

Unidad 1

Selección de arquitecturas y herramientas de programación



El desarrollo web se puede catalogar en dos partes:

- **Back-end o lado del servidor**

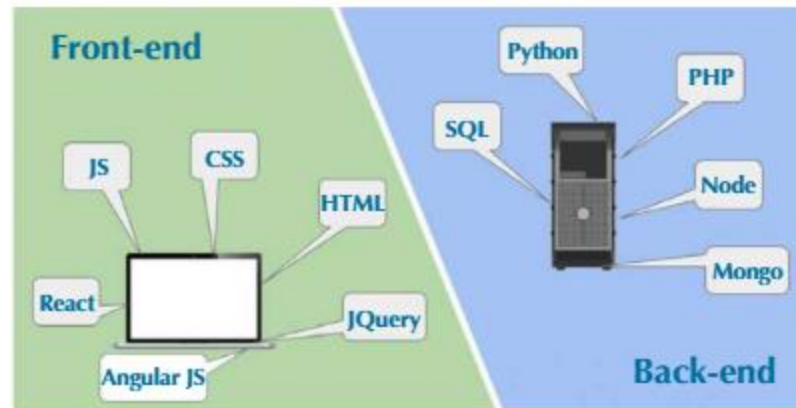
La parte no visible de la web.

Por ejemplo: bases de datos o scripts que se ejecutan en el servidor.

- **Front-end o lado del cliente**

La parte visible de la web.

Por ejemplo: código HTML, CSS, scripts que se ejecutan en el lado del cliente.



Las principales diferencias entre el desarrollo de los lados cliente y servidor son:

	Lado cliente	Lado servidor
Definición	Implica la implementación efectiva de los componentes visuales de una aplicación web.	Implica la implementación efectiva de funcionalidades de una aplicación web, donde se incluyen el acceso a datos, la administración de servidores, etcétera.
Habilidades requeridas	HTML, CSS, SASS, JavaScript...	Python, Ruby, Java, PHP...
Independencia	No puede funcionar de forma independiente excepto en el caso de sitios estáticos.	Funciona de forma independiente respecto del <i>front-end</i> .
Objetivo	Garantizar que todos los usuarios puedan acceder a la aplicación y que siga respondiendo en todos los dispositivos.	Garantizar que la aplicación se ejecute en todos los casos, sea escalable y funcione de manera eficiente con baja latencia y sin fallos.
Equipo de desarrollo	Su trabajo es diseñar y desarrollar la apariencia e interactividad de la aplicación en función de la entrada de usuario.	Su trabajo es proporcionar datos al <i>front-end</i> , vincular páginas, brindar seguridad y soporte a los usuarios.
Frameworks utilizados	AngularJS, React, vue.js...	Django, Flask, CakePHP, Laravel, Ruby on Rails...
Habilidades adicionales	Una buena comprensión del diseño de UI y UX.	Razonamiento lógico y resolución de problemas.

FRAMEWORK – conjunto de herramientas y librerías software que proporcionan numerosas utilidades para el desarrollo de aplicaciones web más escalables y sencillas de mantener, proporcionan un importante ahorro de recursos.

Los frameworks nacieron como librerías, más o menos completas, que tenían una serie de estructuras que permitían al programador tener una base para la creación y el desarrollo de sus proyectos. Hoy, son mucho más que eso, puesto que pueden utilizar lenguajes como TypeScript o JSX, los cuales luego se compilan a JavaScript.

Entre las ventajas que aportan los frameworks, están:

- El coste. Muchos de estos frameworks son de código abierto, con lo cual no hay que realizar ninguna inversión.
- Están suficientemente *probados* y su código suele carecer de errores, puesto que muchos programadores lo utilizan. Además, suelen tener un alto nivel de seguridad y rendimiento.
- Permite desarrollar mucho más *rápido*, puesto que muchas de las estructuras, clases, patrones de diseño, etc., vienen ya incorporadas en el framework.
- Cualquier persona que maneja un framework determinado *puede entender* e incorporarse de forma eficiente y efectiva a un equipo de desarrollo que lo esté utilizando en un proyecto determinado.

