



JS

Debugging Java Script

- Firefox Architecture redesigned for improved
 - Responsiveness
 - Stability
 - Security
- Multi process Architecture
 - Firefox's user interface will run in one process (things like the address bar, the tabs and menus)
 - The content (websites) will run in other processes
 - Advantage????

With multi process architecture, if a website crashes, it doesn't also crash the whole browser

Result of Electrolysis ???



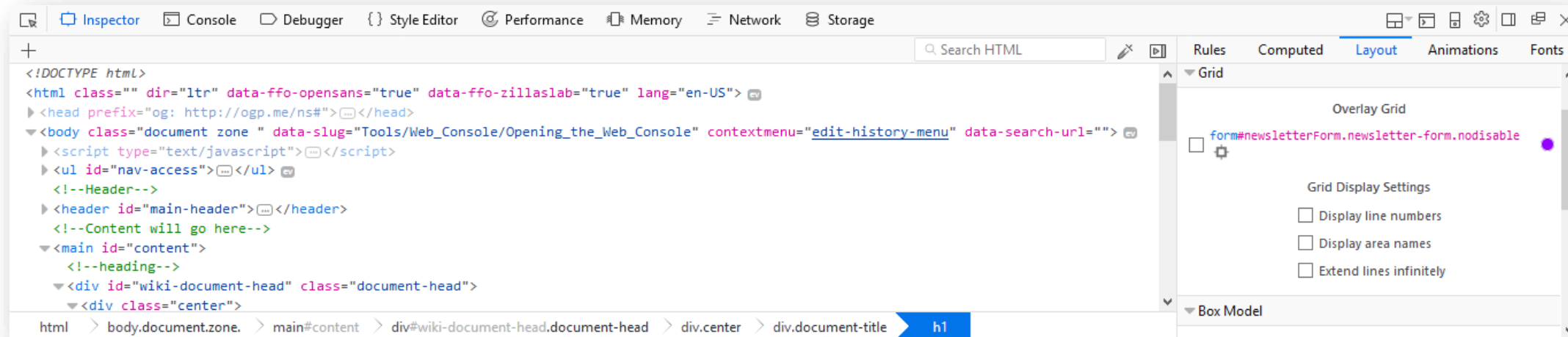
Firebug wasn't designed with multi process in mind

No Firebug , How to debug?

