

# OPTIMISING WHERE IT HURTS

**Jake Archibald**

# WHO AM I?

*Where are we? What year is this?*

# WHO AM I?

- Jake Archibald
- Web Developer at the BBC
- Developer on Glow - the BBC's JavaScript library

# WHO AM I?

- Jake Archibald
- Web Developer
- Developer on **Angular**



# PERFORMANCE

*Does it actually matter?*

YES

# PERFORMANCE & JS

- Expectations have changed

# PERFORMANCE & JS

→ Exp



# PERFORMANCE & JS

- Expectations have changed
- DOM Level 1 - first draft 1997

# PERFORMANCE & JS

No. I'm not interested in Red Dwarf or child pornography so the internet has nothing for me.

Stewart Lee (1995)



# PERFORMANCE & JS

- JavaScript is still single-threaded
- Human beings are multi-threaded
- No-one likes waiting
- As developers, we know performance matters because as users, we know performance matters

# THE FUTURE...

...is NOW! (sort of)



optimis

lurts BBC

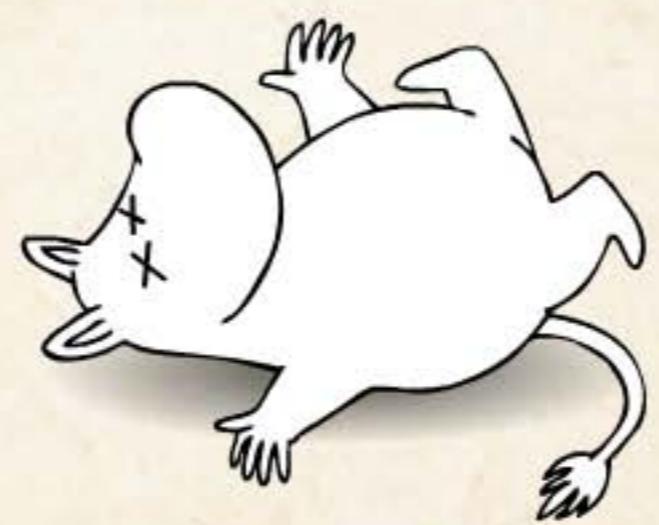


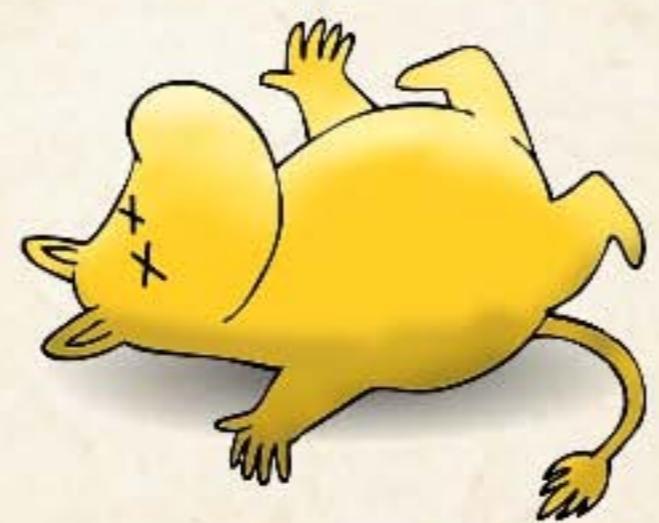
optimus  
lurts BBC



Jake Archibald - Optimising Where it Hurts BBC









Jake Archibald - Optimising Where it Hurts BBC

# JAVASCRIPT

!=

# DOM

# ABSTRACTIONS

*on abstractions on abstractions*

# ABSTRACTIONS

JavaScript Library

JavaScript

JavaScript Engine

The DOM

Browser

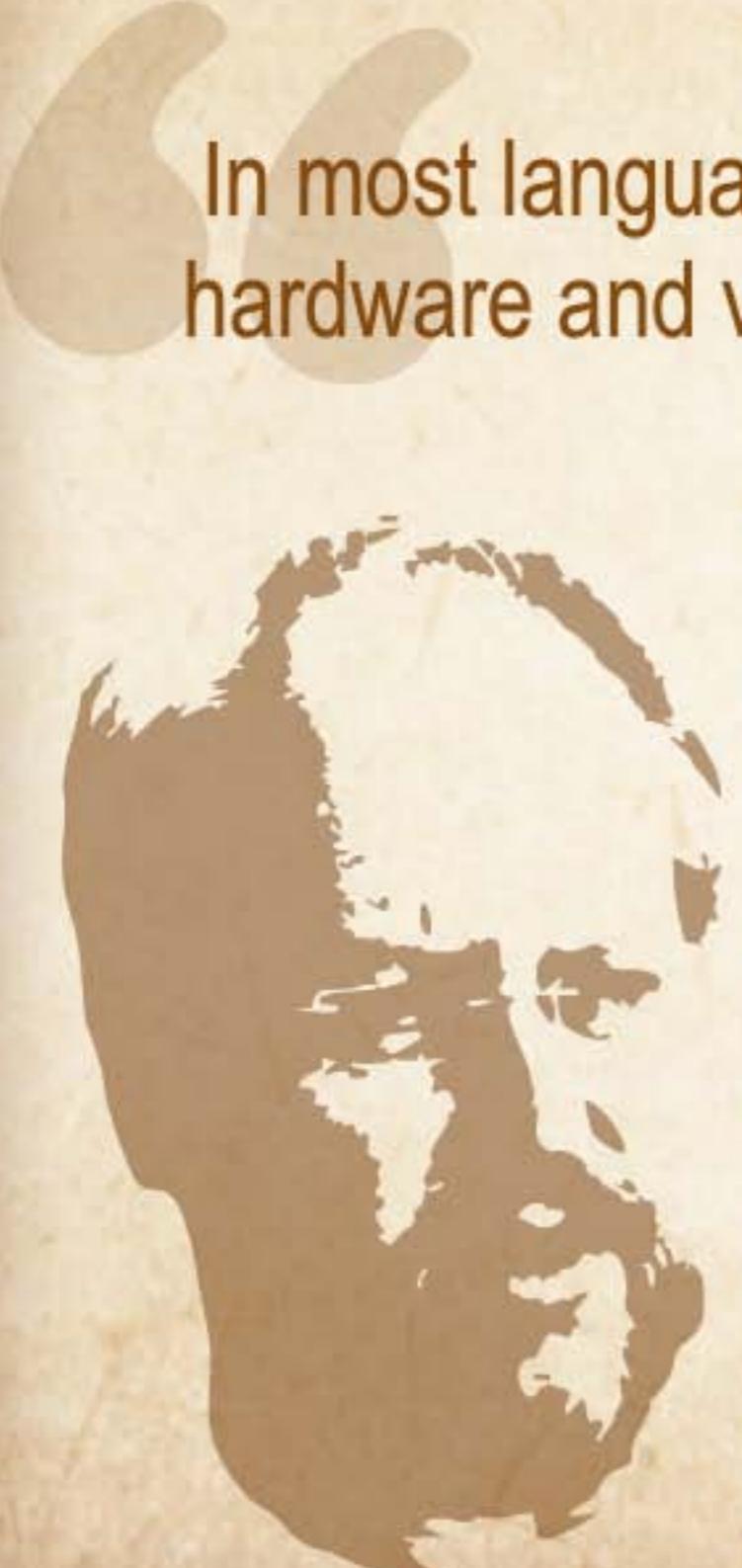
Operating System

Processor

# BITWISE

*Operations*

# BITWISE OPERATIONS



In most languages, [bitwise] operators are very close to the hardware and very fast. In JavaScript, they are very far from the hardware and very slow.

Douglas Crockford

# HEX COLOUR PARSING

```
var colour = parseHexColour('#76400E');

// colour is...
{
  r: 118,
  g: 64,
  b: 14
}
```

# HEX COLOUR PARSING

```
"#76400E".slice(1,3)
```

```
"76"
```

# HEX COLOUR PARSING

"#76400E".slice(3,5)

"76"

"40"

# HEX COLOUR PARSING

"#76400E".slice(5, 7)

"76"

"40"

"0E"

# HEX COLOUR PARSING

```
parseInt("76", 16)  
parseInt("40", 16)  
parseInt("0E", 16)
```

# HEX COLOUR PARSING

```
function parseHexColour(hexColour) {  
    return {  
        r: parseInt(hexColour.slice(1, 3), 16),  
        g: parseInt(hexColour.slice(3, 5), 16),  
        b: parseInt(hexColour.slice(5, 7), 16)  
    }  
}
```

# HEX COLOUR PARSING

```
function parseHexColour(hexColour) {  
    var matches = hexColour.match(/\w\w/g);  
  
    return {  
        r: parseInt(matches[0], 16),  
        g: parseInt(matches[1], 16),  
        b: parseInt(matches[2], 16)  
    };  
}
```

# HEX COLOUR PARSING

```
"#76400E".slice(1)
```

```
parseInt("76400E", 16)
```

```
0x76400E
```

# HEX COLOUR PARSING

**0x76400E >> 16**

# HEX COLOUR PARSING

>> 16

0111 0110 0100 0000 0000 1110  
0x76400E

# HEX COLOUR PARSING

0x**76**~~400E~~ >> 16

0x**76**

# HEX COLOUR PARSING

**0x7640~~0E~~** >> 8

**0x7640** & **0xFF**

**0x76**

**0x40**

# HEX COLOUR PARSING

**0x76400E & 0xFF**

**0x76**

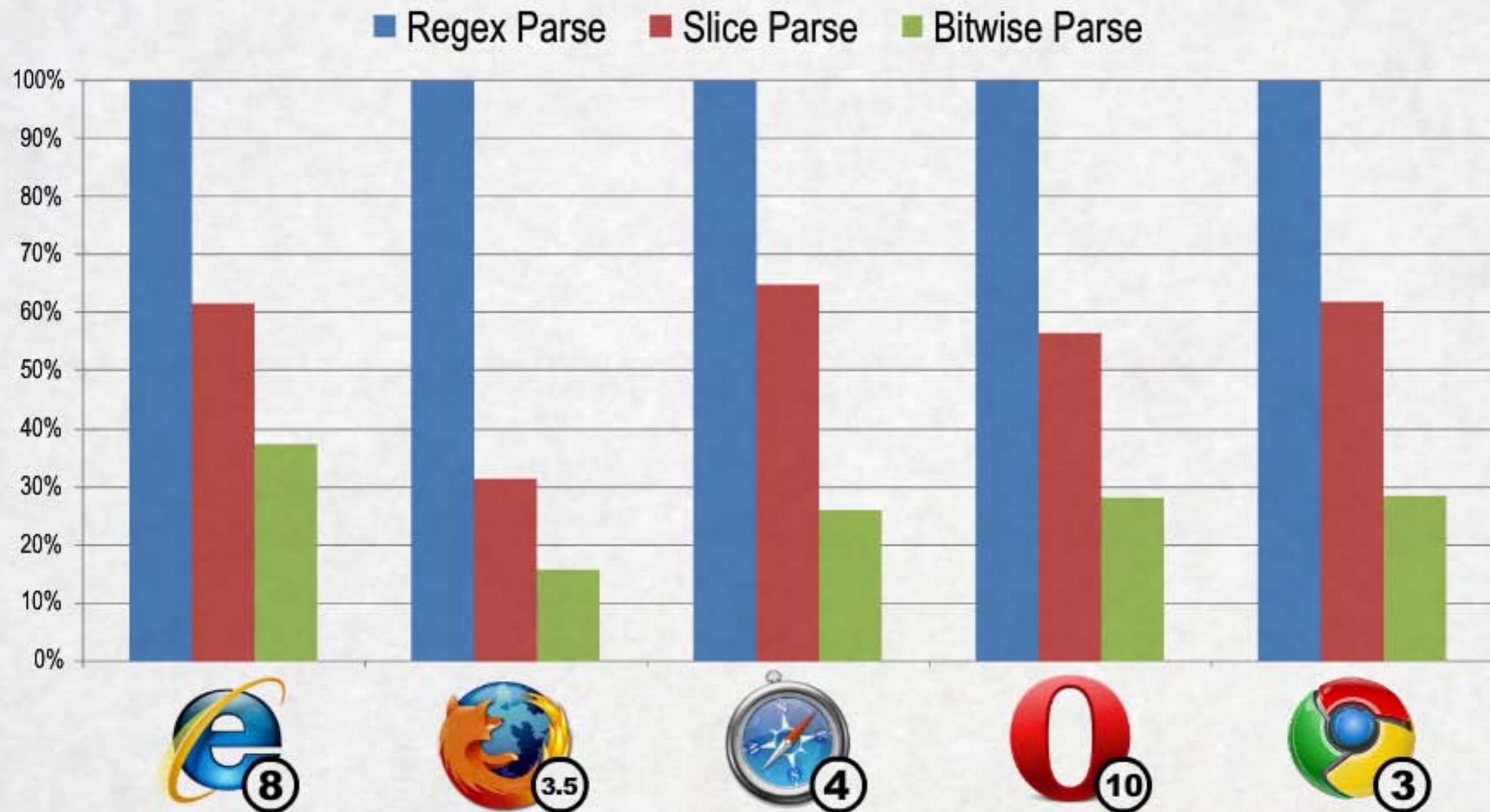
**0x40**

**0x0E**

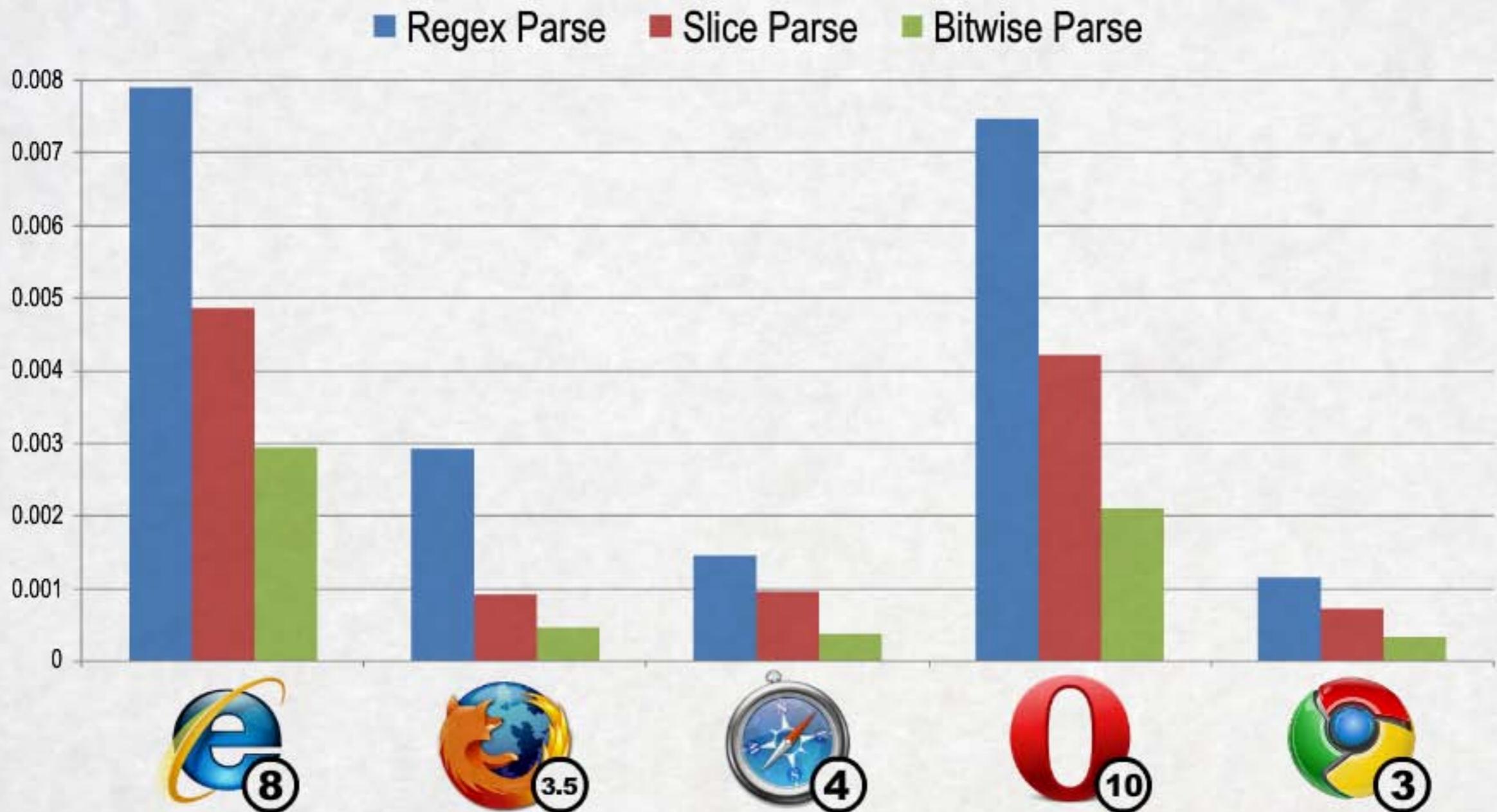
# HEX COLOUR PARSING

```
function parseHexColour(hexColour) {  
    var hex = parseInt(hexColour.slice(1), 16);  
  
    return {  
        r: hex >> 16,  
        g: hex >> 8 & 0xff,  
        b: hex & 0xff  
    };  
}
```

# HEX COLOUR PARSING



# HEX COLOUR PARSING



# OPTIMISE...

*...the right stuff*

# WASTING ENERGY

If the levels of wastage continue, an extra 43m tonnes of carbon dioxide will be pumped into the atmosphere... 65% of UK consumers leave chargers on once a week

British Broadcasting Corporation







# WASTING ENERGY



All the energy saved in switching off your charger for one day  
is used up in one second of car-driving.

