

Inspecting the Developer Tools

Ana Risteska / a.risteska@gmail.com



About me

■ UI design / front-end development / UX @ **netcetera**

■ Member of Free Software Macedonia

■ Part of the Front-End meet ups @ Hacklab KIKA in Skopje

■ Part of the Macedonian Mozilla community

What's in this talk?

■ Developer Tools - what and why?

■ Overview of some of the Firefox and Chrome Developer Tools and their panels

■ ... and what can we do with them to speed up our workflow

■ ... or to learn how our page works

So what are the DevTools exactly?

Visual and on-demand tools that help you
build and **debug** your webpage

DevTools give us control over the building blocks of every web page



What can we do with the DevTools?

- Manipulate the DOM tree
- Manipulate the CSS styles
- Debug JavaScript: find and fix errors
- Analyze performance - loading time of scripts
- Test the responsivity of a web page
- **Learn how our webpage works, so we can know how to make it better!**

Disclaimer: Automatic save-to-disk is not possible

■ ...unless we set up some kind of authoring and saving tools

■ Firefox: Scratchpad, Style Editor etc for writing.
Add-ons for saving: Firediff and others.

■ Chrome: Enabling Workspaces

A bit of history

- 2001 - Netscape Navigator 7 had a JavaScript debugger called Venkman. It couldn't inspect the DOM or show network traffic.
- 2005 - Microsoft Issues IE Developer Toolbar in 2005. DOM inspector and styles
- 2006 - Firebug for Mozilla Firefox - second generation of devtools: has DOM and styles inspector and JavaScript console and debugger
- A few years later Google Chrome and its Developer Tools came around

Finding the devtools in the browser

Right-click menu (context menu) and choose the options:
Inspect Element, or just **Inspect**

Shortcuts: **F12, ctrl+shift+i**
cmd instead of ctrl for Mac users

The DOM tree

On the GUI:

1. First panel: “Inspector” on FF, “Elements” on Chrome
2. Easy to see on which element I am focused on
3. HTML breadcrumbs
4. Searching for nodes in the HTML tree (ctrl+f)
5. Simulate mobile devices (ctrl+shift+m on both browsers)
6. 3D View on Firefox

Actions:

Edit the DOM: add and delete nodes, change text and attributes

2. 5. 1.

Chrome

The screenshot shows the Chrome DevTools interface with the 'Elements' tab selected. The left pane displays the DOM tree of the page, with the 'wrapper' div highlighted. The right pane shows the 'Styles' tab of the styles panel, listing CSS rules applied to the selected element. The 'element.style' section contains rules from 'www.softwareishard.com/style.css:30'. The 'media="screen"' section contains rules from 'www.softwareishard.com/style.css:9'. The 'user agent stylesheet' section contains rules from the user agent's style sheet. The 'Inherited from body' section contains rules from 'www.softwareishard.com/style.css:10'. A green oval highlights the 'Elements' tab and the 'wrapper' node in the DOM tree. A green circle highlights the 'div#Wrapper' entry in the bottom navigation bar.

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 <head profile="http://gmpg.org/xfn/11">...</head>
 <body cz-shortcut-listen="true">
 <script type="text/javascript">...</script>
 ...
 <div id="wrapper">...</div>
 <div id="sk2-footer" style="color:#FFF; background-color:#444;
padding: 3px 2px 3px 2px; border-top: #888 solid 1px;">...</div>
 <script src="http://www.google-analytics.com/urchin.js" type=
"text/javascript">
 </script>
 <script type="text/javascript">
 _uacct = "UA-3586722-1";
 urchinTracker();
 </script>
 <div id="overlay" style="display: none;"></div>
 <div id="lightbox" style="display: none;">...</div>
 </body>
</html>

Styles Computed Event Listeners DOM Breakpoints Properties

Filter

element.style {
}

media="screen" [www.softwareishard.com/
style.css:30](http://www.softwareishard.com/style.css:30)
#wrapper {
 margin: 0 auto;
 text-align: left;
 width: 760px;
 background: #FFFFFF;
 font-size: 0.9em;
 line-height: 1.6em;
 padding: 20px 10px 0 10px;
}
media="screen" [www.softwareishard.com/
style.css:9](http://www.softwareishard.com/style.css:9)
* {
 padding: 0;
 margin: 0;
}
div {
 display: block;
}
Inherited from body
media="screen" [www.softwareishard.com/
style.css:10](http://www.softwareishard.com/style.css:10)
body {
 padding: 0px;
 text-align: left;
 font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;
 font-size: small;
 color: #222222;
 line-height: 150%;
 background: #4A525A;
 background-color: #fff;
}

3.

html body div#Wrapper

Firefox

2. 1.

The screenshot shows the Firefox Developer Tools Inspector panel. The top navigation bar includes tabs for Inspector, Console, Debugger, Style Editor, Performance, Network, Rules, Computed, Fonts, Box Model, and Animations. The Inspector tab is selected and highlighted with a red circle (1). The left sidebar displays the page's DOM structure. A specific input element, `<input id="livesearch" class="search" type="search" ...>`, is selected and highlighted with a red box (2). The status bar at the bottom shows the URL `https://www.google.com/search?rlz=1C1GCEU_enUS904US904&q=site%3Aexample.com+example.com`. The main content area shows the element's properties and styles. Several UI elements are circled with red circles: the search icon in the top right (3), the search bar in the top right (4), the Rules tab (5), the Box Model tab (6), and the three icons in the top right corner of the main content area (7).

3. 4.

6. 5.

CSS Modification

- Inspect styles applied to an element
- Understanding how cascading and inheritance works
- Editing CSS / Designing in the browser
- Ctrl+click on a color value and that will take you to the file where it is defined

Rules Computed Fonts Box Model Animations

Filter Styles

```
element { inline
}
#ghx-header h2 { batch.css:88
  color: #333;
  font-size: 20px;
  font-weight: normal;
  line-height: 1.5;
  margin: 0px;
  word-wrap: break-word;
}

h1:first-child, h2:first-child, h3:first-child, h4:first-child, h5:first-child, h6:first-child { batch.css:23
  margin-top: 0px;
}

p:first-child, ul:first-child, ol:first-child, dl:first-child, h1:first-child, h2:first-child, h3:first-child, h4:first-child, h5:first-child, h6:first-child, blockquote: pre:first-child, form.aui:first-child, table.aui:first-child, .aui-tabs:first-child, .aui-panel:first-child, .aui-group:f {
  margin-top: 0px;
}
```

Firefox

Rules Computed Fonts Box Model Animations

Filter Styles Browser styles

▶ color	> #333
▶ font-family	> Arial,sans-serif
▶ font-size	> 20px
▶ font-weight	> 400
▶ line-height	> 30px
▶ margin-bottom	> 0px
▶ margin-left	> 0px
▶ margin-right	> 0px
▶ margin-top	> 0px
▶ padding-bottom	> 0px
▶ padding-left	> 0px
▶ padding-right	> 0px
▶ padding-top	> 0px
▶ text-transform	> none
▶ word-wrap	> break-word

Styles Computed Event Listeners DOM Breakpoints Properties

Filter

```

element.style {
}

.index>div {
    padding: ▶ 2rem 0;
}

* {
    -moz-box-sizing: border-box;
    -webkit box-sizing: border-box;
    box-sizing: border-box;
}

div {
    display: block;
}

Inherited from header.index
.index {
    text-shadow: 1px 1px 1px ■#0c0d0d;
}

.index, .text-center {
    text-align: center;
}

Inherited from body#gfs
body, button, input, select, textarea {
    font-family: sans-serif;
    color: ■#e3d7bf;
}