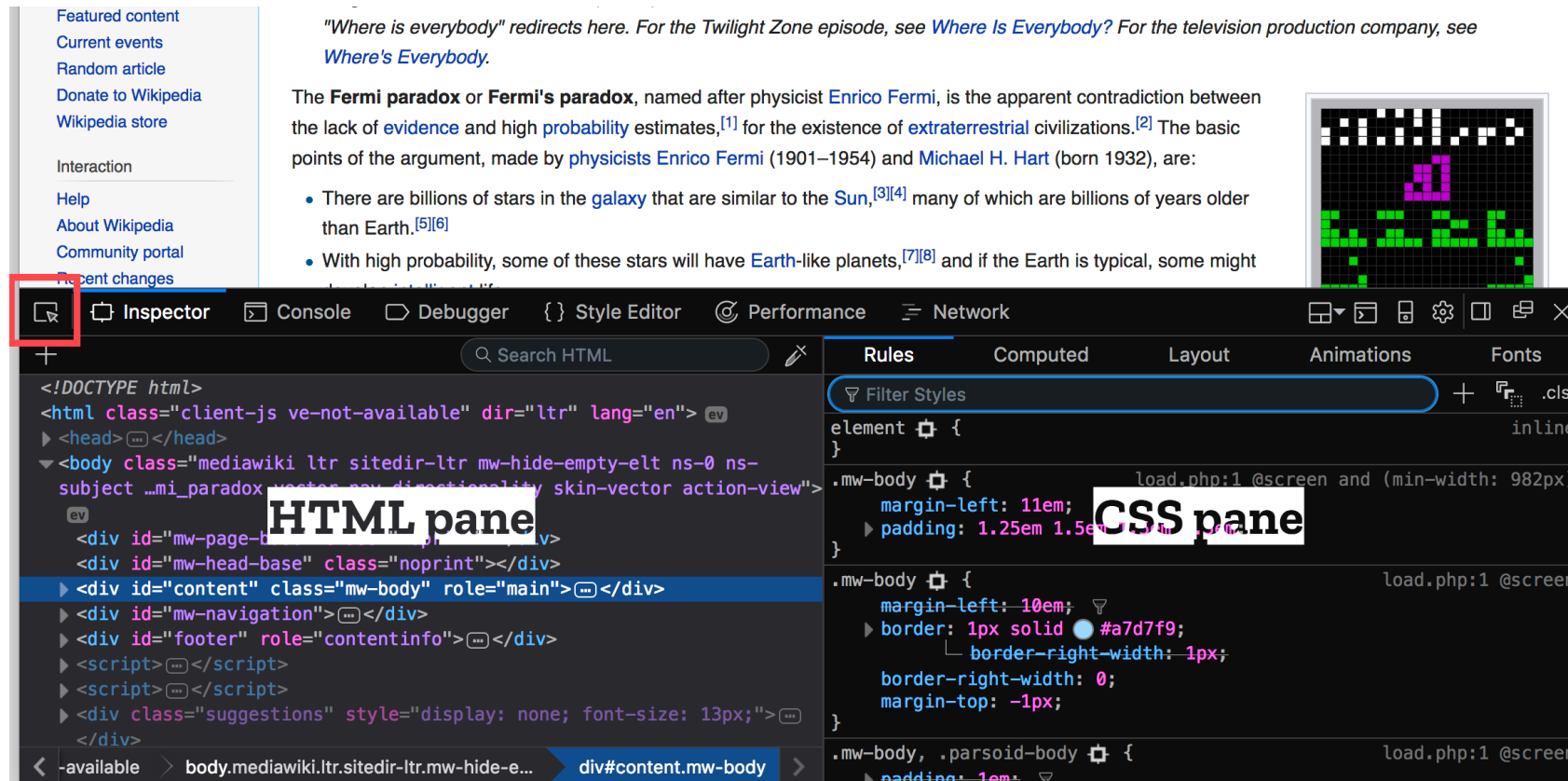


- Firefox Developer Tools
 - Web Console
 - Inspector
 - Debugger
 - Style Editor
 - Performance Tool
 - Network Monitor
 - Storage Inspector

No Firebug, How to debug?



