Tales from the Trenches

Battling Browser Bugs for “Fun” and (Non-)Profit

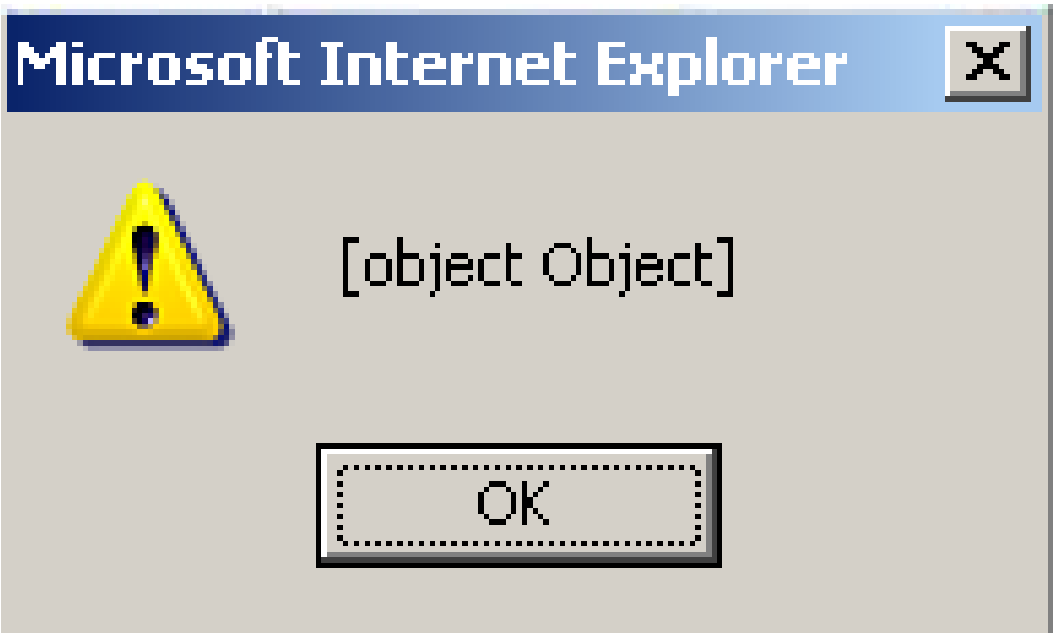
Roan Kattouw

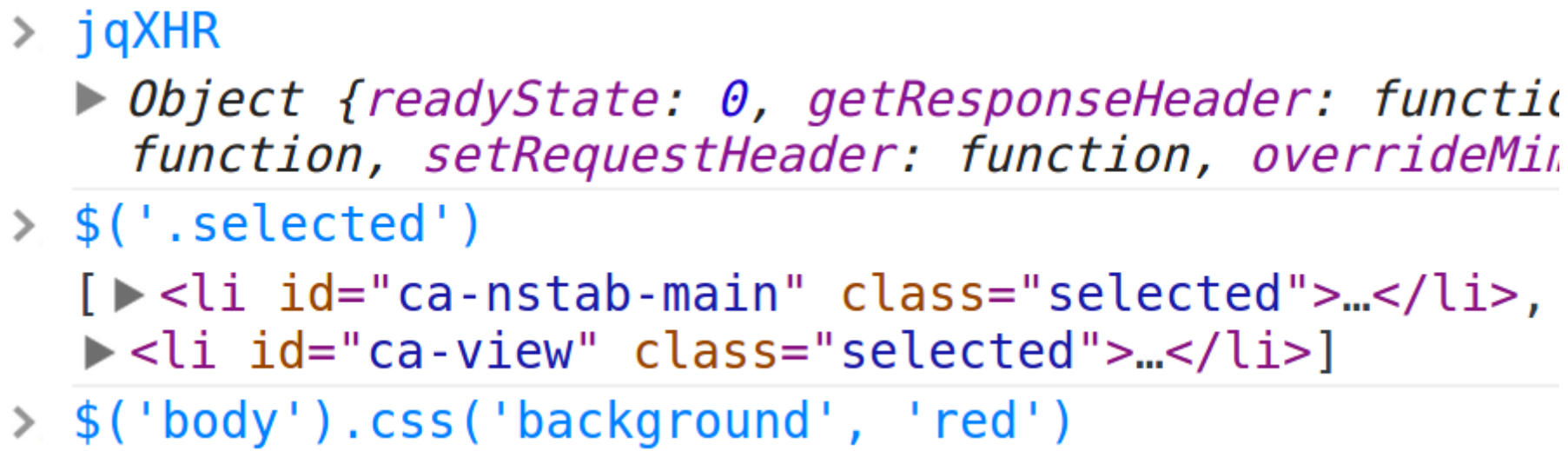
http://tinyurl.com/browserbugsLCA15

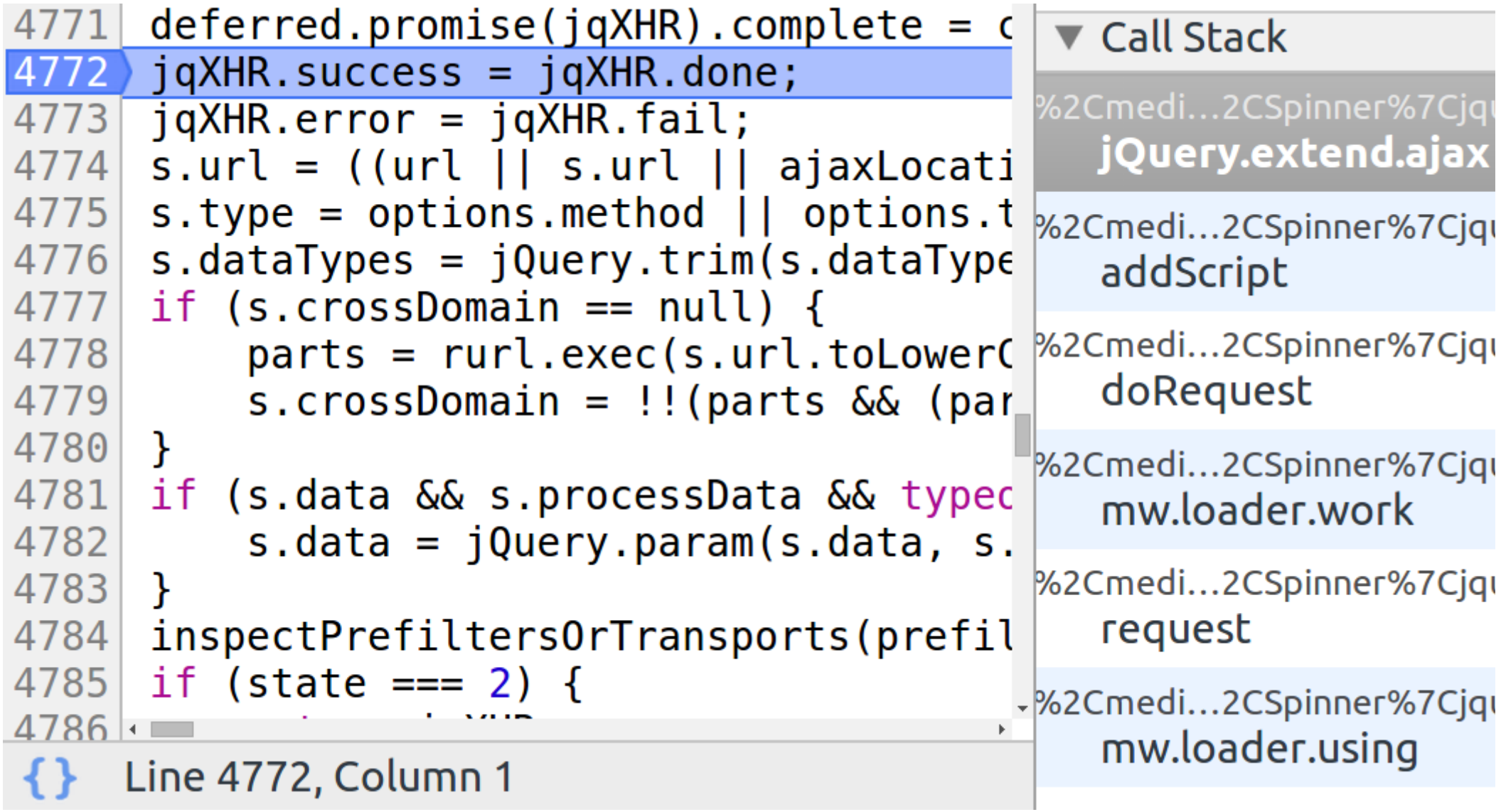
Web development used to be

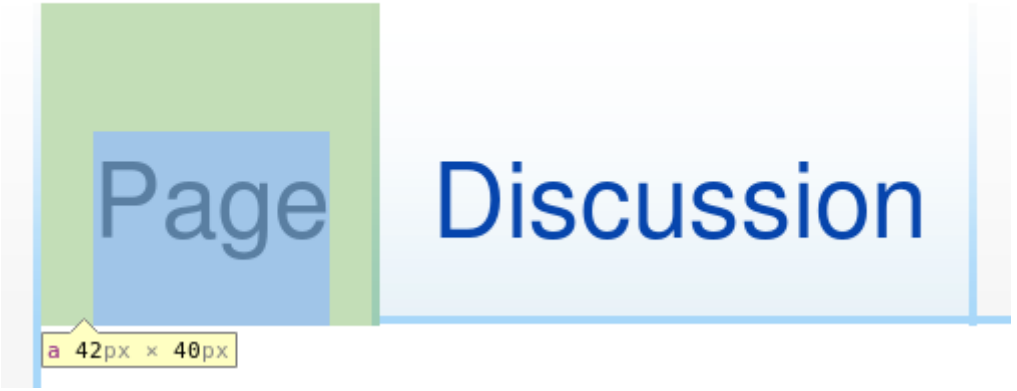
HARD

```
var remove = document.
  getElementsByClassName( 'removeMe' );
for ( var i = 0; i < remove.length; i++ ) {
  remove[i].parentNode.removeChild(
    remove[i]
  );
}



$( '.removeMe' ).remove();
```


Microsoft Internet Explorer
[object Object]
OK


> jqXHR
▶ Object {readyState: 0, getResponseHeader: functi
function, setRequestHeader: function, overrideMi
> $('.selected')
[▶<li id="ca-nstab-main" class="selected">…,
▶<li id="ca-view" class="selected">…]
> $('body').css('background', 'red')


4771 deferred.promise(jqXHR).complete = c
4772 jqXHR.success = jqXHR.done;
4773 jqXHR.error = jqXHR.fail;
4774 s.url = ((url || s.url || ajaxLocati
4775 s.type = options.method || options.t
