



Chrome DevTools



Contenido

1.  Introducción a Chrome DevTools
2.  Abrir DevTools
3.  Elements Panel
4.  Console Panel
5.  Network Panel
6.  Lighthouse
7.  Application Panel
8.  Tips Avanzados

★ 1. Introducción a Chrome DevTools

Chrome DevTools es un conjunto de herramientas de desarrollo web integradas en Google Chrome que nos permite analizar el front (HTML, CSS y JS) de las páginas web. Entre muchas cosas podemos hacer:

-  Inspeccionar y modificar el código en tiempo real (No se guarda)
-  Depurar problemas rápidamente
-  Analizar el rendimiento
-  Simular dispositivos móviles

| El resto de navegadores tienen una aplicación parecida

2. Abrir DevTools

Hay varias formas de acceder a las DevTools:

-  Acceso rápido:
 - Mac: Cmd + Opt + I
 - Windows/Linux: F12 o Ctrl + Shift + I
-  Encima de un elemento: Click derecho → "Inspeccionar"

¿Tenemos abierto el inspector de elementos? ¿Controlamos dónde aparece (Abajo, derecha, ventana separada)?



3. Elements Panel (A)

Herramienta principal para inspeccionar y modificar el DOM:

- Inspeccionar y modificar HTML en tiempo real.
 - Buscar
 - Abrir y cerrar bloques
 - Modificar atributos (nombre y valor)
 - Añadir elementos

| ¿Has editado el HTML de alguna página? Comparte un cambio

🔍 3. Elements Panel (B)

- 🔍 Ver y editar CSS:
 - Añadir/borrar propiedades (colores, alineaciones, tamaños, etc.)
 - Panel Computed: Analizar todas las capas de CSS
 - Experimentar con Flexbox/Grid (En apple.com los div del nav puedo ajustar)
 - Control de animaciones ([Ver en W3S](#)) Activar en los circulitos y ver los pasos
- 💚 Selectores: De etiqueta, único, de clase, de atributo. ¿Cuándo?
- Estados de elementos: Pseudoselectores y pseudoclases (:hover, :active) (En apple.com forzar el estado del button buy al hover)

| ¿Has editado el CSS de alguna página? Comparte un cambio

4. Console Panel

Centro de diagnóstico y debugging:

-  Ver errores y warnings
-  Ejecutar JavaScript en vivo:
`alert('Hola'); body =
document.getElementsByTagName('body'); document.location.host =
'jorgebenitezlopez.com' // redirige la página`
-  Usar `debugger;` para pausar la ejecución, con temporizado:
`setTimeout(function(){ debugger; }, 3000);`

¿Has ejecutado algo de JS? ¿Has visto la consola de instagram? ¿Has cambiado contenido de una página dinámicamente? ¿Entiendes la diferencia entre un método y una propiedad?

5. Network Panel

Analiza el rendimiento de red:

-  Timeline de requests
-  Tamaño de recursos
-  Tiempos de carga. Diferencia entre DOMContentLoaded y Load (Todo el contenido)
-  Simular conexiones lentas con Throttling, simular sin caché
-  Detalles de cada petición. (Por ejemplo para ver qué está devolviendo el back y a veces quiero ver el Preview)

| ¿Has verificado el tiempo real de carga de una página sin cache y estrangulando la conexión?

⚡ 6. Lighthouse

Diferentes tipos de auditoría, te dice en qué falla y referencias para solucionar el problema:

- Performance
- Accessibility
- Best practices
- SEO

| ¿Has auditado una página web? ¿Has consultado los errores?



7. Application Panel

Gestión de datos del lado del cliente (Antes de que el usuario cree una cuenta podemos guardar y recuperar información del usuario):

- 🍪 Cookies (Revisar instalación y gestión. Antes de meter nada debería de avisar)
- 💾 Local Storage vs Session Storage (Diferencia clave: Local persiste a la sesión)
- 📦 Session Storage

| ¿Revisa que no te hayan instalado nada que no hayas autorizado? ¿Has instalado algo?

8. Tips Avanzados

-  Practica con el modo responsive (Hacer pruebas, botón de arriba a la izquierda)
-  Modificar sensores para testing (Tres puntos > More tools > Sensors)
-  Verificar el contraste de colores (En apple en el botón con background-color revisar el color)
-  CSS Overview: Identify potential CSS improvements (Tres puntos > More tools > CSS Improvements)
-  Coverage: Encuentra CSS y JS no utilizado: Analiza código cargado vs utilizado y Optimiza el tamaño de tus páginas (Tres puntos > More tools > Coverage)

 ¡Gracias!