Inspector - UI



HTML Pane



Search

Search HTML

HTML Tree

```
+
<!DOCTYPE html>
<html class="client-js ve-not-available" dir="ltr" lang="en"> ev
  > <head> ... </head>
  ▼ <body class="mediawiki ltr sitedir-ltr mw-hide-empty-elt ns-0 ns-subject ...e-Starfish
    vector-nav-directionality skin-vector action-view"> ev
      <div id="mw-page-base" class="noprint"></div>
      <div id="mw-head-base" class="noprint"></div>
      > <div id="content" class="mw-body" role="main"> ... </div>
      > <div id="mw-navigation"> ... </div>
      > <div id="footer" role="contentinfo"> ... </div>
      > <script> ... </script>
      > <script> ... </script>
      > <div class="suggestions" style="display: none; font-size: 13px;"> ... </div>
    </body>

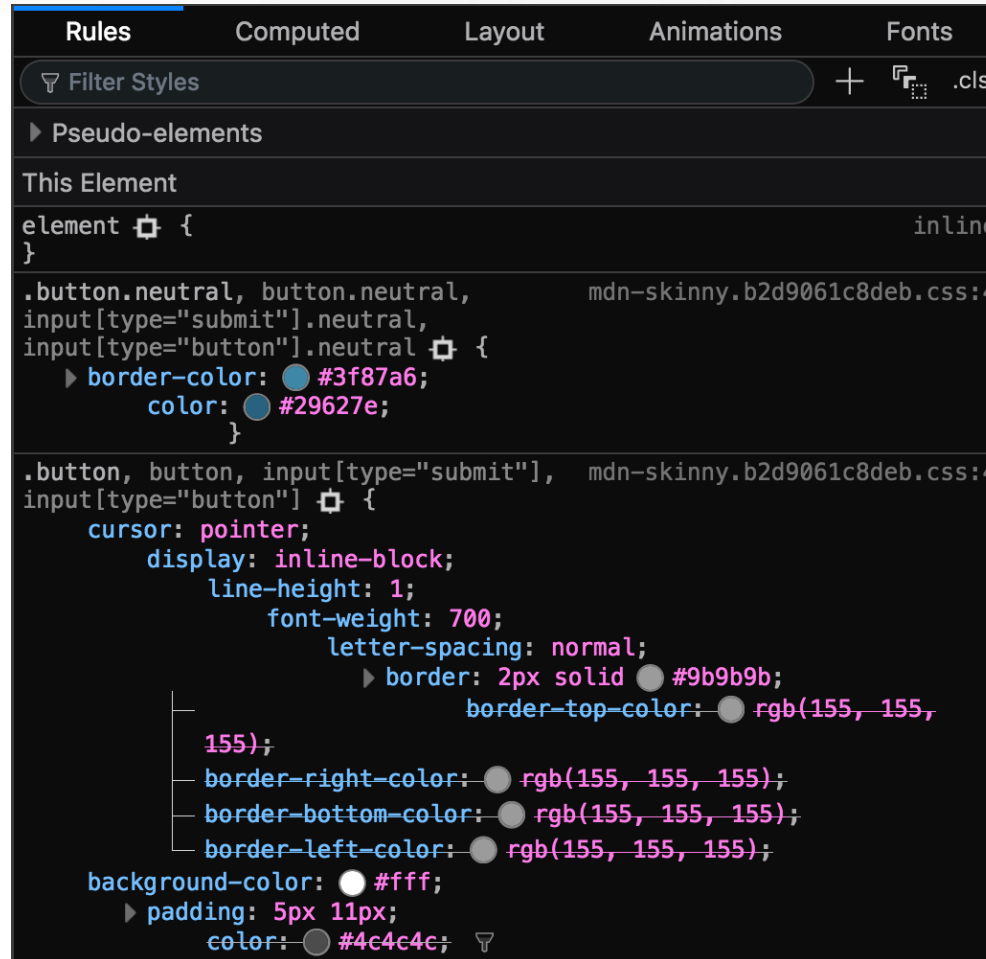
< tml.client-js.ve-not-available > body.mediawiki.ltr.sitedir-ltr.mw-hide-e... div#mw-head-base.noprint >
```

Breadcrumbs

Editing HTML



```
▼ ☐
  ▼ <div id="mw-indicator-pp-default" class="mw-indicator">
    ▶ <a href="/wiki/Wikipedia:Protection_policy#semi" title="This article is semi-protected.">...</a>
  </div>
</div>
  ▼ <h1 id="firstHeading" class="firstHeading" lang="en">Starfish</h1>|
  ▼
  ▶ <div id="bodyContent" class="mw-body-content">...</div>
</div>
▶ <div id="mw-navigation">...</div>
```

- Used to step through JavaScript code and examine or modify its state to help track down bugs