```

Chrome

Styles Computed Event Listeners DOM Breakpoints Properties

margin	-
border	-
padding	32
	709.094 × 293
	32
	-
	-

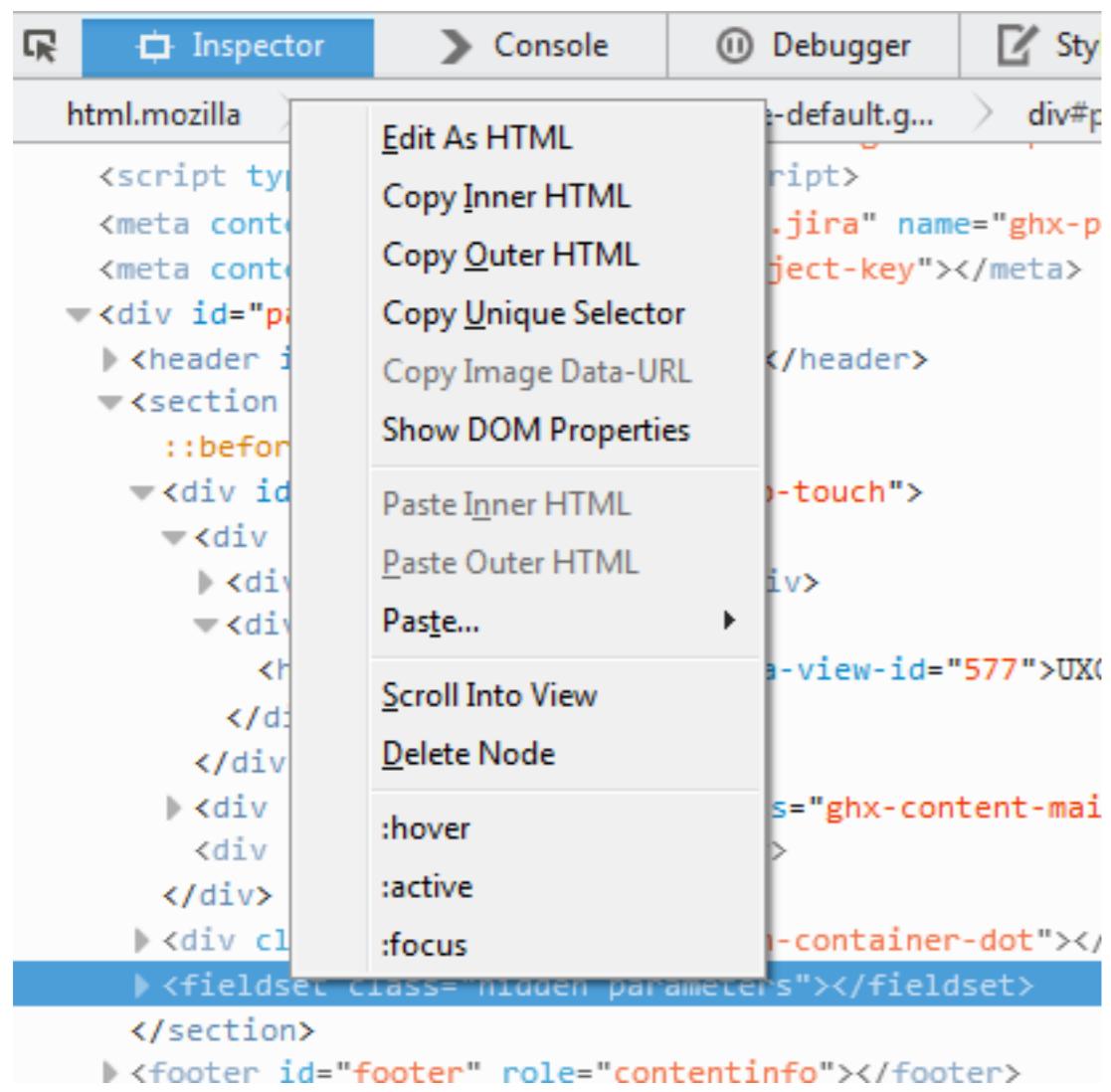
Filter Show all

▶ box-sizing	border-box
▶ color	■rgb(227, 215, 191)
▶ display	block
▶ font-family	sans-serif
▶ font-size	16px
▶ font-style	normal
▶ font-variant	normal
▶ font-weight	normal
▶ height	357px
▶ line-height	24px
▶ padding-bottom	32px
▶ padding-left	0px
▶ padding-right	0px
▶ padding-top	32px
▶ text-align	center

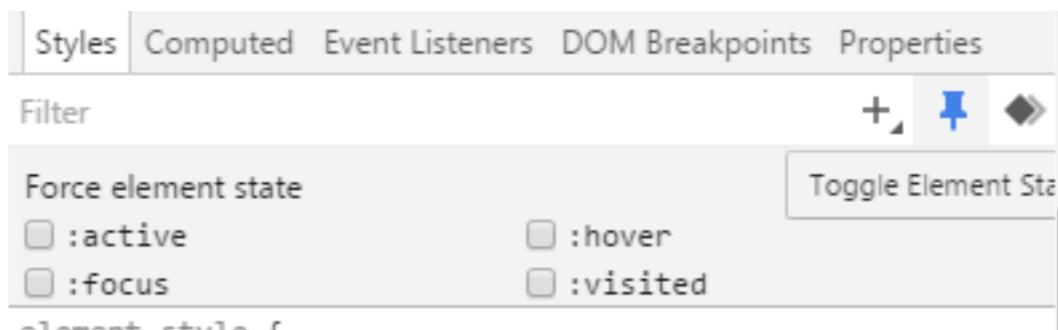
CSS Modification

Pseudo classes

Firefox



Chrome



CSS Modification

The preprocessor workflow - Experimental in Chrome, but by default it shows the file names where the style comes from

Chrome

The screenshot shows the Chrome DevTools Styles tab. The top navigation bar has 'Styles' selected. Below it is a 'Filter' input field and some icons. The main area displays a list of CSS rules:

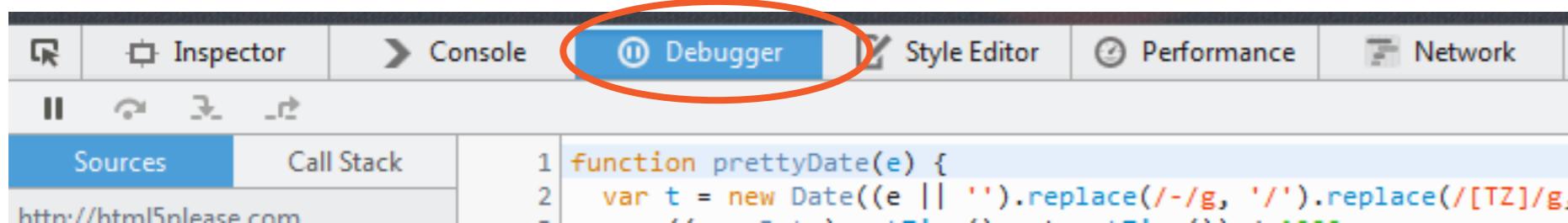
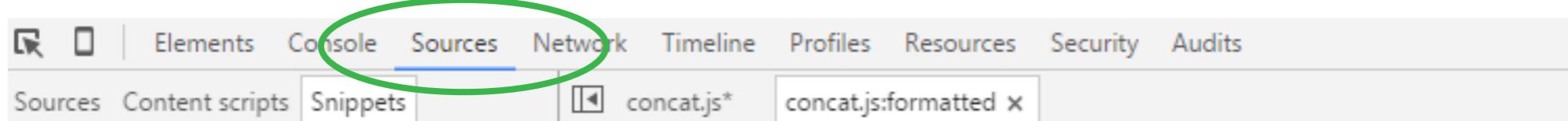
- `element.style { }`
- `#mc-embedded-subscribe-form { padding: 20px 0 5px 0; background: #ccc; text-align: center; }` footer.scss:15
- `form { margin: 0; }` normalize.scss:84
- `*, *::after, *::before { -moz-box-sizing: border-box; box-sizing: border-box; }` normalize.scss:1
- `form { display: block; margin_top: 0em; }` user agent stylesheet
- `Inherited from html`
- `html { font-family: 'Source Sans Pro', sans-serif; font-size: 17px; font-weight: 300; line-height: 1.45; }` normalize.scss:169