4776 s.dataTypes = jQuery.trim(s.dataType
4777 if (s.crossDomain == null) {
4778 parts = rurl.exec(s.url.toLowerC
4779 s.crossDomain = !!(parts && (par
4780 }
4781 if (s.data && s.processData && typec
4782 s.data = jQuery.param(s.data, s.
4783 }
4784 inspectPrefiltersOrTransports(prefil
4785 if (state === 2) {
Line 4772, Column 1
Call Stack
%2Cmedi...2CSpinner%7Cjqu
jQuery.extend.ajax
%2Cmedi...2CSpinner%7Cjqu
addScript
%2Cmedi...2CSpinner%7Cjqu
doRequest
%2Cmedi...2CSpinner%7Cjqu
mw.loader.work
%2Cmedi...2CSpinner%7Cjqu
request
%2Cmedi...2CSpinner%7Cjqu
mw.loader.using




```
▼ <ul>
  ▼ <li id="ca-nstab-main" class="selected">
    ▼ <span>
        <a href="/wiki/VisualEditor" title="V
        [alt-shift-c]" accesskey="c">Page</a>
      </span>
    </li>
  ▼ <li id="ca-talk">
```

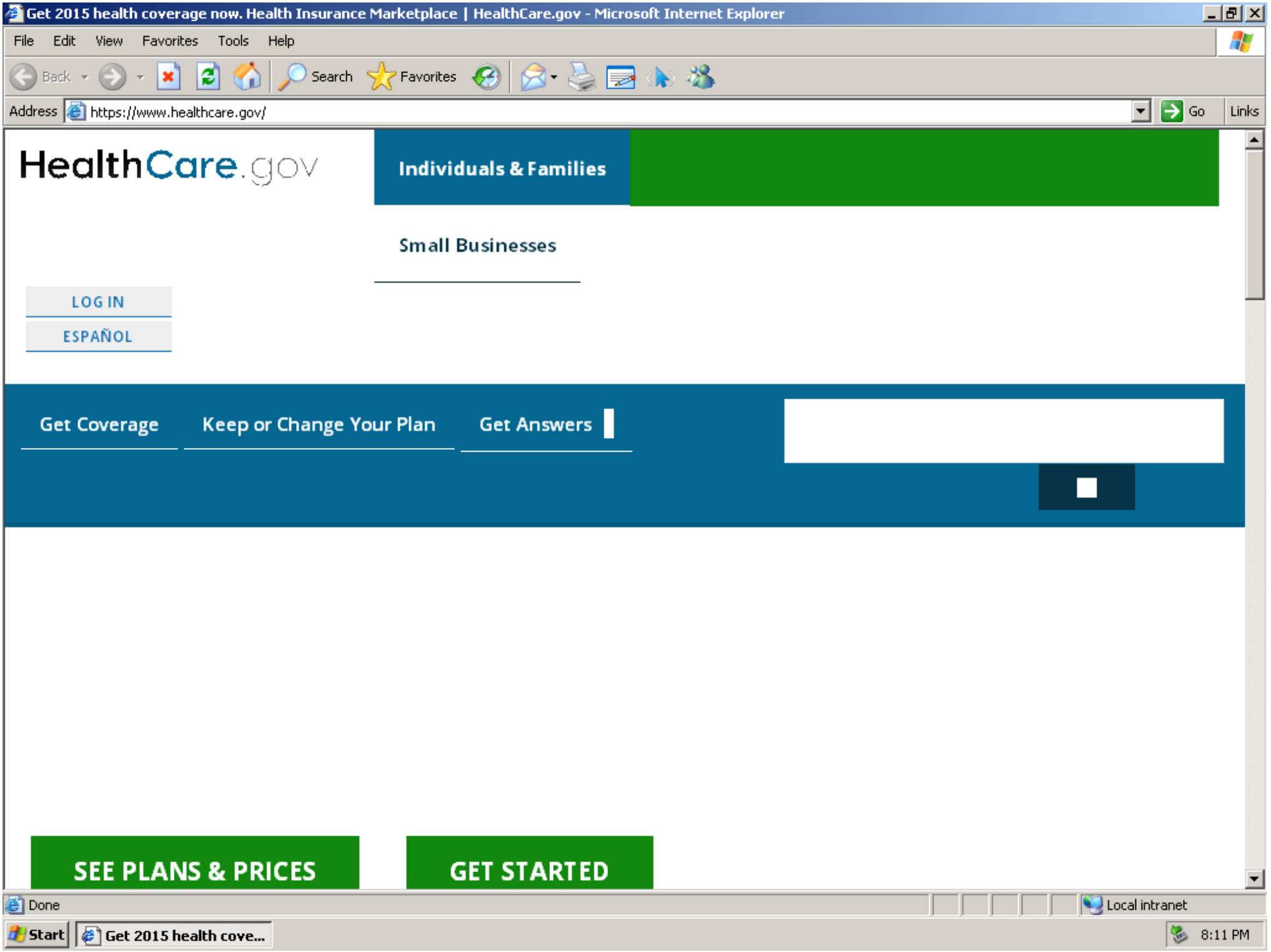
```
<title>W</title>
<title>We</title>
<title>Wel</title>
<title>Welc</title>
<title>Welco</title>
<title>Welcom</title>
<title>Welcome</title>
```

