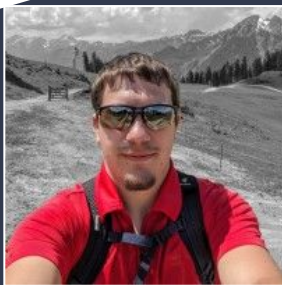


10 Chrome DevTools features you are not using

... but could be!

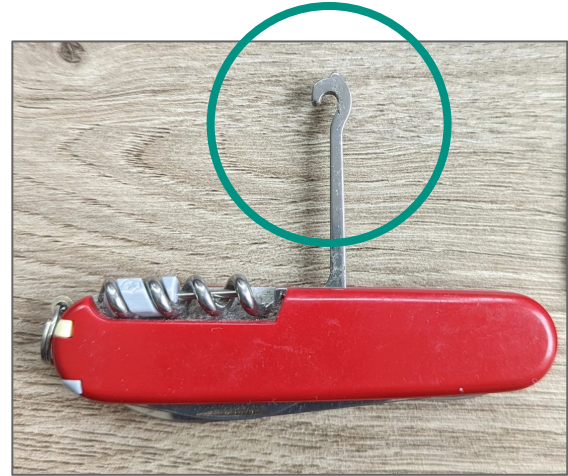


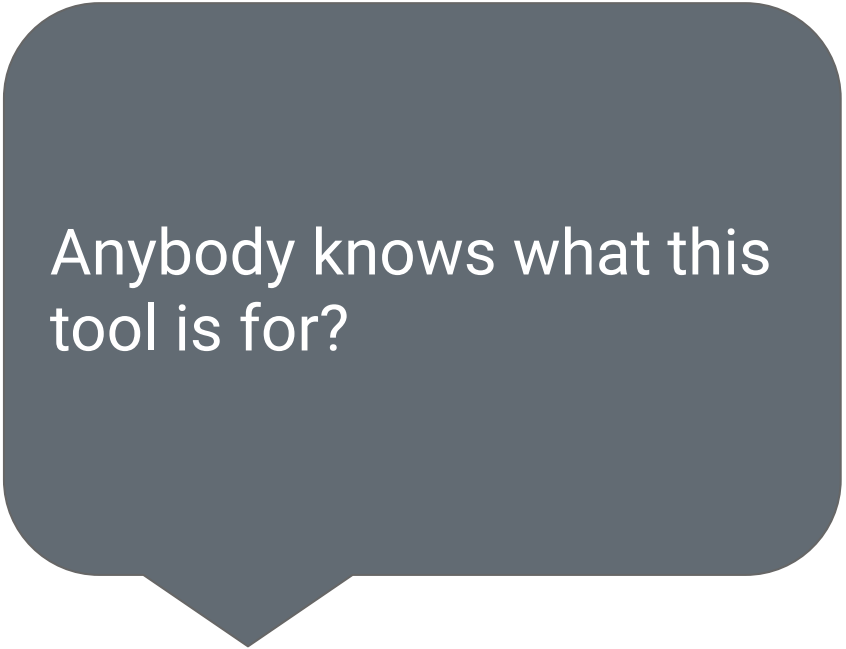
Michael Hablich, Product Manager @ Google
Chrome for Debugging & Testing on the web
hablich@chromium.org, @mhablich

Chrome DevTools is
like a swiss army
knife



It has many
unknown tools ...





Anybody knows what this
tool is for?

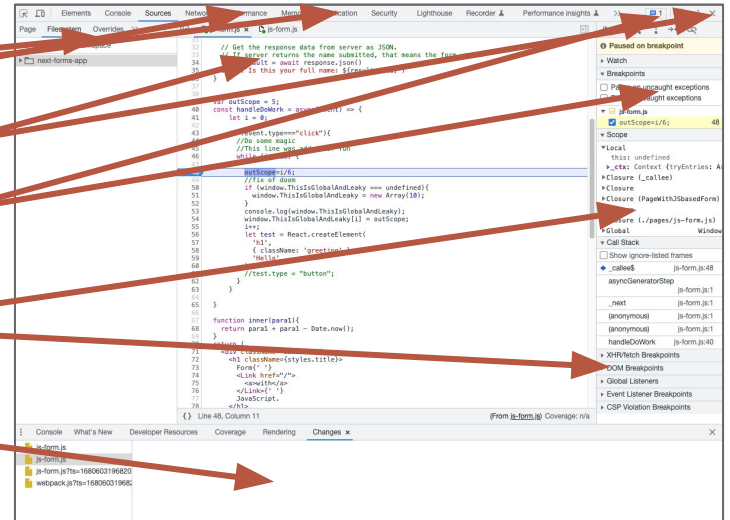
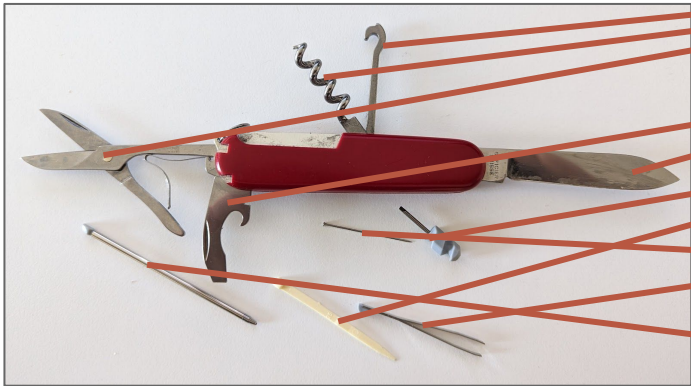
Interactive part!

... that are useful if
one knows their
purpose.





```
Page  Filesystem  Overrides  >>  js-form.js  js-form.js
+ Add folder to workspace
next-forms-app
22 // Get the response data from server as JSON.
23 // If server returns the name submitted, that means the form works.
34 const result = await response.json()
35 alert('Is this your full name: ${result.data}')
36 }
37
38 var outScope = {}
39 const handleWork = async(event) => {
40   let i = 0;
41   if (event.type === 'click'){
42     //Do some magic
43     //This line was added for fun
44     while (i < 10) {
45       outScope[i] = i;
46       if (window.ThisIsGlobalLeaky === undefined){
47         window.ThisIsGlobalLeaky = new Array(10);
48       }
49       console.log(window.ThisIsGlobalLeaky);
50       window.ThisIsGlobalLeaky[i] = outScope;
51       let test = React.createElement(
52         'div',
53         { className: 'greeting' },
54         'Hello'
55       );
56       //Test.type = 'button';
57     }
58   }
59   function innerParallel(){
60     return parallel = parallel - Date.now();
61   }
62   return {
63     <div className="container">
64       <div className="styles.title">
65         Form
66       </div>
67       <div className="styles.body">
68         <div className="styles.button">
69           Submit
70         </div>
71       </div>
72     </div>
73   }
74 }
75
76 {
77   <div className="container">
78     <div className="styles.title">
79       Form
80     </div>
81     <div className="styles.body">
82       <div className="styles.button">
83         Submit
84       </div>
85     </div>
86   </div>
87 }
88
89 {
90   <div className="container">
91     <div className="styles.title">
92       Form
93     </div>
94     <div className="styles.body">
95       <div className="styles.button">
96         Submit
97       </div>
98     </div>
99   </div>
100 }
```