Prof. David MacKay

`typeof NaN == 'number'`

# SCOPING

*Does deep scoping impact performance?*

# THE SCOPE CHAIN

```
var msg = 'Hello!';

function addListenerTo(elm) {
    var attentionGrabber = 'You there!';
    elm.addEventListener('click', function() {
        alert(attentionGrabber + ' ' + msg);
    }, false);
}
```

# THE SCOPE CHAIN

```
var msg = 'Hello!';

function addListenerTo(elm) {
    var attentionGrabber = 'You there!';
    elm.addEventListener('click', function() {
        alert(attentionGrabber + ' ' + msg);
    }, false);
}
```

# THE SCOPE CHAIN

```
var msg = 'Hello!';

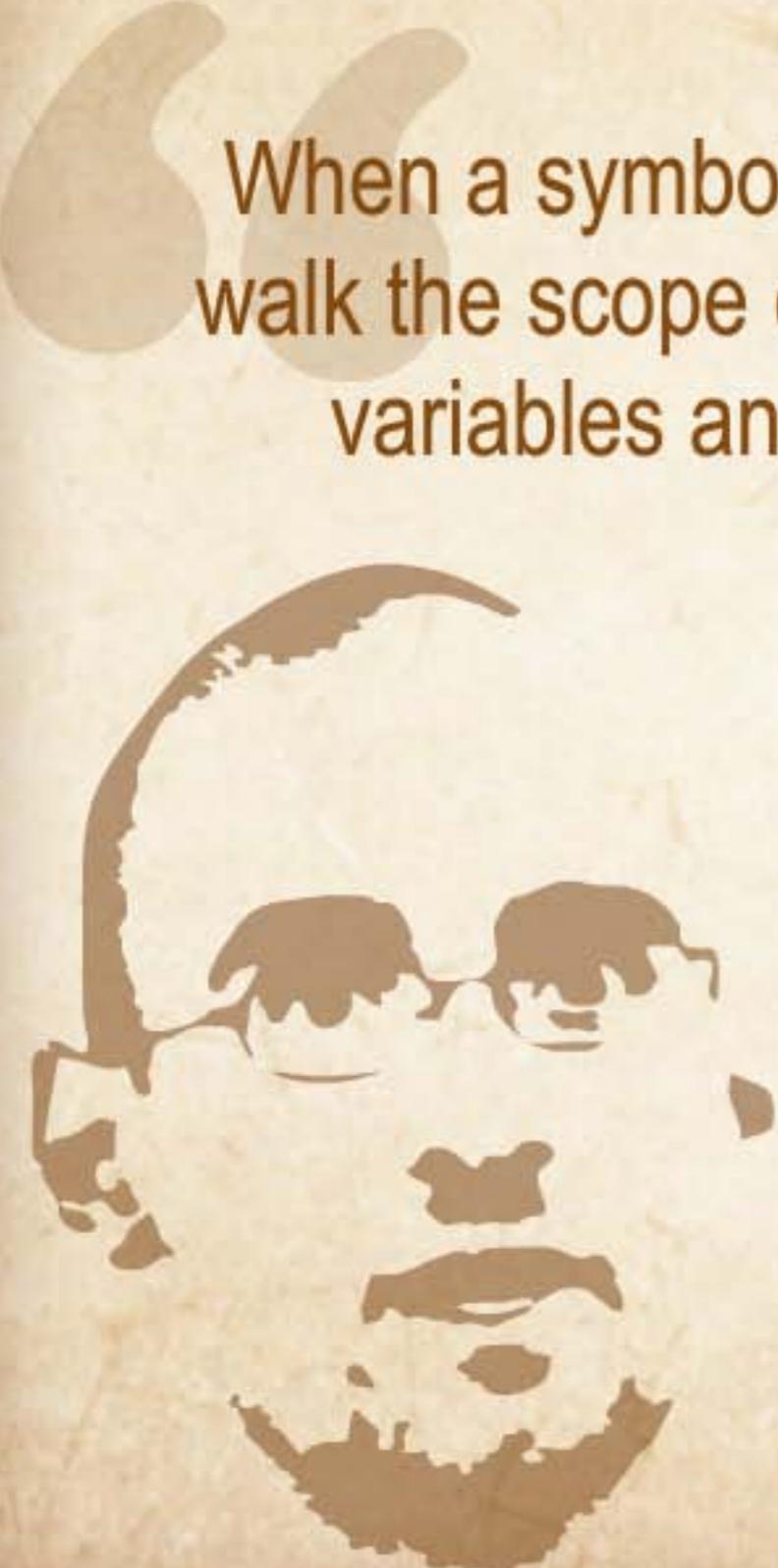
function addListenerTo(elm) {
    var attentionGrabber = 'You there!';
    elm.addEventListener('click', function() {
        alert(attentionGrabber + ' ' + msg);
    }, false);
}
```

# THE SCOPE CHAIN

```
var msg = 'Hello!';

function addListenerTo(elm) {
    var attentionGrabber = 'You there!';
    elm.addEventListener('click', function() {
        alert(attentionGrabber + ' ' + msg);
    }, false);
}
```

# THE SCOPE CHAIN



When a symbol is accessed, the JavaScript engine has to walk the scope chain to find that symbol... Using more local variables and fewer global variables results in better performance.

Nicholas C. Zakas

# BENCHMARKING SCOPING

Frontend engineer working on the YUI library, talking about More accessible user interfaces with ARIA.

Todd's talk.

More about Todd

## Jake Archibald - 4.10pm

Jake worked on the BBC's Glow JavaScript library, talking about:  
Optimising where it hurts.

More about Jake

### Optimising where it hurts

As the amount of JavaScript we use on our pages increases, as does its effect on page performance. Even when browsers manage to produce the same output, how they reach that output can be very different. Some optimisations aren't obvious, others can be easily broken by seemingly unrelated statements. So, how do we get the most out of our code?

In this session we'll challenge assumptions about performance, using practical code to demonstrate the optimisations that really make a difference, and why they make a difference.



Where

Duke Of York's Picturehouse,  
Preston Circus, Brighton, BN1 4

Venue phone number: +44 (0)

When

9am - 6pm 20th Nov

Tags

Tag your photo  
[#fullfrontal](#)

Contact

Confe

All

Jake's talk

Willison - 4.55pm

Acknowledged developer, talking about Web API

# BENCHMARKING SCOPING

```
// vars to read
var a='a',b='b',c='c',d='d',e='e',f='f',g='g',h='h',i='i',j='j',
    k='k',l='l',m='m',n='n',o='o',p='p',q='q',r='r',s='s',t='t',
    u='u',v='v',w='w',x='x',y='y',z='z';
```

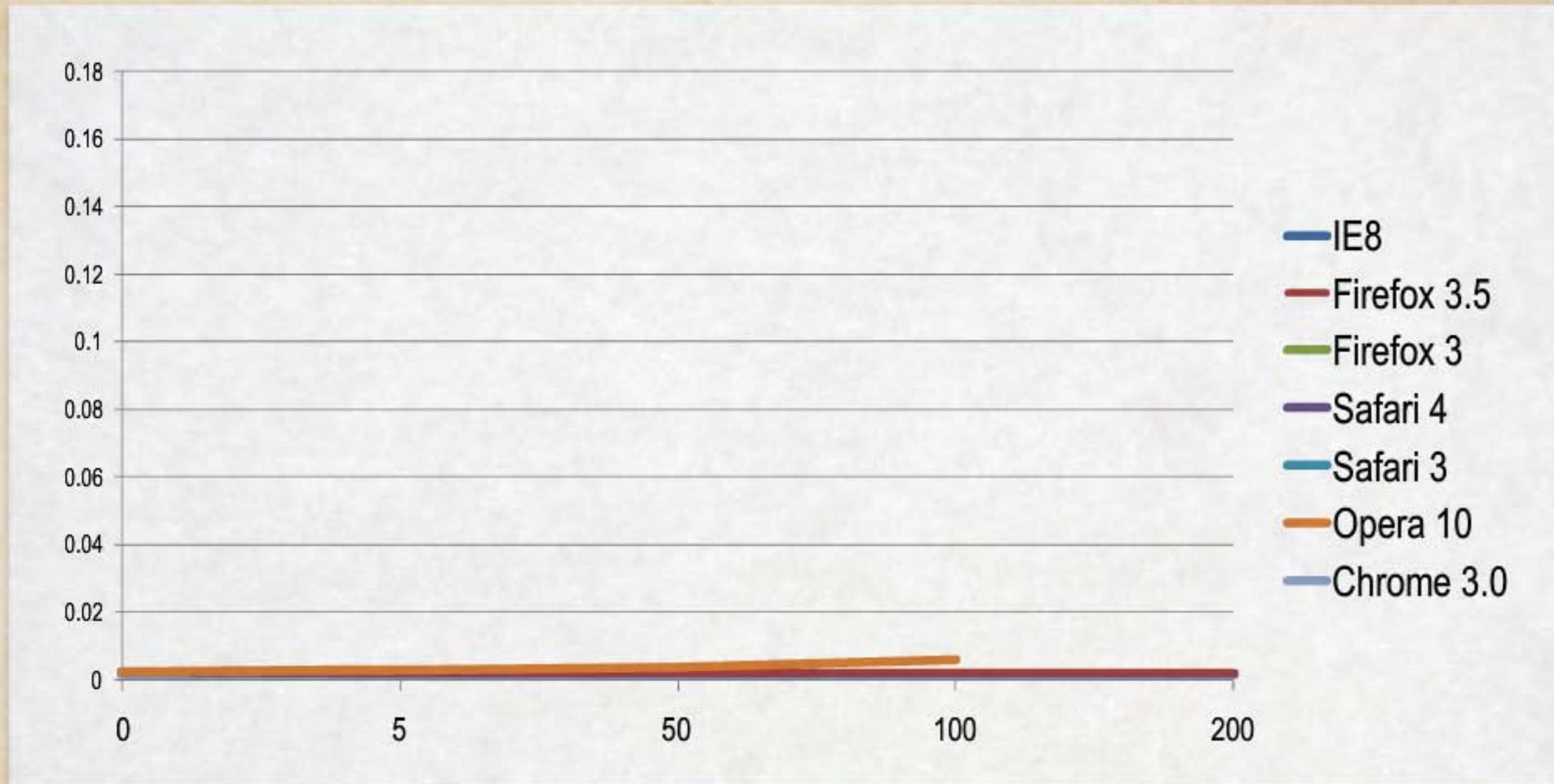
# BENCHMARKING SCOPING

```
(function() { (function() { (function() { (function() { (function() {  
  (function() { (function() { (function() { (function() { (function() {  
    (function() { (function() { (function() { (function() { (function() {  
      (function() { (function() { (function() { (function() { (function() {  
        (function() { (function() { (function() { (function() { (function() {  
          // and so on  
  
scope200 = function() {  
  result = a+b+c+d+e+f+g+h+i+j+k+l+m+n+o+p+q+r+s+t+u+v+w+x+y+z;  
}  
})
```

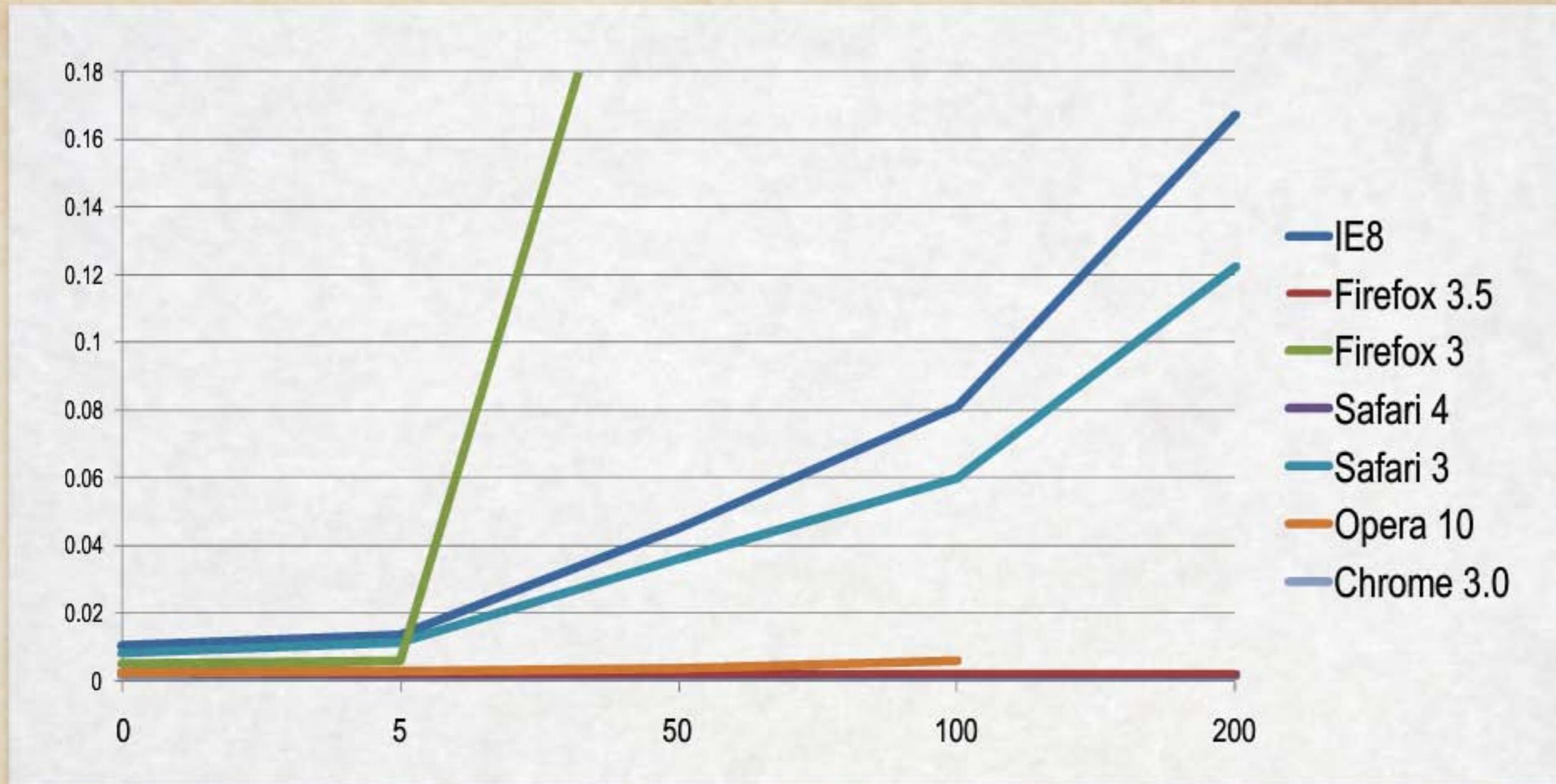
# BENCHMARKING SCOPING

```
// close 100 scopes
}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()}()})}
```