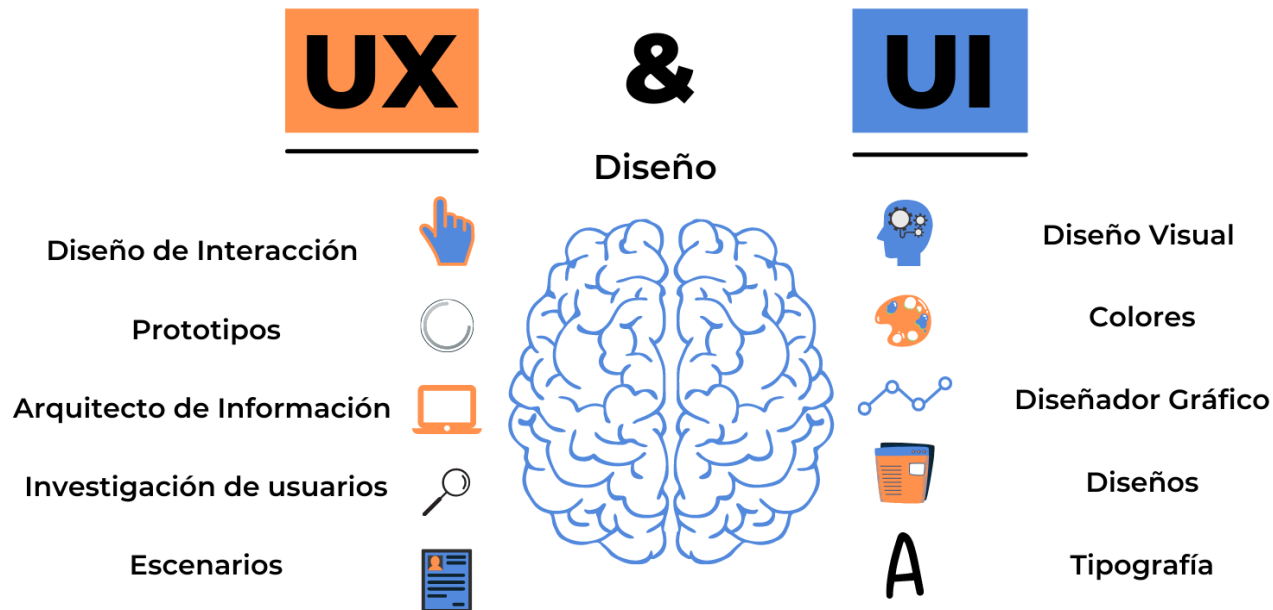
Framework

https://www.youtube.com/watch?v=l0_4tWlvHYw

UX – (User eXperience) experiencia de usuario. Busca una experiencia amigable, sencilla y lo más intuitiva posible

UI – (User Interface) interfaz de usuario. Determina la interacción del usuario con la aplicación a través de los elementos diseñados para ello.

Ambas determinan el grado de satisfacción de uso de una aplicación.



Diseño UX y UI

https://www.youtube.com/watch?v=crs1B_CgDto

2. Front-end (lado cliente)

En la parte front-end (lado cliente) prima la parte creativa y la originalidad. Es el perfil más cercano al diseñador, aunque también se trabaja en el código.

La programación se realizará en lenguajes como:

- **HTML** – no es un lenguaje de programación sino un lenguaje de marcas. Define el contenido que va a tener el documento. La función del navegador web es leer e interpretar todo el contenido, junto con las etiquetas y mostrárselo al cliente. Ventaja: cualquier navegador debería visualizar el contenido de la página web de la misma forma y con el mismo aspecto.
- **CSS** – define la presentación del documento. Es un lenguaje de diseño gráfico y su objetivo es que la página web sea atractiva al usuario. No modifica el comportamiento ni el contenido, sino sólo el aspecto de la página web.
- **JavaScript** – agrega contenido dinámico a las páginas web. Es un verdadero lenguaje de programación.