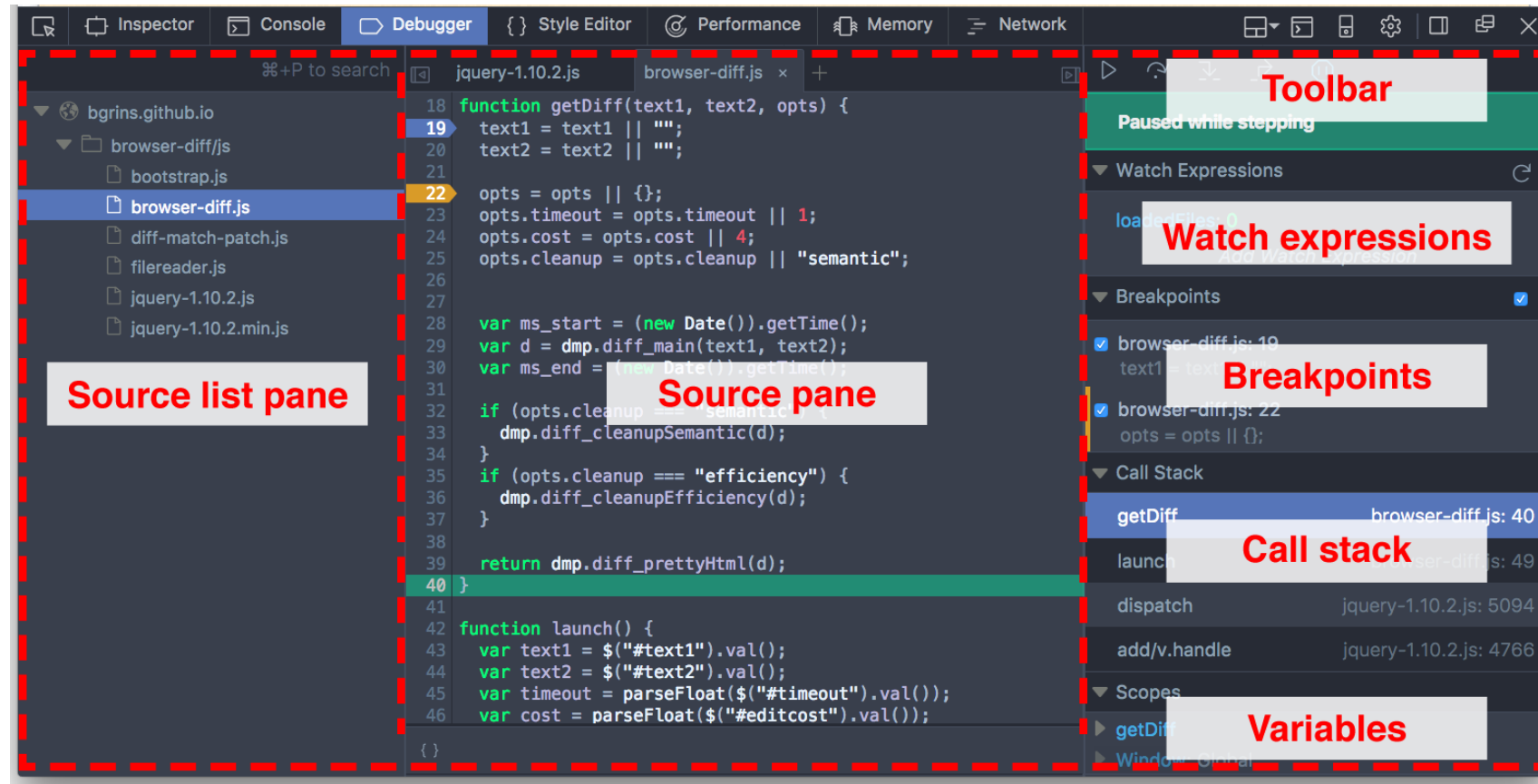
Opening the Debugger



There are three ways to open the debugger:

- select "Debugger" from the Web Developer submenu in the Firefox Menu (or Tools menu if you display the menu bar or are on Mac OS X)
- press the `Ctrl` `Shift` `S` (`Command` `Option` `S` on OS X) keyboard shortcut
- press the menu button (≡), press "Developer", then select "Debugger".

Debugger - UI



Source list Pane



The screenshot displays a web browser's developer tools interface. The **Source list pane** on the left shows a file tree for `bgrins.github.io`, with `browser-diff.js` selected. A red dashed box highlights this pane, and a white box with the text "Source list pane" is overlaid on it. The main code editor shows the `getDiff` function in `browser-diff.js`, with line 19 selected. The right sidebar contains the **Debugger** tab, which is currently **Paused while stepping**. It lists **Watch Expressions** (none), **Breakpoints** (two: line 19 and line 22), and a **Call Stack** with frames for `getDiff`, `launch`, `dispatch`, and `add/v.handle`.

Source Pane



Source pane

```
18 function getDiff(text1, text2, opts) {
19   text1 = text1 || "";
20   text2 = text2 || "";
21
22   opts = opts || {};
23   opts.timeout = opts.timeout || 1;
24   opts.cost = opts.cost || 4;
25   opts.cleanup = opts.cleanup || "semantic";
26
27
28   var ms_start = (new Date()).getTime();
29   var d = dmp.diff_main(text1, text2);
30   var ms_end = (new Date()).getTime();
31
32   if (opts.cleanup === "semantic") {
33     dmp.diff_cleanupSemantic(d);
34   }
35   if (opts.cleanup === "efficiency") {
36     dmp.diff_cleanupEfficiency(d);
37   }
38
39   return dmp.diff_prettyHtml(d);
40 }
41
42 function launch() {
43   var text1 = $("#text1").val();
44   var text2 = $("#text2").val();
45   var timeout = parseFloat($("#timeout").val());
46   var cost = parseFloat($("#editcost").val());
47
48 }
```