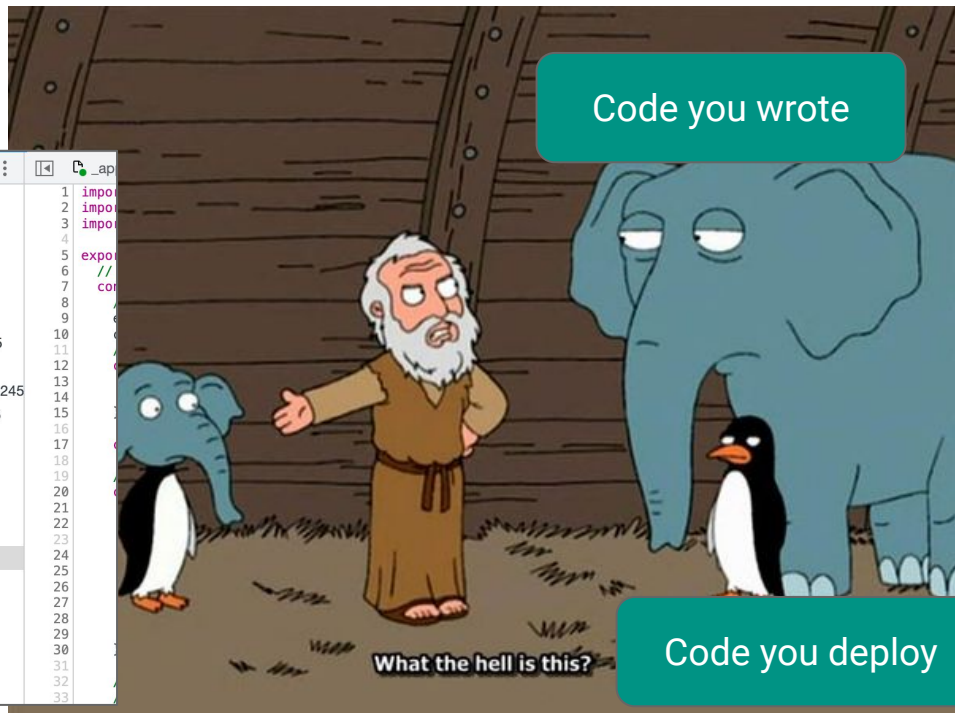
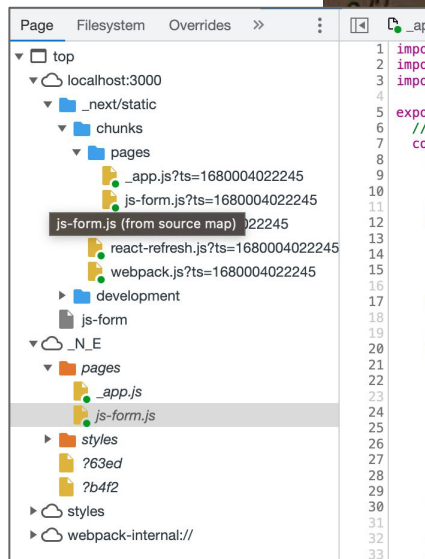
Tame complexity



Problem: Where is my code?!

Page explorer shows a mesh of artifacts

code_authored != code_deployed



Code you wrote

Code you deploy

Solution: Clear separation

Authored/Deployed view creates a clear separation

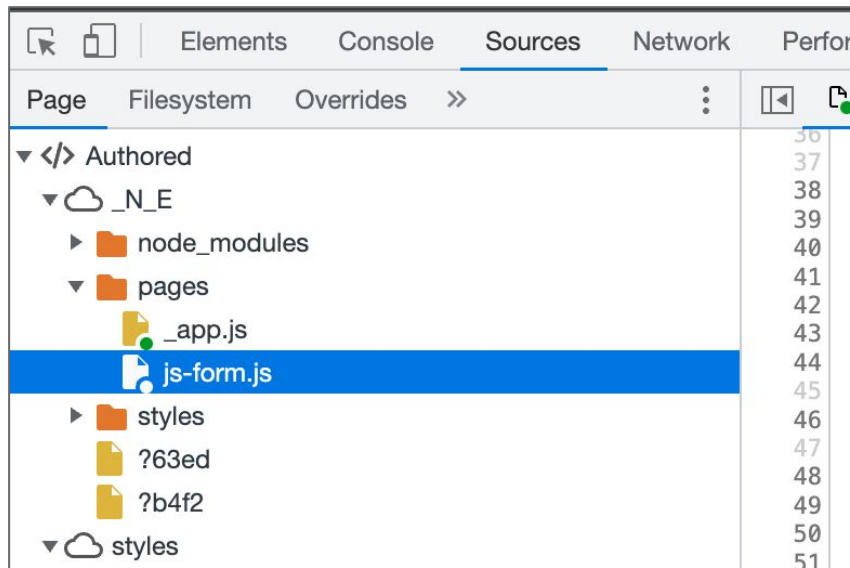
Still experimental, enable it via settings

Settings

- Preferences
- Workspace
- Experiments
- Ignore List
- Devices
- Throttling
- Locations
- Shortcuts

Experiments

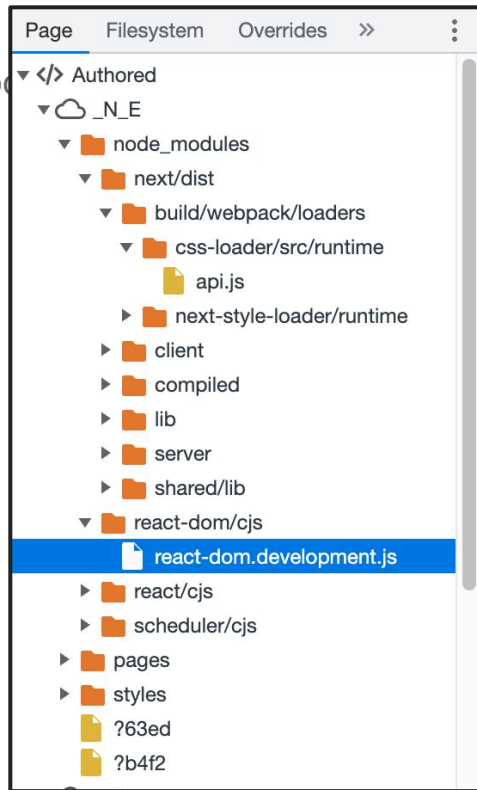
- ☒ Enable CSS Authoring hints for inactive rules, deprecated properties, etc.
- ☒ Enable color picking outside the browser window
- ☒ Group sources into Authored and Deployed trees [? Send feedback](#)
- ☒ Hide ignore-listed code in sources tree view
- ☐ Highlight important DOM properties in the Object Properties viewer
- ☒ Disable the deprecated `Color format` setting (requires reloading DevTools)
- ☐ Enable background page selector (e.g. for prerendering debugging)



Problem: 3rd party resources are all over the place

Modern web applications are importing tons of code

And you will pay for that when you debug (and deploy)



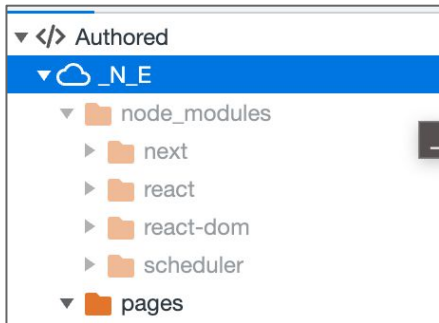
Call Stack	
<code>_callee\$</code>	<code>js-form.js:49</code>
<code>tryCatch</code>	<code>runtime.js:45</code>
<code>invoke</code>	<code>runtime.js:274</code>
<code>prototype.<computed></code>	<code>runtime.js:97</code>
<code>asyncGeneratorStep</code>	<code>js-form.js:1</code>
<code>_next</code>	<code>js-form.js:1</code>
<code>(anonymous)</code>	<code>js-form.js:1</code>
<code>(anonymous)</code>	<code>js-form.js:1</code>
<code>handleDoWork</code>	<code>js-form.js:40</code>
<code>callCallback</code>	<code>react-dom.development.js:3945</code>
<code>invokeGuardedCallbackDev</code>	<code>react-dom.development.js:3994</code>
<code>invokeGuardedCallback</code>	<code>react-dom.development.js:4056</code>
<code>invokeGuardedCallbackAndCatc...</code>	<code>react-dom.development.js:4070</code>
<code>executeDispatch</code>	<code>react-dom.development.js:8243</code>

Solution: Ignore what you are not interested in


Ignore lists enables you to ignore paths:

- Page tree greys-out ignored files and folders
- Stack trace will be cleanup-up too

Might already be prefilled by your framework of choice



▼ Call Stack	
<input type="checkbox"/> Show ignore-listed frames	
▶ _callee\$	js-form.js:49
asyncGeneratorStep	js-form.js:1
_next	js-form.js:1
(anonymous)	js-form.js:1
(anonymous)	js-form.js:1
handleDoWork	js-form.js:40



How often do you face
the problems of mixed
files and 3rd party files?