# SCOPE RESULTS



# SCOPE RESULTS



# VARIABLE DECLARATION

```
var msg = 'spatchcock';

function doStuff() {
    alert(msg);

    if (false) {
        var msg = 'spotted dick';
    }
}

doStuff();
```

# VARIABLE DECLARATION

```
var msg = 'spatchcock';
```

```
function doStuff() {
```

```
    alert('The page at http://www.jakearchibald.co.uk says:')
```

```
    if (
```



```
        undefined
```

```
}
```

```
    }
```

```
}
```

```
doStuff();
```

# VARIABLE DECLARATION

```
var msg = 'spatchcock';

function doStuff() {
    var msg;
    alert(msg);

    if (false) {
        msg = 'spotted dick';
    }
}

doStuff();
```



eval()

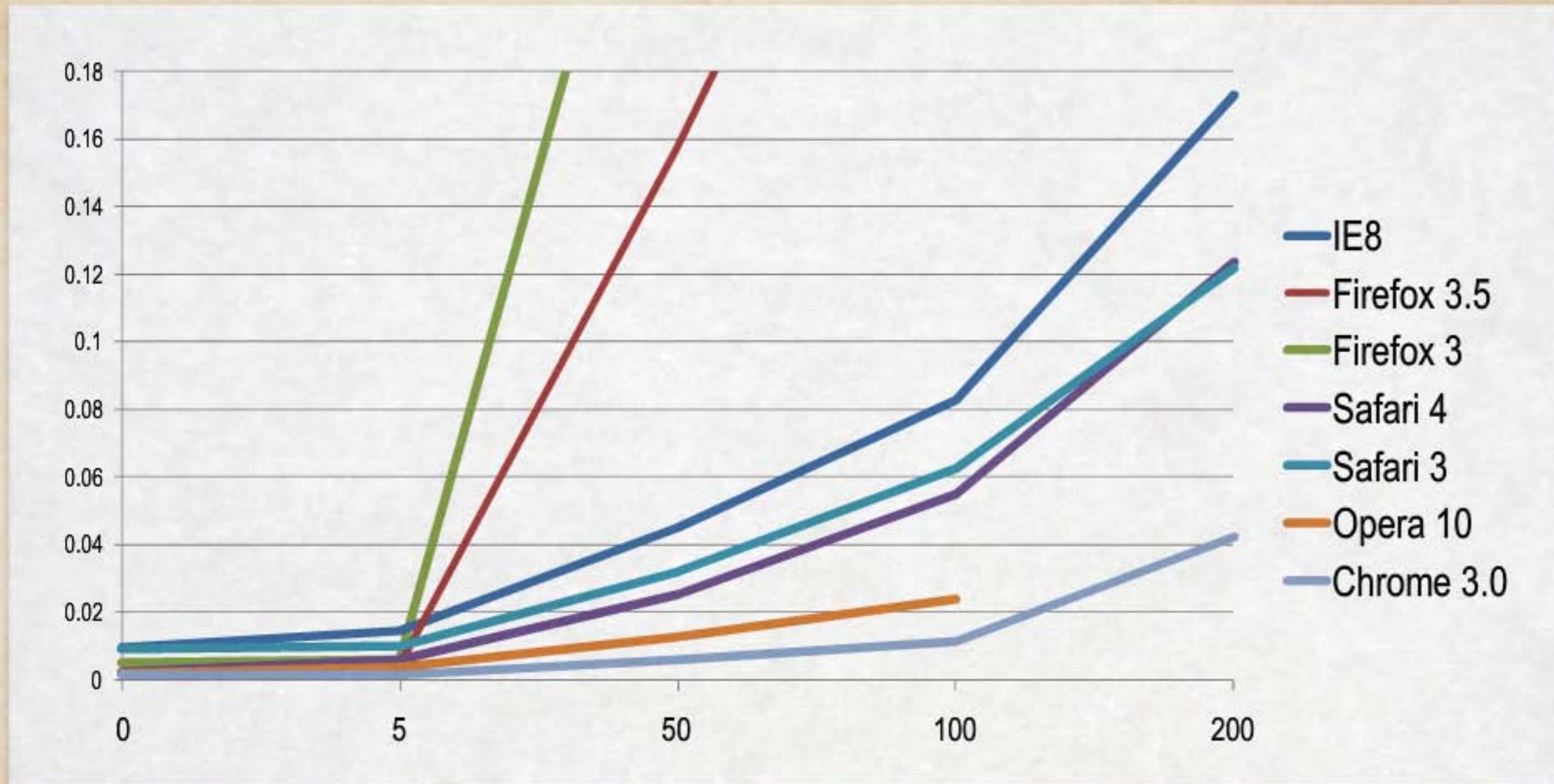
# BENCHMARKING SCOPING

```
(function() { (function() { (function() { (function() { (function() {  
  (function() { (function() { (function() { (function() { (function() {  
    (function() { (function() { (function() { (function() { (function() {  
      (function() { (function() { (function() { (function() { (function() {  
        (function() { (function() { (function() { (function() { (function() {  
          // and so on
```

# BENCHMARKING SCOPING

```
(function() {eval(''); (function() {eval(''); (function() {eval('');  
  (function() {eval(''); (function() {eval(''); (function() {eval('');  
    (function() {eval(''); (function() {eval(''); (function() {eval('');  
      (function() {eval(''); (function() {eval(''); (function() {eval('');  
        // and so on
```

# SCOPE RESULTS WITH EVAL



# SCOPING...

*...and memory*

# LEAKY LISTENER

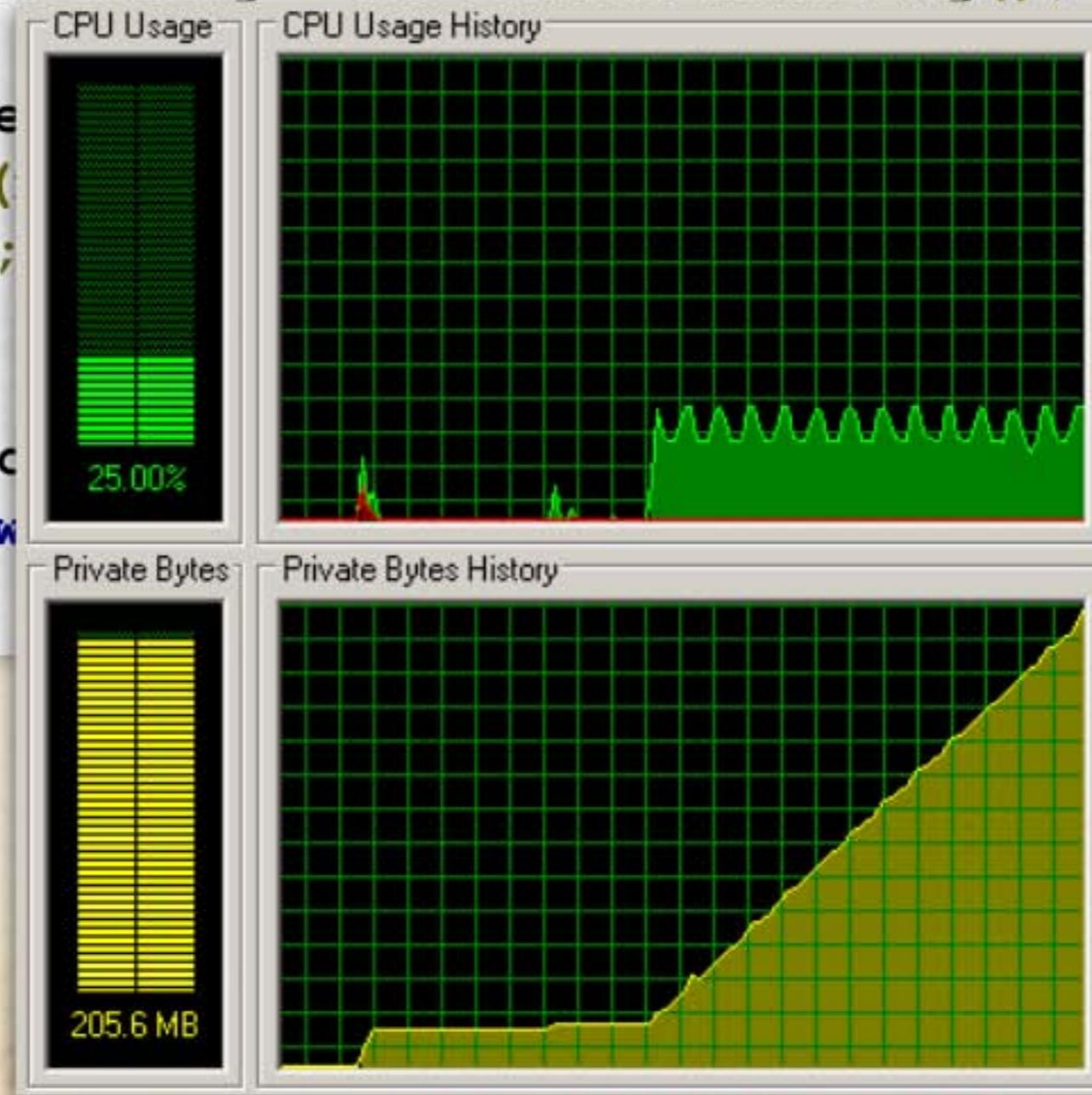
```
function addClickListener(elm) {
    var massiveString = buildMassiveString();

    elm.addEventListener('click', function() {
        alert(massiveString);
    }, false);
}

function buildMassiveString() {
    return new Array( 1024 * 1024 + 1 ).join('#');
}
```

# LEAKY LISTENER

```
function addClickListener(elm) {  
    var massiveString = buildMassiveString();  
  
    elm.addEventListener('click', function() {  
        alert(massiveString);  
    }, false);  
}  
  
function buildMassiveString() {  
    return new Array(1000000).join('#');  
}
```



# LEAKY FUNCTION

```
var functions = [],
    i = 0;

function createFunction() {
    var massiveString = buildMassiveString();

    functions[i++] = function() {
        alert(massiveString);
    };
}
```

# LEAKY FUNCTION

```
var functions = [],
    i = 0;

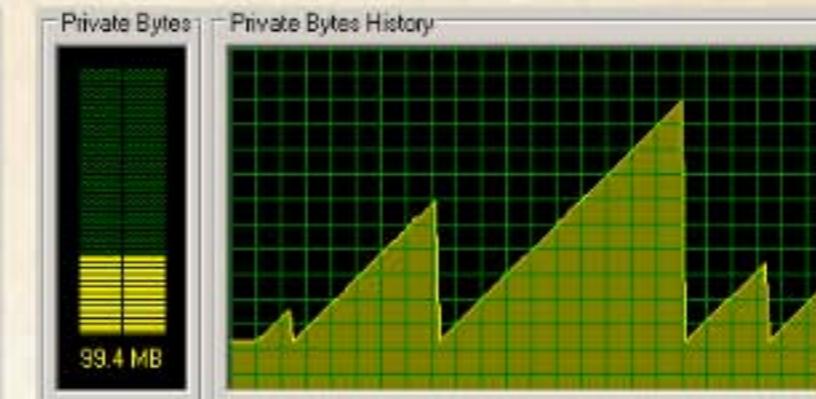
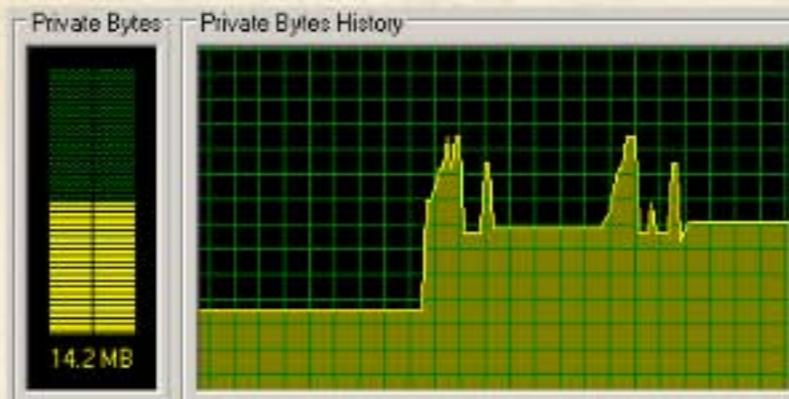
function createFunction() {
    var massiveString = buildMassiveString();

    functions[i++] = function() {
        alert('Scruttocks!');
    };
}
```

# GARBAGE COLLECTION



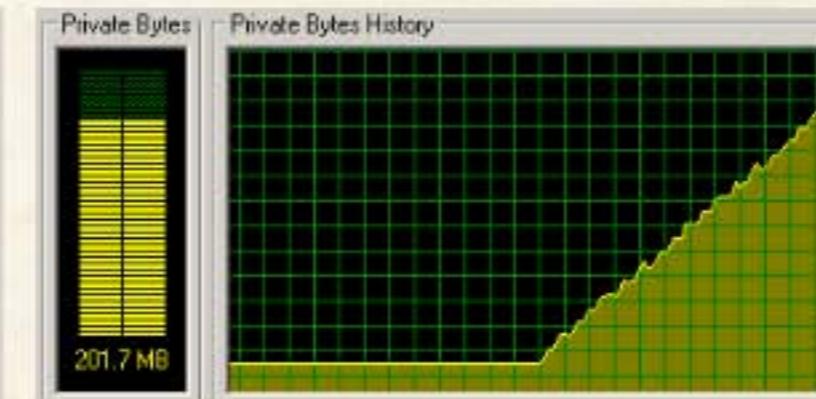
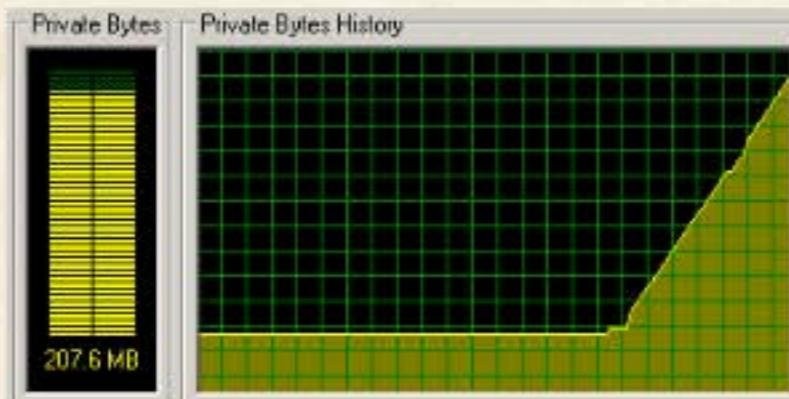
3



