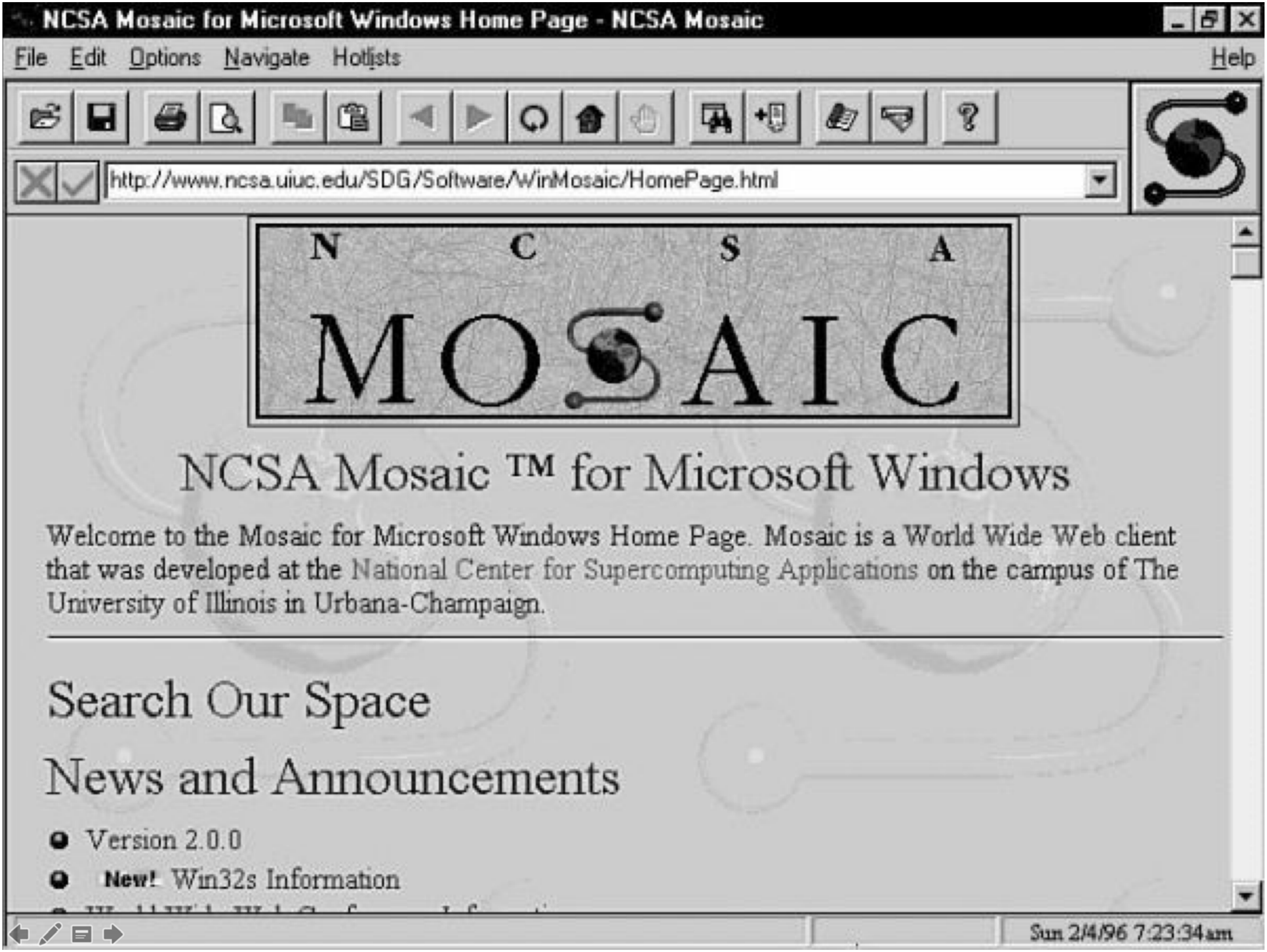
THE WEB'S ECLIPSING SENSATION

HOUDINI

UNLOCKING CSS

By

@PatrickKettner



http://www.ncsa.uiuc.edu/SDG/Software/WinMosaic/HomePage.html

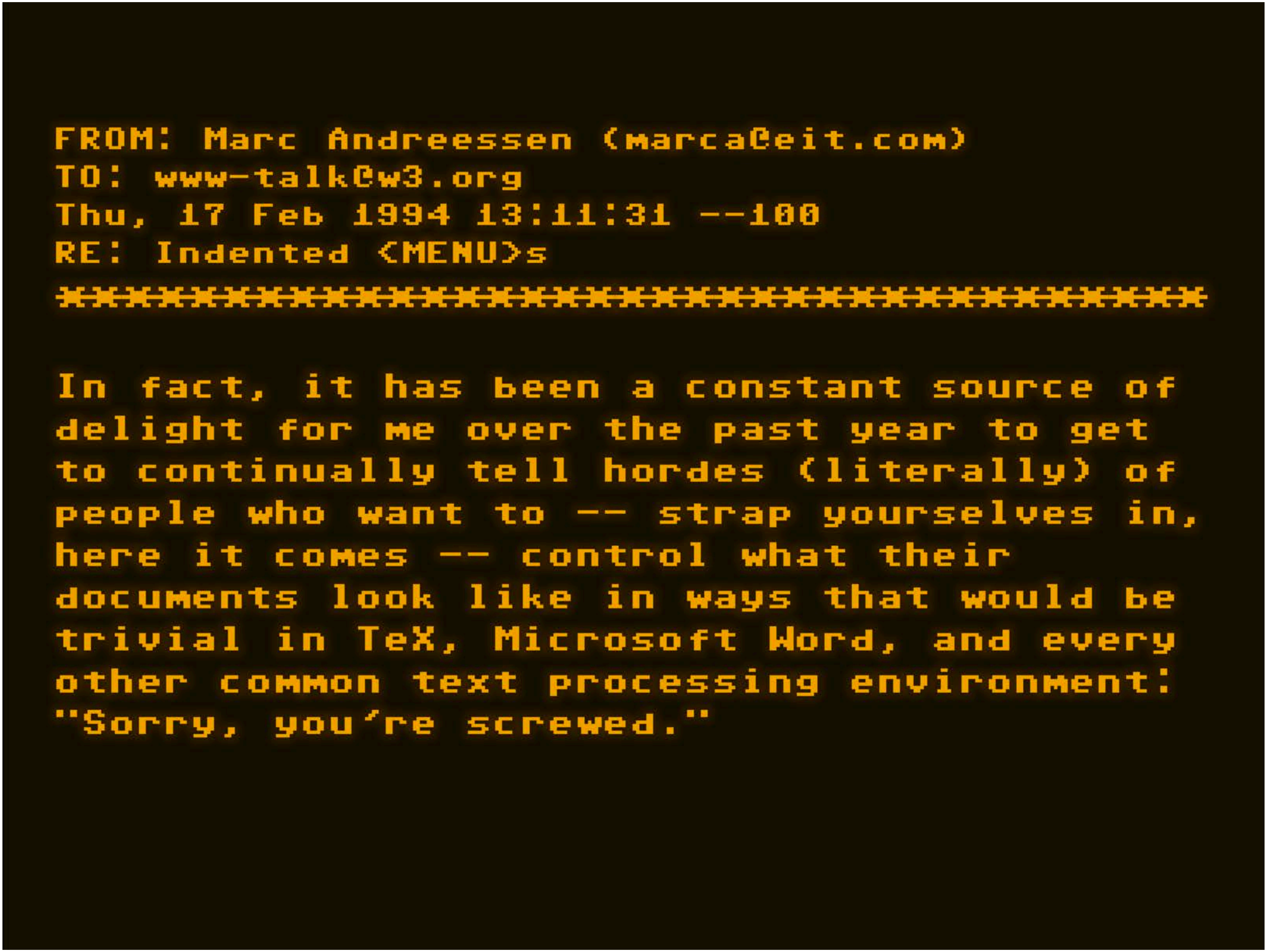
NCSA Mosaic ™ for Microsoft Windows

Welcome to the Mosaic for Microsoft Windows Home Page. Mosaic is a World Wide Web client that was developed at the National Center for Supercomputing Applications on the campus of The University of Illinois in Urbana-Champaign.