Interactive part!

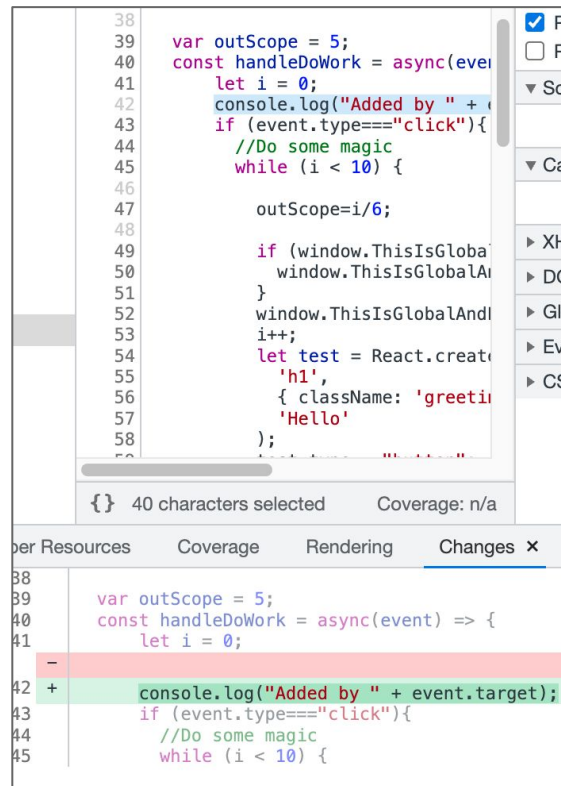
Alternatives to console.log



Problem: Console.log is great, and ...

... it requires changing the code and redeploying the new code.

Requires clean-up



```
38
39  var outScope = 5;
40  const handleDoWork = async(event) => {
41    let i = 0;
42    console.log("Added by " + event.target);
43    if (event.type==="click"){
44      //Do some magic
45      while (i < 10) {
46
47        outScope=i/6;
48
49        if (window.ThisIsGlobalAndI
50          window.ThisIsGlobalAndI
51        }
52        window.ThisIsGlobalAndI
53        i++;
54        let test = React.create
55        'h1',
56        { className: 'greetin
57        'Hello'
58      });
59    }
60  }
61}
```

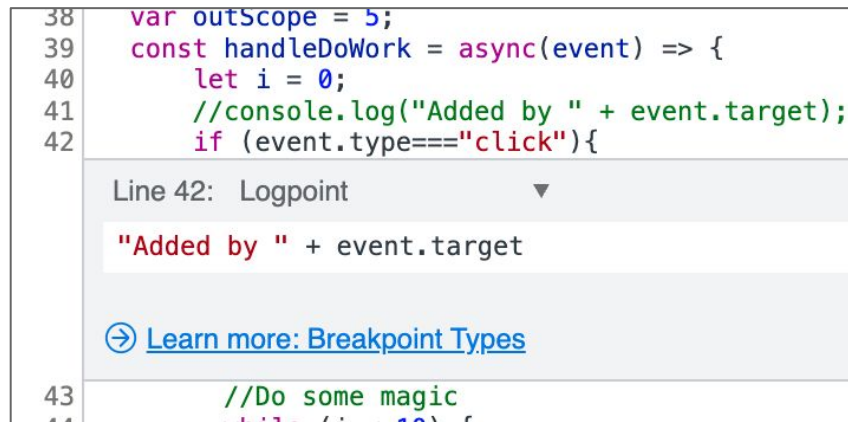
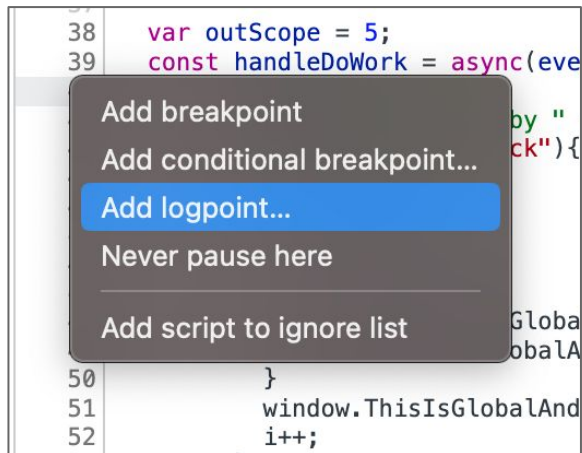
{ } 40 characters selected Coverage: n/a

	per Resources	Coverage	Rendering	Changes
38				
39				
40				
41				
42	-			
43	+			
44				
45				

Solution: Logpoints

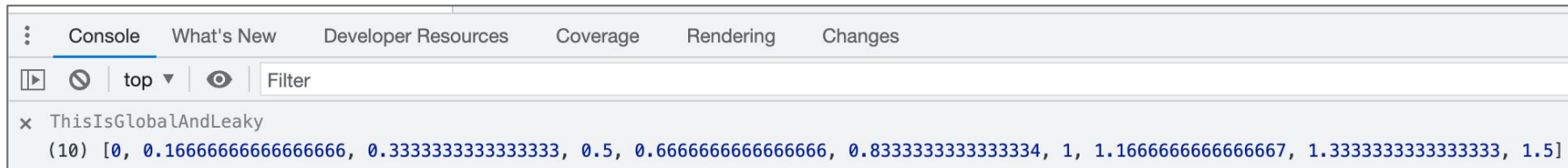
Don't require code changes.

Handling like breakpoints



Solution2: Live expressions

When you just want to observe the latest state of a variable



Problem: Use the current element in the console

Need to find the element again with a selector.

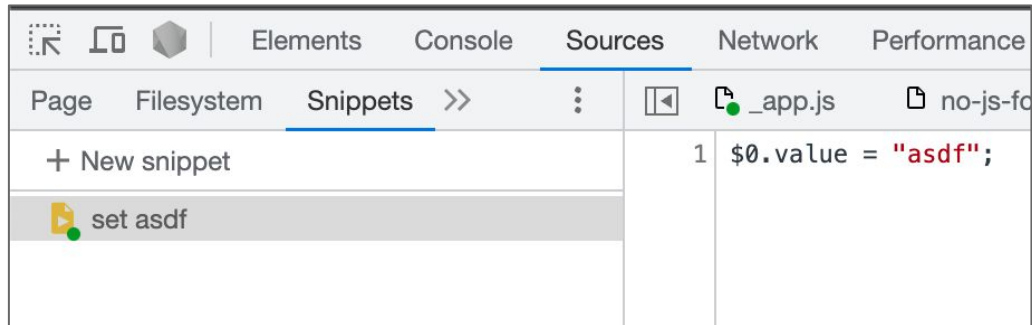
```
> $("#last").value = "asdf";  
< 'asdf'  
> $("#first").value = "asdf";  
< 'asdf'
```

Solution: \$0 always refers to the last, selected element

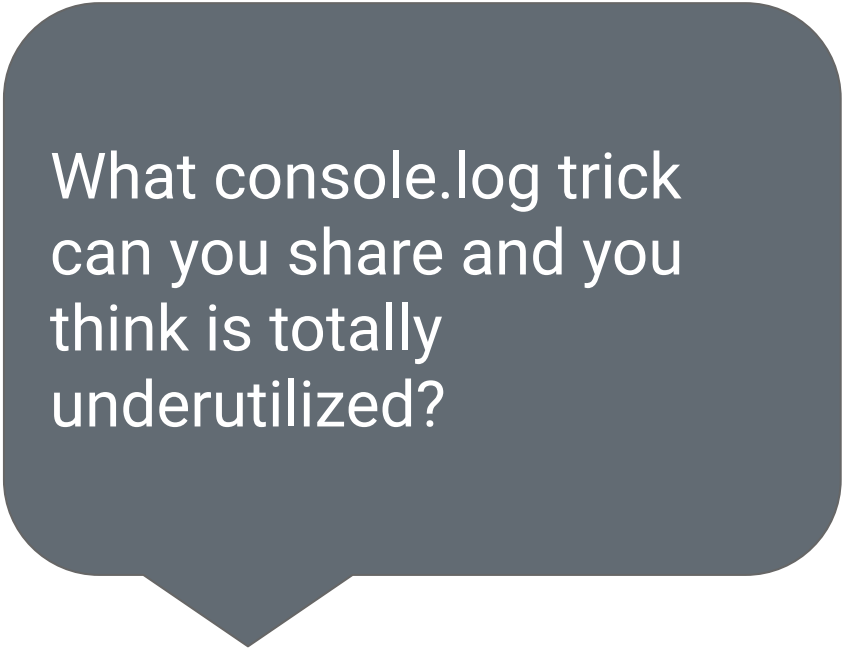
The last Element selected in the Elements panel is stored in \$0

There is also \$1...4.