Debugging scripts

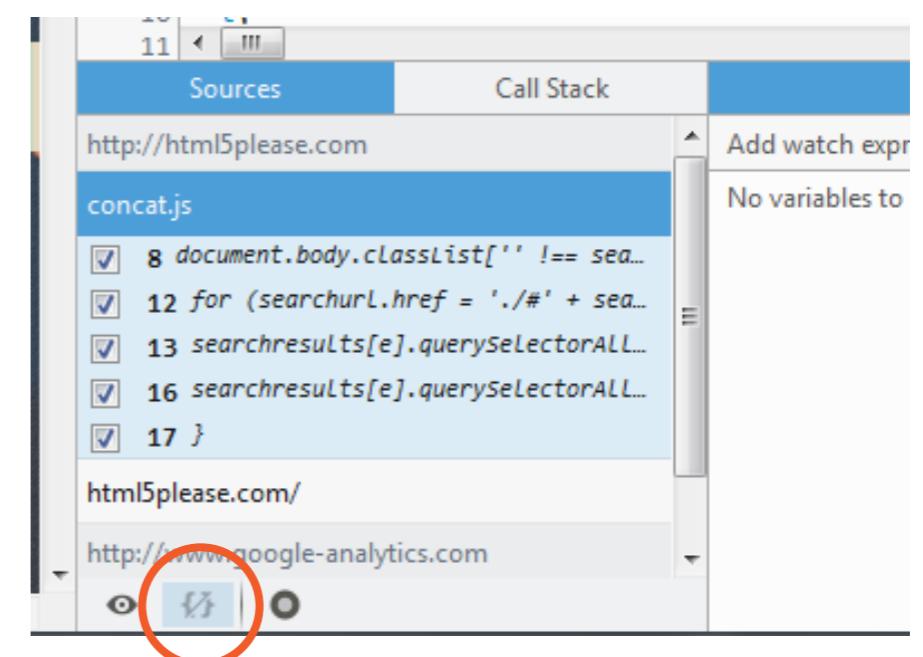
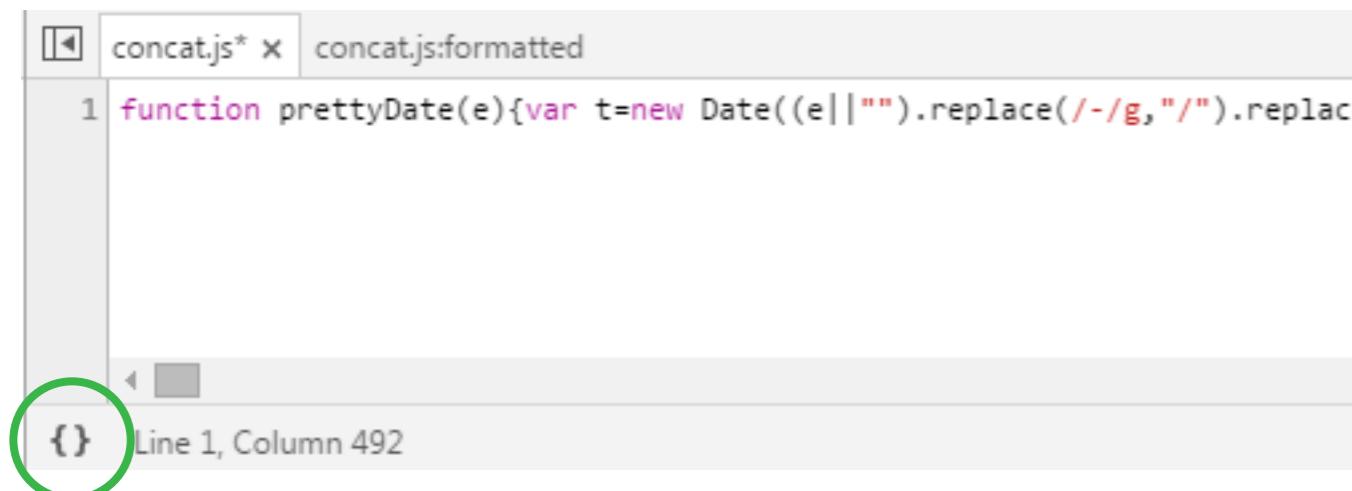
- Debugging stands for finding and fixing logic in our code.
- Pause running JavaScript at various points so you can determine its progress or examine its variable values.
- no need to use brute force alerts like: `alert("ok so far");`

JS in the DevTools

Sources tab in Chrome and Debugger tab in FF

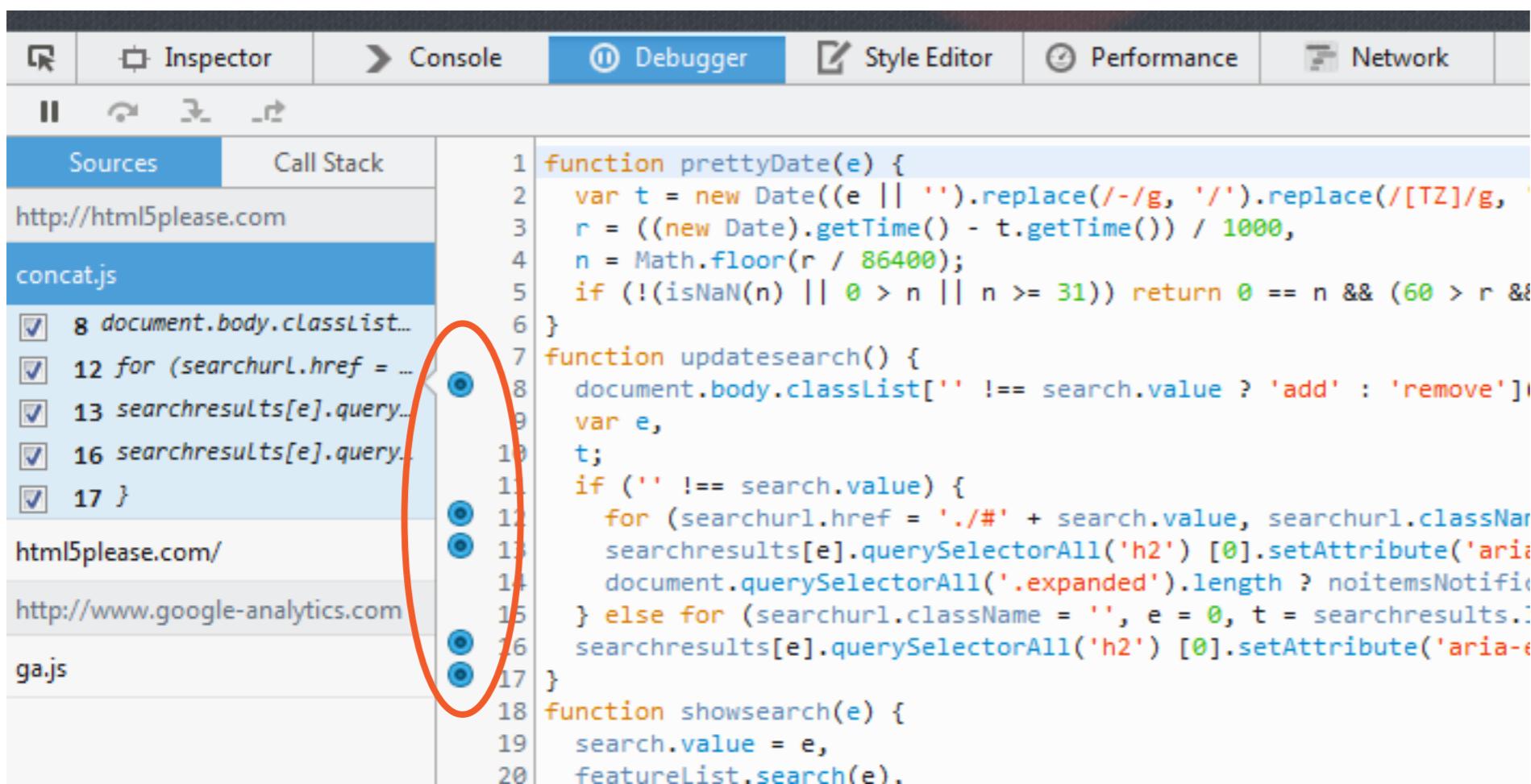


Prettify the JS



Breakpoints: Quintessential feature of debuggers

1. Find the line of source you want to investigate
2. Set a breakpoint
3. Run the program
4. Debugger stops the execution at that point, you poke around



The screenshot shows the 'Debugger' tab of a browser's developer tools. The left sidebar lists several scripts: 'concat.js' (selected), 'document.body.classList...', 'searchresults[e].query...', 'searchresults[e].query...', 'html5please.com/' (disabled), 'http://www.google-analytics.com', and 'ga.js'. The main pane displays the code of 'concat.js' with line numbers 1 through 20. A red oval highlights the first four lines of code, which define the `prettyDate` function. Each highlighted line has a blue circular breakpoint icon to its left.

