

# Referencia Rápida de Chrome DevTools

Elementos, Consola, Red, Rendimiento, depuración

## Elementos

<b>Inspeccionar y editar</b>	
<b>Right-click -&gt; Inspect</b>	Abrir panel Elementos para el elemento
<b>Double-click tag/attribute</b>	Editar HTML en línea
<b>Delete key</b>	Eliminar nodo seleccionado
<b>Ctrl+Z</b>	Deshacer cambio en el DOM
<b>H key</b>	Alternar visibilidad del elemento seleccionado
<b>Drag node</b>	Mover elemento en el árbol DOM

## Panel Estilos

<b>element.style {}</b>	Agregar estilos en línea al elemento
<b>Click property value</b>	Editar valor CSS en vivo
<b>Checkbox next to rule</b>	Activar/desactivar propiedad CSS
<b>:hov button</b>	Forzar pseudo-estados del elemento (:hover, :focus)
<b>.cls button</b>	Agregar/quitar clases CSS
<b>Color swatch</b>	Abrir selector de color
<b>Computed tab</b>	Ver valores CSS calculados finales

## Consola

<b>API de Consola</b>	
<pre>console.log("info"); console.error("error"); console.group("label"); console.time("t");</pre>	<pre>console.warn("warning"); console.table(arrayObj); console.groupEnd(); console.timeEnd("t");</pre>

## Utilidades de Consola

<b>\$0</b>	Elemento actualmente seleccionado en Elementos
<b>\$('#sel') / \$\$('sel')</b>	Atajo de querySelector / querySelectorAll
<b>copy(obj)</b>	Copiar objeto como cadena al portapapeles
<b>clear()</b>	Limpiar salida de la consola
<b>monitor(fn)</b>	Registrar llamadas a la función fn
<b>monitorEvents(el, 'click')</b>	Registrar eventos en el elemento
<b>keys(obj) / values(obj)</b>	Claves / valores del objeto
<b>\$_</b>	Resultado de la última expresión evaluada

## Filtrado

<b>Log levels dropdown</b>	Filtrar por verbose/info/warn/error
<b>Filter text box</b>	Buscar en la salida de la consola
<b>-url:extension</b>	Excluir mensajes por URL de origen
<b>context: dropdown</b>	Filtrar por contexto iframe/worker

## Red

### Funciones del panel Red

<b>Filter bar</b>	Filtrar por tipo: XHR, JS, CSS, Img, Doc, WS
<b>Search (Ctrl+F)</b>	Buscar en todos los cuerpos de solicitud/respuesta
<b>Preserve log</b>	Conservar el log durante la navegación
<b>Disable cache</b>	Omitir caché del navegador con DevTools abierto
<b>Throttling dropdown</b>	Simular 3G lento, 3G rápido, Sin conexión
<b>Block request URL</b>	Clic derecho en solicitud -> Block URL

## Pestañas de detalles de solicitud

<b>Headers</b>	Cabeceras de solicitud/respuesta, código de estado
<b>Payload</b>	Cuerpo POST, parámetros de consulta
<b>Preview</b>	Respuesta formateada (JSON, HTML, imagen)
<b>Response</b>	Cuerpo de respuesta sin procesar
<b>Timing</b>	DNS, conexión, TLS, TTFB, descarga
<b>Initiator</b>	Traza de pila que originó la solicitud
<b>Cookies</b>	Cookies enviadas/recibidas

## Copiar y exportar

<b>Right-click -&gt; Copy as cURL</b>	Copiar solicitud como comando cURL
<b>Copy as fetch</b>	Copiar como JavaScript fetch()
<b>Export HAR</b>	Exportar todas las solicitudes como archivo HAR
<b>Copy response</b>	Copiar cuerpo de respuesta al portapapeles

## Fuentes

### Breakpoints

<b>Click line number</b>	Activar/desactivar breakpoint en línea
<b>Right-click -&gt; Conditional</b>	Pausar solo si la expresión es verdadera
<b>Right-click -&gt; Logpoint</b>	Registrar sin pausar la ejecución
<b>DOM breakpoint</b>	Pausar en subárbol/atributo/eliminación
<b>XHR/Fetch breakpoint</b>	Pausar cuando la URL contiene la cadena
<b>Event listener breakpoint</b>	Pausar en tipos de eventos específicos

## Controles del depurador

<b>F8 / Ctrl+\</b>	Reanudar / Pausar ejecución
<b>F10</b>	Pasar por encima de la llamada de función
<b>F11</b>	Entrar en la llamada de función
<b>Shift+F11</b>	Salir de la función actual
<b>Ctrl+Shift+P -&gt; "never pause"</b>	Deshabilitar todos los breakpoints

## Paneles de depuración

<b>Watch</b>	Monitorear valores de expresiones
<b>Scope</b>	Variables locales, de cierre y globales
<b>Call Stack</b>	Cadena de llamadas a funciones
<b>Snippets</b>	Guardar y ejecutar código JS reutilizable

## Rendimiento

### Grabación

<b>Record button (Ctrl+E)</b>	Iniciar/detener grabación de rendimiento
<b>Reload button</b>	Grabar rendimiento de carga de página
<b>Screenshots checkbox</b>	Capturar línea de tiempo visual
<b>CPU throttle dropdown</b>	Simular reducción de CPU 4x/6x
<b>Network throttle</b>	Simular red lenta durante la grabación

## Leer el Flame Chart

<b>Main track</b>	Ejecución de JavaScript (flame chart)
<b>Network track</b>	Línea de tiempo de solicitudes de red
<b>Frames track</b>	FPS y duración de fotogramas
<b>Timings track</b>	Marcadores FCP, LCP, DCL, Load
<b>Yellow bars</b>	JavaScript (scripting)
<b>Purple bars</b>	Layout / renderizado
<b>Green bars</b>	Pintado / composición