3. Scripts vs Lenguajes de Programación

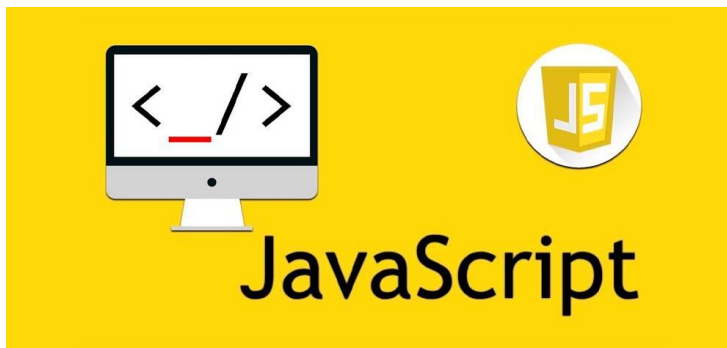
Las principales diferencias entre los scripts y los lenguajes de programación son:

SCRIPTS	LENGUAJES DE PROGRAMACIÓN
Lenguajes interpretados, no necesitan ser compilados	Se suelen compilar
Utilizan componentes ya preexistentes	A veces, se empiezan a desarrollar desde 0
Se suelen incrustar dentro de otros programas	Se pueden ejecutar de forma independiente
Se ejecutan línea a línea. Pueden producirse muchos errores en ejecución	Se pueden ejecutar en bloques
No generan un fichero ejecutable	Generan un fichero ejecutable
Diseñados para ser fáciles de utilizar y programar	No siempre son fáciles de utilizar y programar
Ejemplos: JavaScript, Shell, Perl, PHP, Python, Ruby, etc.	Java, C, C++, Swift, Pascal, etc.

4. Introducción a JavaScript

¿Por qué elegir JavaScript?

- Es muy fácil de implementar: tan solo es necesario colocar el código en un documento HTML y decirle al navegador que es JavaScript.
- Funciona en todos los navegadores que usan los usuarios de la web, incluso cuando están desconectados.
- Permite crear interfaces altamente amigables que mejoran la experiencia del usuario y brindan mucho dinamismo, sin tener que esperar a que el servidor reaccione y muestre otra página.
- Puede cargar contenido de forma asíncrona en el documento si el usuario lo necesita y cuando lo necesite, sin recargar toda la página.
- Puede comprobar lo que es posible hacer en un navegador y reaccionar en consecuencia.
- Puede ayudar a solucionar problemas de incompatibilidades entre navegadores y corregir problemas de diseño con CSS.



Historia de JavaScript

<https://www.youtube.com/watch?v=NmaHJxanMcA>

4. Introducción a JavaScript

Actualmente, las empresas no programan en JavaScript, sino que se basan en un framework de este.

Los frameworks de JavaScript más utilizados son:

- **ReactJS** – framework creado por Facebook que permite realizar aplicaciones web de una forma rápida y eficiente renderizando (dibujando) los componentes del front-end de una forma sencilla y eficaz. Utiliza programación orientada a objetos.
- **AngularJS** – framework creado y mantenido por Google. Ha pasado de ser una librería a ser una plataforma de desarrollo. No es fácil de aprender. Actualmente se programa en TypeScript, superconjunto de JavaScript.
- **Vue.js** – framework de gran ligereza y velocidad de ejecución. Es algo más sencillo de aprender que Angular.

Existen otros frameworks como EmberJS, BackboneJS, MeteorJS, Aurelia.js, Polymer o Mithril.js. Escoger un framework es un proceso importante, puesto que el objetivo es elegir el más popular para que tenga un soporte más amplio, el que tenga más futuro, el más rápido de ejecución, el más fácil de aprender. Dicha elección muchas veces es complicada, ya que la tecnología cambia de forma muy rápida.