Also works in Snippets




```
> $0.value = "asdf";  
< 'asdf'  
  
> $0.value = "asdf";  
< 'asdf'  
  
>
```



What console.log trick
can you share and you
think is totally
underutilized?

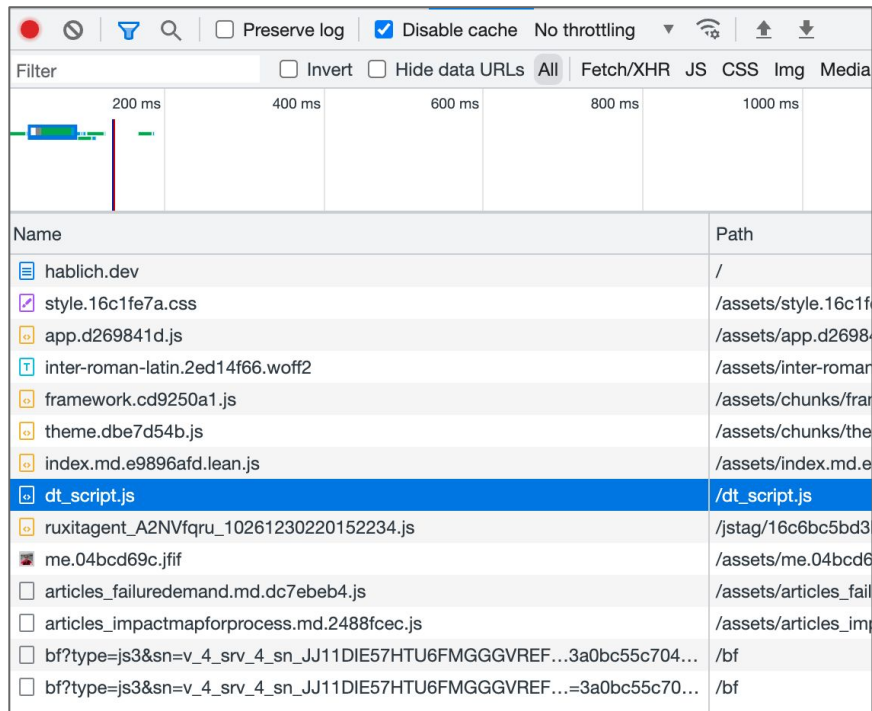
Interactive part!

Useful features I failed to identify a
cohesive theme for



Problem: Seeing the network request chain

Where is this coming from? What does it entail?



The screenshot shows the Chrome DevTools Network tab. The top toolbar includes a red stop button, a filter icon, a search icon, and checkboxes for 'Preserve log' (unchecked), 'Disable cache' (checked), and 'No throttling' (selected). The filter dropdown is set to 'All'. Below the toolbar is a timeline view with a vertical red line at 200 ms. The main panel displays a list of network requests with columns for 'Name' and 'Path'. The request 'dt_script.js' is highlighted in blue.

Name	Path
hablich.dev	/
style.16c1fe7a.css	/assets/style.16c1fe7a.css
app.d269841d.js	/assets/app.d269841d.js
inter-roman-latin.2ed14f66.woff2	/assets/inter-roman-latin.2ed14f66.woff2
framework.cd9250a1.js	/assets/chunks/framework.cd9250a1.js
theme.dbe7d54b.js	/assets/chunks/theme.dbe7d54b.js
index.md.e9896afd.lean.js	/assets/index.md.e9896afd.lean.js
dt_script.js	/dt_script.js
ruxitagent_A2NVfqr_u10261230220152234.js	/jstag/16c6bc5bd3ruxitagent_A2NVfqr_u10261230220152234.js
me.04bcd69c.jfif	/assets/me.04bcd69c.jfif
articles_faileddemand.md.dc7ebeb4.js	/assets/articles_faileddemand.md.dc7ebeb4.js
articles_impactmapforprocess.md.2488fcec.js	/assets/articles_impactmapforprocess.md.2488fcec.js
bf?type=js3&sn=v_4_srv_4_sn_JJ11DIE57HTU6FMGGGVREF...3a0bc55c704...	/bf
bf?type=js3&sn=v_4_srv_4_sn_JJ11DIE57HTU6FMGGGVREF...=3a0bc55c704...	/bf

Solution: Seeing the network request chain

- Use the Initiator Tab
- Hover over the Network item and hold SHIFT

× Headers Preview Response **Initiator** Timing

▼ **Request call stack**

- (anonymous) @ [dt_script.js:2244](#)
- (anonymous) @ [dt_script.js:2280](#)
- (anonymous) @ [dt_script.js:2281](#)

▼ **Request initiator chain**

- ▼ <https://hablich.dev/>
 - ▼ https://hablich.dev/dt_script.js
 - ▼ https://js-cdn.dynatrace.com/jstag/16c6bc5bd3b/ruxitagent_...
 - https://bf33032bms.bf.dynatrace.com/bf?type=js3&sn=v_4_sr...
 - https://bf33032bms.bf.dynatrace.com/bf?type=js3&sn=v_4_sr...

Name

- hablich.dev
- style.16c1fe7a.css
- app.d269841d.js
- inter-roman-latin.2ed14f66.woff2
- framework.cd9250a1.js
- theme.dbe7d54b.js
- index.md.e9896afd.lean.js
- dt_script.js
- ruxitagent_A2NVfqr_10261230220152234.js
- me.04bcd69c.jiff
- articles_faileddemand.md.dc7eb4.js
- articles_impactmapforprocess.md.2488fcec.js
- bf?type=js3&sn=v_4_srv_4_sn_JJ11DIE57HTU6FMGGGVREF...3a0bc55c704c9d8f&crc=3159633315&en=0my2gprh&end=1
- bf?type=js3&sn=v_4_srv_4_sn_JJ11DIE57HTU6FMGGGVREF...3a0bc55c704c9d8f&crc=748901241&en=0my2gprh&end=1

Problem: CORS error due to misconfigured headers

The screenshot shows the Chrome DevTools interface for the URL `cors-demo-devtools.glitch.me/`. The **Network** panel is active, displaying a list of requests:


Name	Status	Type	Initiator	Size	Time	Waterfall
cors-demo-devt...	200	document	Other	1.1 kB	245...	
style.css	200	stylesheet	(index)	401 B	136...	
todos.json	CORS error	fetch	(index):22	0 B	239...	

Below the table, it shows 3 requests, 1.5 kB transferred, and 1.1 kB resources. The **Console** panel shows the following error:

```
✖ Access to fetch at 'https://third-party-server.glitch.me/todos.json' from origin 'https://cors-demo-devtools.glitch.me:1s.json' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource. If an opaque response serves your needs, set the request's mode to 'no-cors' to fetch the resource with CORS disabled.
✖ ▶ GET https://third-party-server.glitch.me/todos.json net::ERR_FAILED 200 (index):22
✖ ▶ Uncaught (in promise) TypeError: Failed to fetch at (index):22:5
```

I want to modify the response headers of some request, and there is no quick way for me to do that.