3.5



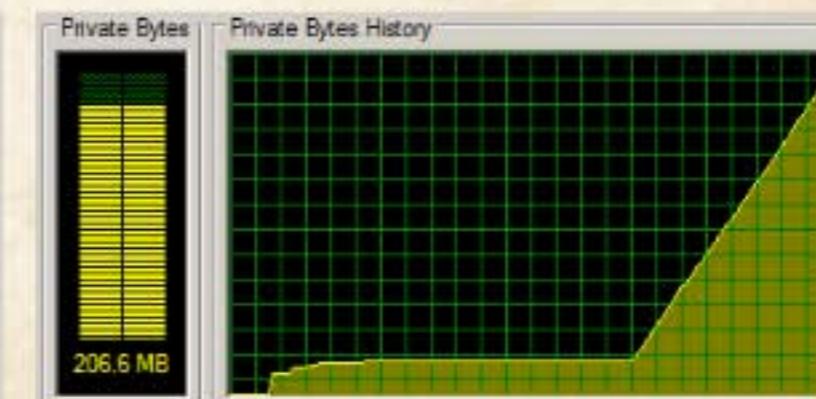
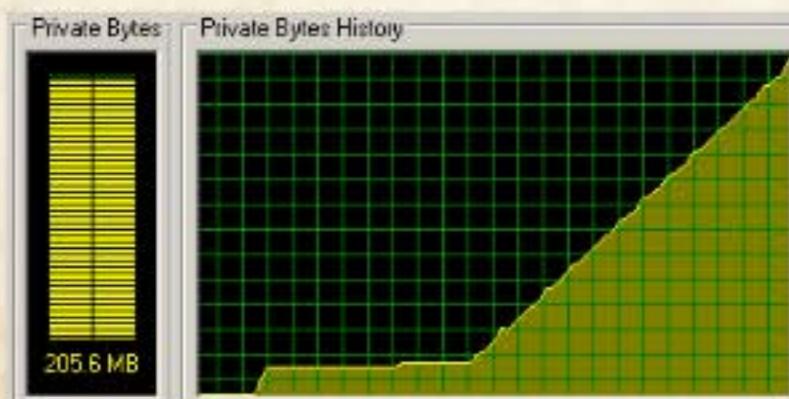
4



10



8



3



eval()

# LEAKY FUNCTION

```
var functions = [],
    i = 0;

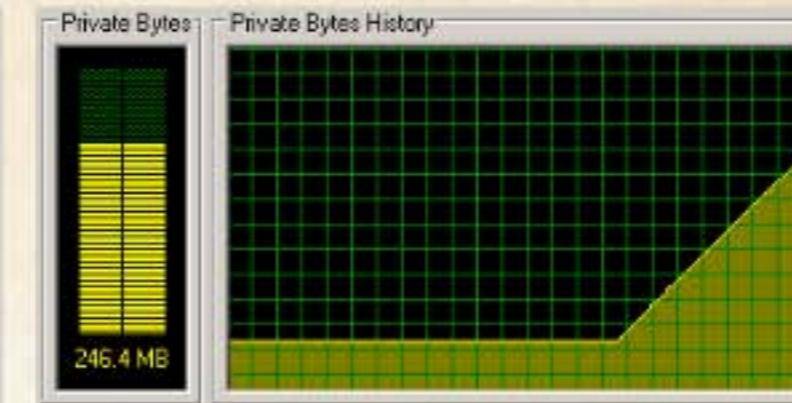
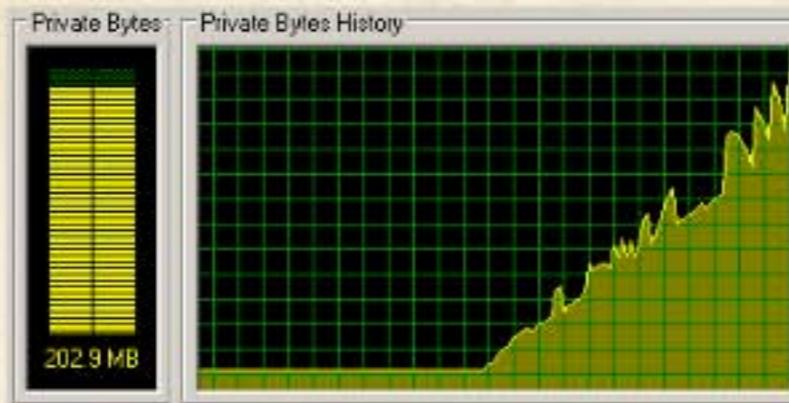
function createFunction() {
    var massiveString = buildMassiveString();

    functions[i++] = function() {
        eval('');
        alert('Scruttocks!');
    };
}
```

# SCOPE MEMORY WITH EVAL



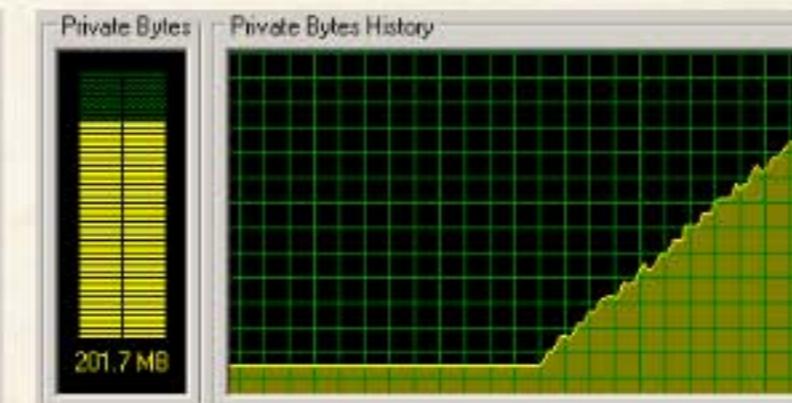
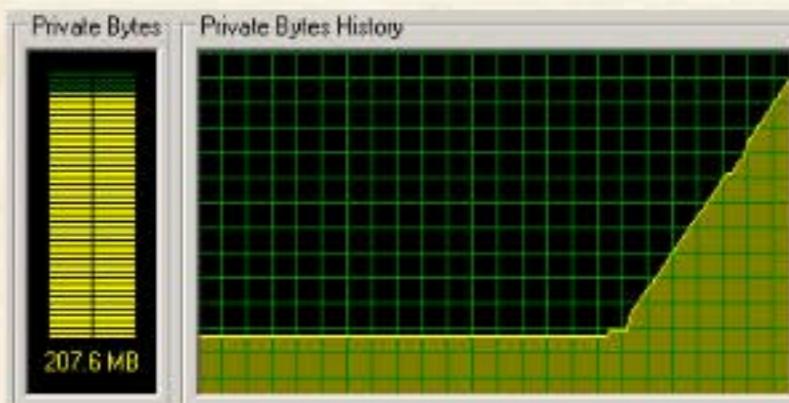
3



3.5



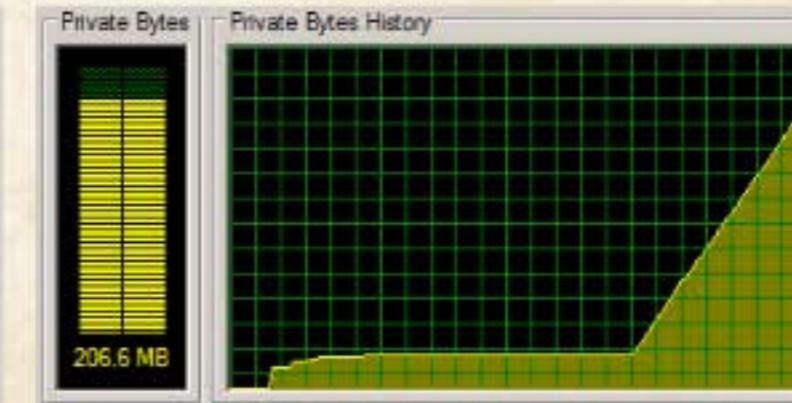
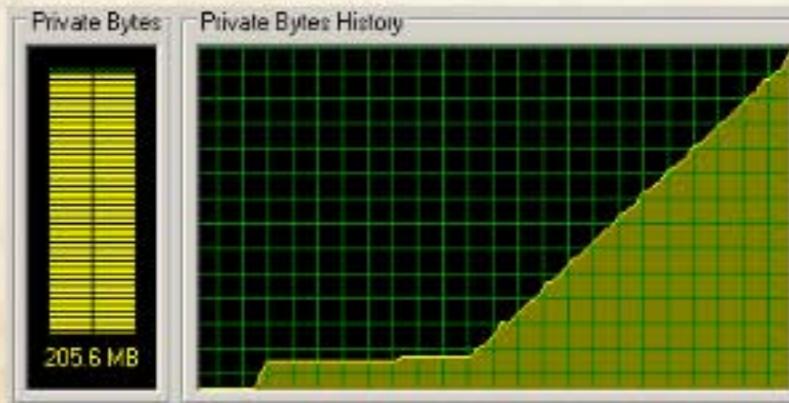
4



10

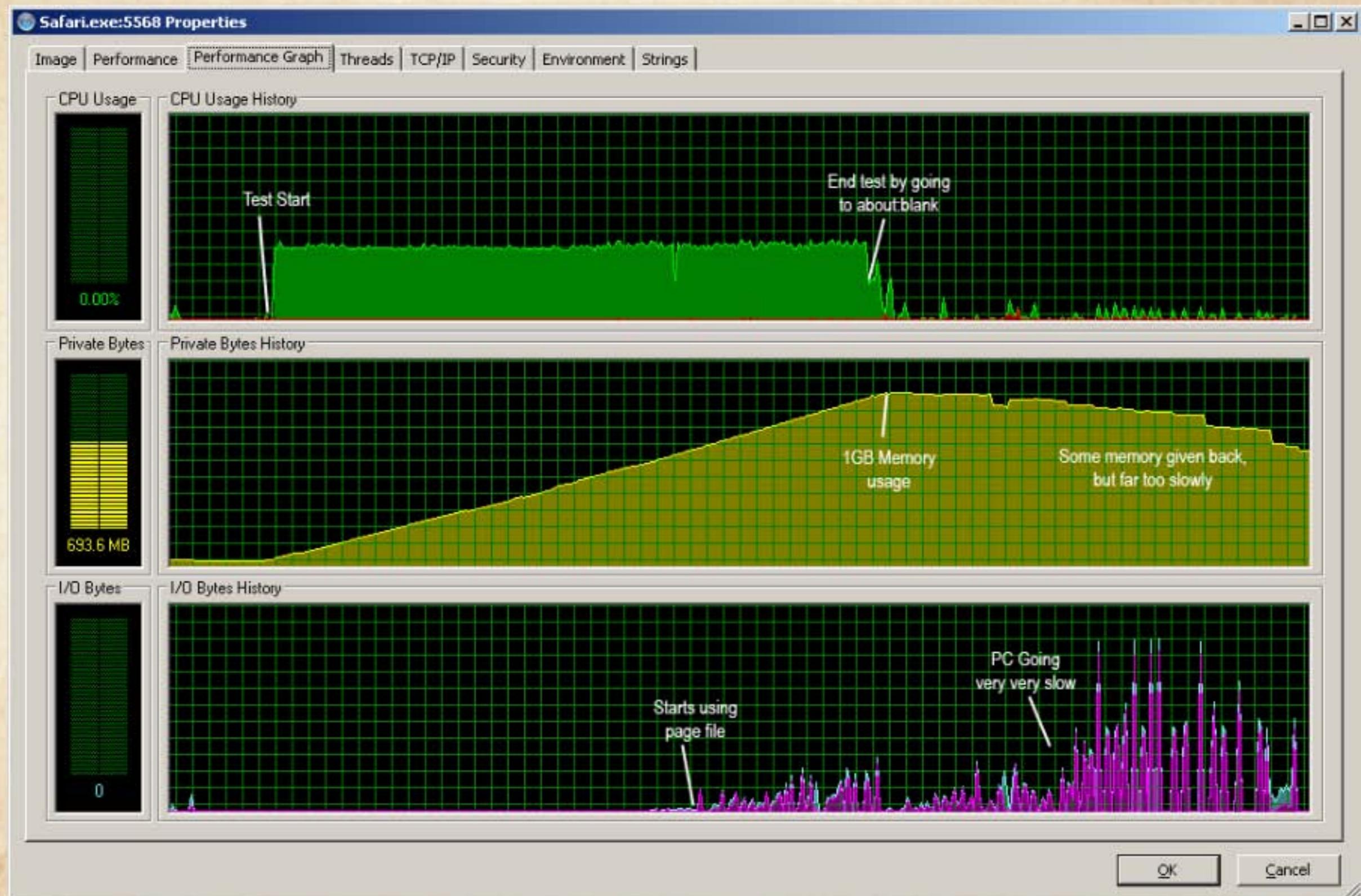


8



3

# SCOPE MEMORY WITH EVAL



# LEAKY FUNCTION SOLUTION

```
function addClickListener(elm) {  
    var massiveString = buildMassiveString();  
  
    elm.addEventListener('click', function() {  
        alert('Scruttocks!');  
    }, false);  
}
```

# LEAKY FUNCTION SOLUTION

```
function addClickListener(elm) {  
    var massiveString = buildMassiveString();  
  
    elm.addEventListener('click', clickListener, false);  
}  
  
function clickListener() {  
    alert('Scruttocks!');  
}
```

# CREATING

*Elements*

# CREATING ELEMENTS

```
<div class="generatedDiv" id="num0">
  <div></div>
</div>

function createElms() {
  var div;

  for (var i = 0, len = 100; i < len; i++) {
    div = document.createElement('div');
    div.id = 'num' + i;
    div.className = 'generatedDiv';
    div.appendChild( document.createElement('div') );
    container.appendChild(div);
  }
}
```

# CREATING ELEMENTS

```
<div class="generatedDiv" id="num0">
  <div></div>
</div>
```

```
function createElms() {
  var str = '';

  for (var i = 0, len = 100; i < len; i++) {
    str +=
      '<div class="generatedDiv" id="num' + i + '">' +
      '<div></div>' +
      '</div>';
  }
  container.innerHTML = str;
}
```