¿Qué framework JS elegir?

https://www.youtube.com/watch?v=l0_4tWlvHYw

9 frameworks JS

<https://raygun.com/blog/popular-javascript-frameworks/>

5. Herramientas para el desarrollo en JavaScript

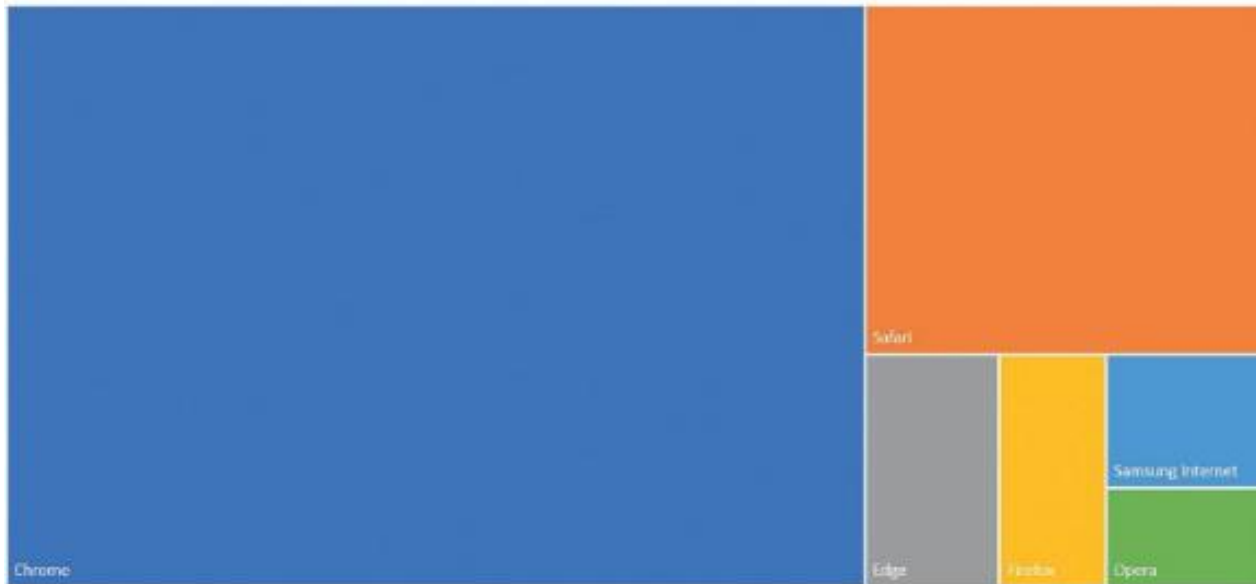
Las herramientas imprescindibles son:

1. **NAVEGADOR** – sirve para ejecutar las aplicaciones en su contexto. Según el porcentaje de uso, Google Chrome tiene una posición dominante:

Chrome	Safari	Edge	Firefox	Samsung	Opera
65,87%	18,62%	4,12%	3,26%	2,88%	2,12%

Cuota de mercado de navegadores - Junio 2022

■ Chrome ■ Safari ■ Edge ■ Firefox ■ Samsung Internet ■ Opera



A la hora de programar los desarrolladores deben trabajar con varios navegadores. Si sólo se centran en Chrome se pierden un 35% de la cuota de mercado.

Por otra parte, optimizar las aplicaciones para todos los navegadores, es técnicamente muy complicado y tiene poco sentido ya que el esfuerzo no compensa con el número de usuarios que se alcanza.

Se recomienda probar para los 4 navegadores con mayor cuota de mercado, optimizándola todo lo posible para el navegador dominante.

También es fundamental probar diferentes configuraciones del navegador para ajustarlo al mayor número de tamaños de pantalla posibles, para ello, se pueden utilizar extensiones de los navegadores (plugins).

5. Herramientas para el desarrollo en JavaScript

2. **EDITOR DE CÓDIGO** – herramienta que se utiliza para escribir el código y que incluyen características que facilitan el trabajo de los desarrolladores como: comprobación de errores, autocompletado, resaltado de la sintaxis, etc.

Como recomendación de algunos editores muy populares pueden incluirse los siguientes:

- Editores de código: Atom, Sublime Text, Bracktes, Coda...
- Entornos de desarrollo integrados (IDE): Visual Studio, Eclipse, Netbeans, Webstorm...
- Editores *online*: CodePen, JSFiddle, JS Bin...



La mejor forma de elegir un editor de código es probar todos los que se pueda y así poder tomar una decisión.

No es necesario centrarse en un único editor, se pueden utilizar varios si con eso se está cómodo y se es más productivo.

