By @blittle

Summary

Google Chrome

Chrome Developer Tools

Chromium

OpenWest 2014

Bret Little - @little_bret

http://github.com/blittle

Basics

Internal Chrome Pages

A chomplete list of internal Chrome URLs is available at chrome://about/

Flags of interest

- `chrome://flags` - Experimental features that may cause instability in the browser.
  - `Enable Developer Tools experiments` - Useful for new additions to the developer tools.
  - `Enable Experimental JavaScript` - Enable ECMAScript 6 additions.
  - `Enable experimental Web Platform features` - Enable experimental features in Blink
- `chrome://inspect` - Device inspection

Navigation

Load the developer tools by pressing `F12`.

The developer tools are separated into eight tabs in addition to a settings dialog:

- Elements - Inspect and manipulate the DOM tree and associated styles and event listeners.
- Network - Monitor and examine network requests.
- Sources - Examine, edit, and debug source code.
- Timeline - Display and record activity as it runs, including events, script activity, page rendering, and memory usage.
- Profiles - Record CPU and memory profiles.
- Resources - Examine cookies, local storage, indexedDB, and web sql.
- Audits - Execute page performance and CSS usage audits.
- Console

Navigate between the separate tabs by pressing `ctr` + `[` and `ctr` + `]`

The Console is available at from any tab by pressing the `escape` key or `ctr` + `~`

More keyboard shortcuts

Settings

The settings dialog is available by pressing the cog icon in the top right corner of the dev tools or with the keyboard shortcut `?`

Full settings documentation

Options to note

- Disable cache (while DevTools is open)

- Enable JavaScript and CSS source maps
- Experiments - Will only appear if enabled (chrome://flags/#enable-devtools-experiments)[chrome://flags/#enable-devtools-experiments]
- Workspace

Elements

Styles

- The toggle state button allows you to toggle CSS pseudo anchor state.
- Clicking the source file name or `ctr` + `click` the property will load the associated source file.
- Properties that are ~~crossed out~~ are either overriden or unknown to Chrome.

Inspect the following element and toggle its state.

State changes

Increment property values with the following shortcuts:

Increment Value	key-stroke
.1	`alt` + `up` & `alt` + `down`
1	`up` & `down`
10	`shift` + `up` & `shift` + `down`

LESS and SASS Source Maps

Chrome supports mapping the transpiled CSS rules back to the original SASS or LESS source. Make sure CSS source maps are enabled within the settings dialog.

This Button is Styled With Less

DOM Breakpoints

DOM Breakpoints are useful to break on DOM mutation events, such as when a node is removed, modified, or its attributes are changed. You can view all current DOM breakpoints on the `DOM Breakpoint` tab.

Find which script is modifying the following element by using DOM Breakpoints:

0

Event Listeners

Inspect event listeners associated with individual DOM nodes. Use the filter icon to restrict the event listeners to only the selected node.

Find the onclick handler definition for the button below:

Alert Me

Computed Properties

Sometimes CSS properties are dynamically computed based upon the context of the style. For example an property value with an `em` unit is proportional to the parent element.

Inspect the element below and analyze the defined versus computed properties:

Here is some calculated content

Network

WebSockets

Chrome developer tools contain a websockets view for analyzing websocket message frames.

- Green messages represent client to server (upload) messages while white represent server to client (download).
- The frames do not automatically update. Click on the request within the name column to refresh the frames.
- Right click to copy the data out of the frames.

Inspect the frames sent within the following websocket connection:

Message: Send Message

Sources

Profiles

Audits

Mobile

Console

Extensions