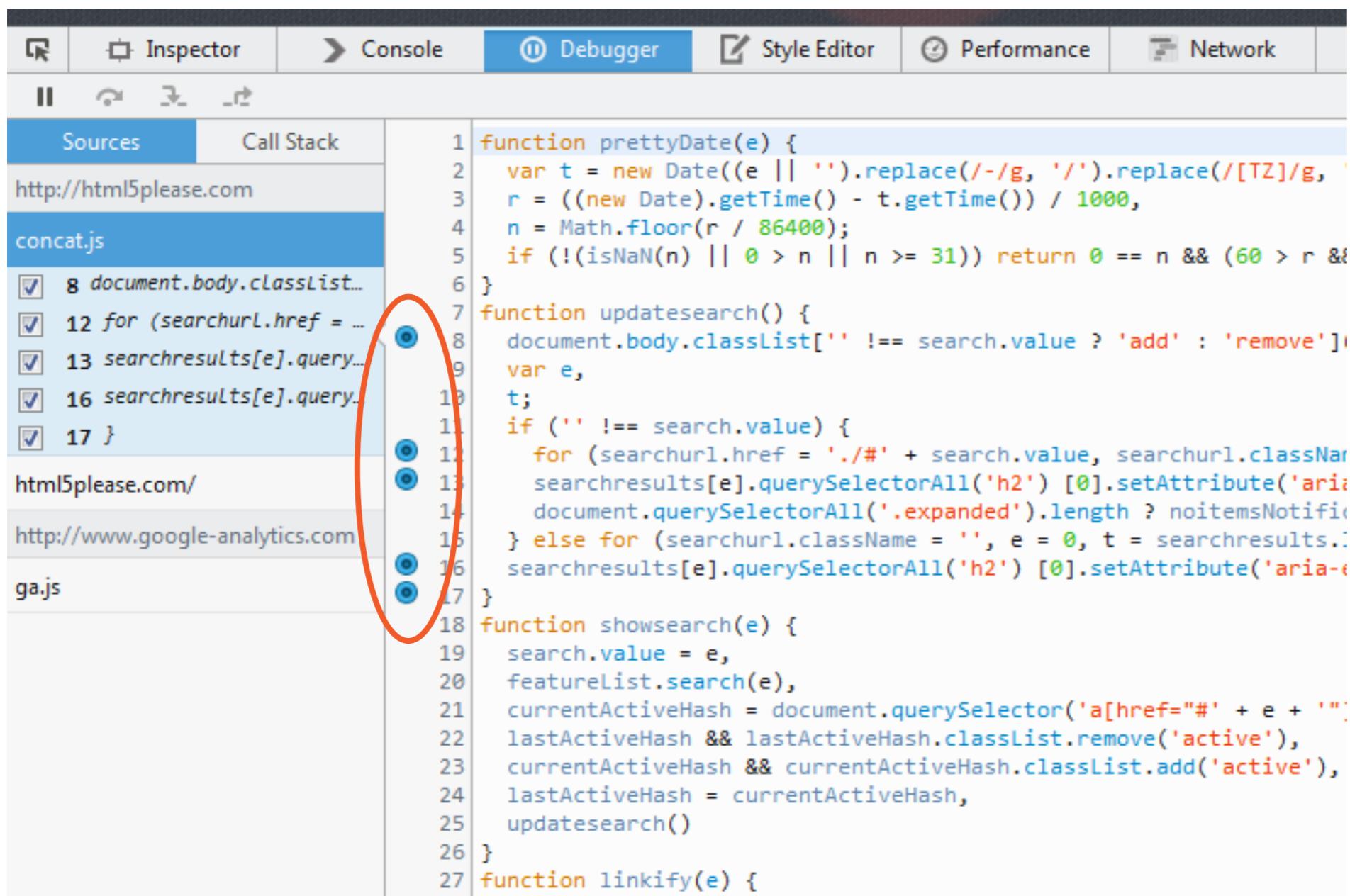
```
function prettyDate(e) {
  var t = new Date((e || '')).replace(/-/g, '/').replace(/[TZ]/g, '')
  r = ((new Date).getTime() - t.getTime()) / 1000,
  n = Math.floor(r / 86400);
  if (!(isNaN(n) || 0 > n || n >= 31)) return 0 == n && (60 > r &&
}
function updateSearch() {
  document.body.classList['' != search.value ? 'add' : 'remove']
  var e,
  t;
  if ('' != search.value) {
    for (searchurl.href = './#' + search.value, searchurl.className =
      searchresults[e].querySelectorAll('h2')[0].setAttribute('aria-
      document.querySelectorAll('.expanded').length ? noitemsNotific
    } else for (searchurl.className = '', e = 0, t = searchresults.
      searchresults[e].querySelectorAll('h2')[0].setAttribute('aria-
    }
  function showSearch(e) {
    search.value = e,
    featureList.search(e),
  }
}
```

Types of breakpoints and how to use them

Locate a specific method with **ctrl+shift+o** in Chrome / **ctrl + d** in FF

Add breakpoint in the JS (or debugging inside-out)