5. Herramientas para el desarrollo en JavaScript

- 3. INTÉRPRETE DE JAVASCRIPT** – por la naturaleza del JavaScript no es necesario realizar ninguna instalación adicional para el intérprete ya que viene integrado en los navegadores web.

Debemos comprobar que esté habilitado:

- **Chrome:** entrar en la **Configuración**, seleccionar **Seguridad y privacidad**, clicar en **Configuración de sitios** y después en **JavaScript** y finalmente seleccionar **Los sitios pueden usar JavaScript**.
- **Safari:** acceder a **Herramientas**, clicar en **Preferencias**, acceder a **Seguridad** y activar la casilla etiquetada como **Activar JavaScript**.
- **Edge:** acceder al menú de **Opciones** y desplegar **Mostrar opciones avanzadas**. En la sección de **Privacidad** clicar en **Configuración de contenido**. Navegar hasta la sección **JavaScript** y finalmente clicar en **Permitir que todos los sitios ejecuten JavaScript (recomendado)**.
- **Firefox:** hacer clic en **Herramientas**, luego en **Opciones** y en la pestaña **Contenido** hacer clic en la casilla **Activar JavaScript**.

En todos los casos, debe reiniciarse el navegador para tener un intérprete que es capaz de ejecutar los programas JavaScript.

6. Herramientas de programación en JavaScript

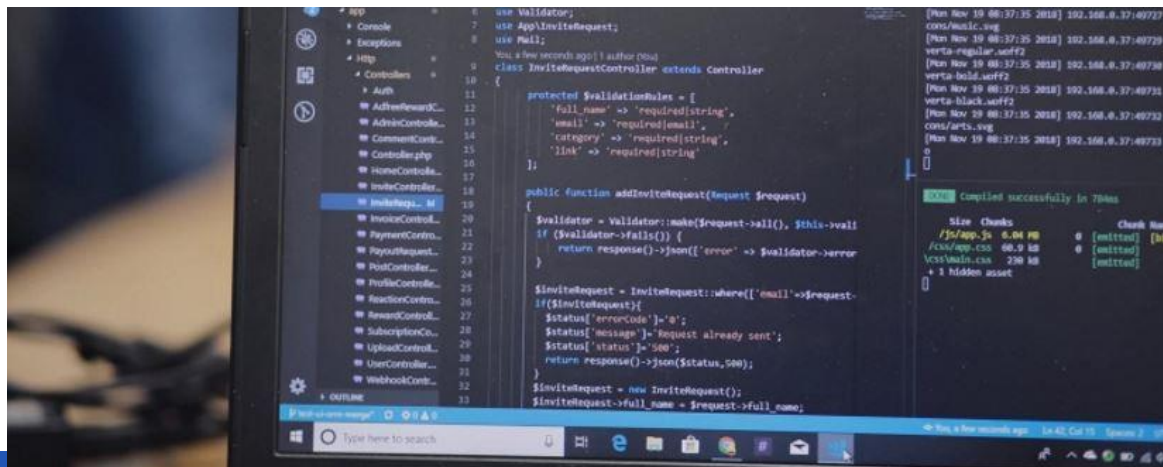
Existen múltiples alternativas a la hora de elegir una herramienta de programación en JavaScript.

Se puede usar desde un simple editor de texto, herramientas online o un IDE (entorno integrado de desarrollo).

1. **HERRAMIENTAS ONLINE** - <https://www.tutorialspoint.com/codingground.htm>

2. **IDE CON SISTEMA DE CONTROL DE VERSIONES –**

- **ATOM** - <https://github.com/atom>
- **SUBLIME TEXT** - <https://www.sublimetext.com/>
- **VISUAL STUDIO CODE** - <https://code.visualstudio.com/>



7. Integración de JavaScript en HTML

JavaScript se complementa al código HTML de una página web.

Existen dos opciones:

- Integrar el código JavaScript dentro de los archivos HTML
- Tener separado del HTML el código JavaScript en archivos con extensión js



Lo más limpio y eficaz es tener el código JavaScript fuera de los archivos HTML por las ventajas que ofrece.