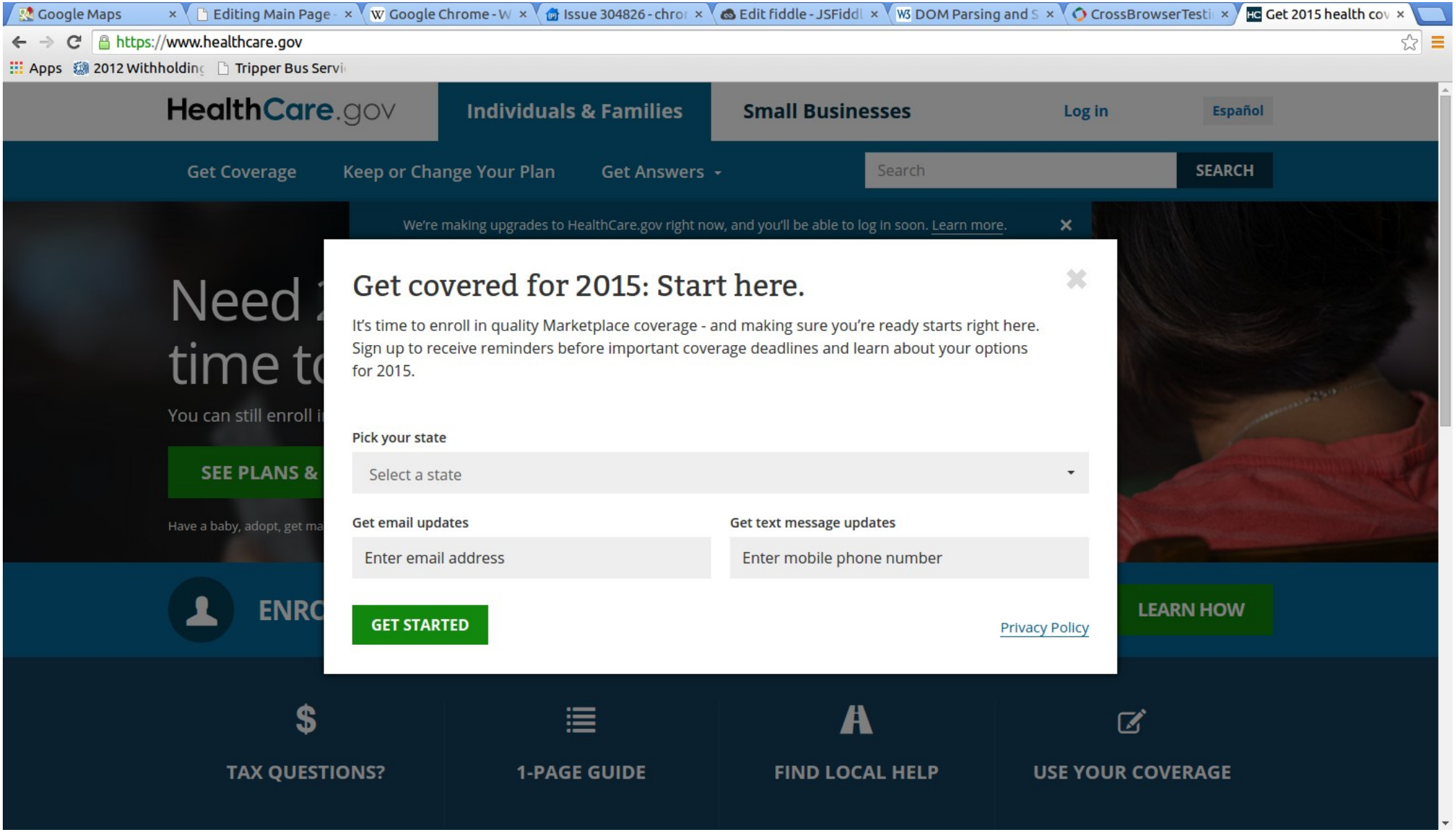
```
$( 'body' ).animate( {
  'scrollTop': 1000
} );
```

```
.button {
  border: 1px #c9c9c9 solid;
  border-radius: 0.3em;
  transition: border-color
    100ms ease-in-out;
}

.button:hover {
  border-color: #aaaaaa;
}
```

The modern web is **EASY**

Browsers used to be
GARBAGE


Get 2015 health coverage now. Health Insurance Marketplace | HealthCare.gov - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address https://www.healthcare.gov/
Go Links
HealthCare.gov
Individuals & Families
Small Businesses
LOG IN
ESPAÑOL
Get Coverage
Keep or Change Your Plan
Get Answers
SEE PLANS & PRICES
GET STARTED
Done
Local intranet
Start
Get 2015 health cove...
8:11 PM


Google Maps
Editing Main Page
Google Chrome - W
Issue 304826 - chror
Edit fiddle - JSFiddl
DOM Parsing and S
CrossBrowserTesti
Get 2015 health cov
https://www.healthcare.gov
Apps
2012 Withholding
Tripper Bus Servi
HealthCare.gov
Individuals & Families
Small Businesses
Log in
Español
Get Coverage
Keep or Change Your Plan
Get Answers
Search
SEARCH
We're making upgrades to HealthCare.gov right now, and you'll be able to log in soon. Learn more.
Need
time to
You can still enroll i
SEE PLANS &
Have a baby, adopt, get ma
ENRO
LEARN HOW
Get covered for 2015: Start here.
It's time to enroll in quality Marketplace coverage - and making sure you're ready starts right here. Sign up to receive reminders before important coverage deadlines and learn about your options for 2015.
Pick your state
Select a state
Get email updates
Enter email address
Get text message updates
Enter mobile phone number
GET STARTED
Privacy Policy
TAX QUESTIONS?
1-PAGE GUIDE
FIND LOCAL HELP
USE YOUR COVERAGE