Firefox



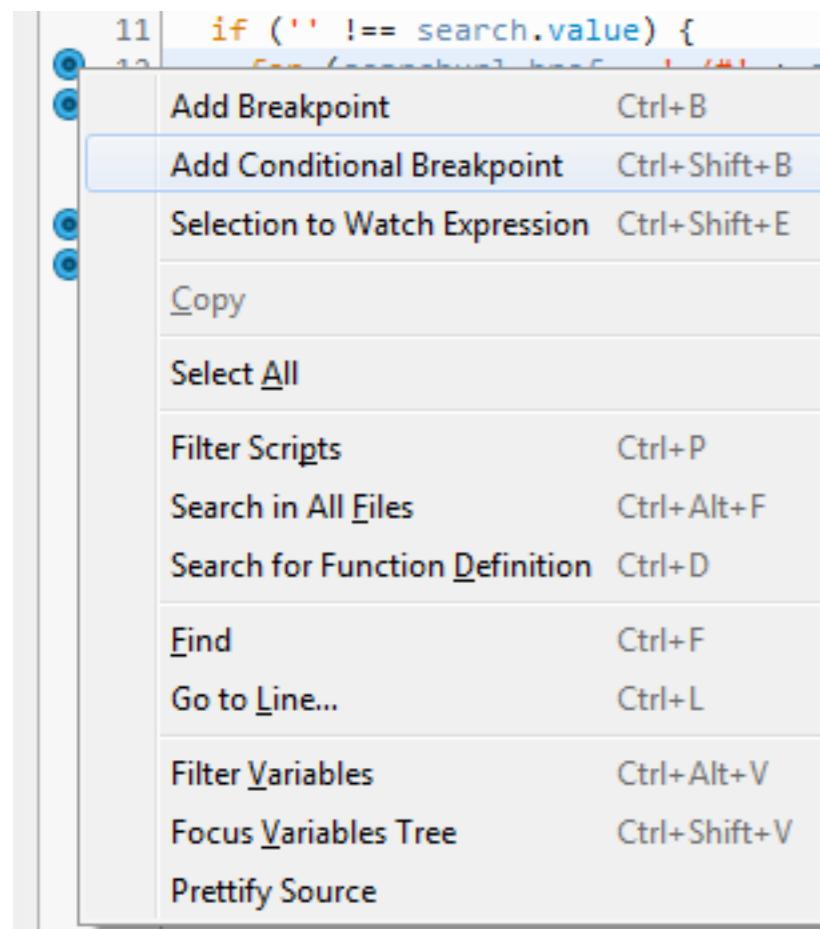
The screenshot shows the Firefox Developer Tools interface with the 'Debugger' tab selected. In the left sidebar under 'Sources', several files are listed: 'concat.js' (selected), 'document.body.classList...', 'searchresults[e].query...', 'searchresults[e].query...', and 'ga.js'. The 'concat.js' file is expanded, showing its source code. A red oval highlights a group of five circular breakpoints placed on the first few lines of code. The code itself is a snippet of JavaScript related to date formatting and search functionality.

```
function prettyDate(e) {
  var t = new Date((e || '')).replace(/-/g, '/').replace(/[TZ]/g, '')
  r = ((new Date).getTime() - t.getTime()) / 1000,
  n = Math.floor(r / 86400);
  if (!(isNaN(n) || 0 > n || n >= 31)) return 0 == n && (60 > r &&
}
function updateSearch() {
  document.body.classList['' != search.value ? 'add' : 'remove']
  var e,
  t;
  if ('' != search.value) {
    for (searchurl.href = './#' + search.value, searchurl.className =
      searchresults[e].querySelectorAll('h2')[0].setAttribute('aria-
      document.querySelectorAll('.expanded').length ? noitemsNotific
    } else for (searchurl.className = '', e = 0, t = searchresults.
      searchresults[e].querySelectorAll('h2')[0].setAttribute('aria-
    }
  }
  function showSearch(e) {
    search.value = e,
    featureList.search(e),
    currentActiveHash = document.querySelector('a[href="#' + e + '"]
    lastActiveHash && lastActiveHash.classList.remove('active'),
    currentActiveHash && currentActiveHash.classList.add('active'),
    lastActiveHash = currentActiveHash,
    updateSearch()
  }
  function linkify(e) {
```

Types of breakpoints and how to use them

Add conditional breakpoint

Firefox

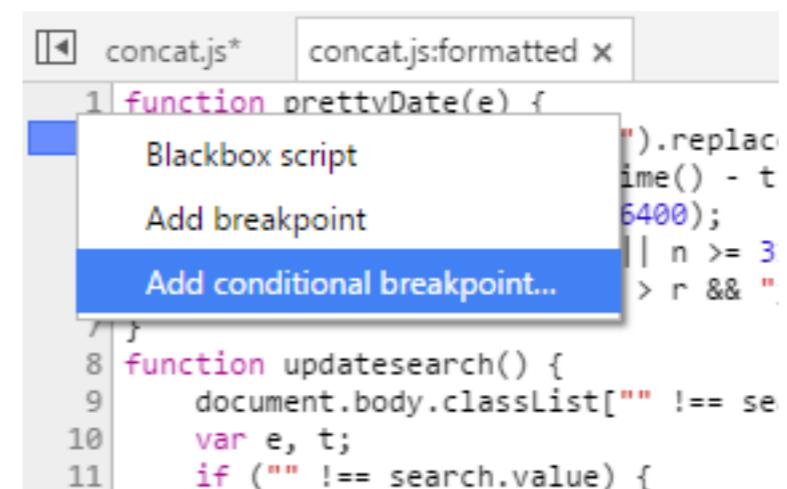


The screenshot shows the context menu for a line of code in Firefox DevTools. The menu items include:

- 11 if ('' !== search.value) {
- 12 e = e.replace(/\n/g);
- 13 if (e.length > 0) {
- 14 search.value, searchurl.classList.add('ended');
- 15 e.length ? noitemsNotEqual : 0;
- 16 t = searchresults.querySelector('h2').classList.remove('active');
- 17 Hash.classList.add('active');
- 18 }
- 19 t = document.querySelector('#' + e + '');
- 20 t.classList.remove('active');
- 21 Hash.classList.add('active');
- 22 }
- 23 return t;
- 24 }
- 25 }
- 26 }
- 27 }
- 28 }
- 29 }
- 30 }
- 31 }
- 32 }
- 33 }
- 34 function e(t, r, n) {
- 35 var s = e.resolve(t);

The "Add Conditional Breakpoint" option is highlighted in the menu.

Chrome



The screenshot shows the context menu for a line of code in Chrome DevTools. The menu items include:

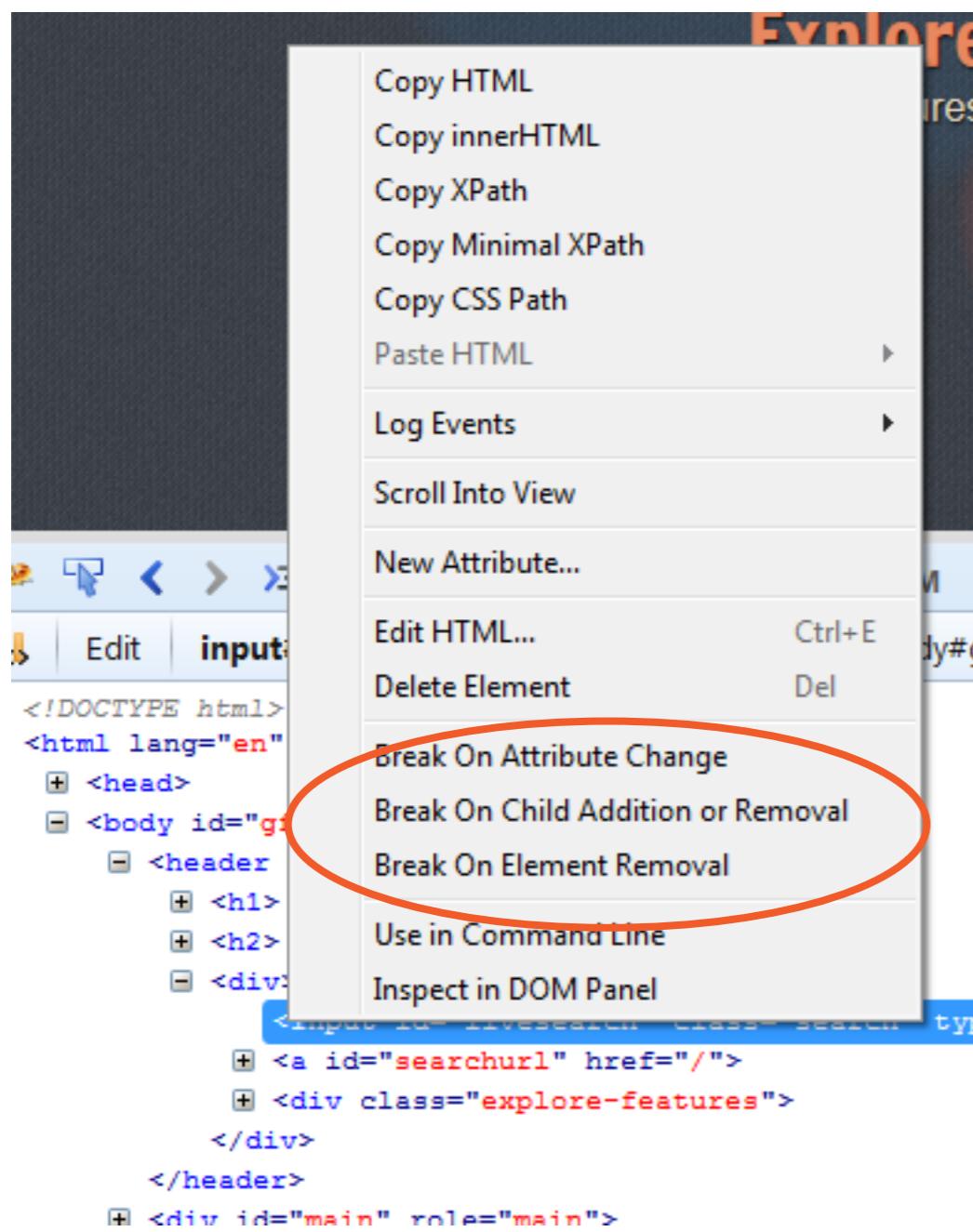
- concat.js*
- concat.js:formatted x
- 1 function prettyDate(e) {
- 2 e.replace(/\n/g);
- 3 if (e.length - t > 6400) {
- 4 | n >= 3;
- 5 Add conditional breakpoint...
- 6 }
- 7 }
- 8 function updateSearch() {
- 9 document.body.classList['' !== se]
- 10 var e, t;
- 11 if ('' !== search.value) {

The "Add conditional breakpoint..." option is highlighted in the menu.