Paused while stepping

Watch Expressions

loadedFiles: 0

Add Watch Expression

Breakpoints

browser-diff.js: 19
text1 = text1 || "";

browser-diff.js: 22
opts = opts || {};

Call Stack

getDiff browser-diff.js: 40

launch browser-diff.js: 49

dispatch jquery-1.10.2.js: 5094

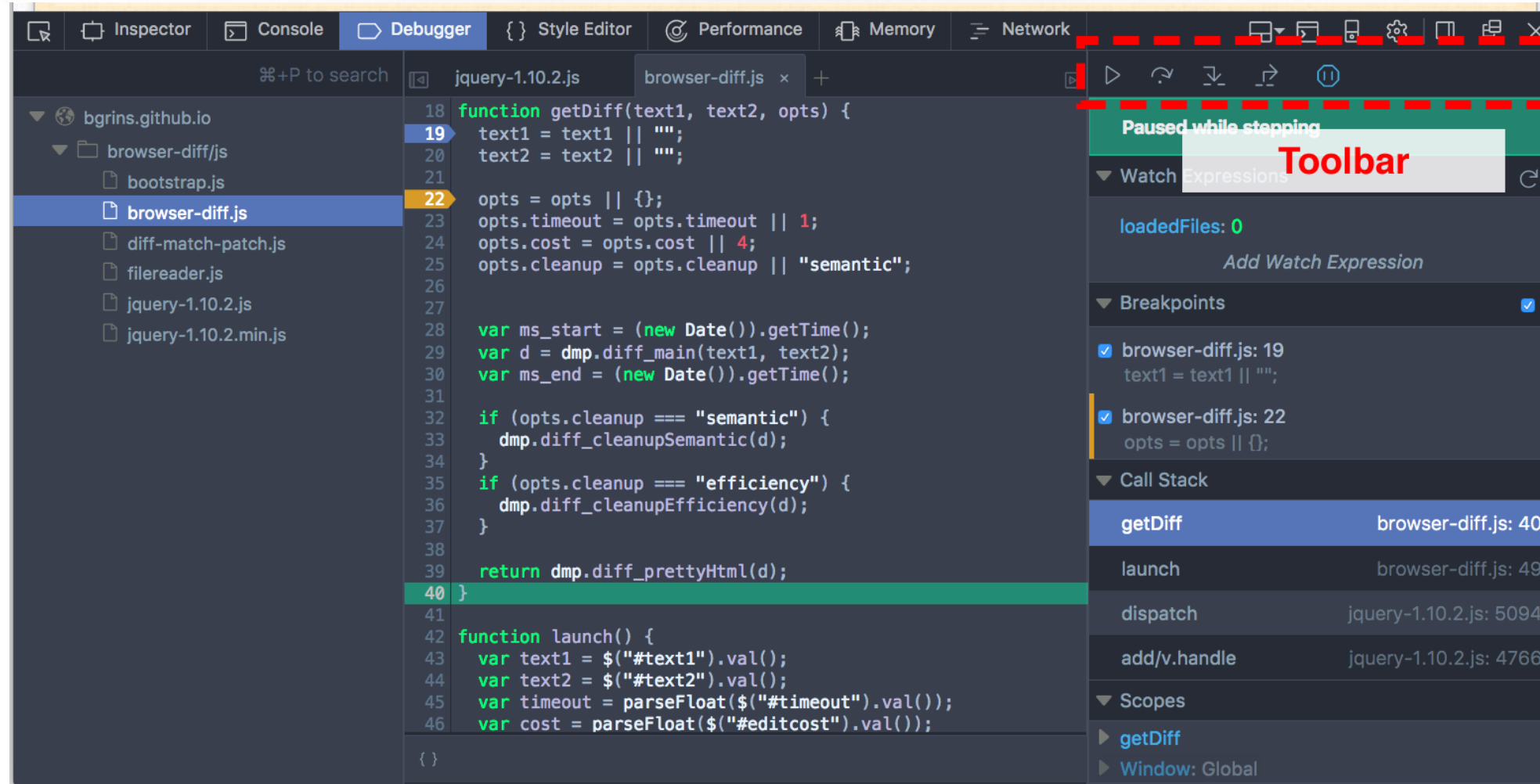
add/v.handle jquery-1.10.2.js: 4766

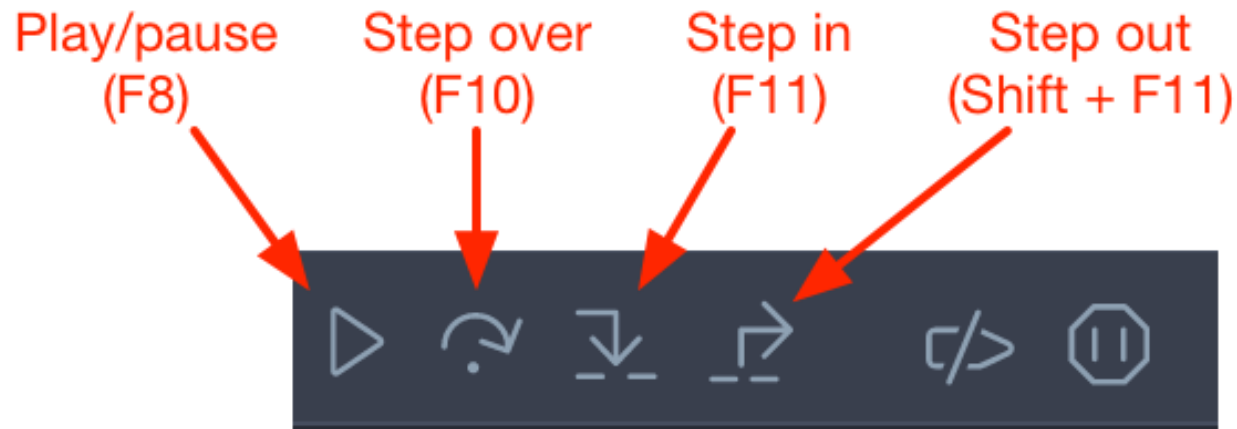
Scopes

getDiff

Window: Global

Tool Bar





Play: run to the next breakpoint

Step over: advance to the next line in the same function.

Step in: advance to the next line in the function, unless on a function call, in which case enter the function being called

Step out: run to the end of the current function

Setting Breakpoints



- in the **source pane**, click on the line number for the line you want to break at
- in the **source pane**, activate the context menu while on the line you want to break at, and select "Add breakpoint"
- in the **source pane**, highlight the line you want to break at and press Ctrl+B (Windows/Linux) or Command+B (Mac OS X)

Each breakpoint is shown in two places in the debugger:

- the **breakpoints list** shows the filename and line number for the breakpoint
- the line in the **source pane** is marked with a blue arrow, or an orange arrow if the breakpoint is **conditional**.

Breakpoint List



Paused while stepping

Watch Expressions

loadedFiles: 0

Add Watch Expression

Breakpoints

- ☒ browser-diff.js: 19
text1 = text1 || "";
- ☒ browser-diff.js: 22
opts = opts || {};