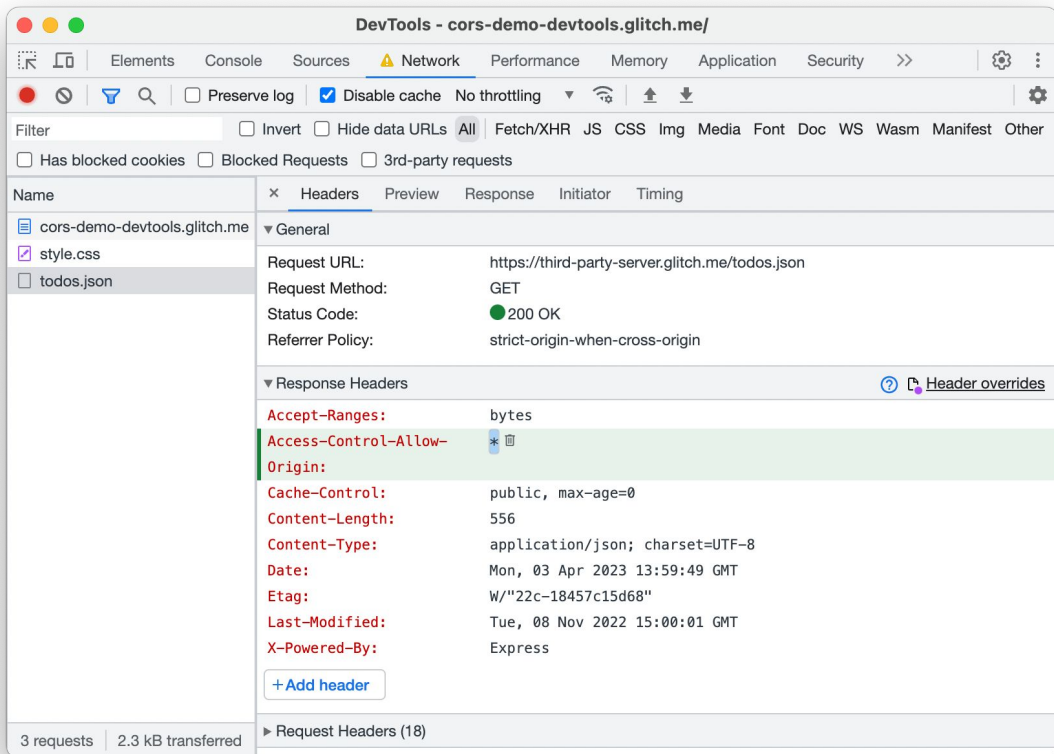
Problem: I want to experiment with different values for response headers

×	Headers	Preview	Response	Initiator	Timing	Cookies
▼ General						
Request URL:		https://www.dynatrace.de/				
Request Method:		GET				
Status Code:		 200 OK				
Remote Address:		18.185.35.211:443				
Referrer Policy:		strict-origin-when-cross-origin				
▼ Response Headers		<input type="checkbox"/> Raw				
Cache-Control:		public				
Connection:		keep-alive				
Content-Encoding:		gzip				
Content-Security-Policy:		frame-ancestors 'self'				
Content-Type:		text/html; charset=utf-8				
Date:		Mon, 03 Apr 2023 13:51:18 GMT				
Etag:		W/"6422ec79-3c49e:dtagent10261230220152234aZds:dtagent10261230220152234aZds:dtagent10261230220152234aZds"				
Last-Modified:		Tue, 28 Mar 2023 13:32:39 GMT				
Server:		nginx				
Server-Timing:		dtSInfo;desc="0", dtRpid;desc="-1865770864"				
Transfer-Encoding:		chunked				
Vary:		Accept-Encoding				
X-Aws-Layer:		content				
X-Aws-Region:		eu-central-1a				
X-Cache-Status:		HIT				
X-Frame-Options:		sameorigin				
X-Instance-Revision:		b9b2a2a597				
X-Oneagent-Js-Injection:		true				
X-Ruxit-Js-Agent:		true				

e.g.

- CORS (cross-origin resource sharing)
- Content-Security-Policy
- Cross-Origin Isolation headers
- Permissions-Policy headers
- Feature flags
- ...

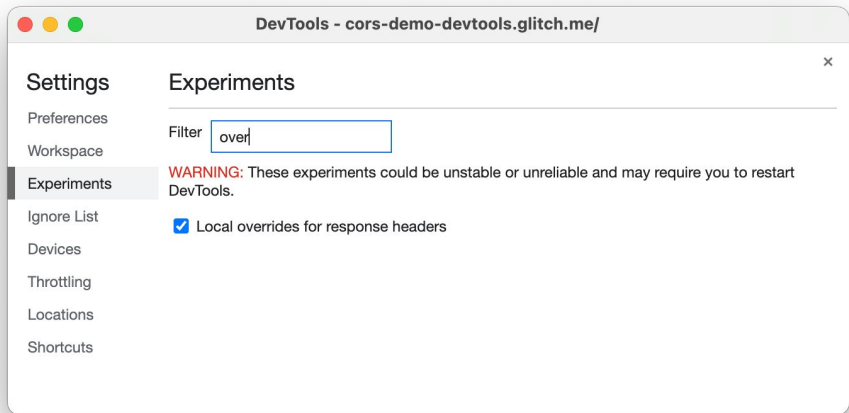
Solution: Header Overrides



1. Edit response headers directly in Chrome DevTools
2. Refresh page/request
3. Chrome behaves as if server was sending modified headers

Header Overrides – Disclaimer

Enable DevTools experiment
(switched on by default in Chrome 113)



What two features would
you love to have in
DevTool's network panel

except throttling all network traffic ?

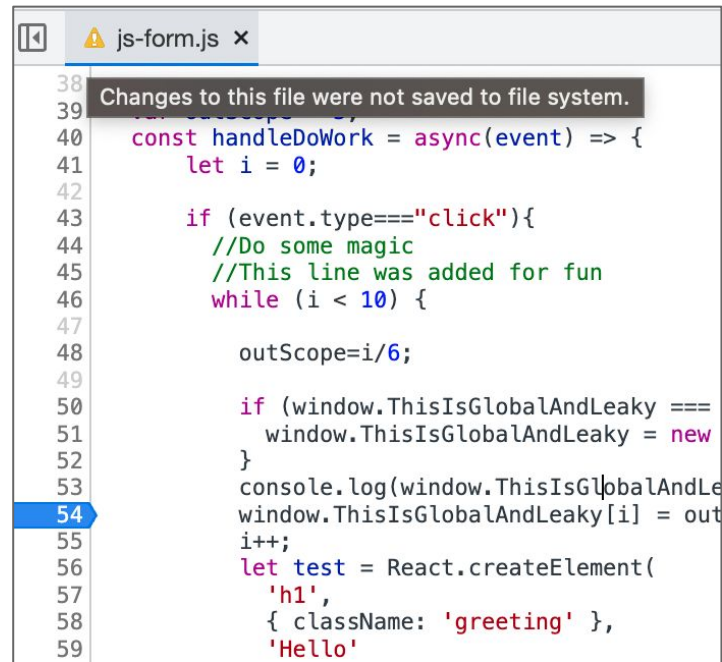
Interactive part!

