

# Referência Rápida Chrome DevTools

Elements, Console, Network, Performance, depuração

## Elements

### Inspecionar e Editar

<b>Right-click -&gt; Inspect</b>	Abrir painel Elements para o elemento
<b>Double-click tag/attribute</b>	Editar HTML inline
<b>Delete key</b>	Excluir nó selecionado
<b>Ctrl+Z</b>	Desfazer alteração no DOM
<b>H key</b>	Alternar visibilidade do elemento selecionado
<b>Drag node</b>	Mover elemento na árvore DOM

### Painel Styles

<b>element.style {}</b>	Adicionar estilos inline ao elemento
<b>Click property value</b>	Editar valor CSS ao vivo
<b>Checkbox next to rule</b>	Ativar/desativar propriedade CSS
<b>:hov button</b>	Forçar pseudo-estados do elemento (:hover, :focus)
<b>.cls button</b>	Adicionar/remover classes CSS
<b>Color swatch</b>	Abrir seletor de cores
<b>Computed tab</b>	Ver valores CSS computados finais

## Console

### API do Console

<code>console.log("info");</code>	<code>console.warn("warning");</code>
<code>console.error("error");</code>	<code>console.table(arrayOrObj);</code>
<code>console.group("label");</code>	<code>console.groupEnd();</code>
<code>console.time("t");</code>	<code>console.timeEnd("t");</code>

### Utilitários do Console

<b>\$0</b>	Elemento selecionado no painel Elements
<b>\$('.sel') / \$\$('sel')</b>	Atalho para <code>querySelector()</code> / <code>querySelectorAll()</code>
<b>copy(obj)</b>	Copiar objeto como string para a área de transferência
<b>clear()</b>	Limpar saída do console
<b>monitor(fn)</b>	Registrar chamadas à função <code>fn</code>
<b>monitorEvents(el, 'click')</b>	Registrar eventos no elemento
<b>keys(obj) / values(obj)</b>	Chaves / valores do objeto
<b>\$_</b>	Resultado da última expressão avaliada

### Filtros

<b>Log levels dropdown</b>	Filtrar por verbose/info/warn/error
<b>Filter text box</b>	Pesquisar saída do console
<b>-url:extension</b>	Excluir mensagens por URL de origem
<b>context: dropdown</b>	Filtrar por contexto iframe/worker

## Network

### Recursos do Painel Network

<b>Filter bar</b>	Filtrar por tipo: XHR, JS, CSS, Img, Doc, WS
<b>Search (Ctrl+F)</b>	Buscar em todos os corpos de requisição/resposta
<b>Preserve log</b>	Manter log durante navegação
<b>Disable cache</b>	Ignorar cache do browser enquanto DevTools estiver aberto
<b>Throttling dropdown</b>	Simular Slow 3G, Fast 3G, Offline
<b>Block request URL</b>	Clique direito na requisição -> Block URL

## Abas de Detalhes da Requisição

<b>Headers</b>	Headers de requisição/resposta, código de status
<b>Payload</b>	Corpo do POST, parâmetros de query
<b>Preview</b>	Resposta formatada (JSON, HTML, imagem)
<b>Response</b>	Corpo bruto da resposta
<b>Timing</b>	DNS, conexão, TLS, TTFB, download
<b>Initiator</b>	Stack trace que originou a requisição
<b>Cookies</b>	Cookies enviados/recebidos

### Copiar e Exportar

<b>Right-click -&gt; Copy as cURL</b>	Copiar requisição como comando cURL
<b>Copy as fetch</b>	Copiar como JavaScript <code>fetch()</code>
<b>Export HAR</b>	Exportar todas as requisições como arquivo HAR
<b>Copy response</b>	Copiar corpo da resposta para área de transferência

## Sources

### Breakpoints

<b>Click line number</b>	Ativar/desativar breakpoint de linha
<b>Right-click -&gt; Conditional</b>	Parar somente quando expressão for verdadeira
<b>Right-click -&gt; Logpoint</b>	Registrar sem pausar execução
<b>DOM breakpoint</b>	Parar em modificação de sub-árvore/atributo/remoção
<b>XHR/Fetch breakpoint</b>	Parar quando URL contiver string
<b>Event listener breakpoint</b>	Parar em tipos de eventos específicos

### Controles do Depurador

<b>F8 / Ctrl+\</b>	Continuar / Pausar execução
<b>F10</b>	Passar sobre próxima chamada de função
<b>F11</b>	Entrar na chamada de função
<b>Shift+F11</b>	Sair da função atual
<b>Ctrl+Shift+P -&gt; "never pause"</b>	Desativar todos os breakpoints

### Painéis de Depuração

<b>Watch</b>	Monitorar valores de expressões
<b>Scope</b>	Variáveis locais, closure e globais
<b>Call Stack</b>	Cadeia de chamadas de função
<b>Snippets</b>	Salvar e executar código JS reutilizável

## Performance

### Gravação

<b>Record button (Ctrl+E)</b>	Iniciar/parar gravação de performance
<b>Reload button</b>	Gravar performance de carregamento
<b>Screenshots checkbox</b>	Capturar linha do tempo visual
<b>CPU throttle dropdown</b>	Simular lentidão de CPU 4x/6x
<b>Network throttle</b>	Simular rede lenta durante gravação

### Lendo o Flame Chart

<b>Main track</b>	Execução JavaScript (flame chart)
<b>Network track</b>	Linha do tempo de requisições de rede
<b>Frames track</b>	FPS e duração dos frames
<b>Timings track</b>	Marcadores FCP, LCP, DCL, Load
<b>Yellow bars</b>	JavaScript (scripting)
<b>Purple bars</b>	Layout / renderização
<b>Green bars</b>	Pintura / composição