Modern browsers
JUST WORK

Well...
not **REALLY**

Browsers ~~used to be~~ are

GARBAGE

Modern browsers

~~JUST~~ WORK

kind of

in common cases

Modern browsers

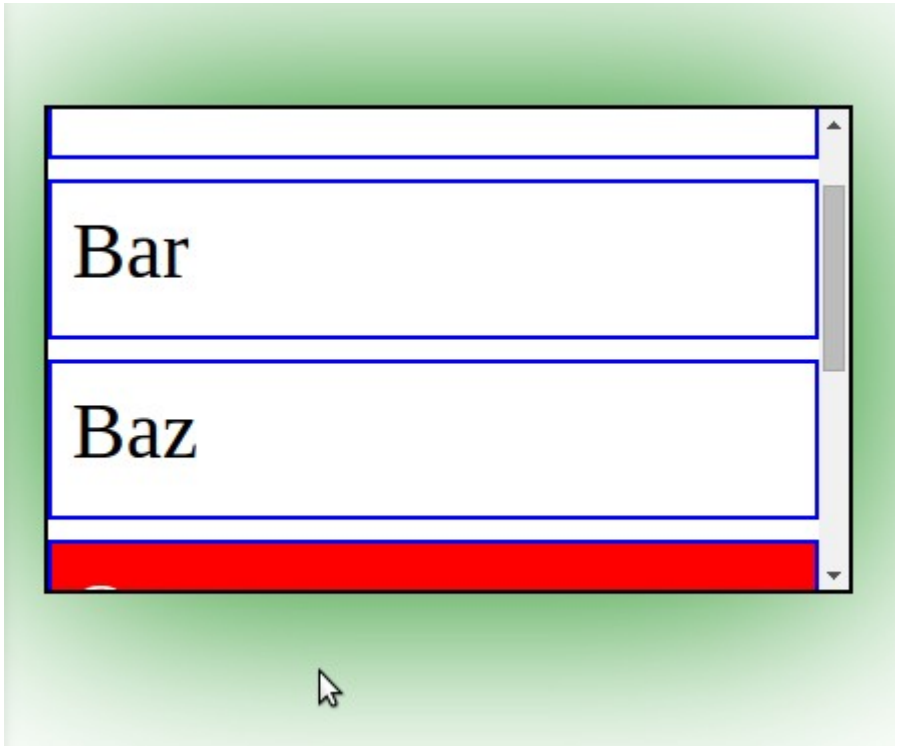
are incredibly complex

Modern browsers

have LOTS of bugs

Let's talk about Chrome

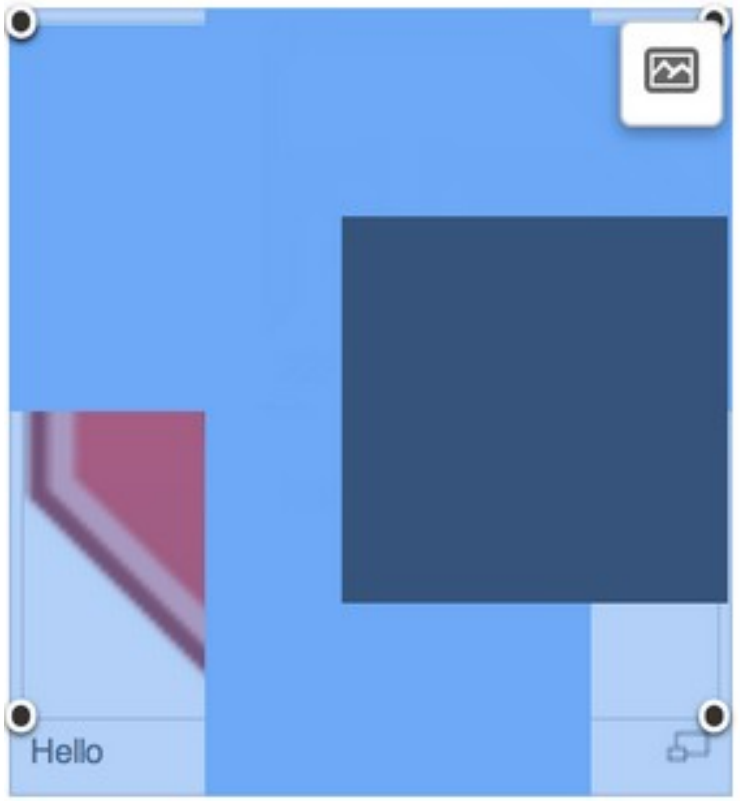
Interactive box-shadow



FIXED now

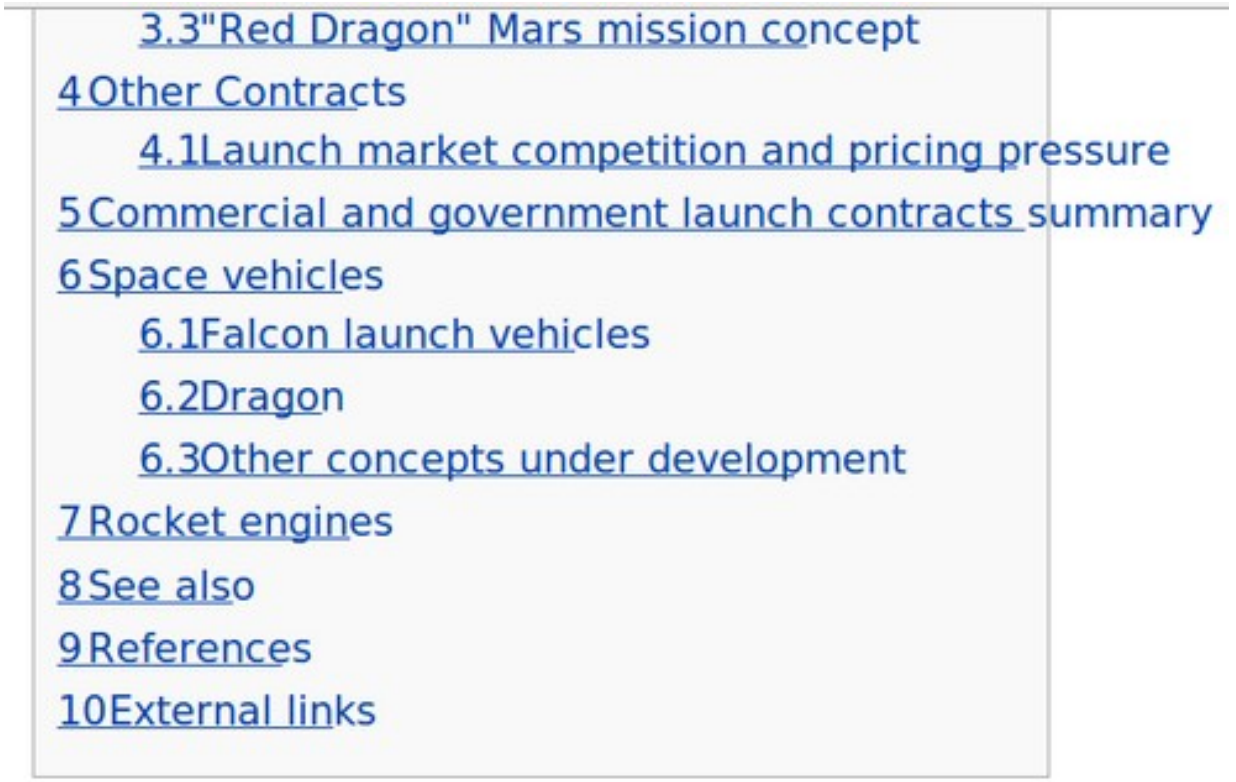
https://code.google.com/p/chromium/issues/detail?id=314291

Image rendering is hard



https://code.google.com/p/chromium/issues/detail?id=308046

Font rendering is harder



FIXED in Chrome 33 (Feb 2014)

https://code.google.com/p/chromium/issues/detail?id=236298

gc.sandboxes.app.economist.com/issues/20131102/uk.11691.a621fa9d/leaders/806431.cristinaandrsquos.come-
Apps Read Later TP-LINK Wireless Ex Daisy Road Super Hu DD-WRT (build 130
Leaders The Economist November 2nd 2013

Argentina's mid-term election

Cristina's come-uppance

President Fernández should build bridges to her opponents—or risk leaving office early