7. Integración de JavaScript en HTML

ACTIVIDAD 1: Ejemplo JavaScript

Ejemplo de JavaScript en ficheros js separados

Contenido del archivo *index.html*:

```
<!DOCTYPE html>
<html>
<title>Myfpschool</title>
<head>
<script src="script.js"></script>
</head>
<body>
</body>
</html>
```

A continuación, se muestra el contenido del archivo *script.js*:

```
function diAlgo()
{
    alert("hola");
}
diAlgo();
```

Ejemplo de JavaScript con código dentro del HTML

Detalle del archivo *index.html*:

```
<!DOCTYPE html>
<html>
<title>Myfpschool</title>
<head>
<script>
    function diAlgo()
    {
        alert("Hola");
    }
</script>
</head>
<body>

<script>
diAlgo();
</script>

</body>
</html>
```

- ✓ El código JavaScript está contenido dentro de las etiquetas `<script>` y `</script>`. El código JavaScript se puede colocar tanto dentro de la etiqueta `<head>` como dentro de la etiqueta `<body>`. Nuestro consejo es que se sitúe en el mismo sitio. No es buena práctica diseminar el código por toda una página porque luego es imposible de entender y mantener. Es posible ver versiones antiguas de JavaScript con etiquetas del tipo: `<script type="text/javascript">`.

7. Integración de JavaScript en HTML

Como se puede observar, aunque el resultado es el mismo, la forma de distribuir el código HTML y el JavaScript no es igual. El mejor consejo es utilizar archivos separados para HTML y JavaScript, salvo que las líneas de código de este último sean mínimas y no vayan a ser modificadas prácticamente nunca.

Las ventajas de tener el JavaScript en un fichero separado son que las páginas cargarán mucho más rápido, se independiza el HTML del código y, como se puede adivinar, el JavaScript será mucho más fácil de mantener.

Los proyectos suelen tener los scripts en una carpeta aparte de nombre `script` o `js` para tener los archivos más ordenados.



1. Modificación del contenido de una página web

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Modificando el código HTML</h1>
5     <p id="prueba">Modificando el contenido</p>
6     <button type="button" onclick="document.getElementById('prueba').innerHTML = 'CAMBIANDO el contenido!'">¡Dale!</button>
7   </body>
8 </html>
```

Modificando el código HTML

Modificando el contenido

¡Dale!



Modificando el código HTML

CAMBIANDO el contenido!

¡Dale!

2. Cambiar atributos de objetos HTML

```
1  <!DOCTYPE html>
2  <html>
3      <body>
4          <h1>Cambio de imágenes con JavaScript</h1>
5          
6          <p>Haz click sobre la imagen para cambiarla</p>
7          <script>
8              function cambiaPic() {
9                  var image = document.getElementById('imgJS')
10                 if (image.src.match("js")) {
11                     image.src="img/html.png";
12                 } else {
13                     image.src="img/js.png";
14                 }
15             }
16         </script>
17     </body>
18 </html>
```

Cambio de imágenes con JavaScript

JavaScript



Haz click sobre la imagen para cambiarla



Cambio de imágenes con JavaScript

HTML



Haz click sobre la imagen para cambiarla

3. Cambiar el estilo CSS