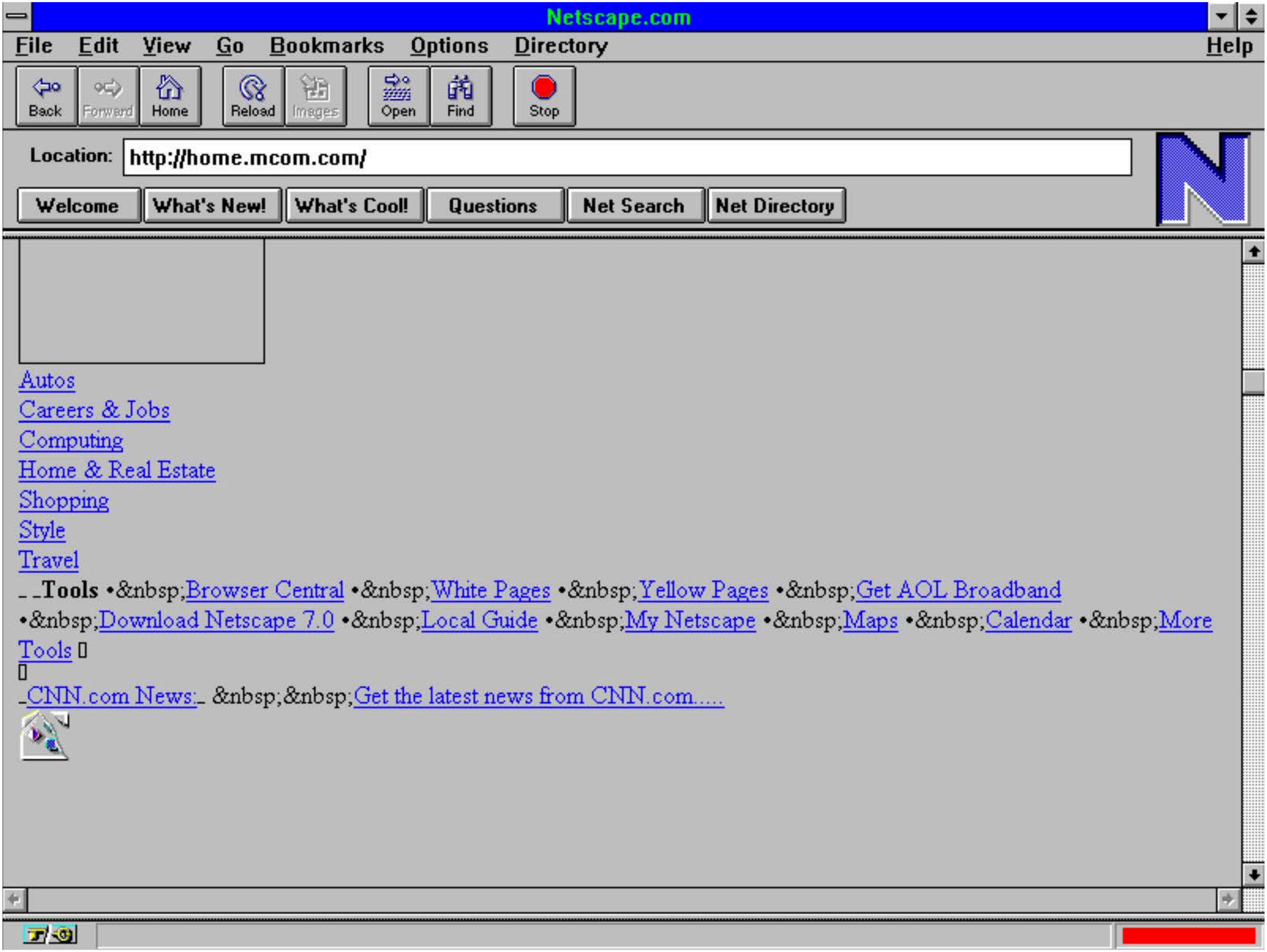
Search Our Space

News and Announcements

- Version 2.0.0
- **New!** Win32s Information



```
FROM: Marc Andreessen (marca@eit.com)
TO: www-talk@w3.org
Thu, 17 Feb 1994 13:11:31 --100
RE: Indented <MENU>s
***********************************

In fact, it has been a constant source of
delight for me over the past year to get
to continually tell hordes (literally) of
people who want to -- strap yourselves in,
here it comes -- control what their
documents look like in ways that would be
trivial in TeX, Microsoft Word, and every
other common text processing environment:
"Sorry, you're screwed."
```