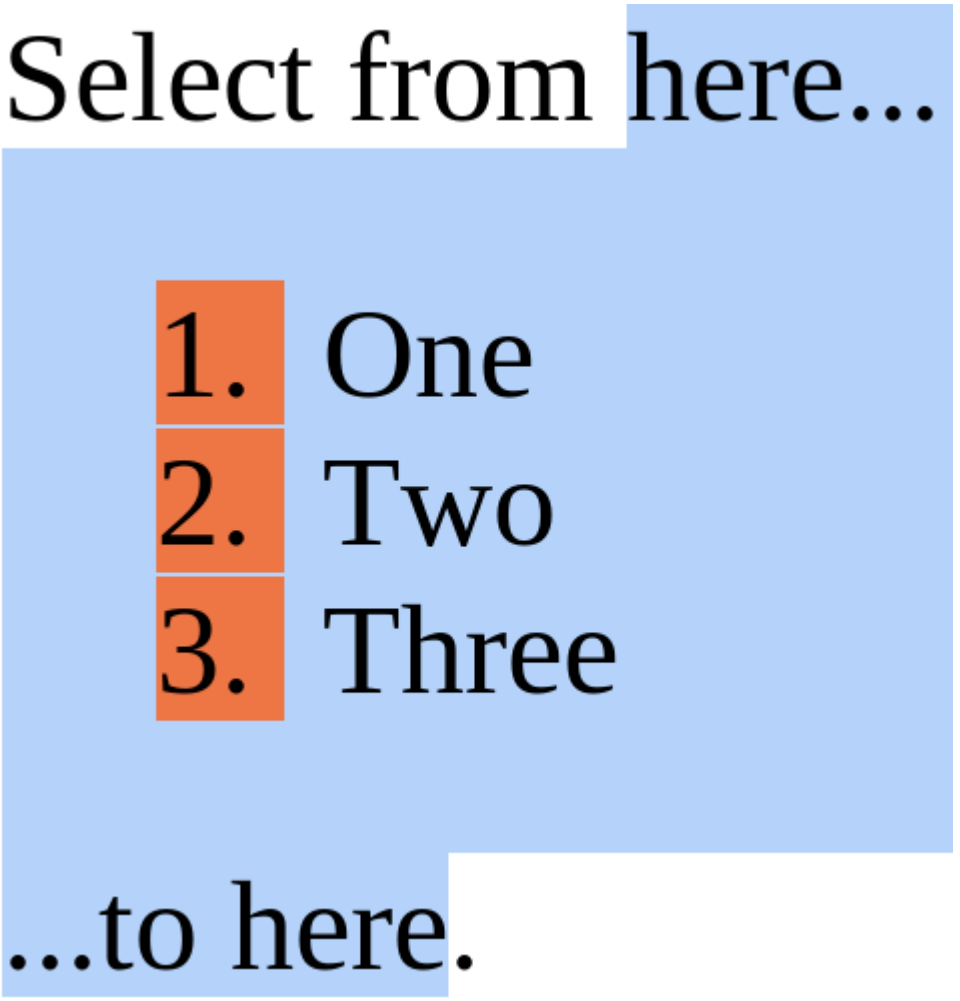
POWER in Argentina is like mercury. It drains swiftly from troubled leaders, flooding towards their most likely successors. That is the prospect facing the president, Cristina Fernández de Kirchner, after a mid-term congressional election on October 27th. Having driven her country's economy close to a precipice, she still has two more years in office. They look as if they will be bumpy ones—even assuming she makes a full recovery from the head injury that prevented her campaigning during the last four weeks.

On paper, Ms Fernández did not fare badly in the election. She retained a narrow majority in Congress. Her opponents are divided three ways. Her group within Argentina's all-embracing Peronist movement remains the country's largest single political force. But it won only 33% of the vote, down from the 54% she secured in winning a second term in 2011. In politically crucial Buenos Aires province, it was trounced by a rival Peronist list led by Sergio Massa, who served as Ms Fernández's cabinet chief before breaking with her.

The election has killed off any lingering hope the president might have had of lifting term limits to allow her to run again in 2015, a measure that would need the backing of two-thirds of Congress. And she has no obvious successor. To a more consensual leader, none of this would matter much. But Ms Fernández and her late husband and predecessor, Néstor Kirchner, have ruled Argentina since 2003 through permanent confrontation—with bondholders, the IMF, political opponents, the media and, lately, the judiciary. Their main weapon was a booming economy. They were fortunate to preside over a surge in the world prices of farm exports from the bountiful Pampas. They shovelled the proceeds into public employment, loss-making state companies and welfare

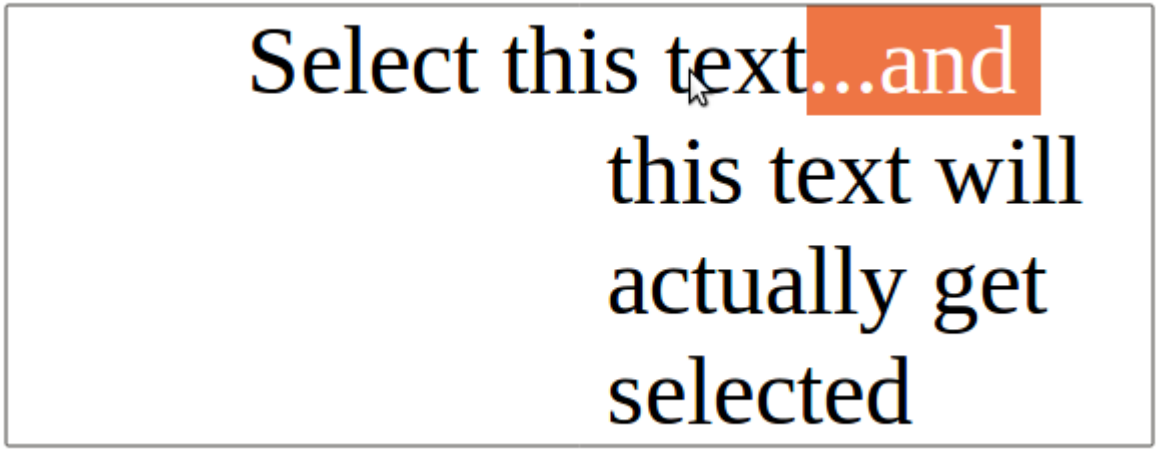
1 of 3
Elements N
** ECON healt
1 if (ECON)
setPoliciesFromSe
{} Line 1, Colu
Console Search
<top fr
The key "tar
Bootstrapper
This event m
This registr
Using perfor
Bootstrapper
Document was
Application
Application

Why respect CSS3 selection colors?