# CREATING ELEMENTS

```
<div class="generatedDiv" id="num0">
  <div></div>
</div>

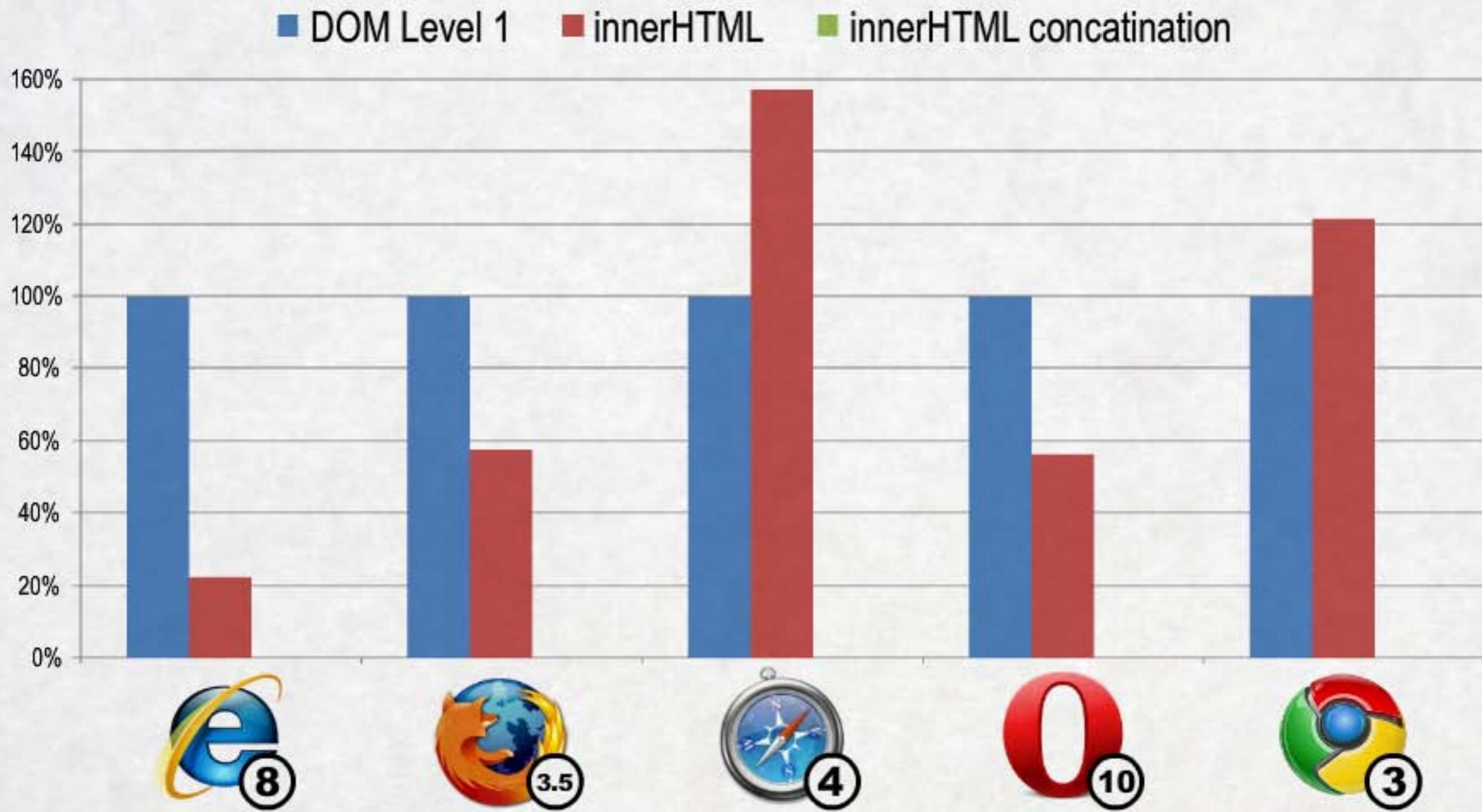
function createElms() {
  for (var i = 0, len = 100; i < len; i++) {
    container.innerHTML +=
      '<div class="generatedDiv" id="num' + i + '">' +
      '<div></div>' +
      '</div>';
  }
}
```

# CREATING ELEMENTS

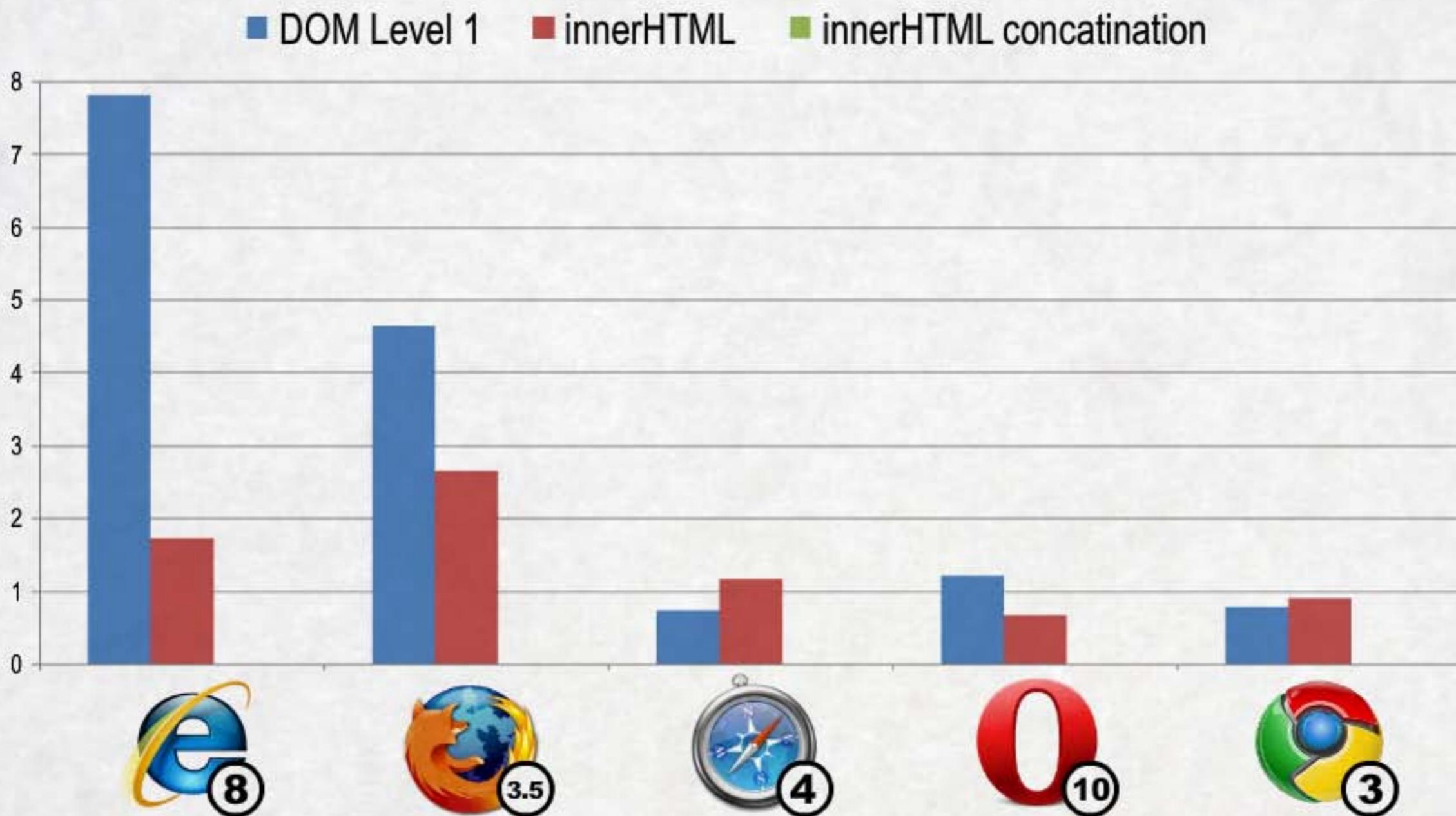
```
<div class="generatedDiv" id="num0">  
  <div></div>  
</div>
```

```
container.innerHTML = container.innerHTML +  
  '<div class="generatedDiv" id="num' + i + '">' +  
  '<div></div>' +  
  '</div>;'
```

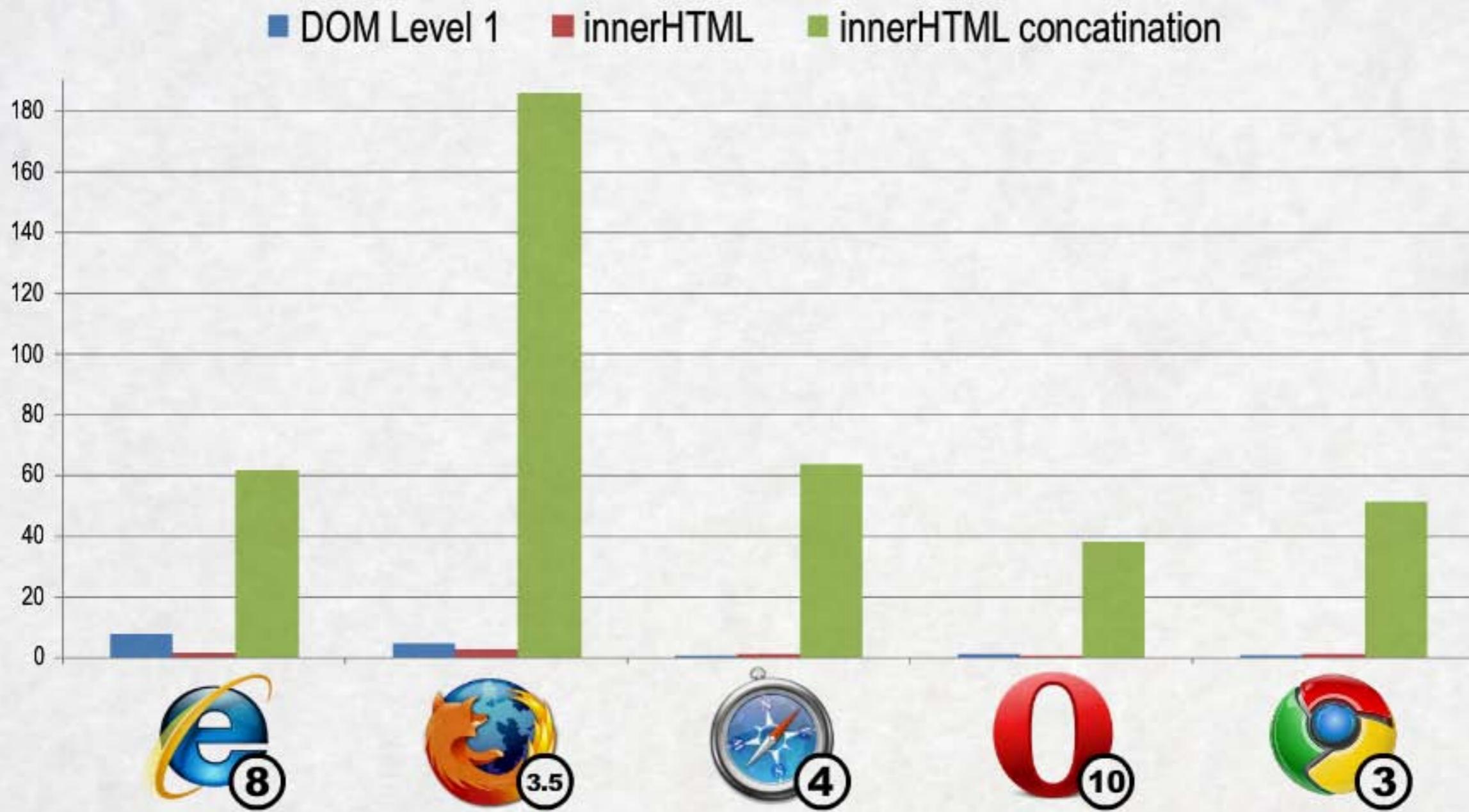
# CREATING ELEMENTS



# CREATING ELEMENTS



# CREATING ELEMENTS



# CREATING ELEMENTS

“Your node will be inserted into the first empty child.

BBC Glow (draft documentation)



Header Bar: JAWS 11.0.1

http://www.freedomscientific.com/jaws/jaws11/jaws-products/jaws11

Change Page for Low Vision

Freedom Scientific

Skip to Main Content

Search

Home | Products | Purchase | Support | Training | About Us | Vision Loss Facts

## JAWS for Windows® Screen Reading Software

"I've used JAWS for the past 15 years, and I use it for just about everything. If I wasn't for JAWS, I would not be able to practice law. Law is very, very vital to my job and what I do on a daily basis."

- Attorney John Pangaro [Listen to more](#)

[Read praise for JAWS Tandem™](#)

### What's New in JAWS 11

#### Powerful access to screen content

Get started today working with all your Windows and IBM Linux environments® applications using JAWS 11, the world's most popular screen reader. Even users for computer users whose visual loss prevents them from seeing screen content, JAWS reads aloud what's on the PC screen.

#### Compatible with the most frequently-used workplace and classroom applications

JAWS enables you to work with Lotus Symphony®, a suite of OEM tools for word processing, spreadsheets, and presentation creation and with Lotus Notes® by IBM. JAWS also is compatible with Microsoft Office Suite, MSN Messenger®, Cisco® WebEx® fact, Acrobat® Acrobat Reader, Internet Explorer®, Firefox™ – and many more applications that are used on a regular basis on the job and in school.

#### Unmatched braille support

With a nonrefreshable braille display, JAWS also provides braille output in addition to text instead of speech. An array of versatile features and customizable options lets you tailor JAWS for your individual needs and preferences.



Extras!

- [JAWS Quick Start Guide](#)
- [Learn JAWS online with new Webinar training](#)
- [JAWS Headquarters: Explore all things JAWS from stories of people who use JAWS to how to get JAWS news via RSS feeds](#)

**FREE 32-bit DEMO of JAWS 11**  
(93.4 MB FTP download)

**FREE 64-bit DEMO of JAWS 11**  
(97 MB FTP download)

A full working version of FBReader DAISY player now comes for FREE with JAWS

## JAWS® Screen Reading

past 15 years, and I use it for  
it wasn't for JAWS, I would not  
JAWS is very, very vital to my  
daily basis."

[Listen to more](#)

## JAWS Tandem™

11

## Screen content

with all your Microsoft and IBM  
applications using JAWS®, the  
screen reader. Developed for  
vision loss prevents them from  
JAWS reads aloud what's on the

most frequently-used



Extras!

# CREATING ELEMENTS

```
function createElms() {  
    var str = '';  
  
    for (var i = 0, len = 100; i < len; i++) {  
        str +=  
            '<div class="generatedDiv" id="num' + i + '">' +  
            '<div></div>' +  
            '</div>;  
    }  
    container.innerHTML = str;  
    // or if you'd prefer to use a library...  
    glow.dom.create(str).appendTo(container);  
}
```

# BACK / FWD

*Page caching*

# BACK / FORWARD CACHE

Google X

Index of /jsperformance/backforwardcache/

http://www.jakearchibald.co.uk/jsperformance/backforwardcache/

## Index of /jsperformance/backforwardcache

	<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
	<a href="#">Parent Directory</a>		-	
	<a href="#">1.html</a>	25-Aug-2009 13:26	2.5K	
	<a href="#">empty.html</a>	25-Aug-2009 13:00	645	

Apache/2.2 Server at www.jakearchibald.co.uk Port 80

Open 1.html

# BACK / FORWARD CACHE

Google X X X

Back / Forward Cache

http://www.jakearchibald.co.uk/jsperformance/backforwardcache/1.html

## Back / Forward Cache

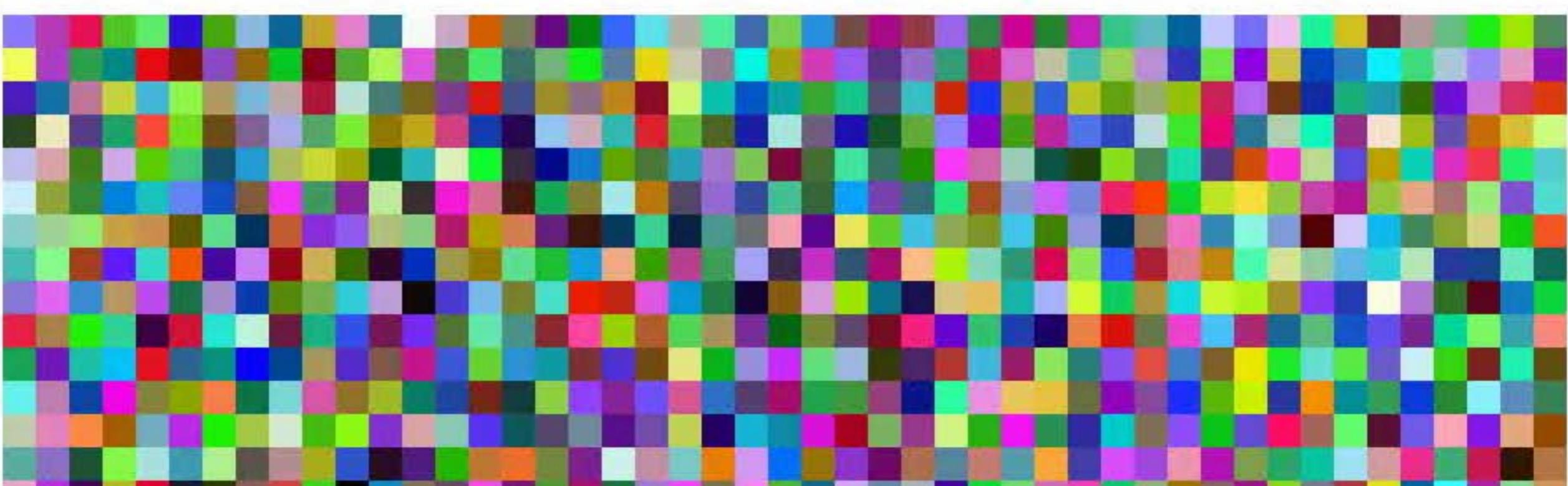
When this page loads, it waits for 5 seconds, generates a random number, then generates a load of divs of random colour.