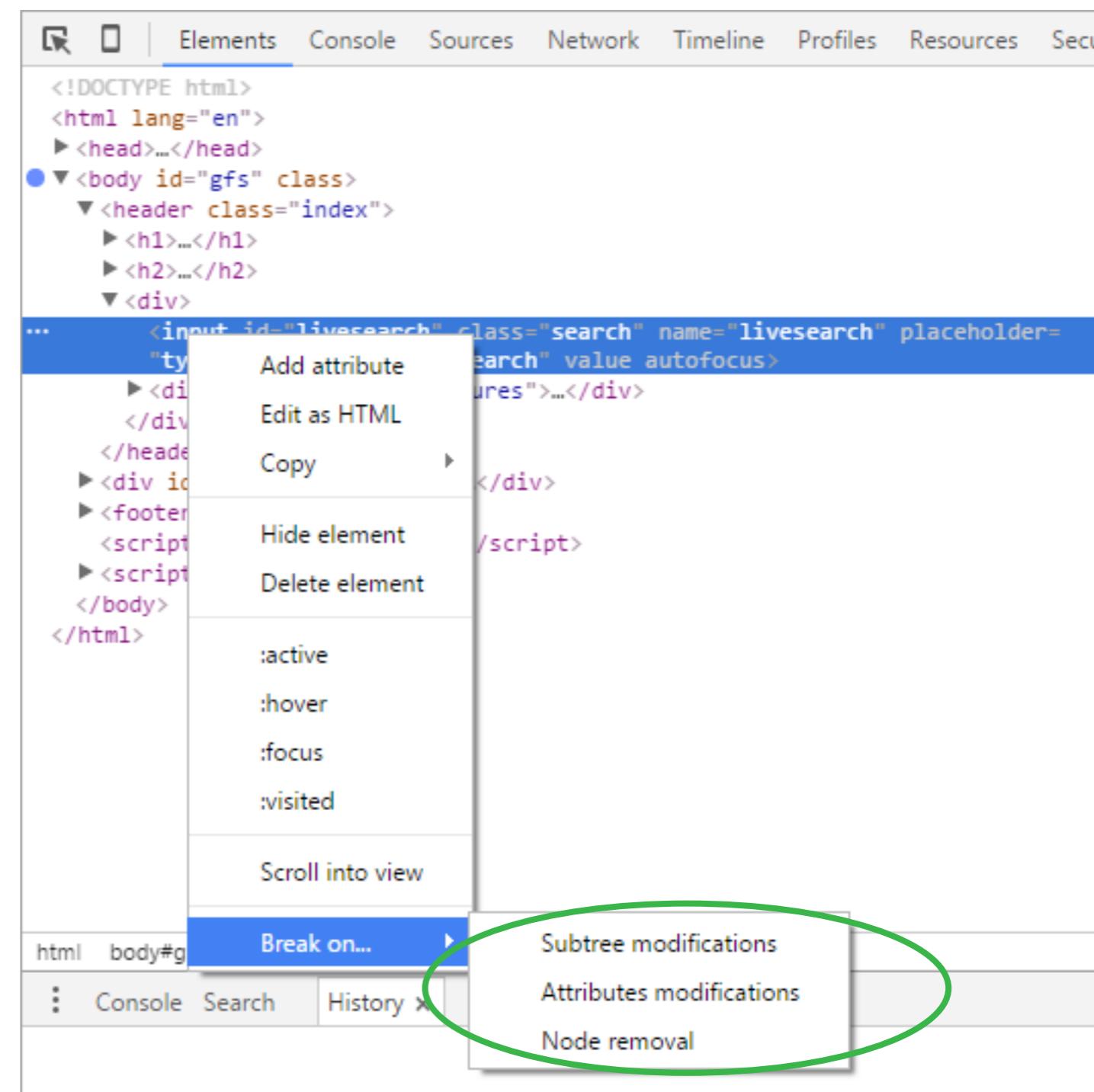
Types of breakpoints and how to use them

Add DOM breakpoint (or debugging outside-in)

Firefox

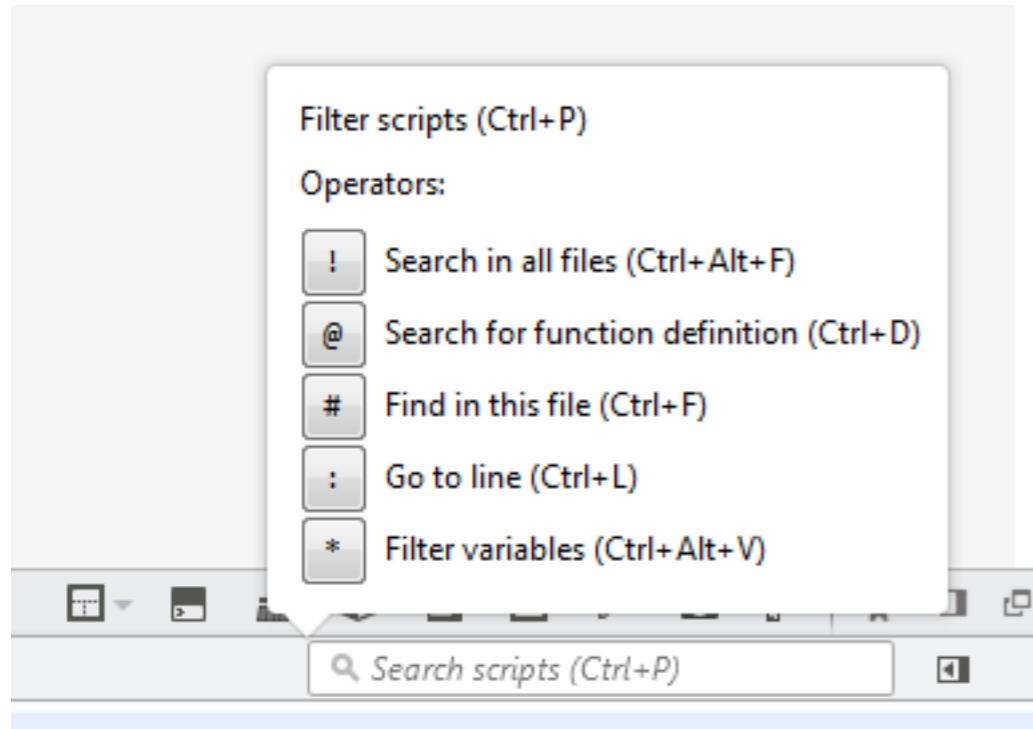


Chrome



Shortcuts

Firefox



Chrome

Ctrl+o gives the list of files that are available for that particular project (also css)