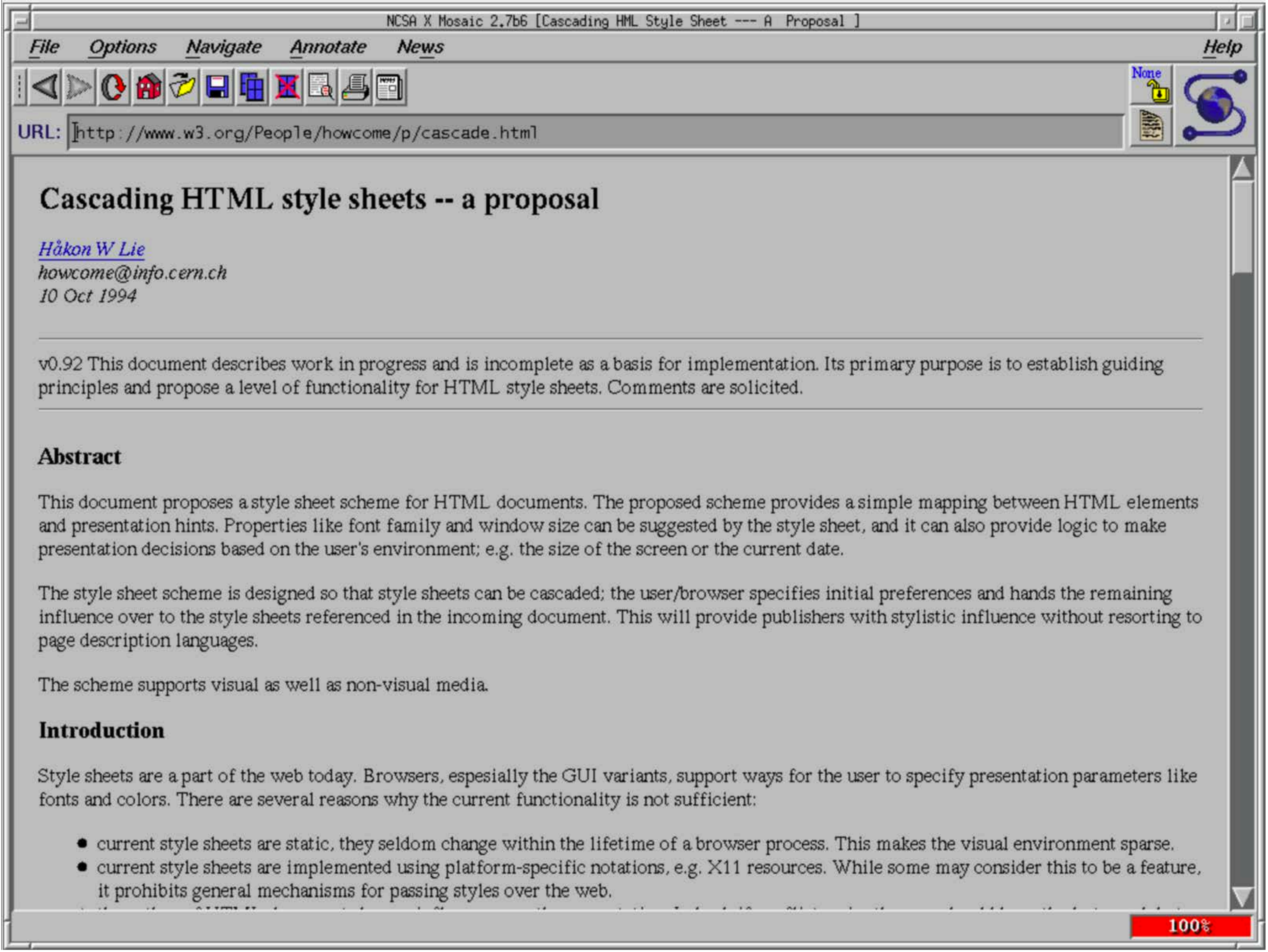
Location: http://home.mcom.com/

Autos
Careers & Jobs
Computing
Home & Real Estate
Shopping
Style
Travel

_ _**Tools** • Browser Central • White Pages • Yellow Pages • Get AOL Broadband • Download Netscape 7.0 • Local Guide • My Netscape • Maps • Calendar • More Tools ▯
▯

CNN.com News: Get the latest news from CNN.com.....

<center>



NCSA X Mosaic 2.7b6 [Cascading HML Style Sheet --- A Proposal]
File
Options
Navigate
Annotate
News
Help
None
URL: http://www.w3.org/People/howcome/p/cascade.html
100%

Cascading HTML style sheets -- a proposal

Håkon W Lie
howcome@info.cern.ch
10 Oct 1994

v0.92 This document describes work in progress and is incomplete as a basis for implementation. Its primary purpose is to establish guiding principles and propose a level of functionality for HTML style sheets. Comments are solicited.

Abstract

This document proposes a style sheet scheme for HTML documents. The proposed scheme provides a simple mapping between HTML elements and presentation hints. Properties like font family and window size can be suggested by the style sheet, and it can also provide logic to make presentation decisions based on the user's environment; e.g. the size of the screen or the current date.

The style sheet scheme is designed so that style sheets can be cascaded; the user/browser specifies initial preferences and hands the remaining influence over to the style sheets referenced in the incoming document. This will provide publishers with stylistic influence without resorting to page description languages.