Call Stack

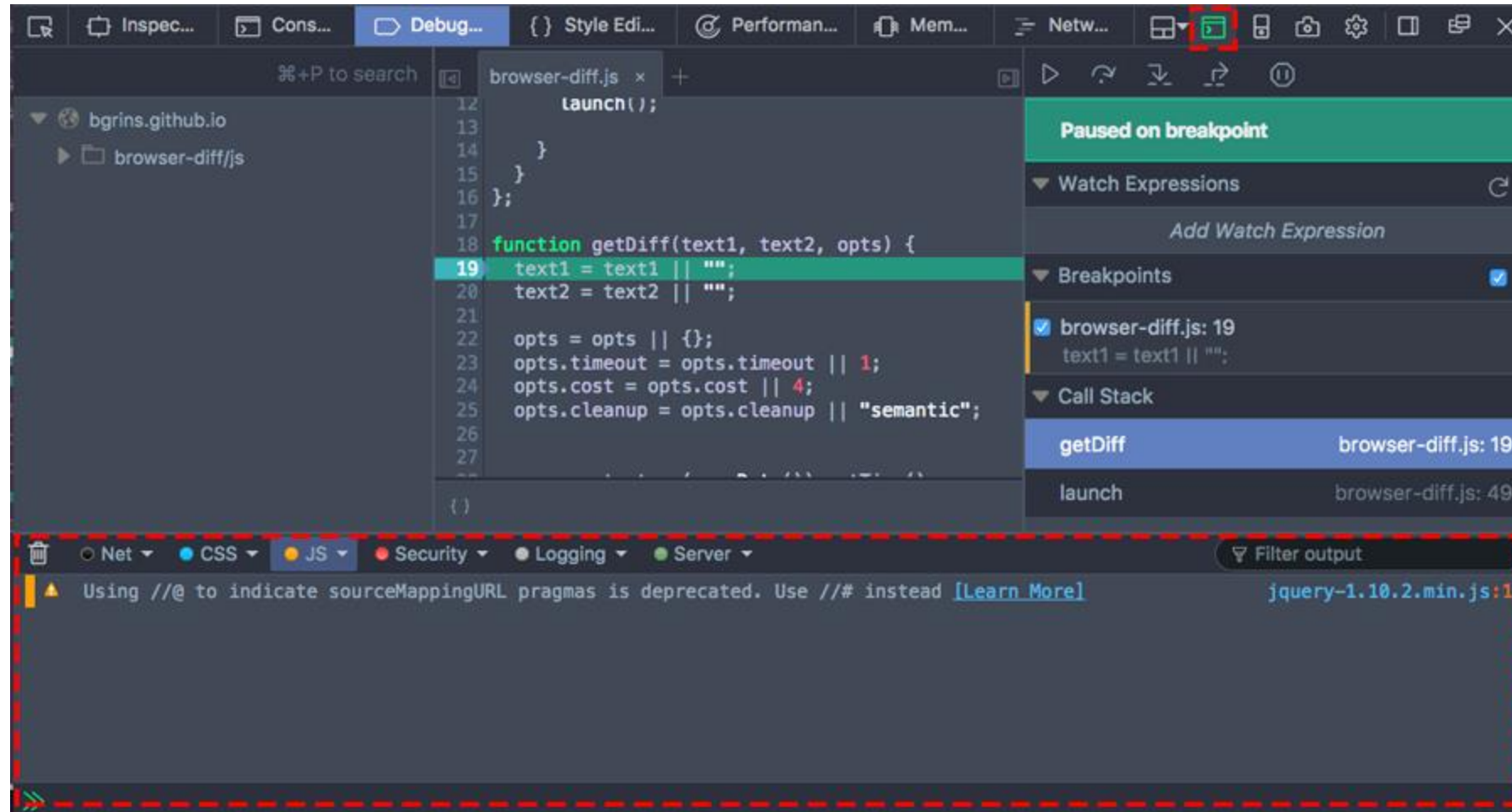
getDiff	browser-diff.js: 40
launch	browser-diff.js: 49
dispatch	jquery-1.10.2.js: 5094
add/v.handle	jquery-1.10.2.js: 4766

