejo correcto de estas herramientas es imprescindible para el desarrollo en JavaScript

```
> texto
8
> texto2
"3"
> |
Uncaught ReferenceError: texto5 is not defined
>
```



Shift + Ctrl + J

Objeto console

- Ofrece una interfaz para comunicarnos por código con la consola
 - No es estándar, pero viene incluida en la mayor parte de los navegadores

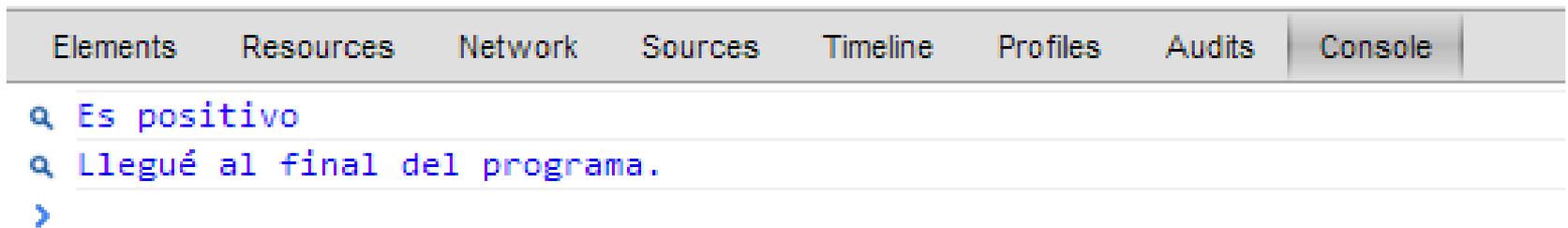
```
/* AVISOS (WARNING) */  
console.warn("Mensaje");  
  
/* ERRORES */  
console.error("Mensaje");  
  
/* INFORMACIÓN (LOG) */  
console.log("Mensaje");  
  
/* DEPURACIÓN (DEBUG) */  
console.debug("Mensaje");
```

Objeto console : Ejemplo

```
var numero=prompt("Introduzca un número positivo");

if(numero>0){
  console.debug("Es positivo");
  hazCosas();
}
else{
  console.debug("Es negativo");
  hazOtrasCosas();
}

console.debug("Llegué al final del programa");
```



PARA SABER MÁS: Ofuscadores de código

- Al ejecutarse en cliente cualquiera que visualice la página puede ver el contenido de un script
- Los [ofuscadores de código](#) hacen que el código sea difícil de interpretar por humanos sin perder funcionalidad
 - Elimina saltos de línea, espacios en blanco, cambio de nombre de variables,