## Bottom-Up e Resumo

<b>Summary tab</b>	Distribuição de tempo: scripting, rendering, etc.
<b>Bottom-Up tab</b>	Funções mais custosas primeiro
<b>Call Tree tab</b>	Hierarquia de chamadas da raiz até folhas
<b>Event Log tab</b>	Lista cronológica de eventos

## Application

### Armazenamento

<b>Local Storage</b>	Ver/editar pares chave-valor por origem
<b>Session Storage</b>	Ver/editar armazenamento de sessão
<b>IndexedDB</b>	Inspecionar object stores e registros
<b>Cookies</b>	Ver/editar/excluir cookies por domínio
<b>Cache Storage</b>	Inspecionar caches do Service Worker
<b>Clear storage</b>	Limpar tipos de armazenamento selecionados em lote

### Service Workers e Manifest

<b>Service Workers</b>	Ver registro, status, push/sync
<b>Update on reload</b>	Forçar atualização do SW a cada reload
<b>Bypass for network</b>	Ignorar SW e ir direto para a rede
<b>Manifest</b>	Ver detalhes do manifesto do web app
<b>Offline checkbox</b>	Simular modo offline

## Lighthouse

### Executando Auditorias

<b>Mode: Navigation</b>	Auditoria completa de carregamento
<b>Mode: Timespan</b>	Auditar interações ao longo do tempo
<b>Mode: Snapshot</b>	Auditar estado atual da página
<b>Categories</b>	Performance, Acessibilidade, Melhores Práticas, SEO
<b>Device</b>	Simulação Mobile ou Desktop

### Métricas Principais

<b>FCP (First Contentful Paint)</b>	Tempo até primeiro conteúdo visível
<b>LCP (Largest Contentful Paint)</b>	Tempo até maior elemento visível
<b>TBT (Total Blocking Time)</b>	Soma do tempo de bloqueio de tarefas longas
<b>CLS (Cumulative Layout Shift)</b>	Pontuação de estabilidade visual
<b>SI (Speed Index)</b>	Rapidez com que o conteúdo é populado visualmente
<b>INP (Interaction to Next Paint)</b>	Responsividade à entrada do usuário

## Atalhos

### Abrir DevTools

<b>F12 / Ctrl+Shift+I</b>	Abrir/fechar DevTools
<b>Ctrl+Shift+J</b>	Abrir painel Console
<b>Ctrl+Shift+C</b>	Abrir Elements + modo inspeção
<b>Ctrl+Shift+M</b>	Alternar barra de dispositivos (responsivo)

### Navegação e Busca

<b>Ctrl+Shift+P</b>	Menu de Comandos (executar qualquer ação)
<b>Ctrl+P</b>	Abrir arquivo (Go to File)
<b>Ctrl+Shift+F</b>	Buscar em todos os arquivos
<b>Ctrl+G</b>	Ir para número de linha no Sources
<b>Ctrl+[ / Ctrl+] ]</b>	Alternar entre painéis

# Referência Rápida Chrome DevTools

## Edição e Console

<b>Ctrl+Shift+D</b>	Alternar posição de encaixe do DevTools
<b>Ctrl+L (Console)</b>	Limpar saída do console
<b>Shift+Enter (Console)</b>	Entrada multilinha
<b>Esc</b>	Alternar gaveta do console
<b>Ctrl+K (Console)</b>	Limpar console

## Debug Mobile

### Barra de Dispositivos

<b>Ctrl+Shift+M</b>	Alternar barra de dispositivos
<b>Device dropdown</b>	Dimensões predefinidas de celular/tablet
<b>Responsive mode</b>	Redimensionar viewport livremente
<b>DPR dropdown</b>	Alterar proporção de pixels do dispositivo
<b>Throttling</b>	Simular CPU + rede mobile
<b>Show media queries</b>	Visualizar breakpoints CSS

### Depuração Remota (Android)

<b>1. Enable USB debugging</b>	Settings -> Developer Options no dispositivo
<b>2. Connect USB</b>	Conectar dispositivo Android ao computador
<b>3. chrome://inspect</b>	Abrir no Chrome desktop
<b>4. Click Inspect</b>	Abrir DevTools para página mobile

Requer Chrome tanto no desktop quanto no dispositivo Android

### Substituição de Sensores

<b>Geolocation</b>	Substituir coordenadas GPS
<b>Orientation</b>	Simular orientação do dispositivo
<b>Touch</b>	Simular eventos de toque
<b>Idle detection</b>	Substituir API de detecção de inatividade

## Padrões Comuns

### Depurar Problemas de Rede

```
// In Console: monitor all fetch requests
const origFetch = window.fetch;
window.fetch = (...args) => {
  console.log('fetch:', args);
  return origFetch(...args);
};
```

### Fluxo de Performance

<b>1. Lighthouse audit</b>	Identificar principais problemas de performance
<b>2. Performance recording</b>	Encontrar tarefas longas no flame chart
<b>3. Coverage tab</b>	Encontrar CSS/JS não utilizado (Ctrl+Shift+P -> Coverage)
<b>4. Network waterfall</b>	Identificar recursos bloqueantes
<b>5. Rendering tab</b>	Visualizar paint/layout (Ctrl+Shift+P -> Rendering)

### Snippets Úteis no Console

```
// List all event listeners on element
getEventListeners($0);

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