[Go to another page](#). If your browser supports back/forward caching you should be able to click back and return to this page without a 5 second wait. The random number will not be regenerated and the divs will be the same colour they were before.

- [Add empty unload listener](#)

window load event fired  
My favourite number is: 814

5+ Seconds to Load



# BACK / FORWARD CACHE



# BACK / FORWARD CACHE

Google X X X

Back / Forward Cache

http://www.jakearchibald.co.uk/jsperformance/backforwardcache/1.html

## Back / Forward Cache

When this page loads, it waits for 5 seconds, generates a random number, then generates a load of divs of random colour.

[Go to another page](#). If your browser supports back/forward caching you should be able to click back and return to this page without a 5 second wait.

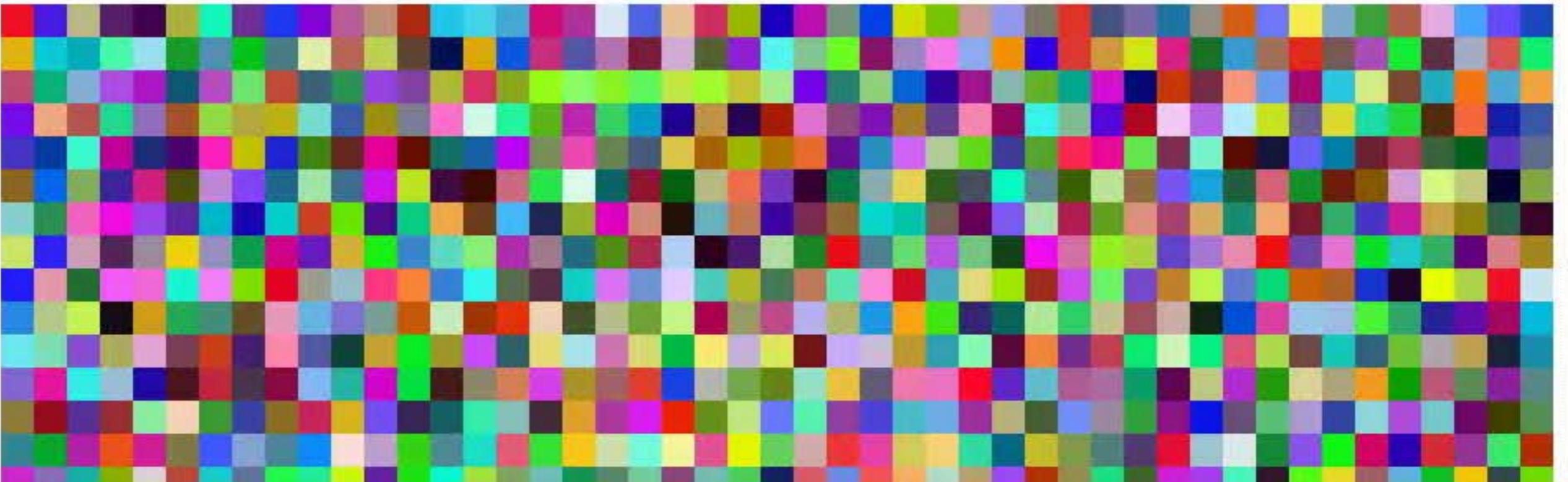
The random number will not be regenerated and the divs will be the same colour they were before.

- [Add empty unload listener](#)

window load event fired

My favourite number is: 385

5+ seconds to load again



# BACK / FORWARD CACHE

Index of /jsperformance/backforwardcache - Opera

File Edit View Bookmarks Widgets Tools Help

Index of /jsperformance... + ⌂ ⌂ ⌂

Index of /jsperformance/backforwardcache/ http://www.jakearchibald.co.uk/jsperformance/backforwardcache/ Google

## Index of /jsperformance/backforwardcache

Icon	Name	Last modified	Size	Description
[DIR]	<a href="#">Parent Directory</a>		-	
[TXT]	<a href="#">1.html</a>	25-Aug-2009 13:26	2.5K	
[TXT]	<a href="#">empty.html</a>	25-Aug-2009 13:00	645	

Apache/2.2 Server at www.jakearchibald.co.uk Port 80

Open 1.html

# BACK / FORWARD CACHE

Back / Forward Cache - Opera

File Edit View Bookmarks Widgets Tools Help

Back / Forward Cache

http://www.jakearchibald.co.uk/jsperformance/backforwardcache/1.html

Google

## Back / Forward Cache

When this page loads, it waits for 5 seconds, generates a random number, then generates a load of divs of random colour.

[Go to another page](#). If your browser supports back/forward caching you should be able to click back and return to this page without a 5 second wait. The random number will not be regenerated and the divs will be the same colour they were before.

• [Add empty unload listener](#)

window load event fired  
My favourite number is: 708

5+ seconds to load



100%

# BACK / FORWARD CACHE

This page is intentionally left blank - Opera

File Edit View Bookmarks Widgets Tools Help

This page is intentionally... +

Back Forward Stop Refresh Home http://www.jakearchibald.co.uk/jsperformance/backforwardcache/empty.htm Google

**This page is intentionally left blank**

[Previous page](#), or just click back.



Navigate to another page, now click back

# BACK / FORWARD CACHE

Back / Forward Cache - Opera

File Edit View Bookmarks Widgets Tools Help

Back / Forward Cache

http://www.jakearchibald.co.uk/jsperformance/backforwardcache/1.html

Google

## Back / Forward Cache

When this page loads, it waits for 5 seconds, generates a random number, then generates a load of divs of random colour.

[Go to another page](#). If your browser supports back/forward caching you should be able to click back and [return to this page](#) without a 5 second wait. The random number will not be regenerated and the divs will be the same colour they were before.

- [Add empty unload listener](#)

window load event fired  
My favourite number is: 708

Page opens instantly



http://www.jakearchibald.co.uk/jsperformance/backforwardcache/

# PAGE CACHE SUPPORT



**X**



**✓**



**✓**



**✓**



**X**

# PAGE CACHE SUPPORT



**X**

**X**

**X**

**X**

**X**

```
window.onunload = function() {};
```

# BACK / FORWARD CACHE

Index of /jsperformance/backforwardcache - Opera

File Edit View Bookmarks Widgets Tools Help

Index of /jsperformance... + ...

Index of /jsperformance/backforwardcache/ http://www.jakearchibald.co.uk/jsperformance/backforwardcache/ Google

## Index of /jsperformance/backforwardcache

Icon	Name	Last modified	Size	Description
[DIR]	<a href="#">Parent Directory</a>		-	
[TXT]	<a href="#">1.html</a>	25-Aug-2009 13:26	2.5K	
[TXT]	<a href="#">empty.html</a>	25-Aug-2009 13:00	645	

Apache/2.2 Server at www.jakearchibald.co.uk Port 80

Open 1.html

# BACK / FORWARD CACHE

Back / Forward Cache - Opera

File Edit View Bookmarks Widgets Tools Help

Back / Forward Cache

http://www.jakearchibald.co.uk/jsperformance/backforwardcache/1.html

Google

## Back / Forward Cache

When this page loads, it waits for 5 seconds, generates a random number, then generates a load of divs of random colour.

[Go to another page](#). If your browser supports back/forward caching you should be able to click back and return to this page without a 5 second wait. The random number will not be regenerated and the divs will be the same colour they were before.

• [Add empty unload listener](#)

window load event fired  
My favourite number is: 708

5+ seconds to load



100%

# BACK / FORWARD CACHE

Back / Forward Cache - Opera

File Edit View Bookmarks Widgets Tools Help

Back / Forward Cache

http://www.jakearchibald.co.uk/jsperformance/backforwardcache/1.html

Google

## Back / Forward Cache

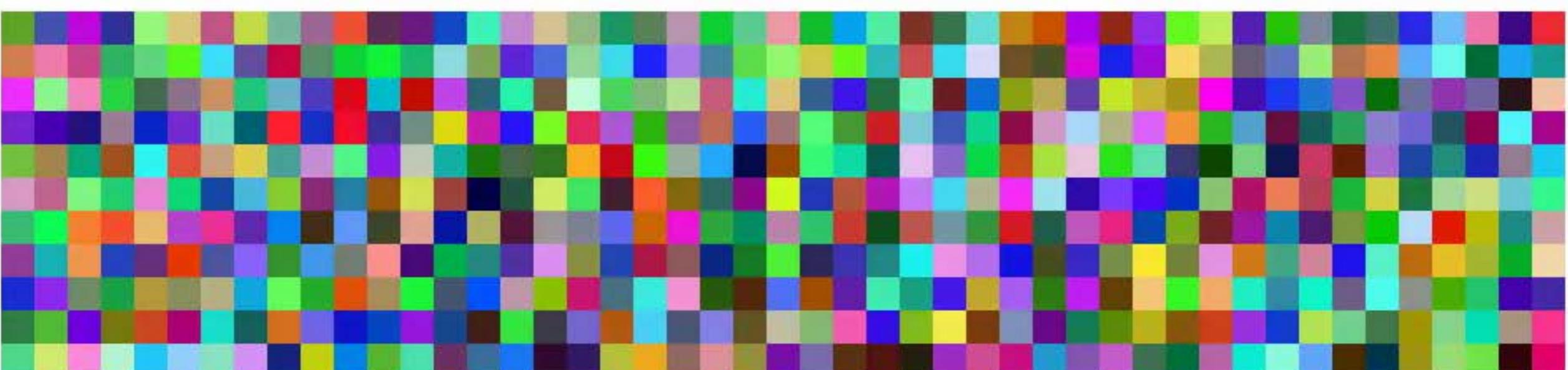
When this page loads, it waits for 5 seconds, generates a random number, then generates a load of divs of random colour.

[Go to another page](#). If your browser supports back/forward caching you should be able to click back and return to this page without a 5 second wait. The random number will not be regenerated and the divs will be the same colour they were before.

- [Add empty unload listener](#)

window load event fired  
My favourite number is: 218  
unload listener added

### Add unload listener



100%

# BACK / FORWARD CACHE

This page is intentionally left blank - Opera

File Edit View Bookmarks Widgets Tools Help

This page is intentionally... +

Back Forward Stop Refresh Home http://www.jakearchibald.co.uk/jsperformance/backforwardcache/empty.htm Google

**This page is intentionally left blank**

[Previous page](#), or just click back.



Navigate to another page, now click back

# BACK / FORWARD CACHE

Back / Forward Cache - Opera

File Edit View Bookmarks Widgets Tools Help

Back / Forward Cache

http://www.jakearchibald.co.uk/jsperformance/backforwardcache/1.html

Google

## Back / Forward Cache

When this page loads, it waits for 5 seconds, generates a random number, then generates a load of divs of random colour.

[Go to another page](#). If your browser supports back/forward caching you should be able to click back and return to this page without a 5 second wait. The random number will not be regenerated and the divs will be the same colour they were before.

• [Add empty unload listener](#)

window load event fired  
My favourite number is: 708

5+ seconds to load



100%

# BACK / FORWARD CACHE

```
jQuery( window ).bind( 'unload' , function() {  
    // do stuff  
});
```

# BACK / FORWARD CACHE

```
/*@cc_on
    jQuery( window ).bind( 'unload', function() {
        // do stuff
    });
@*/
```

# SELECTING

*Elements*

DOM Level 3

document.evaluate

XPath

~~VM Levels~~

~~document evaluate~~

~~Path~~

WHATWG

document.querySelectorAll

CSS Selectors

# QUERYSELECTION SUPPORT



✓ **0**



✓ **10**



✓ **3.5**



✓ **3.1**



✓ **8**

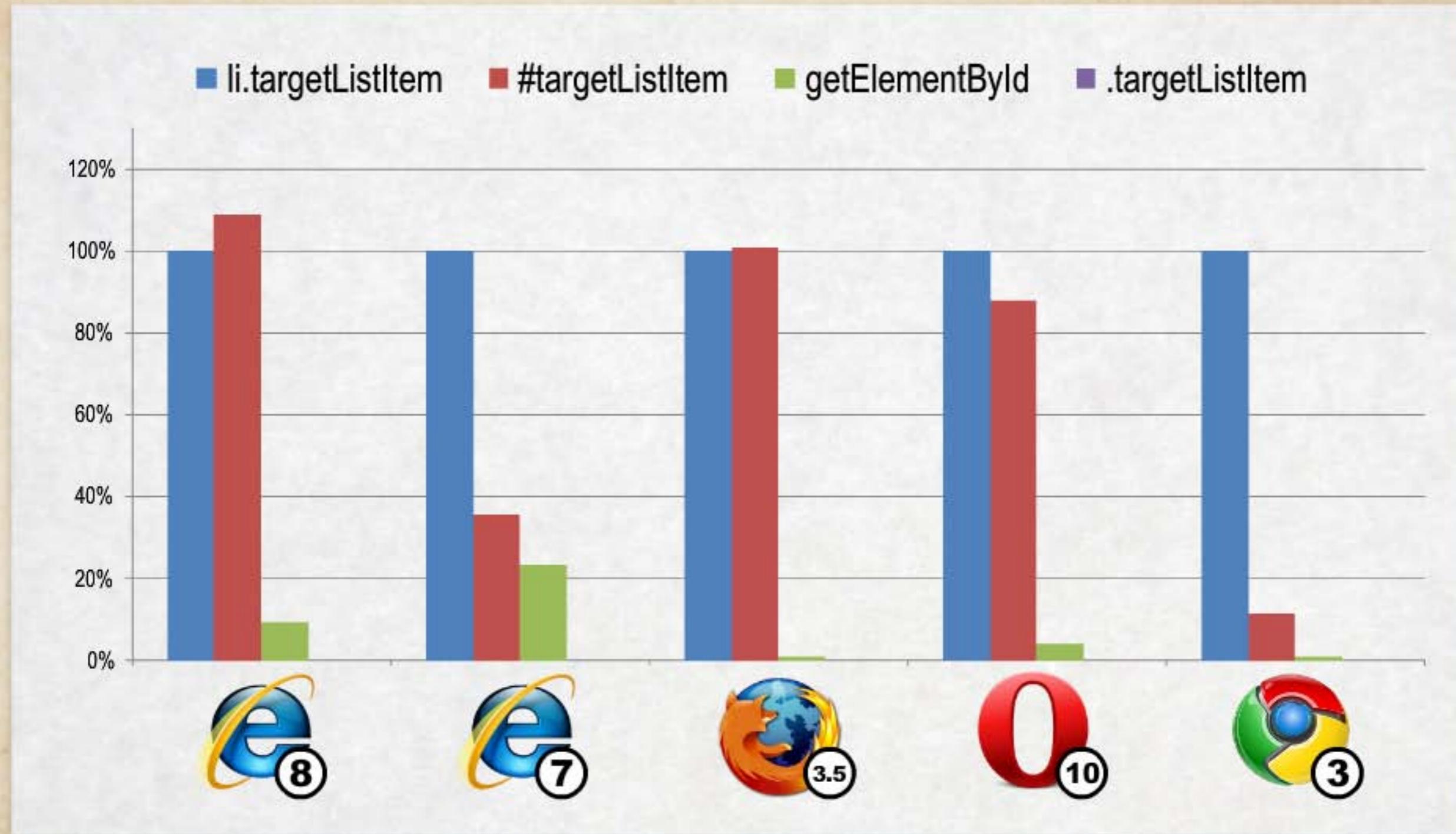
# SELECTORS...