Scopes

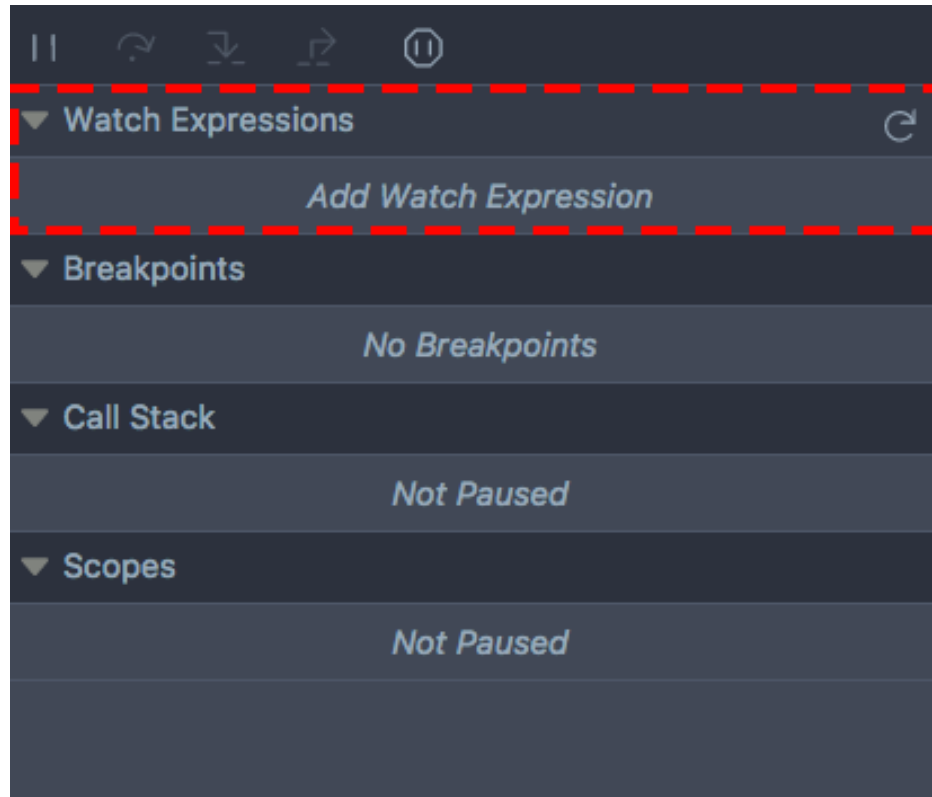
- getDiff
- Window: Global

- Normal :
 - It is associated with a line
 - when the program reaches that line, the debugger pauses.
- Conditional :
 - Has a condition associated with it, which is represented as an expression
 - When the program reaches the line, the debugger pauses only if the expression evaluates to true

Breakpoint List



Watch Expressions



Watch Expressions



browser-diff.js x +

```
1 var dmp = new diff_match_patch();
2 var loadedFiles = 0;
3 var FILEREADER_OPTS = {
4   on: {
5     load: function (e, file) {
6       var el = (loadedFiles % 2) ? "#text2" : "#text1";
7       loadedFiles++;
8       $(el).val(e.target.result);
9     },
10    error: function (e, file) {},
11    groupend: function () {
12      launch();
13    }
14  }
15 };
16 };
17
18 function getDiff(text1, text2, opts) {
19   text1 = text1 || "";
20   text2 = text2 || "";
21
22   opts = opts || {};
23   opts.timeout = opts.timeout || 1;
24   opts.cost = opts.cost || 4;
25
26 }
```

Paused on breakpoint

Watch Expressions

text1: textarea#text1
jQuery11020049510603483288285: 5
__proto__: HTMLTextAreaElementPrototype
Add Watch Expression

Breakpoints

browser-diff.js: 19
text1 = text1 || "";

Call Stack

getDiff browser-diff.js: 19
launch browser-diff.js: 49
dispatch jquery-1.10.2.js: 5094
add/v.handle jquery-1.10.2.js: 4766

Call Stack



Paused while stepping

Watch Expressions

loadedFiles: 0

Add Watch Expression

Breakpoints

Call stack

- getDiff browser-diff.js: 40
- launch browser-diff.js: 49
- dispatch jquery-1.10.2.js: 5094
- add/v.handle jquery-1.10.2.js: 4766

Scopes

- getDiff
- Window: Global

Scopes



The screenshot displays the Chrome DevTools interface with the 'Debugger' tab active. The left sidebar shows the file structure of 'bgrins.github.io', with 'browser-diff.js' selected. The main editor shows the code for 'browser-diff.js', with the 'getDiff' function paused at line 19. The right sidebar shows the 'Paused while stepping' message, 'Watch Expressions' (loadedFiles: 0), 'Breakpoints' (two breakpoints at lines 19 and 22), 'Call Stack' (showing 'getDiff' at line 40), and 'Scopes' (highlighted with a red dashed box, showing 'getDiff' and 'Window: Global').

Pretty-print



- <https://developer.mozilla.org/en-US/docs/Tools>
- <https://hacks.mozilla.org/2016/12/firebug-lives-on-in-firefox-devtools/>



Thank You

Customer Engagement Reimagined

Our Locations: U.S.A | U.K | China | Costa Rica | India | Mauritius | Philippines | Poland | Singapore