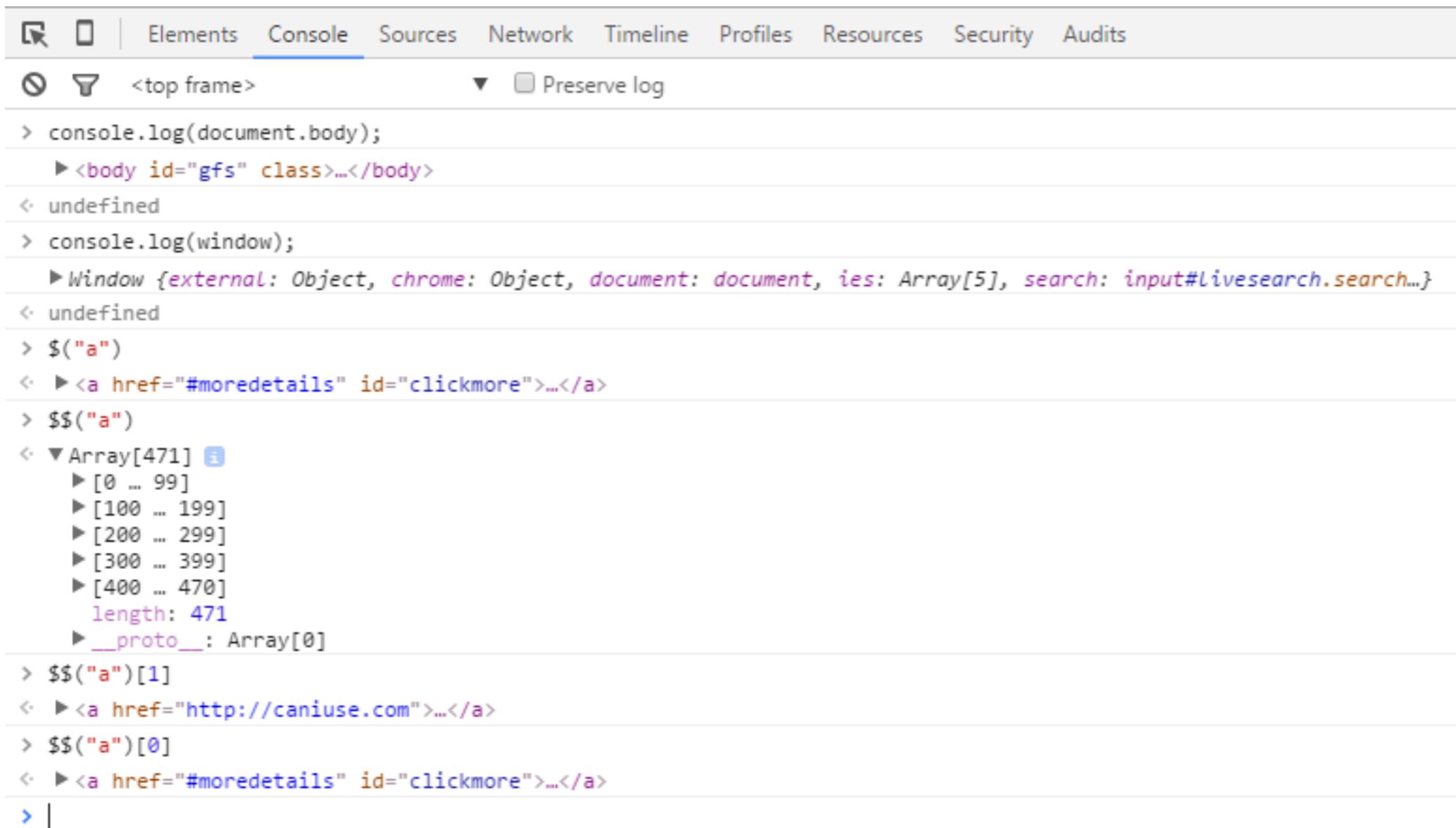
Ctrl+l go to line

Ctrl+f= searches for every instance of a method or variable name

Using the console

Console lets you use standard JavaScript statements and Console-specific commands while a page is live in the browser to help you debug the page.

Chrome



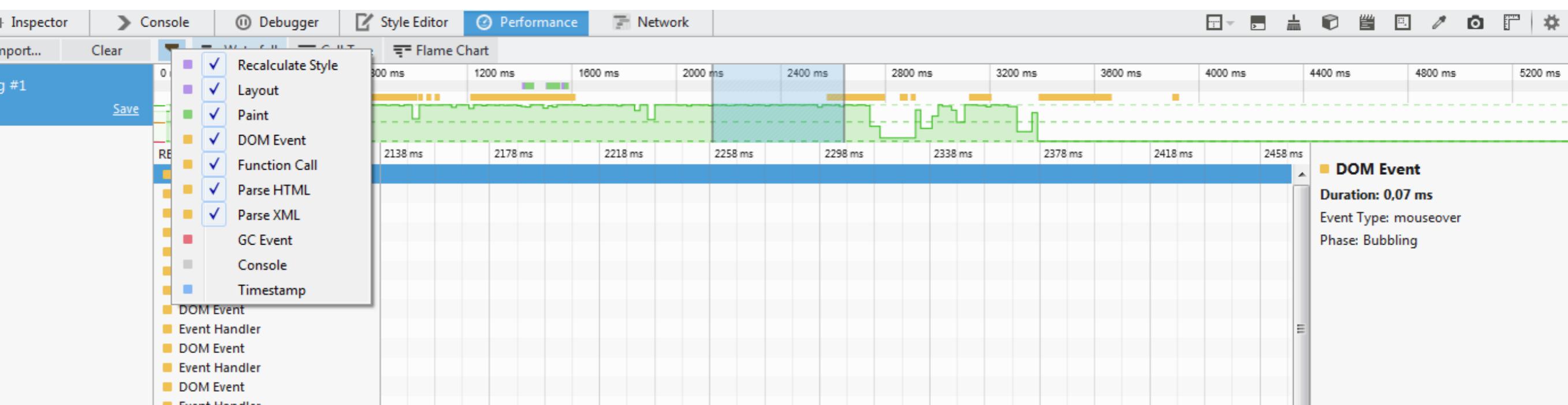
The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. The console output is as follows:

```
> console.log(document.body);
▶ <body id="gfs" class="...</body>
<- undefined
> console.log(window);
▶ Window {external: Object, chrome: Object, document: document, ies: Array[5], search: input#livesearch.search...}
<- undefined
> $("a")
<- ▶ <a href="#moredetails" id="clickmore">...</a>
> $$("a")
<- ▶ Array[471] ⓘ
▶ [0 ... 99]
▶ [100 ... 199]
▶ [200 ... 299]
▶ [300 ... 399]
▶ [400 ... 470]
  length: 471
▶ __proto__: Array[0]
> $$("a")[1]
<- ▶ <a href="http://caniuse.com">...</a>
> $$("a")[0]
<- ▶ <a href="#moredetails" id="clickmore">...</a>
> |
```

Performance

Every marker in the timeline represents the type of work a browser is doing at a particular point of time