```
1  <!DOCTYPE html>
2  <html>
3      <body>
4          <h1>Cambio de estilos CSS</h1>
5          <p id="mytxt">¡¡Hola clase!!</p>
6          <button type="button" onclick="CambioCss()">¡Dale!</button>
7          <script>
8              function CambioCss() {
9                  var x = document.getElementById("mytxt");
10                 x.style.fontSize = "25px";
11                 x.style.color = "red";
12             }
13         </script>
14     </body>
15 </html>
```

Cambio de estilos CSS

¡¡Hola clase!!

¡Dale!



Cambio de estilos CSS

¡¡Hola clase!!

¡Dale!

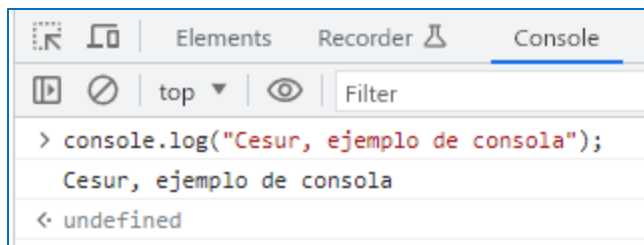
9. Comunicación de JavaScript con el exterior

Existen varias formas para que el código JS se comunice con el usuario:

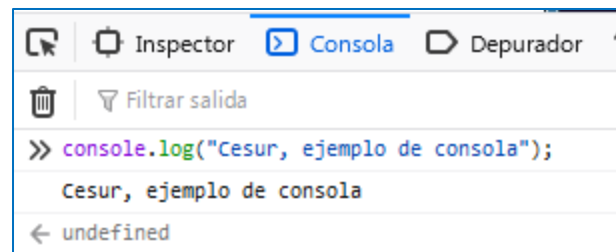
1) ESCRIBIR EN LA CONSOLA DEL NAVEGADOR UTILIZANDO `console.log()`

Esta opción sólo la pueden utilizar los desarrolladores.

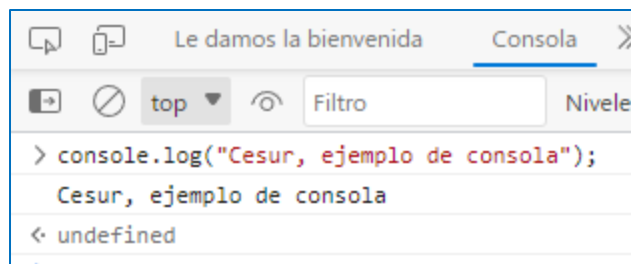
Para acceder a la consola, basta con pulsar F12 y elegir Consola en su navegador.



Consola Chrome



Consola Firefox



Consola Edge

9. Comunicación de JavaScript con el exterior

2) ESCRIBIR EN EL HTML UTILIZANDO EL ATRIBUTO `innerHTML`

```
1  <!DOCTYPE html>
2  <html>
3      <body>
4          <p>El resultado de la suma de 5 + 6 es:</p>
5          <p id="texto"></p>
6          <script>
7              document.getElementById("texto").innerHTML = 5 + 6;
8          </script>
9      </body>
10 </html>
```

El resultado de la suma de 5 + 6 es:

11

9. Comunicación de JavaScript con el exterior

3) GENERAR DIRECTAMENTE HTML UTILIZANDO EL MÉTODO `document.write()`

```
1  <!DOCTYPE html>
2  <html>
3      <body>
4          <p>El siguiente texto está escrito con JS:</p>
5          <script>
6              document.write("<h2>¡¡Hola clase!!</h2>");
7          </script>
8      </body>
9  </html>
```

El siguiente texto está escrito con JS:

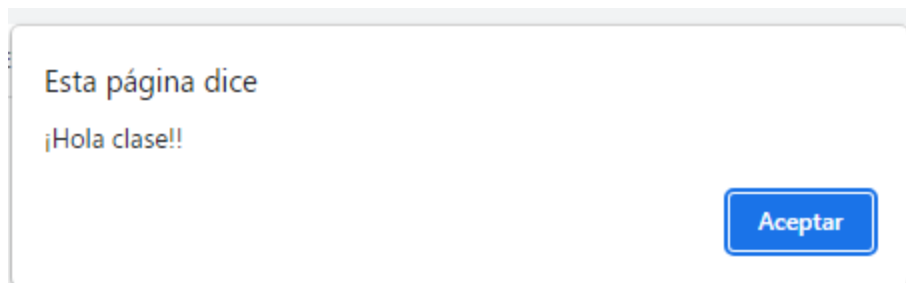
¡¡Hola clase!!

9. Comunicación de JavaScript con el exterior

4) GENERAR UN MENSAJE DE ALERTA UTILIZANDO EL MÉTODO `window.alert()` o `alert()`

Muestra un diálogo emergente en el navegador

```
1  <!DOCTYPE html>
2  <html>
3      <body>
4          <p>El siguiente texto está escrito con JS:</p>
5          <script>
6              window.alert("¡Hola clase!!");
7          </script>
8      </body>
9  </html>
```



Gracias