*...in JavaScript libraries*

# SIZZLE BENCHMARK

```
Sizzle("li.targetListItem") ;  
  
Sizzle("#targetListItem") ;  
  
document.getElementById("targetListItem") ;  
  
Sizzle(".targetListItem") ;
```

# SIZZLE BENCHMARK

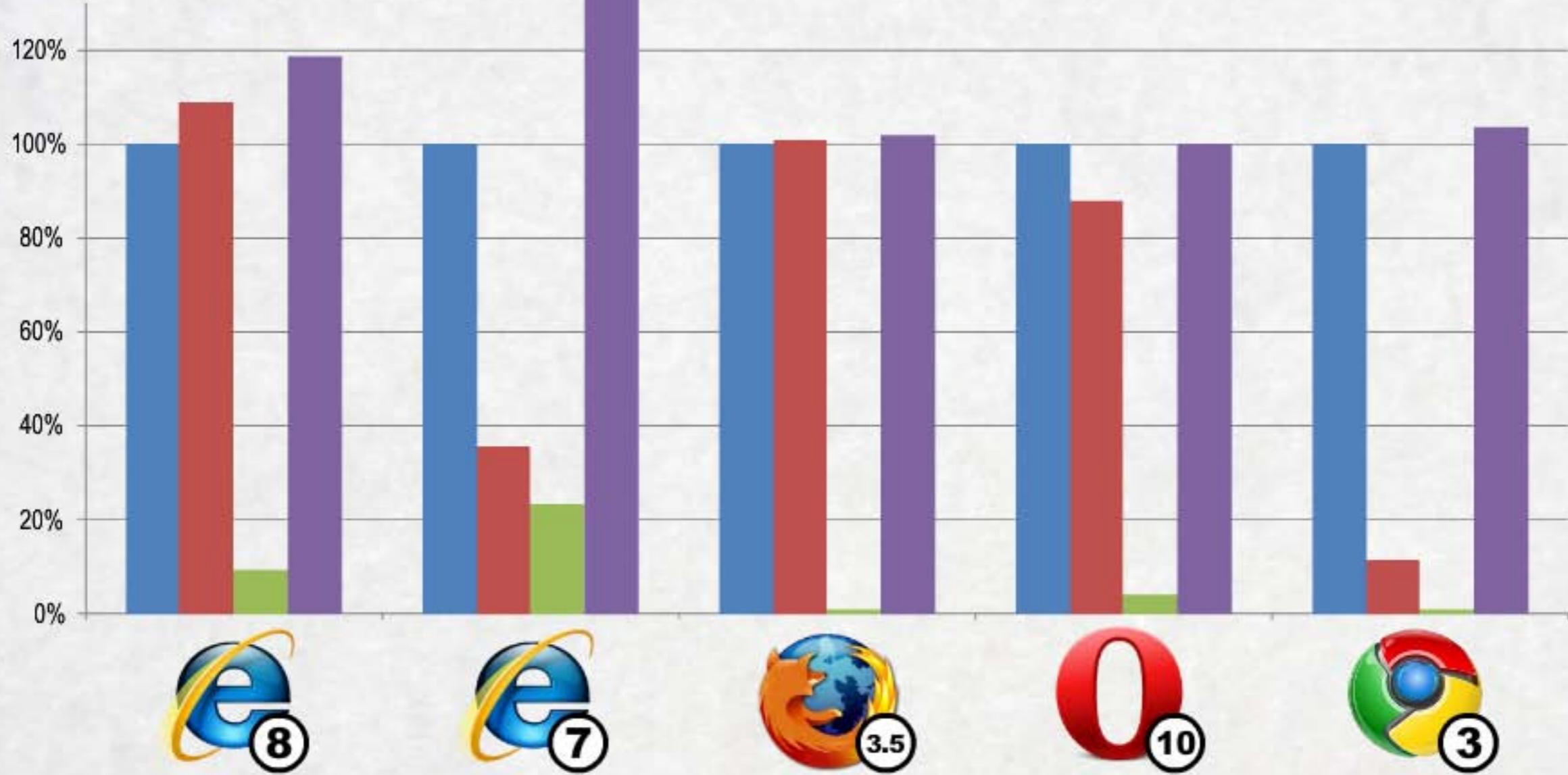


# SIZZLE BENCHMARK

DK

1244%

■ li.targetListItem ■ #targetListItem ■ getElementById ■ .targetListItem



# SIZZLE BENCHMARK

IE8

11.22ms

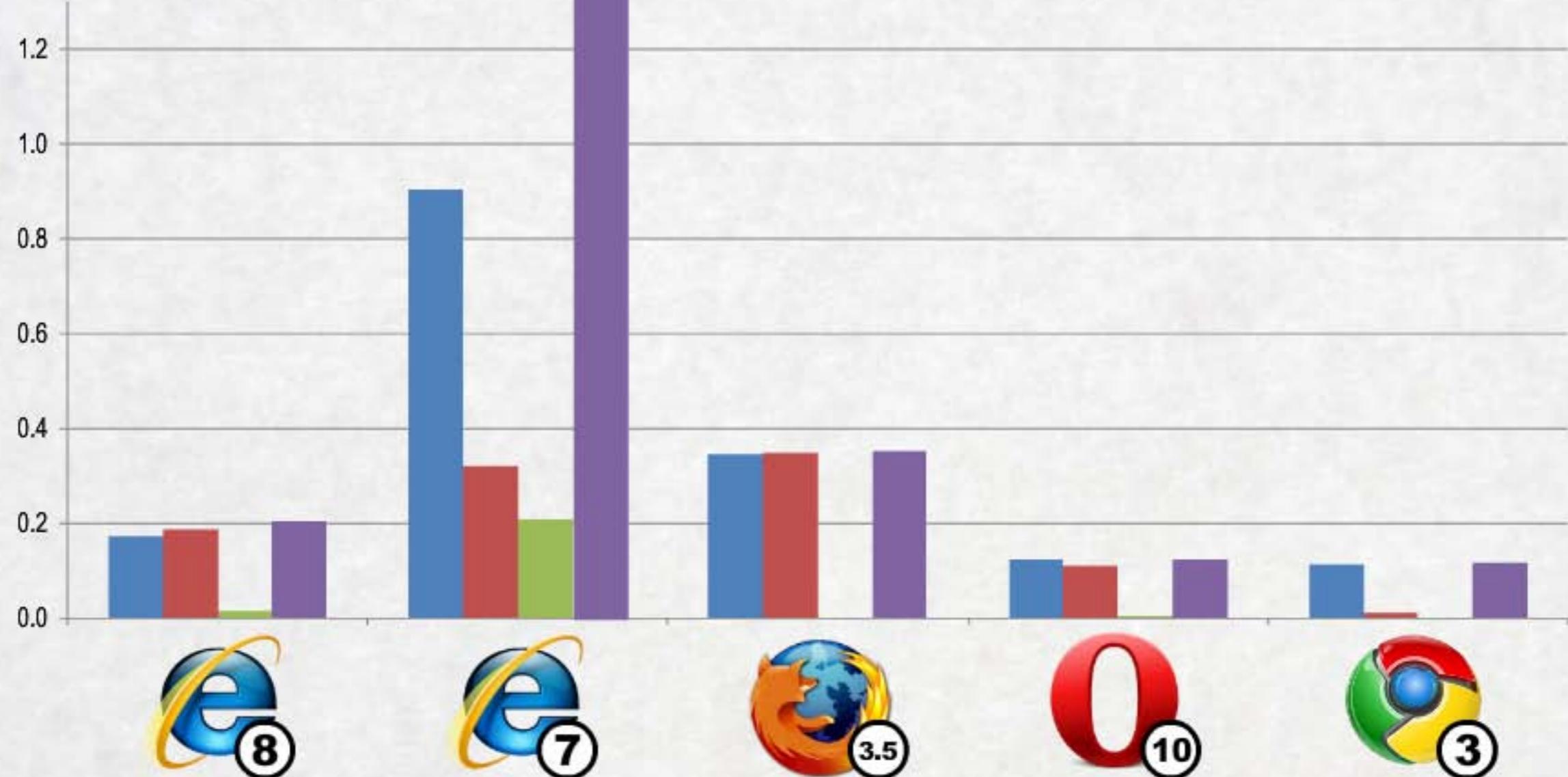
IE7

getElementsBy

.targetListIte

li.targetListIte

#tar



evaluate

getElementsByClassName

getElementById

querySelectorAll

getElementsByTagName

~~evaluate~~

~~getElementsByTagName~~

~~getElementById~~

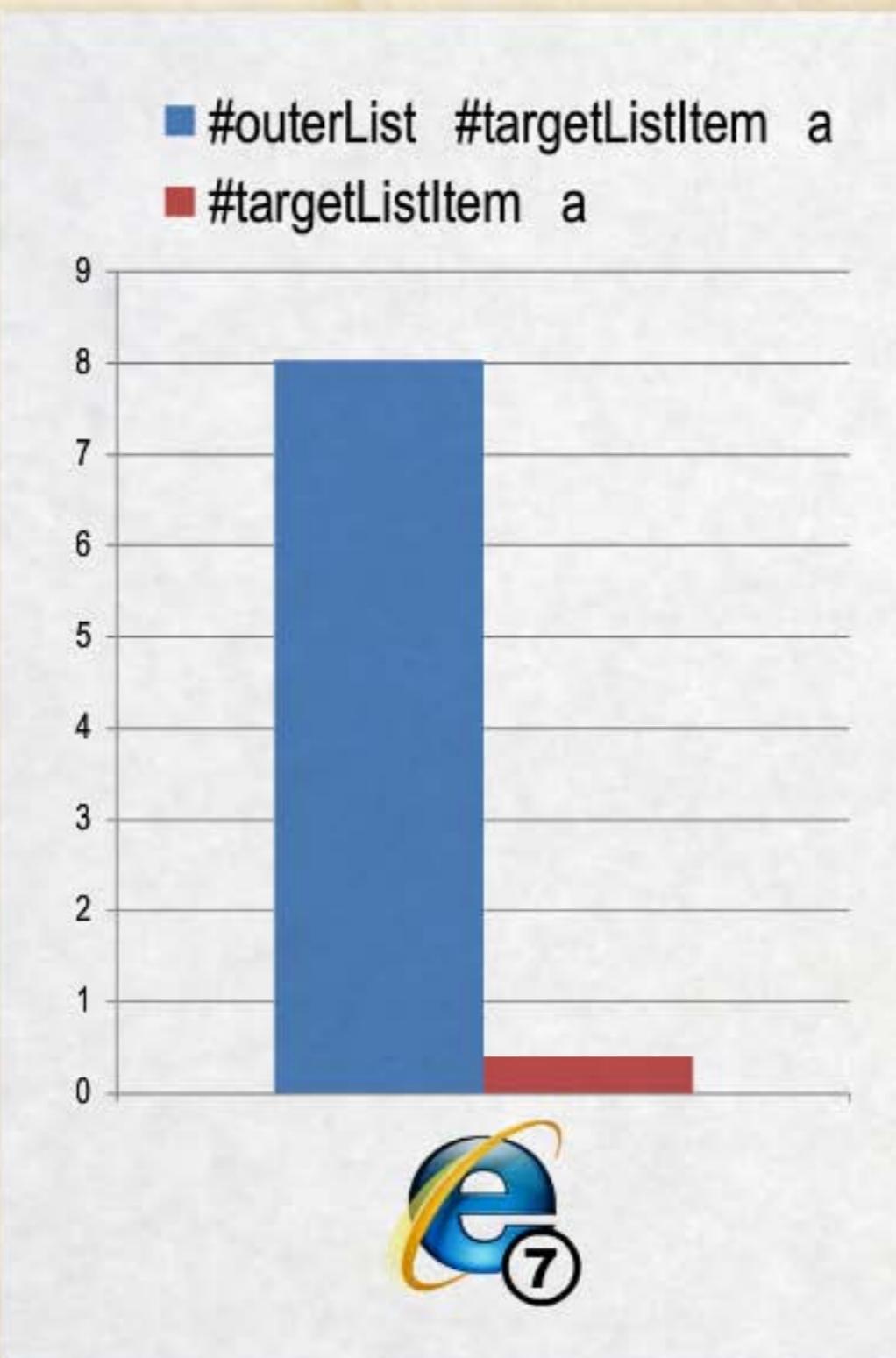
~~querySelectorAll~~

getElementsByTagName

# SIZZLE BENCHMARK

```
// avoid:  
Sizzle(".headline");  
  
// instead:  
Sizzle("#articlePromos h2.headline");
```

# SIZZLE BENCHMARK



# CUSTOM CSS

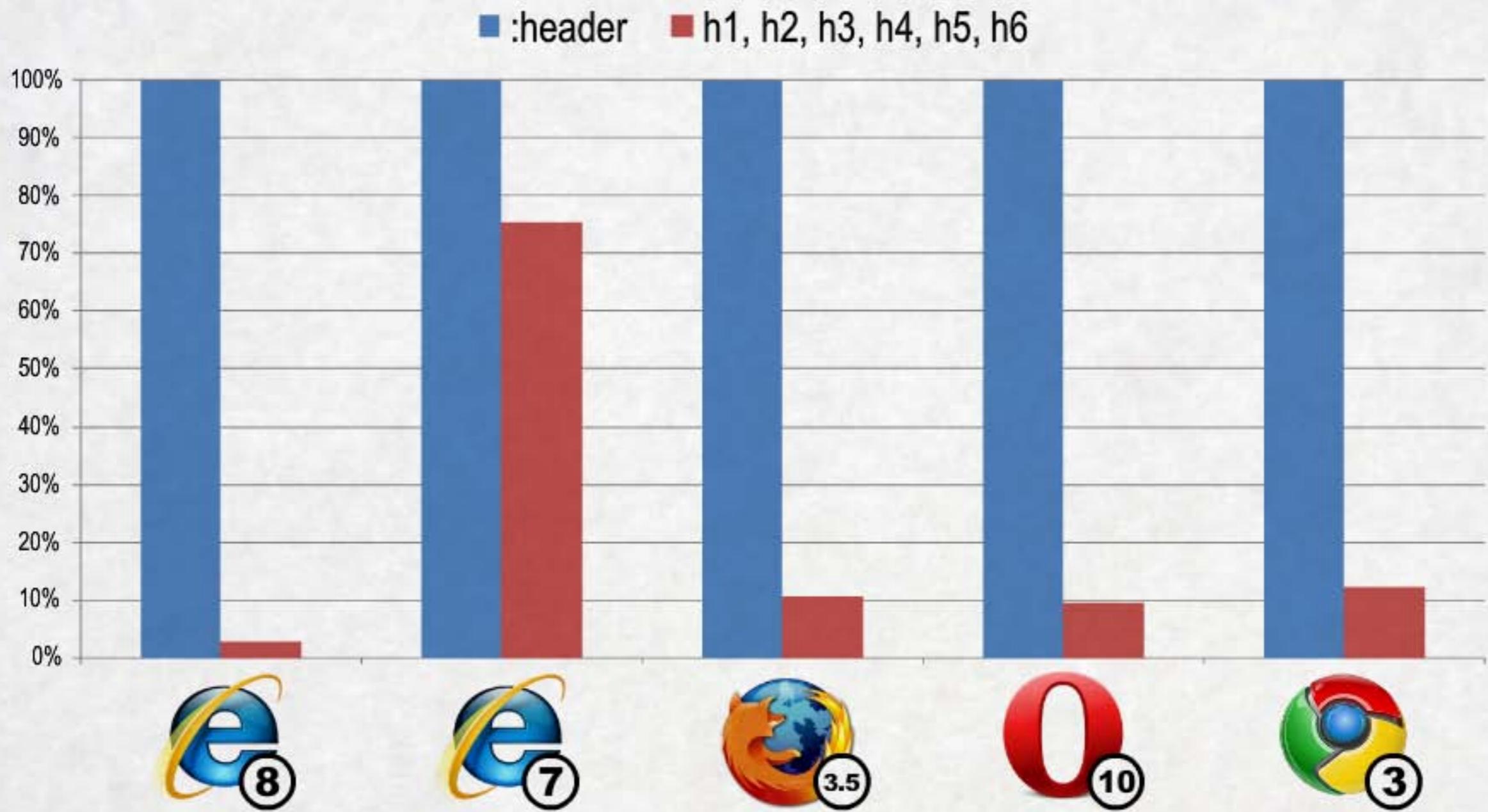
*Non-standard Selectors*

# CUSTOM SELECTORS

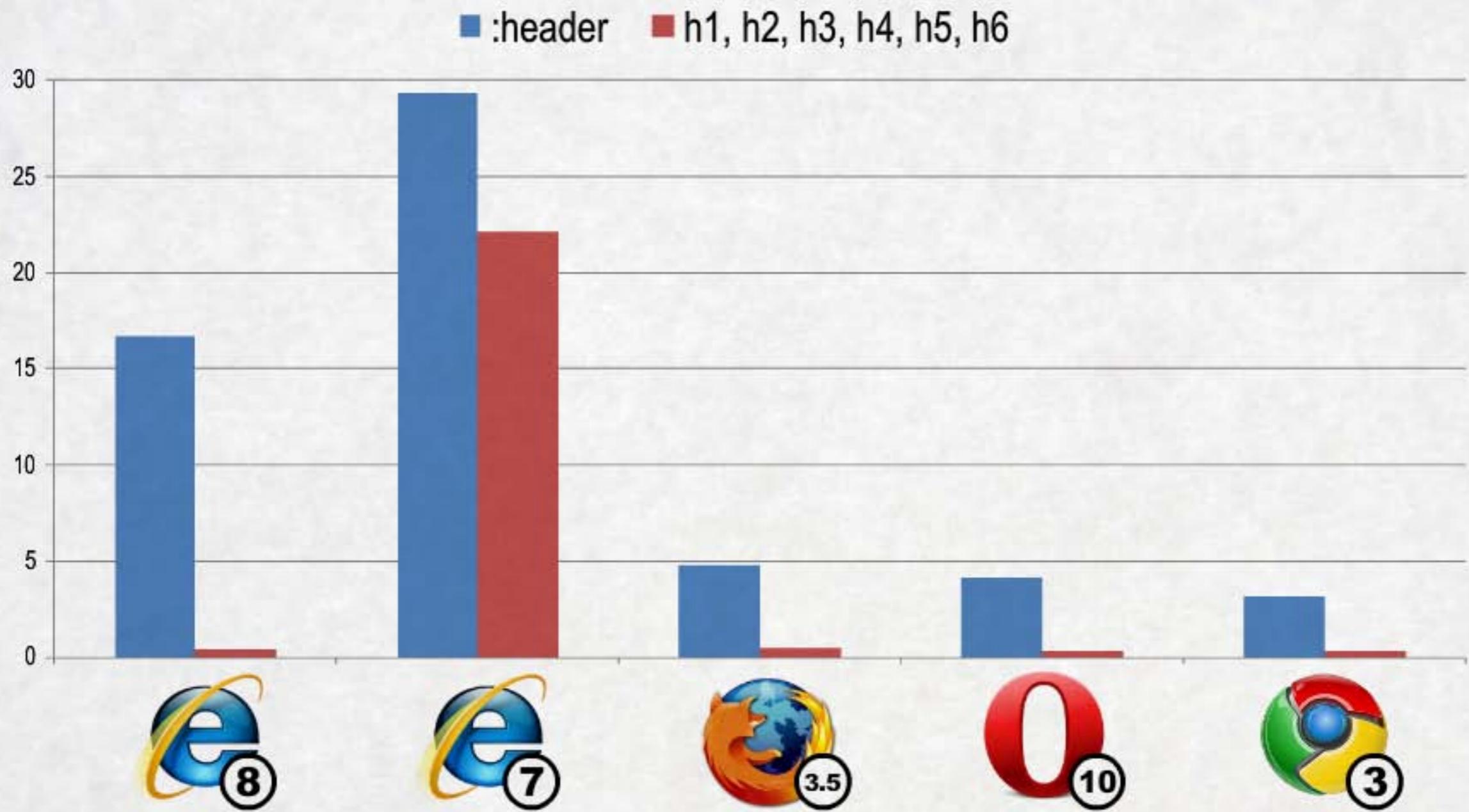
```
// Standard CSS
Sizzle("h1, h2, h3, h4, h5, h6");

// Special Sizzle CSS
Sizzle(":header");
```

# CUSTOM SELECTORS



# CUSTOM SELECTORS



# BENCHMARKING

*Writing your own tests*

# BENCHMARKING

The screenshot shows two browser windows side-by-side. The left window is Mozilla Firefox displaying a 'Selectors Test' page. The right window is Microsoft Internet Explorer displaying the same test page. Both pages show the results of a CSS selector performance test.

**Mozilla Firefox (Left):**

**selectors-test.html**

This tests a few simple CSS select...

**Console**

Profile (17941.005ms, 3634003 calls)

Function	Calls	Perf.
PSEUDO()	1813000	40.7
filter()	1000	29.9
header()	1813000	28.3
Sizzle()	1000	0.54
Sizzle()	1000	0.17

>>>

Done

**Windows Internet Explorer (Right):**

**Selectors Test - Windows Internet Explorer**

This tests a few simple CSS selectors using Sizzle @ e0f5cbc75d12aa78f3ef30930414b2f88da7b2b8

File Find Disable View Outline Images Cache Tools Validate

Browser Mode: IE8 Compat View Document Mode: IE7 Standards

HTML CSS Script Profiler

Start Profiling Current View: Functions Report 2

Function	Count	Inclusive Time (ms)	Exclusive Ti...	URL
CLASS	1,000	500.00	500.00	
isXML	2,000	62.50	62.50	
speedTestSelector	1	703.13	31.25	
Sizzle	1,000	671.88	31.25	
String.replace	4,000	31.25	31.25	
RegExp.exec	10,000	15.63	15.63	
TAG	1,000	15.63	15.63	
filter	1,000	593.75	15.63	
onclick	1	703.13	0.00	
Date	2	0.00	0.00	
Array	3,000	0.00	0.00	

Internet 100%

# BENCHMARKING

- Isolate badly performing code





# BENCHMARKING

- Isolate badly performing code
- Time the script using Date objects

```
var duration,  
    start = new Date();  
  
thingToTest();  
  
duration = new Date() - start;
```

# BENCHMARKING

- Isolate badly performing code
- Time the script using Date objects
- Ensure your test run long enough to iron out timer errors

```
var loopCount = 5000,  
    duration,  
    start = new Date();  
  
while (loopCount--) {  
    thingToTest();  
}  
  
duration = new Date() - start;
```

# BENCHMARKING

- Isolate badly performing code
- Time the script using Date objects
- Ensure your test run long enough to iron out timer errors
- And... turn developer tools off

```
var loopCount = 5000,  
    duration,  
    start = new Date();  
  
while (loopCount--) {  
    thingToTest();  
}  
  
duration = new Date() - start;
```

# BENCHMARKING

The screenshot shows a web browser window with the title bar "Google" and the tab "Creating Elements". The address bar contains the URL "http://www.jakearchibald.co.uk/jsperformance/creatingelements/2.html". The main content area displays the heading "Creating Elements" and a descriptive paragraph about creating 100 divs. Below the paragraph is a bulleted list of methods, followed by three performance results. The bottom of the browser window shows the footer "jakearchibald.co.uk/jsperformance/creatingelements/2.html#".

## Creating Elements

Here I create 100 divs, with a unique ID & same className, each containing a single div `<div class="generatedDiv" id="num1"><div></div></div>` and add them to the document.

- [w3Method](#)
- [innerHTMLAppend](#) (Warning - may hang your browser for minutes)
- [innerHTMLConcat](#)

404ms (w3Method)  
415ms (w3Method)  
483ms (w3Method)

jakearchibald.co.uk/jsperformance/creatingelements/2.html#

# BENCHMARKING

Creating Elements

http://www.jakearchibald.co.uk/jsperformance/creatingelements/2.html

## Creating Elements

Here I create 100 divs, with a unique class="generatedDiv" id="num1"><div>