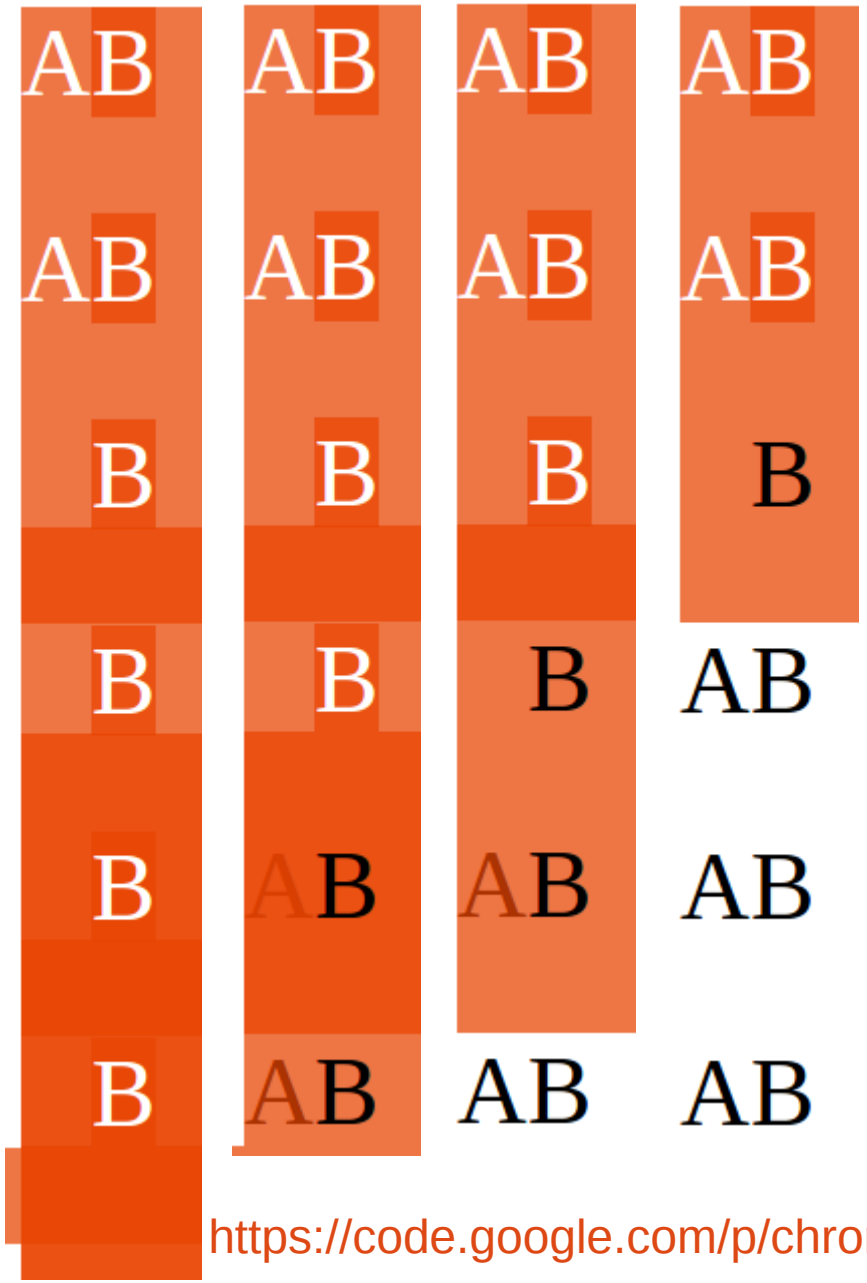
https://code.google.com/p/chromium/issues/detail?id=304826

Because selection is also hard



https://code.google.com/p/chromium/issues/detail?id=269556

...very hard actually



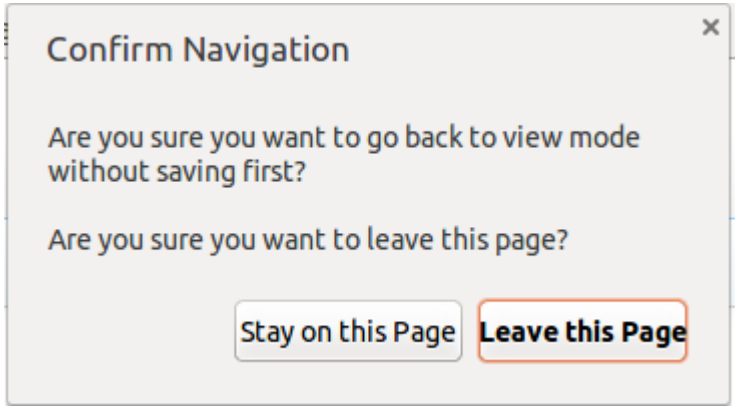
https://code.google.com/p/chromium/issues/detail?id=298175

Dead scrollbars deserve memorials

Workaround: change CSS properties in “right” order

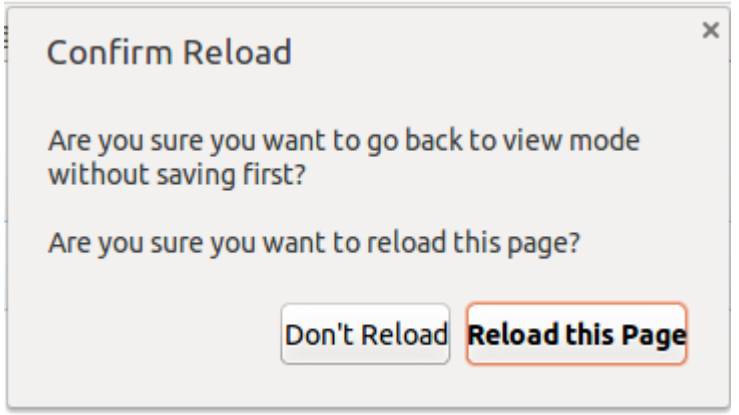
- Remove `height: Npx;`
- Force reflow
- Remove `overflow-y: hidden;`

https://code.google.com/p/chromium/issues/detail?id=387290



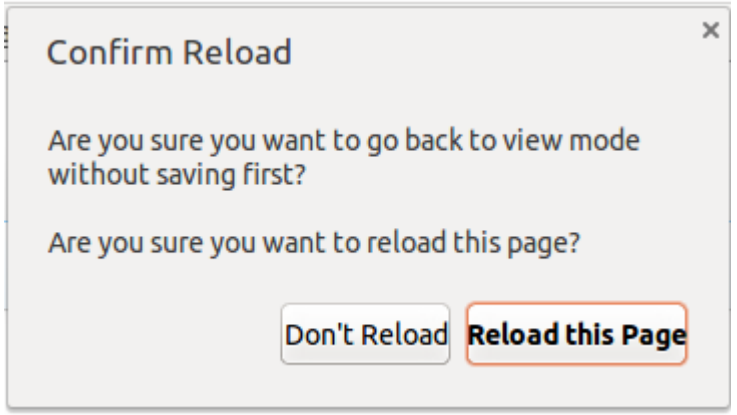
onbeforeunload dialog

Pretty standard
But what if you **reload** instead?



There's a dialog for that!

Now click **Reload this page**
Then **close the tab**



Same dialog
But I tried to **close** the tab...

Reload this Page: closes tab (?)
Don't Reload: continues reload (!!)

https://code.google.com/p/chromium/issues/detail?id=254202

Saga #1
splice()

Inserting items

```
> a = [ 'a', 'b', 'c', 'd' ];
["a", "b", "c", "d"]
> a.splice( 2, 0, 'x', 'y' ); a;
["a", "b", "x", "y", "c", "d"]
```

What if you have an array of items?

```
> a = [ 'a', 'b', 'c', 'd' ];
["a", "b", "c", "d"]
> b = [ 'x', 'y', 'z' ];
["x", "y", "z"]
> a.splice.apply( a, [2, 0].concat( b ) ); a;
["a", "b", "x", "y", "z", "c", "d"]
```

But what if you have a **LOT**?

```
> a = [ 'a', 'b', 'c', 'd' ];
["a", "b", "c", "d"]
> b = new Array( 500 );
▶ Array[500]
> a.splice.apply( a, [2, 0].concat( b ) ); a;
▶ Array[504]
```

How far can you go?

- What happens if you pass too many arguments?
- How many is “too many”?

ECMAScript spec

Says nothing

Chrome

131072 ($=2^{17}$)

RangeError: Maximum call stack size exceeded

Firefox

524288 ($=2^{19}$)

RangeError: arguments array passed to Function.prototype.apply is too large

Opera 12 and below

$2097152 (=2^{21})$

Error: Function.prototype.apply: argArray is too large

Safari

65536 ($=2^{16}$)

RangeError: Maximum call stack size exceeded

IE10+

$262144 (=2^{18})$

Error: Out of stack space

IE9 and below

No apparent limit

Got up to 33554432 ($=2^{25}$)

Error: Out of memory (!!)