Firefox



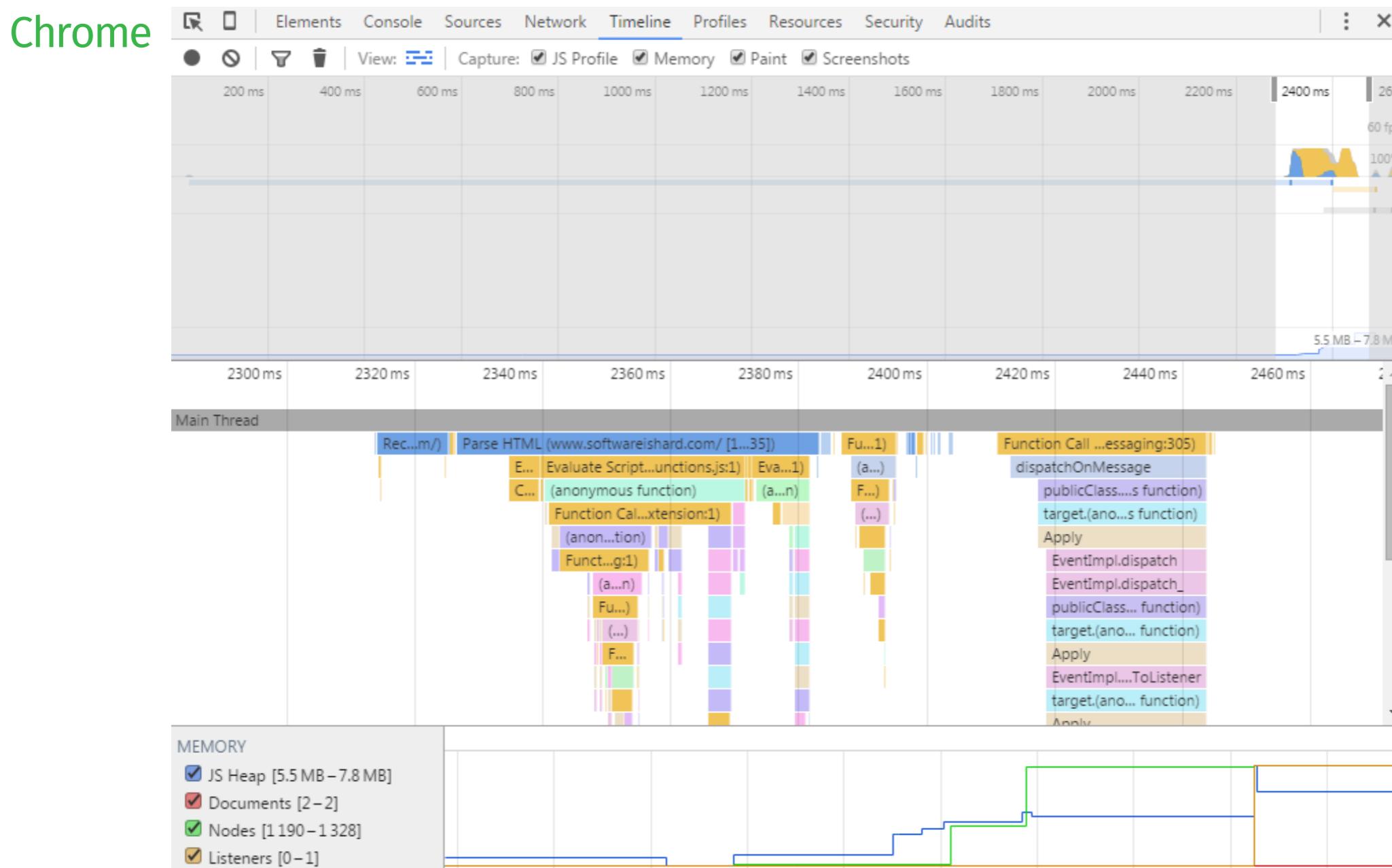
Styling and layout

Actual painting on the screen

Computation, typically JS

Performance

Every marker in the timeline represents the type of work a browser is doing at a particular point of time



Network

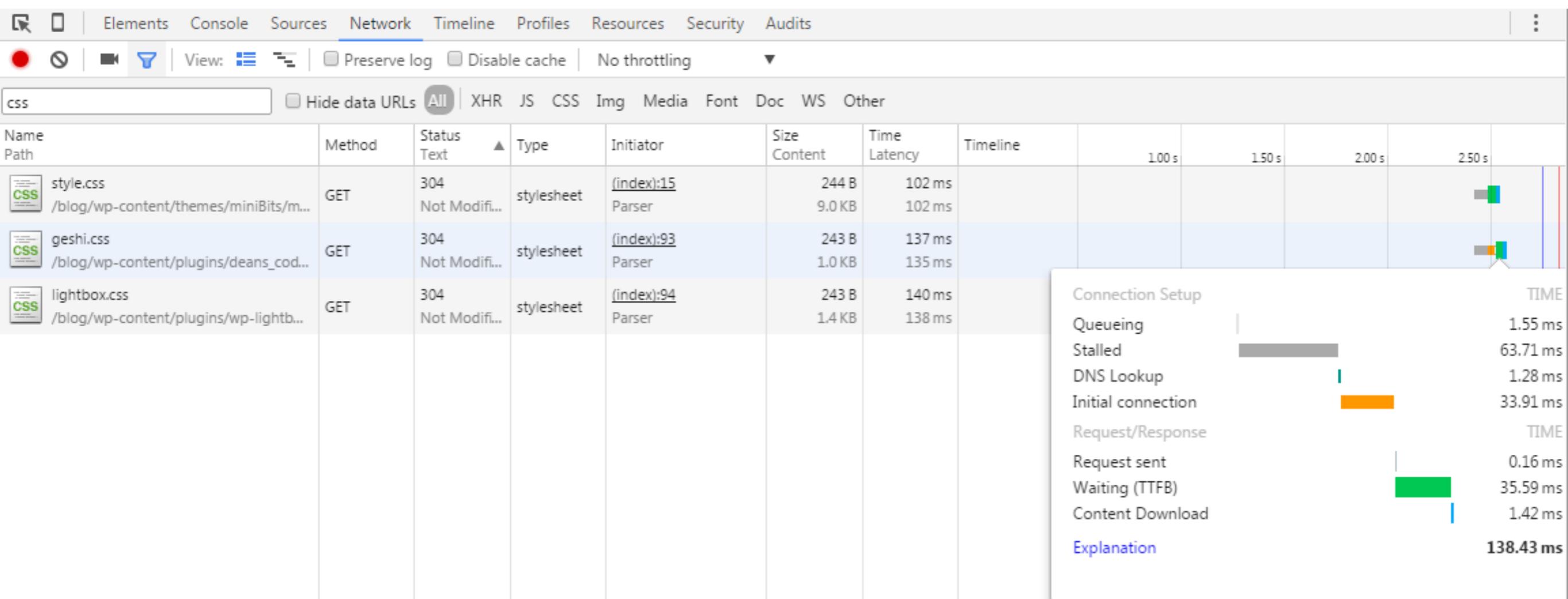
Record + Details View

Check the request method types

Analyse response headers

Filter and search for particular resource and analyze its performance

Chrome



Network

Record + Details View

Check the request method types

Analyse response headers

Filter and search for particular resource and analyze its performance

Firefox

Method	File	Domain	Type	Transferred	Size	Time	Headers	Cookies	Params	Response	Tin
200 GET	css?family=Francois+One Open+Sans:40...	fonts.googleapis.com	css	0,38 KB	1,01 KB	→ 90 ms	DNS resolution: → 0 ms				
200 GET	_utm.gif?utmwv=5.6.7&utms=1&ut...	www.google-analytics.c...	gif	0,03 KB	0,05 KB	→ 45 ms	Connecting: → 45 ms				
304 GET	/	html5please.com	html	38,13 KB	164,84 KB	→ 1 ms	Sending: → 0 ms				
304 GET	concat.js	html5please.com	js	8,22 KB	23,46 KB	→ 1 ms	Waiting: → 0 ms				
304 GET	ga.js	www.google-analytics.c...	js	15,65 KB	42,07 KB	→ 3 ms	Receiving: → 0 ms				
304 GET	6cfbd373bb1dd5f26c066c1fc7e68288?...	www.gravatar.com	jpeg	—	2,46 KB	→ 1 ms					
304 GET	21dbcdbf46f8ee3774c1b9b3efadfa9e?...	www.gravatar.com	jpeg	—	2,62 KB	→ 1 ms					
304 GET	56e38d8bcb52c9674dad204a2e8fc804...	www.gravatar.com	jpeg	—	1,90 KB	→ 1 ms					
304 GET	6025224?v=3&s=48	avatars3.githubusercontent...	jpeg	—	1,77 KB	→ 1 ms					
304 GET	denim.png?1436049899	html5please.com	png	—	26,50 KB	→ 1 ms					
304 GET	HTML5-logo.png	html5please.com	png	—	1,14 KB	→ 3 ms					
304 GET	c64a99fb7785e6cf9588a8777ba083?...	www.gravatar.com	png	—	5,36 KB	→ 2 ms					

Learn more and stay updated

developer.mozilla.org

developer.chrome.com/devtools

umaar.com/dev-tips/



@MozDevNet

@FirefoxDevTools

@googledevs

@umaar

@addyosmani