- [w3Method](#)
- [innerHTMLAppend \(Warning - m\)](#)
- [innerHTMLConcat](#)

404ms (w3Method)  
415ms (w3Method)  
483ms (w3Method)

The screenshot shows the Google Chrome Developer Tools open in a new tab. The title bar says "Developer Tools - http://www.jakearchibald.co.uk/jsperform...". The main area displays the DOM tree under the "Elements" tab. The tree shows the following structure:

- <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-gb">
- <head>
- <body>
  - <div id="heading">
  - <p>
  - <ul>
    - <li>
    - <li>
    - <li>
  - </ul>
  - <pre id="output">
  - <div id="container">
  - <script type="text/javascript">

On the right side of the developer tools, there are panels for "Styles", "Metrics", and "Properties". The "Styles" panel shows a rule for li elements: "display: list-item;". The "Metrics" panel shows performance metrics for the page.

# BENCHMARKING

The screenshot shows a web browser window with the title bar "Creating Elements". The address bar contains the URL "http://www.jakearchibald.co.uk/jsperformance/creatingelements/2.html". The main content area displays the heading "Creating Elements" and a descriptive paragraph about creating 100 divs. Below the paragraph is a bulleted list of methods. At the bottom of the page, there is a table with performance data.

## Creating Elements

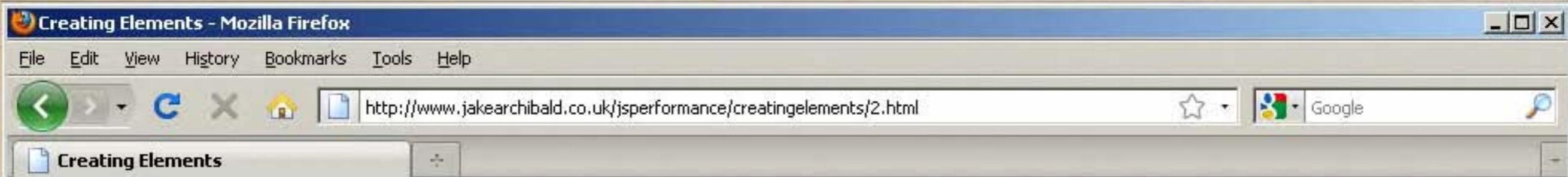
Here I create 100 divs, with a unique ID & same className, each containing a single div `<div class="generatedDiv" id="num1"><div></div></div>` and add them to the document.

- [w3Method](#)
- [innerHTMLAppend](#) (Warning - may hang your browser for minutes)
- [innerHTMLConcat](#)

Method	Time (ms)
w3Method	404ms
w3Method	415ms
w3Method	483ms
w3Method	1390ms
w3Method	1719ms
w3Method	1401ms

jakearchibald.co.uk/jsperformance/creatingelements/2.html#

# BENCHMARKING



## Creating Elements

Here I create 100 divs, with a unique ID & same className, each containing a single div `<div class="generatedDiv" id="num1"><div></div></div>` and add them to the document.

- [w3Method](#)
- [innerHTMLAppend](#) (Warning - may hang your browser for minutes)
- [innerHTMLConcat](#)

483ms (w3Method)

485ms (w3Method)

488ms (w3Method)

# BENCHMARKING

Creating Elements - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Back Forward Stop Home http://www.jakearchibald.co.uk/jisperformance/creatingelements/2.html Google

Creating Elements

## Creating Elements

Here I create 100 divs, with a unique ID & same className, each containing a single div `<div class="generatedDiv" id="num1"><div></div></div>` and add them to the document.

- [w3Method](#)
- [innerHTMLAppend](#) (Warning - may hang your browser for minutes)

Console HTML CSS Script DOM Net

Edit body < html

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-gb">
  <head>
  <body>
    <div id="_firebugConsole" style="display: none;" FirebugVersion="1.4.2"/>
  </body>
</html>
```

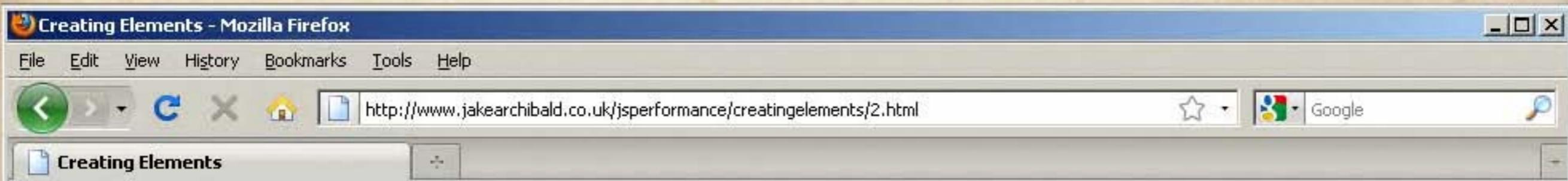
Style Layout DOM

```
body {
  -x-system-font: none;
  font-family: normal verdana, sans-serif;
  font-size: 0.9em;
  font-size-adjust: none;
  font-stretch: normal;
  font-style: normal;
  font-variant: normal;
  font-weight: normal;
  line-height: 1.5;
}
```

2.html (line 11)

Done

# BENCHMARKING



## Creating Elements

Here I create 100 divs, with a unique ID & same className, each containing a single div `<div class="generatedDiv" id="num1"><div></div></div>` and add them to the document.

- [w3Method](#)
- [innerHTMLAppend](#) (Warning - may hang your browser for minutes)
- [innerHTMLConcat](#)

483ms (w3Method)

485ms (w3Method)

488ms (w3Method)

2878ms (w3Method)

# WRAPPING UP

- Optimise where it hurts
- Don't believe everything you read
- Keep persistent functions shallow-scoped
- Eval and unload listeners are enemies of performance
- Non-standard features are fast! (`innerHTML`)
- ...except when they aren't (`:header`)
- Write your own tests and publicise your findings
- Praise be to the BATTLECRAB

# WRAPPING UP

- Optimise with `script` events
- Don't believe everything you read
- Keep persistent functions low-scoped
- Eval and unload listeners at the end of series of performance
- Non-standard features are fast!
- . except when they are not
- to



# ...AND BREATHE

- [jake.archibald@bbc.co.uk](mailto:jake.archibald@bbc.co.uk)
- <http://jakearchibald.co.uk/jsperformance/>
- <http://twitter.com/jaffathecake/>
- <http://www.bbc.co.uk/blogs/webdeveloper/>
- Any questions?



**IT'S OVER!**