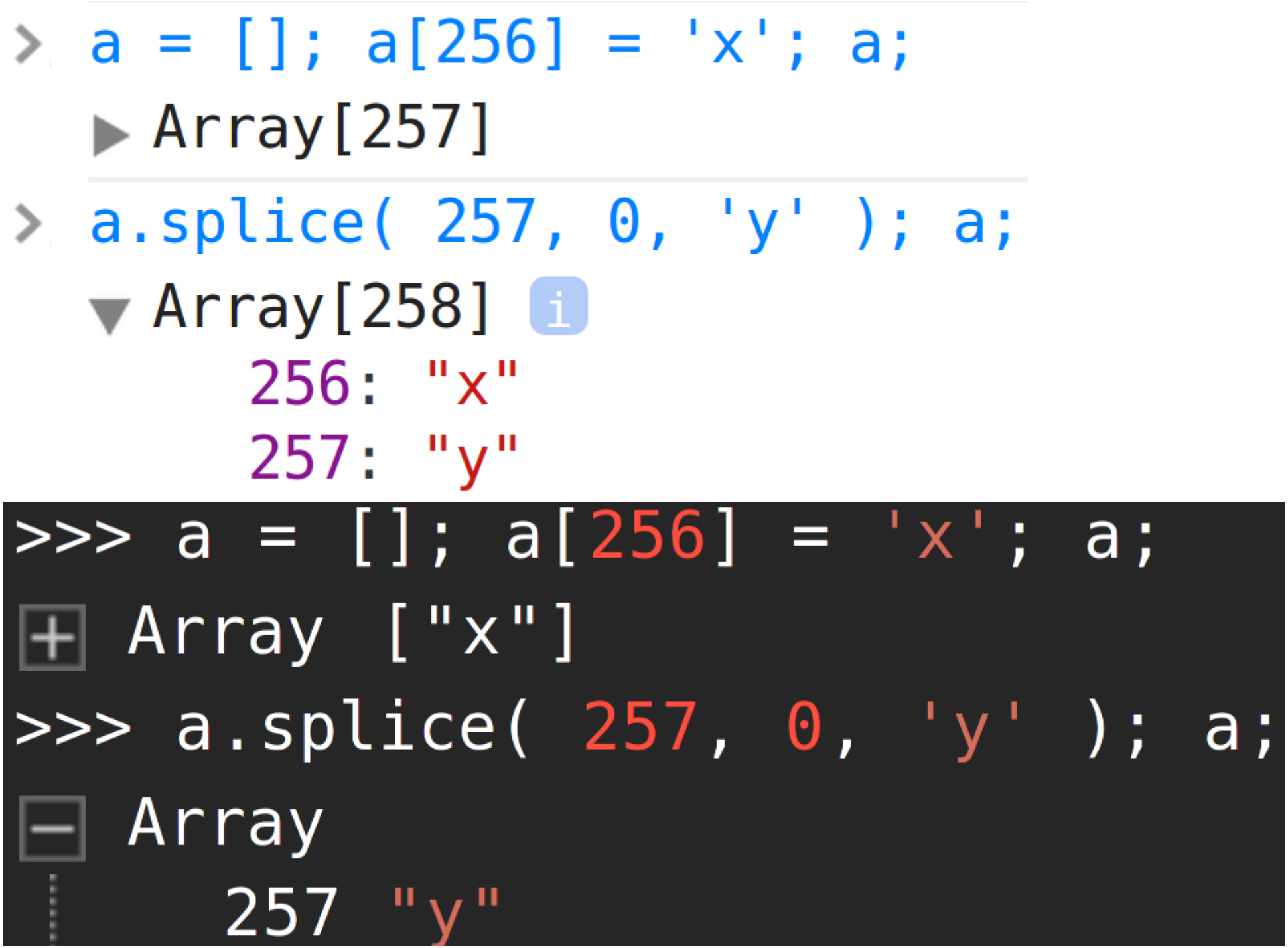
Solution: splice.apply() in batches of 1024

Note assumption:
no crash === correctness

Basic operations would never be implemented incorrectly, right?

```
> a = []; a[6] = 'x'; a;
  [undefined × 6, "x"]
> a.splice( 7, 0, 'y' ); a;
  [undefined × 6, "x", "y"]
```

```
>>> a = []; a[6] = 'x'; a;
⊞ Array ["x"]
>>> a.splice( 7, 0, 'y' ); a;
⊟ Array
    6 "x"
    7 "y"
```


> a = []; a[256] = 'x'; a;
▶ Array[257]
> a.splice(257, 0, 'y'); a;
▼ Array[258] i
256: "x"
257: "y"
>>> a = []; a[256] = 'x'; a;
Array ["x"]
>>> a.splice(257, 0, 'y'); a;
Array
257 "y"

```
function isMyBrowserAPieceOfGarbage() {
  var n = 256, a = [];
  a[n] = 'a';
  a.splice( n + 1, 0, 'b' );
  return a[n] !== 'a';
}
```

Let's talk about Firefox

Underline? Overline!

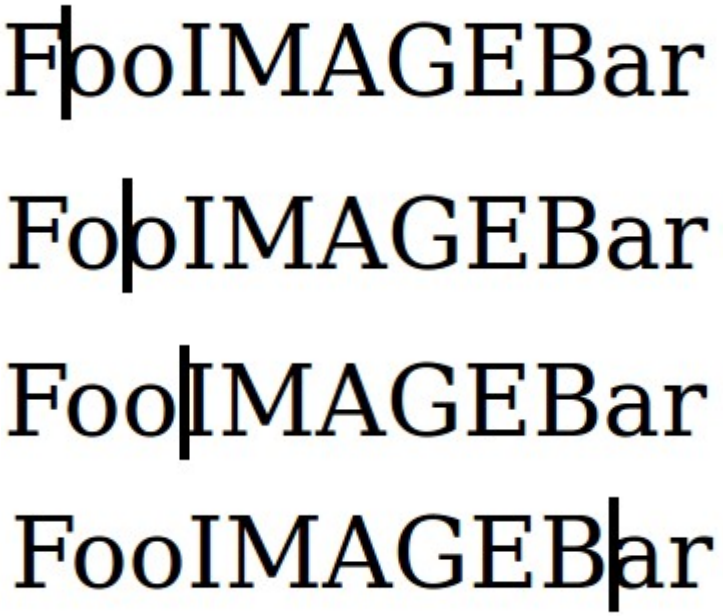
FIXED in Firefox 33

https://bugzilla.mozilla.org/show_bug.cgi?id=727125

Unfollowable but active

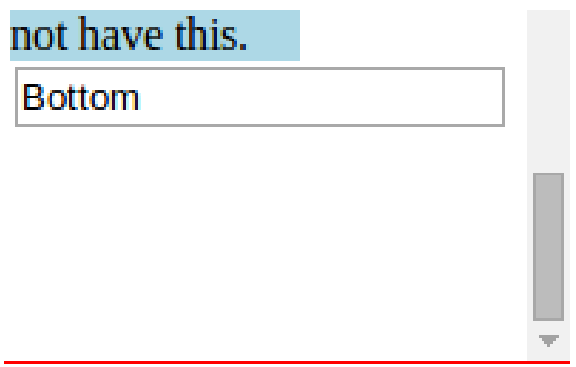
https://bugzilla.mozilla.org/show_bug.cgi?id=924087

Cursoring over “invisible” things



https://bugzilla.mozilla.org/show_bug.cgi?id=989012

Who needs bottom padding?