Problem: Fixes made in DevTools need to be copied over to the source repo

Assume you have source maps up and running

You are hunting and fixing a bug

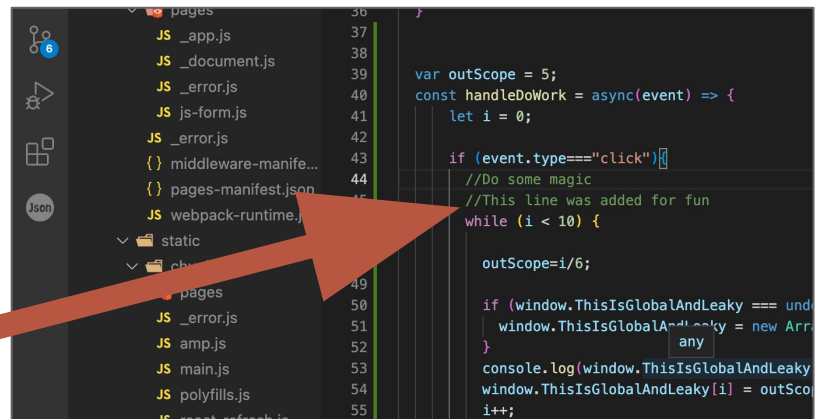
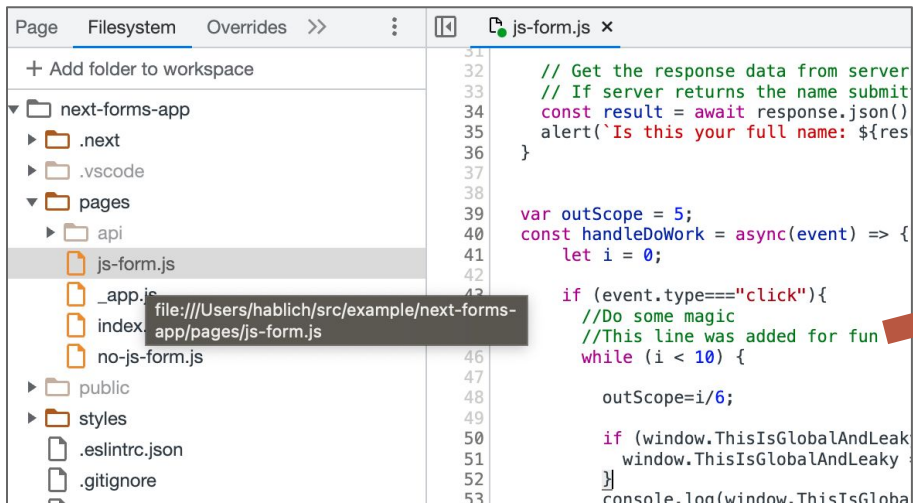
Now you want to persist that bug fix into the actual source code



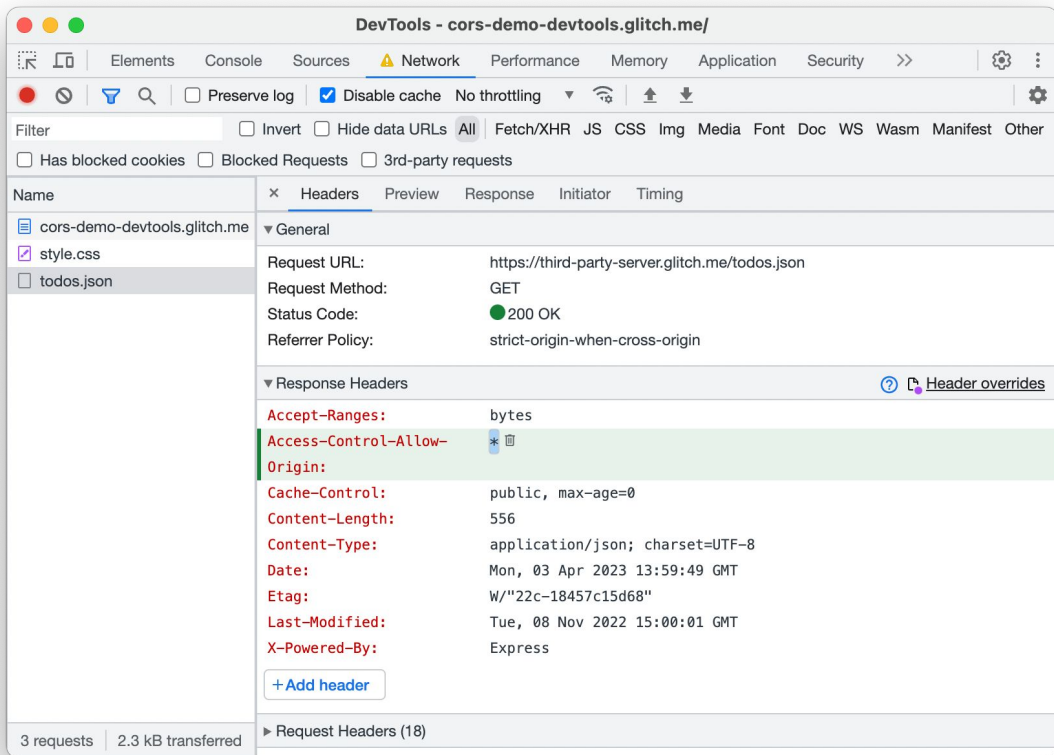
```
js-form.js x
Changes to this file were not saved to file system.
38
39
40 const handleDoWork = async(event) => {
41   let i = 0;
42
43   if (event.type==="click"){
44     //Do some magic
45     //This line was added for fun
46     while (i < 10) {
47
48       outScope=i/6;
49
50       if (window.ThisIsGlobalAndLeaky ===
51         window.ThisIsGlobalAndLeaky = new
52       )
53       console.log(window.ThisIsGlobalAndLeaky)
54       window.ThisIsGlobalAndLeaky[i] = out
55       i++;
56       let test = React.createElement(
57         'h1',
58         { className: 'greeting' },
59         'Hello'
```

Solution: Fixes made in DevTools need to be copied over to the source repo


Workspaces enable you to save a change that you make in Devtools to a local copy of the same file on your computer



Solution2: Local overrides, when you can't modify the source locally



1. Edit response headers directly in Chrome DevTools
2. Refresh page/request
3. Chrome behaves as if server was sending modified headers



When do you make
changes and fixes to your
code in DevTools?

Interactive part!

Problem: Reproducing the bug takes multiple steps

Most of the time one needs to do a few steps before the bug reproduces

That can get tiresome (error-prone) quite fast

Issue 1426337: Sources tab > "Search in all files" option missing

Reported by [REDACTED] on Tue, Mar 21, 2023, 8:35 AM GMT+1 (10 days ago)

Steps to reproduce the problem:

1. Open dev tools
2. Go to Sources tab
3. On the sidebar, right-click on "top" page, nothing happens

Problem Description:

On right click on the "top" page, it should display the "Search in all files" option.
It works fine on 110.0.5481.77

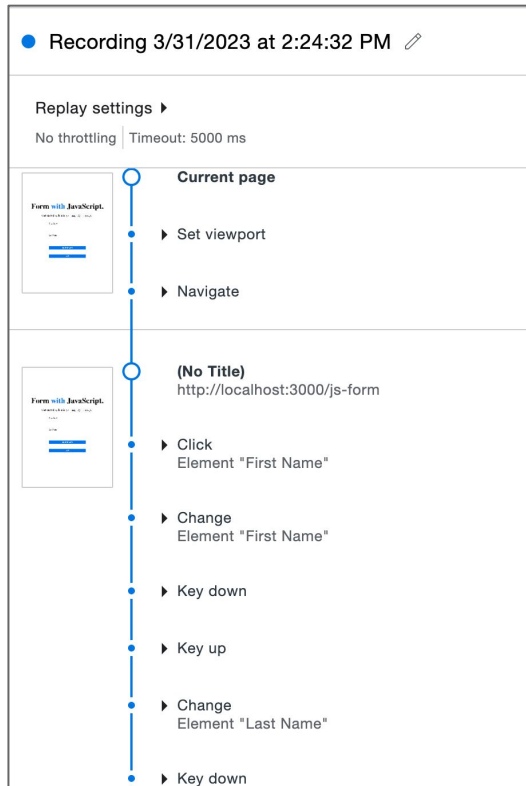
Additional Comments:

Solution: Use the Recorder to create reproducers


With the Recorder panel you can record interactions and replay them.

Also works with Performance profiling.


Reproducers also can be shared.




The screenshot displays the Recorder panel interface. At the top, it indicates 'Recording 3/31/2023 at 2:24:32 PM' with a pencil icon for editing. Below this, the 'Replay settings' section shows 'No throttling' and 'Timeout: 5000 ms'. The main area lists recorded interactions, each preceded by a thumbnail of the page state. The first interaction is labeled 'Current page' and includes 'Set viewport' and 'Navigate'. The second interaction is labeled '(No Title)' with the URL 'http://localhost:3000/js-form' and includes a sequence of actions: 'Click Element "First Name"', 'Change Element "First Name"', 'Key down', 'Key up', 'Change Element "Last Name"', and 'Key down'.

