Chrome DevTools Cheatsheet

A comprehensive guide to Chrome DevTools, covering essential features for debugging, profiling, and optimizing web applications.

Elements Panel

Inspecting Elements

Inspect Element	Right-click on a webpage element and select 'Inspect' to open DevTools with that element selected.
Selecting Elements	Use the 'Select an element in the page to inspect it' tool (Ctrl+Shift+C) to click on any element and view its HTML and CSS.
Editing HTML	Double-click on an element in the 'Elements' panel to edit its HTML directly. Changes are reflected immediately.
Editing Attributes	Modify element attributes by double-clicking on them in the 'Elements' panel. For example, change `src` of an `img` tag.
Forcing Element State	Right-click on an element and select 'Force State' (e.g., `:active`, `:hover`, `:focus`, `:visited`) to see how it looks in different states.
Copying Elements	Right-click on an element and choose 'Copy' to copy the element as HTML, JS path, CSS selector, etc.
Deleting Elements	Select an element and press 'Delete' or right-click and select 'Delete element' to remove it from the page.

Styles Pane

Inspecting Styles	The 'Styles' pane shows CSS rules applied to the selected element, in order of specificity.
Adding New Styles	Click the '.cls' button to add a new class to the element, or click the '+' button to add a new style rule.
Editing Styles	Click on a style property or value to edit it directly. Changes are reflected immediately.
Computed Styles	The 'Computed' tab shows the final computed values of all CSS properties for the selected element.
Filtering Styles	Use the filter box to search for specific CSS properties or values.
Box Model	The box model diagram visually represents the element's padding, border, and margin.
CSS Overview	Provides an overview of the CSS used on the page, including colors, fonts, and unused CSS rules.

Console Panel

Logging Messages

`console.log(message)`	Logs a generic message to the console.
`console.info(message)`	Logs an informational message to the console.
`console.warn(message)`	Logs a warning message to the console.
`console.error(message)`	Logs an error message to the console.
`console.debug(message)`	Logs a debug message to the console. Often hidden unless debug level is enabled.
`console.table(data)`	Displays tabular data as a table.
`console.assert(condition, message)`	Logs an error message if the assertion is false.

Evaluating Expressions

Evaluating JS Expressions	Type JavaScript code directly into the console and press Enter to execute it.
Accessing Variables	Access variables defined in your JavaScript code directly from the console.
`$_`	Represents the result of the last expression evaluated in the console.
`$0 - $4`	Represent the last 5 elements selected in the Elements panel, with `$0` being the most recently selected.
`$(selector)`	Equivalent to `document.querySelector(selector)`. Returns the first element that matches the CSS selector.
`$$(selector)`	Equivalent to `document.querySelectorAll(selector)`. Returns an array of all elements that match the CSS selector.
`inspect(object)`	Opens the specified object in the Elements panel for inspection.

Sources Panel

Debugging JavaScript

Setting Breakpoints	Click in the gutter (line number area) of the code editor to set a breakpoint. Execution will pause when the line is reached.
Conditional Breakpoints	Right-click in the gutter and select 'Add conditional breakpoint...' to set a breakpoint that only triggers when a condition is true.
Stepping Through Code	Use the stepping controls (Step over, Step into, Step out, Step) to navigate through your code execution.
Scope Pane	The 'Scope' pane shows the values of variables in the current scope (local, closure, global).
Watch Expressions	Add expressions to the 'Watch' pane to monitor their values as you step through the code.
Call Stack	The 'Call Stack' pane shows the history of function calls that led to the current execution point.
Break on...	Right-click an element in the Elements panel, and select 'Break on...' to pause execution when the element's attributes are modified, the node is removed, or the subtree changes.

Code Navigation

Open File	Use Ctrl+P (Cmd+P on macOS) to open any file in your project.
Go to Line/Column	Use Ctrl+G (Cmd+G on macOS) to go to a specific line and column number in the current file.
Search in Files	Use Ctrl+Shift+F (Cmd+Shift+F on macOS) to search for text across all files in your project.
Pretty Print	Click the `{}` button to format minified or compressed code for better readability.
Ignoring Code	Right-click on a script in the Sources panel and select "Add script to ignore list" to prevent the debugger from stepping into it.

Network Panel

Analyzing Network Requests

Recording Network Activity	The Network panel records all HTTP requests made by the browser. Ensure the record button is enabled (red).
Filtering Requests	Use the filter bar to filter requests by type (e.g., XHR, CSS, Images, Media, Fonts, WS, Other) or by URL/domain.
Sorting Requests	Click on column headers (e.g., Name, Status, Type, Size, Time) to sort requests.
Request Details	Click on a request to view detailed information, including Headers, Payload, Preview, Response, Timing, Cookies, and Initiator.
Timing Tab	The 'Timing' tab provides a detailed breakdown of the time spent on each phase of the request lifecycle.
Waterfall View	The waterfall chart provides a visual representation of the timing of all network requests.
Simulating Network Conditions	Use the 'Throttling' dropdown to simulate different network speeds (e.g., Slow 3G, Fast 3G, Offline) to test your application's performance under various conditions.

Advanced Features

Preserve Log	Enable 'Preserve log' to keep network requests across page reloads.
Disable Cache	Enable 'Disable cache' to force the browser to fetch resources from the server every time, useful for testing cache invalidation.
Import/Export HAR Files	Import and export HAR (HTTP Archive) files to share network activity recordings with others or analyze them later.
Override Requests	Right-click a request and select 'Override content' to locally modify the response.
Block Requests	Right-click a request and select 'Block request URL' to simulate unavailable resources.