RESOLVED INVALID

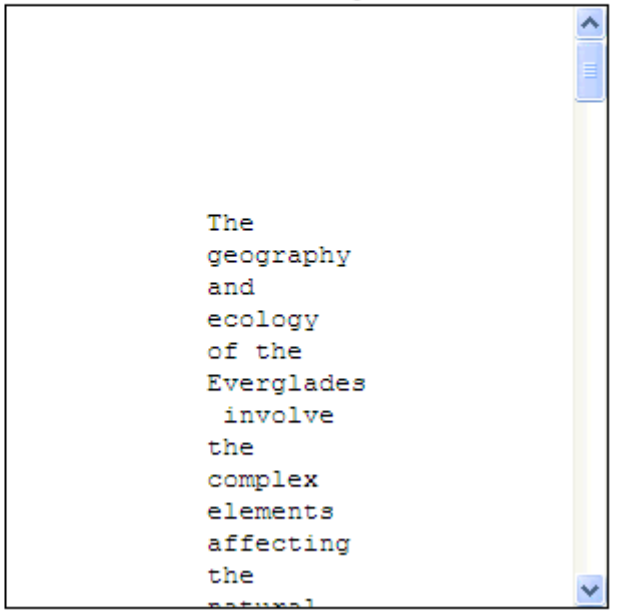
https://bugzilla.mozilla.org/show_bug.cgi?id=748518

Textareas, is who!

```
<textarea style="width: 100px; height: 100px; padding: 100px; border: 1px solid black;">
```

Firefox

Internet Explorer



FIXED in FF 30 (2014-06-10)

Filed 2002-07-16 (!)

https://bugzilla.mozilla.org/show_bug.cgi?id=157846

Saga #2
HTML parsing

Input: HTML string
Desired output: DOM

This is easy for **fragments**

```
> html = '<p>Hello</p><pre>World</pre>';
"<p>Hello</p><pre>World</pre>"
> $.parseHTML( html );
[<p>Hello</p>, <pre>World</pre>]
> $('<div>').html( html );
[▼<div>                    ]
      <p>Hello</p>
      <pre>World</pre>
    </div>
```

But I have a **document**

```
"<!DOCTYPE html>
<html prefix="dc: http://purl.org/d
http://mediawiki.org/rdf/"
about="http://www.mediawiki.org/wik
1044293"><head prefix="mwr:
http://www.mediawiki.org/wiki/Speci
property="mw:articleNamespace" cont
rel="dc:replaces" resource="mwr:rev
property="dc:modified" content="201
<meta about="mwr:user/35927" proper
```

iframe hack

```
function createDocument( html ) {
    var newDocument, iframe;
    iframe = document.createElement( 'iframe' );
    document.body.appendChild( iframe );
    newDocument = iframe.contentDocument;
    newDocument.open();
    // Party like it's 1995!
    newDocument.write( html );
    newDocument.close();
    iframe.parentNode.removeChild( iframe );
    return newDocument;
}
```

.contentWindow.document for IE8

breaks in IE8 after yield

createHTMLDocument()

```
function createDocument( html ) {
   var newDocument = document.implementation.
      createHTMLDocument( '' );
                Why not pass html here?
   // Regex the doctype and html tags out *barf*
   html = html.replace(
      /^\s*(?:<!doctype[^>]*>)?\s*<html[^>]*>/i,
      ''
   );
   html = html.replace( /<\/html>\s*$/i, '' );

   newDocument.documentElement.innerHTML = html;
   return newDocument;
}
```

The **`DOMImplementation.createHTMLDocument()`** method creates a new HTML `Document`.

Syntax

```
document.implementation.createHTMLDocument(title);
```

Parameters

title Optional

Is a `DOMString` containing the title to give the new HTML document.

“We're not doing anything weird, we're doing exactly what the W3C says!”

DOMParser

```
function createDocument( html ) {
   var parser = new DOMParser();
   return parser.parseFromString(
      html, 'text/html'
   );
}
```

DOMParser HTML support

- Firefox 12 *(April 2012)*
- IE 10 *(October 2012)*
- Chrome 30 *(October 2013)*
- Opera 17 *(October 2013)*
- Safari 7.1 *(September 2014)*

** It is currently unclear what the URL of the returned document should be.

Results for a test case:

	Gecko	Opera	Chrome
document.location	null		
document.URL	unsupported	unsupported	""
document.documentURI	Page URL	null	null

Anne van Kesteren suggests using the default, about:blank.

In any case, the returned *document*'s *content type* must be the *type* argument. Additionally, the *document* must have a *URL* value equal to the URL of the *active document*, a *location* value of `null`.

- Firefox: behaves to spec
- Chrome: URL is null
- IE: URL is undefined, location errors

Let's talk about IE

Text background in RTL

Background: Foo

Background and dir=rtl:

Background and dir=ltr: Foo

Background and rtl from CSS:

Background and ltr from CSS: Foo

http://jsfiddle.net/v939T/3/

Measuring superscript position


```
sup, sub { display: inline-block; }
```

http://jsfiddle.net/k3QV2/8/

Style attribute normalization

setAttribute('style', ...)	getAttribute('style')
color: red;	color: red;
color: #ffddff;	color: rgb(255, 221, 255);
font-family: look-no-semicolon	font-family: look-no-semicolon;
font-family:look-no-space	font-family: look-no-space;
invalid-css;	empty string
empty string	null

Workaround:

- Parse as XML
- Copy `style` to `data-unmangled-style`
- Serialize XML DOM back to string
- Parse string as HTML
- Read/write `data-unmangled-style` instead of `style`

http://jsfiddle.net/zwcrn6dv/1/

DOM serialization

```
doc.documentElement
.outerHTML
```

<pre> parsing is weird

```
> div.innerHTML = '<pre>\n\nFoo</pre>';
div.childNodes[0].childNodes[0];
"
Foo"
```

Note *In the HTML syntax, a leading newline character immediately following the pre element start tag is stripped.*

<pre> serialization is broken

```
> div.innerHTML = '<pre>\n\nFoo</pre>';
  div.childNodes[0].childNodes[0];
  "
  Foo"
> div.innerHTML
  "<pre>
  Foo</pre>"
>
```

Very broken

```
> div.innerHTML = div.innerHTML;
"<pre>

Foo</pre>"
> div.innerHTML = div.innerHTML;
"<pre>
Foo</pre>"
> div.innerHTML = div.innerHTML;
"<pre>Foo</pre>"
```

```
function isMyBrowserAPieceOfGarbage() {
  var div =
    document.createElement( 'div' );
  div.innerHTML = '<pre>\n\n</pre>';
  return div.innerHTML ===
    '<pre>\n</pre>';
}
```

Broken in all browsers except...
Opera 12 and below

Workaround:
add newlines to DOM
in the right places

```
function fixupNewlines( $element ) {
   $element.find( 'pre, textarea, listing' )
      .each( function () {
         var child = this.firstChild;
         if (
            child.nodeType === Node.TEXT_NODE &&
            child.data.charAt( 0 ) === '\n'
         ) {
            child.insertData( 0, '\n' );
         }
      } );
}
```

```
function fixupNewlines( $element ) {
   $element.find( 'pre, textarea, listing' )
      .each( function () {
         var matches,
            child = this.firstChild;
         if ( child.nodeType === Node.TEXT_NODE ) {
            matches = child.data.match(
               /^(\r\n|\r|\n)/
            );
            if ( matches && matches[1] ) {
               child.insertData( 0, matches[1] );
            }
         }
      } );
}
```

So browsers suck?

Browsers are still pretty awesome

http://krinkle.github.io/dom-ce/

Thank you!

http://tinyurl.com/browserbugsLCA15