● Recording 3/31/2023 at 2:24:32 PM 

Replay settings ▶
No throttling | Timeout: 5000 ms

 **Current page**

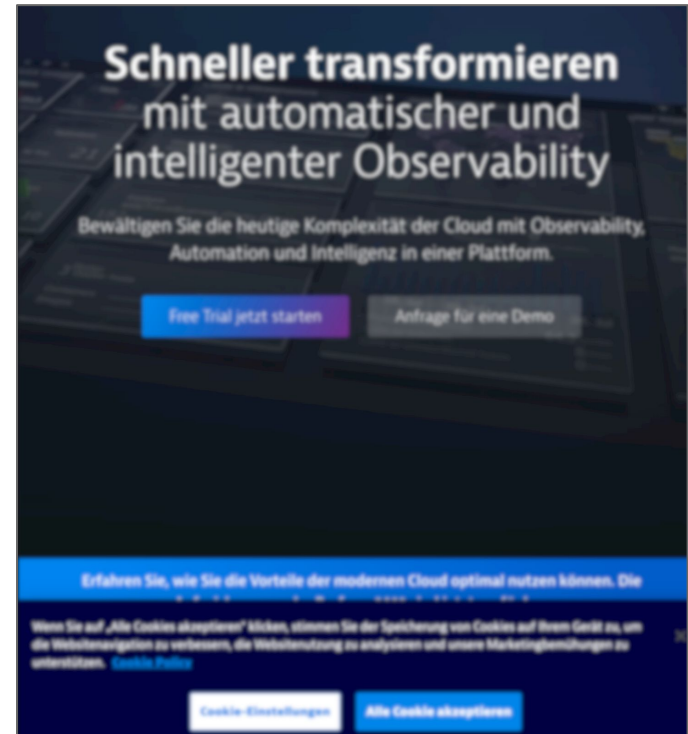
- ▶ Set viewport
- ▶ Navigate

 **(No Title)**
http://localhost:3000/js-form

- ▶ Click Element "First Name"
- ▶ Change Element "First Name"
- ▶ Key down
- ▶ Key up
- ▶ Change Element "Last Name"
- ▶ Key down

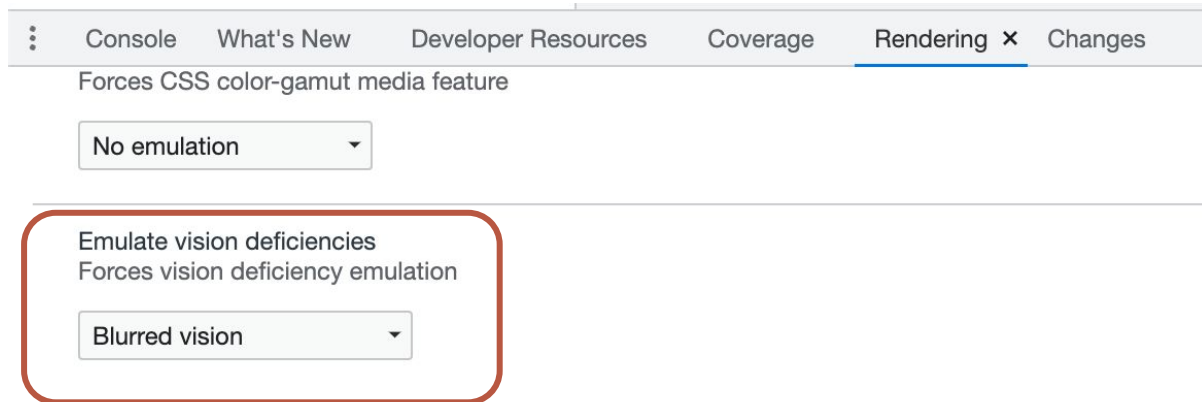
Problem: How does my page look for vision impaired people?

Turns out, that there are tons of people that have a color vision deficiency e.g. around 4 % of population have red-green blindness.



Solution: Rendering->CVD emulation

The Rendering drawer has all sorts of emulations and stuff including a color vision deficiency emulation.



Problem: WebAssembly memory is an array

It is an Array and debugged as an Array.

Navigating MegaBytes of data is ... complicated.

Identifying where a variable is stored is not great either.

```
2  const n = 60; n = 60
3  const b1 = new Uint8Array(n); b1 = Uint8A
4  const b2 = new Uint8Array(n+10); b2 = Uint
5
6  const str = 'This is a string in the ArrayE
7  for (var i = 0; i < str.length; ++i) {
8      b1[i] = str.charCodeAt(i); b1 = Uint8Ar
9      b2[i] = i + 33; b2 = Uint8Array(70)
10 }
11
12 // Opening this buffer in the Memory Inspe
13 // will open the same view as for opening c
14 // b1.
15 const buffer = b1.buffer; buffer = ArrayB
16
17 for (var i = str.length; i < n; ++i) { st
18     b1[i] = i;
19     b2[i] = n - i - 1;
20 }
21 }
22 runDemo();
```

▼ b1: Uint8Array(60)

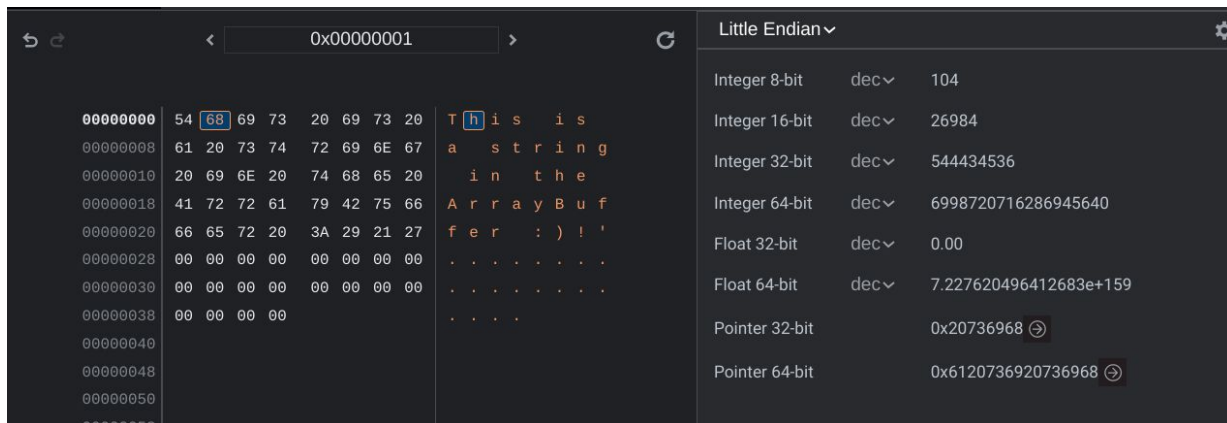
0:	84
1:	104
2:	105
3:	115
4:	32
5:	105
6:	115
7:	32
8:	97
9:	32
10:	115
11:	116
12:	114
13:	105
14:	110
15:	103
16:	32
17:	105

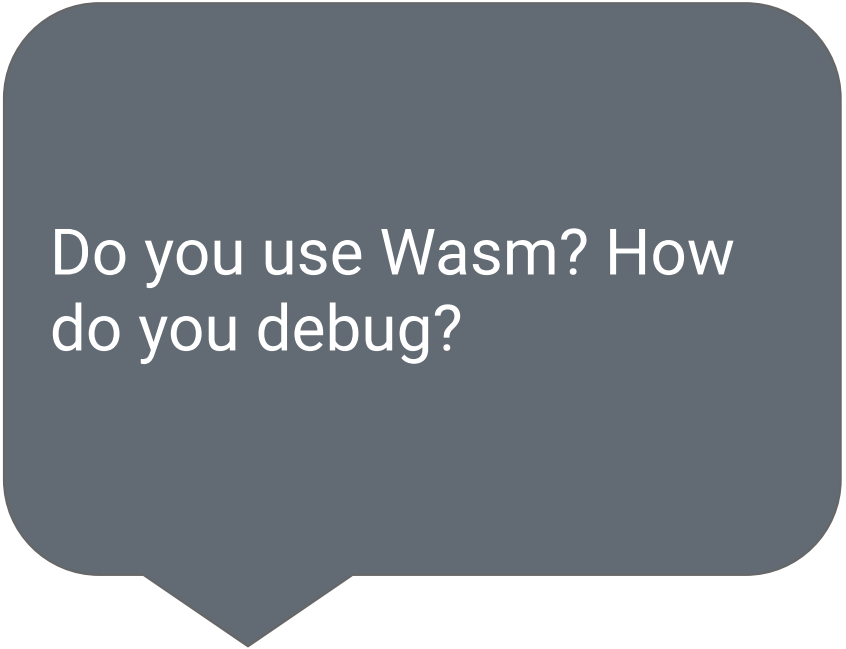
Solution: Linear memory inspector

Displays an ArrayBuffer like a hex editor.