## Bottom-Up y Resumen

<b>Summary tab</b>	Desglose de tiempo: scripting, renderizado, etc.
<b>Bottom-Up tab</b>	Funciones más costosas primero
<b>Call Tree tab</b>	Jerarquía de llamadas de raíz a hoja
<b>Event Log tab</b>	Lista cronológica de eventos

## Aplicación

### Almacenamiento

<b>Local Storage</b>	Ver/editar pares clave-valor por origen
<b>Session Storage</b>	Ver/editar almacenamiento de sesión
<b>IndexedDB</b>	Inspeccionar almacenes de objetos y registros
<b>Cookies</b>	Ver/editar/eliminar cookies por dominio
<b>Cache Storage</b>	Inspeccionar cachés del Service Worker
<b>Clear storage</b>	Limpiar tipos de almacenamiento seleccionados

### Service Workers y Manifest

<b>Service Workers</b>	Ver registro, estado, push/sync
<b>Update on reload</b>	Forzar actualización del SW en cada recarga
<b>Bypass for network</b>	Omitir SW e ir directamente a la red
<b>Manifest</b>	Ver detalles del manifest de la web app
<b>Offline checkbox</b>	Simular modo sin conexión

## Lighthouse

### Ejecutar auditorías

<b>Mode: Navigation</b>	Auditoría de carga completa de página
<b>Mode: Timespan</b>	Auditar interacciones del usuario a lo largo del tiempo
<b>Mode: Snapshot</b>	Auditar estado actual de la página
<b>Categories</b>	Rendimiento, Accesibilidad, Buenas Prácticas, SEO
<b>Device</b>	Simulación de móvil o escritorio

### Métricas clave

<b>FCP (First Contentful Paint)</b>	Tiempo hasta el primer contenido visible
<b>LCP (Largest Contentful Paint)</b>	Tiempo hasta el elemento visible más grande
<b>TBT (Total Blocking Time)</b>	Suma del tiempo bloqueado por tareas largas
<b>CLS (Cumulative Layout Shift)</b>	Puntuación de estabilidad visual
<b>SI (Speed Index)</b>	Velocidad con que se rellena visualmente el contenido
<b>INP (Interaction to Next Paint)</b>	Capacidad de respuesta a la entrada del usuario

## Atajos

### Abrir DevTools

<b>F12 / Ctrl+Shift+I</b>	Alternar DevTools abierto/cerrado
<b>Ctrl+Shift+J</b>	Abrir panel Consola
<b>Ctrl+Shift+C</b>	Abrir Elementos + modo inspección
<b>Ctrl+Shift+M</b>	Alternar barra de dispositivos (responsive)

### Navegación y búsqueda

<b>Ctrl+Shift+P</b>	Menú de comandos (ejecutar cualquier acción)
<b>Ctrl+P</b>	Abrir archivo (Ir al archivo)
<b>Ctrl+Shift+F</b>	Buscar en todas las fuentes
<b>Ctrl+G</b>	Ir al número de línea en Sources
<b>Ctrl+[ / Ctrl+]</b>	Cambiar entre paneles

# Referencia Rápida de Chrome DevTools

## Edición y Consola

<b>Ctrl+Shift+D</b>	Alternar posición de acoplamiento de DevTools
<b>Ctrl+L (Console)</b>	Limpiar salida de la consola
<b>Shift+Enter (Console)</b>	Entrada multilínea
<b>Esc</b>	Alternar cajón de la consola
<b>Ctrl+K (Console)</b>	Limpiar consola

## Depuración móvil

### Barra de dispositivos

<b>Ctrl+Shift+M</b>	Alternar barra de dispositivos
<b>Device dropdown</b>	Dimensiones preestablecidas de teléfono/tablet
<b>Responsive mode</b>	Redimensionar el viewport libremente
<b>DPR dropdown</b>	Cambiar relación de píxeles del dispositivo
<b>Throttling</b>	Simular CPU + red de móvil
<b>Show media queries</b>	Visualizar breakpoints CSS

### Depuración remota (Android)

<b>1. Enable USB debugging</b>	Settings -> Developer Options en el dispositivo
<b>2. Connect USB</b>	Conectar dispositivo Android al ordenador
<b>3. chrome://inspect</b>	Abrir en Chrome de escritorio
<b>4. Click Inspect</b>	Abrir DevTools para la página móvil

Requiere Chrome en el escritorio y en el dispositivo Android

### Anular sensores

<b>Geolocation</b>	Anular coordenadas GPS
<b>Orientation</b>	Simular orientación del dispositivo
<b>Touch</b>	Simular eventos táctiles
<b>Idle detection</b>	Anular la API de detección de inactividad

## Patrones comunes

### Depurar problemas de red

```
// En Consola: monitorear todas las solicitudes fetch
const origFetch = window.fetch;
window.fetch = (...args) => {
  console.log('fetch:', args);
  return origFetch(...args);
};
```

### Flujo de rendimiento

<b>1. Lighthouse audit</b>	Identificar los principales problemas de rendimiento
<b>2. Performance recording</b>	Encontrar tareas largas en el flame chart
<b>3. Coverage tab</b>	Encontrar CSS/JS sin usar (Ctrl+Shift+P -> Coverage)
<b>4. Network waterfall</b>	Identificar recursos bloqueantes
<b>5. Rendering tab</b>	Visualizar pintura/layout (Ctrl+Shift+P -> Rendering)

### Fragmentos de consola útiles

```
// Listar todos los event listeners del elemento
getEventListeners($0);

// Monitorear cambios de layout
new PerformanceObserver(l => l.getEntries().forEach(
  e => console.log('CLS:', e)
)).observe({ type: 'layout-shift', buffered: true });
```