Supports all sorts of memory interpretations.

Combine it with DWARF debugging to also identify variables in the memory





Do you use Wasm? How
do you debug?

Interactive part!

Call to action approaching!

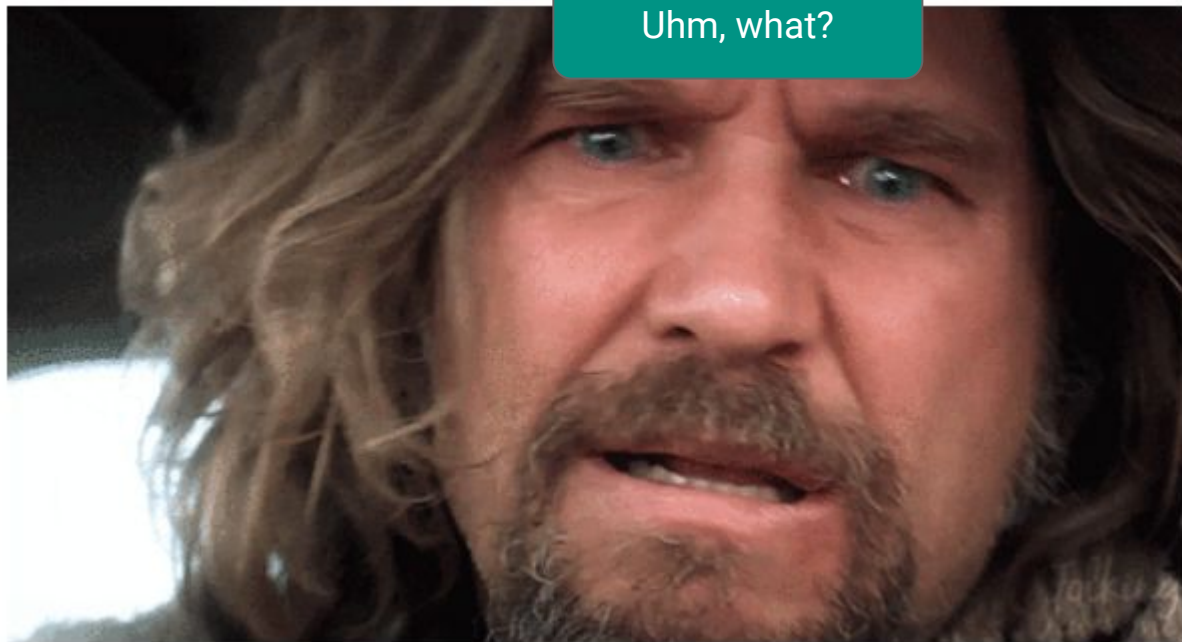


Problem: How to reach the DevTools team?

I have an idea

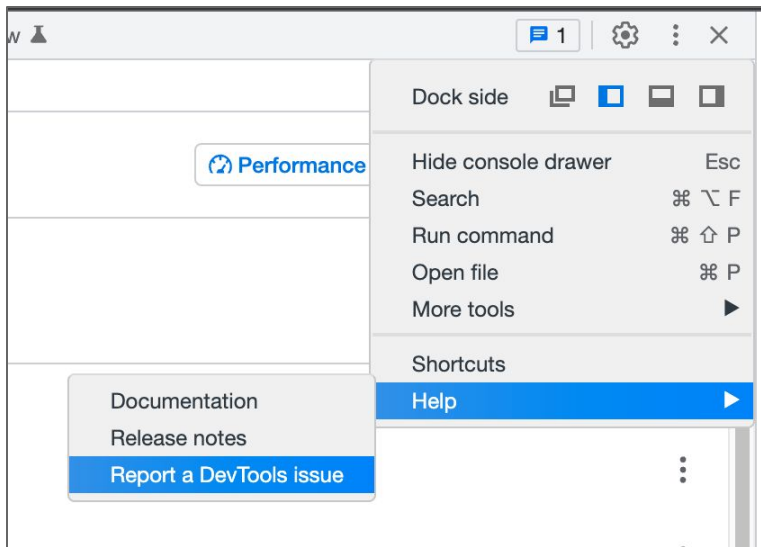
I found a bug

I have an opinion

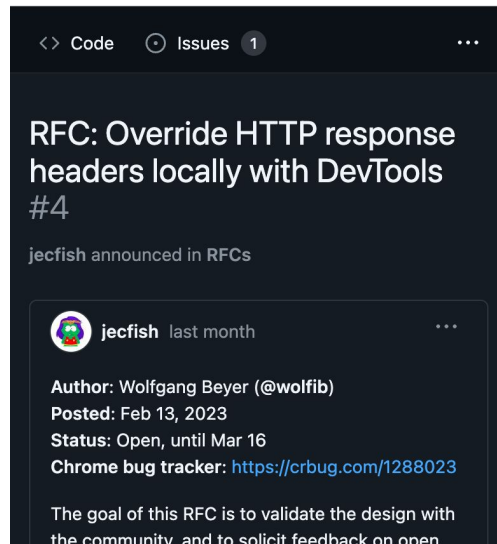


Solution: Multiple channels

File a bug via DevTools

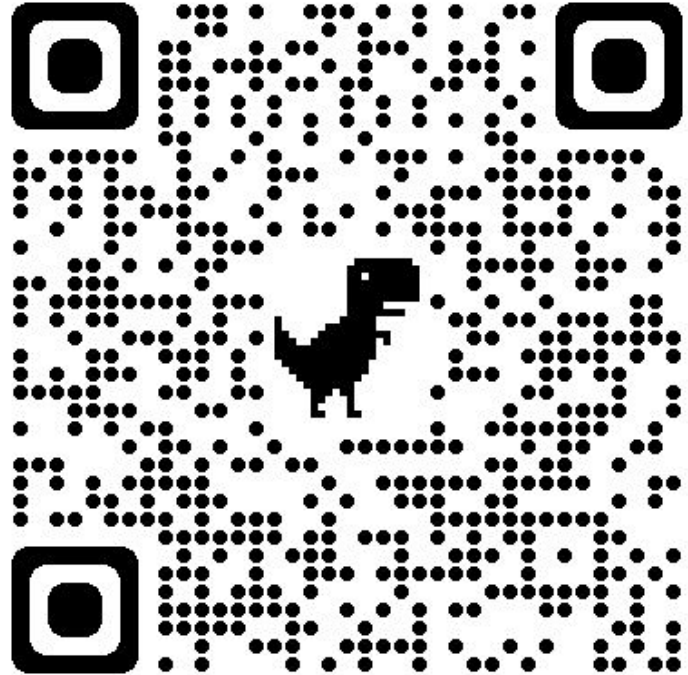


Participate in RFCs



Do you want to know more?

A lot more in-depth information of the mentioned features can be found on the [Chrome DevTools documentation website](#).



Fin

Questions?!



goo.gle/NewInDevTools



goo.gle/devtools-rfc



[@ChromeDevTools](https://twitter.com/ChromeDevTools)



goo.gle/devtools-report-issue