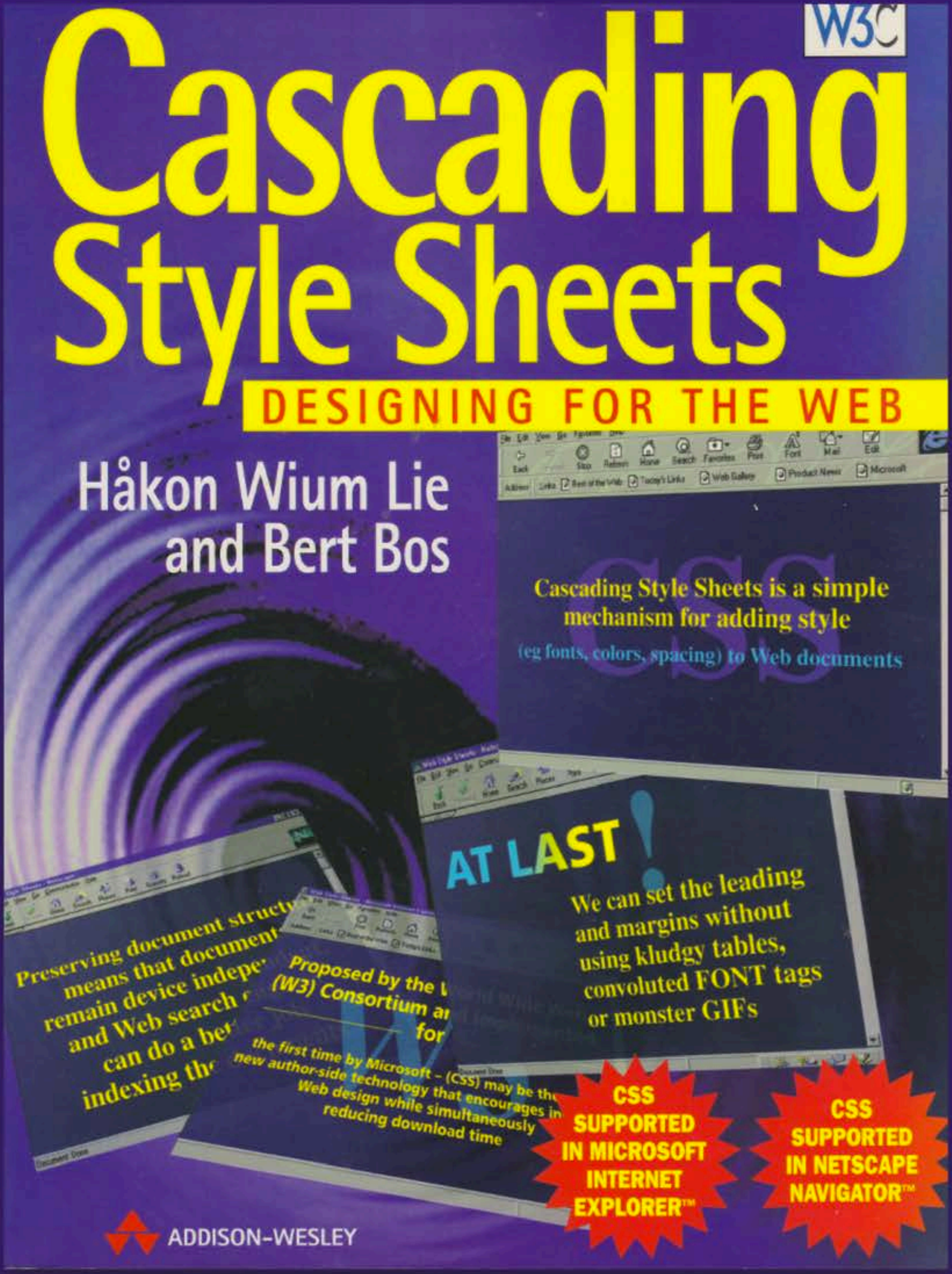
The scheme supports visual as well as non-visual media.

Introduction

Style sheets are a part of the web today. Browsers, espesially the GUI variants, support ways for the user to specify presentation parameters like fonts and colors. There are several reasons why the current functionality is not sufficient:

- current style sheets are static, they seldom change within the lifetime of a browser process. This makes the visual environment sparse.
- current style sheets are implemented using platform-specific notations, e.g. X11 resources. While some may consider this to be a feature, it prohibits general mechanisms for passing styles over the web.

Hakon
Lie
SPEA



W3C

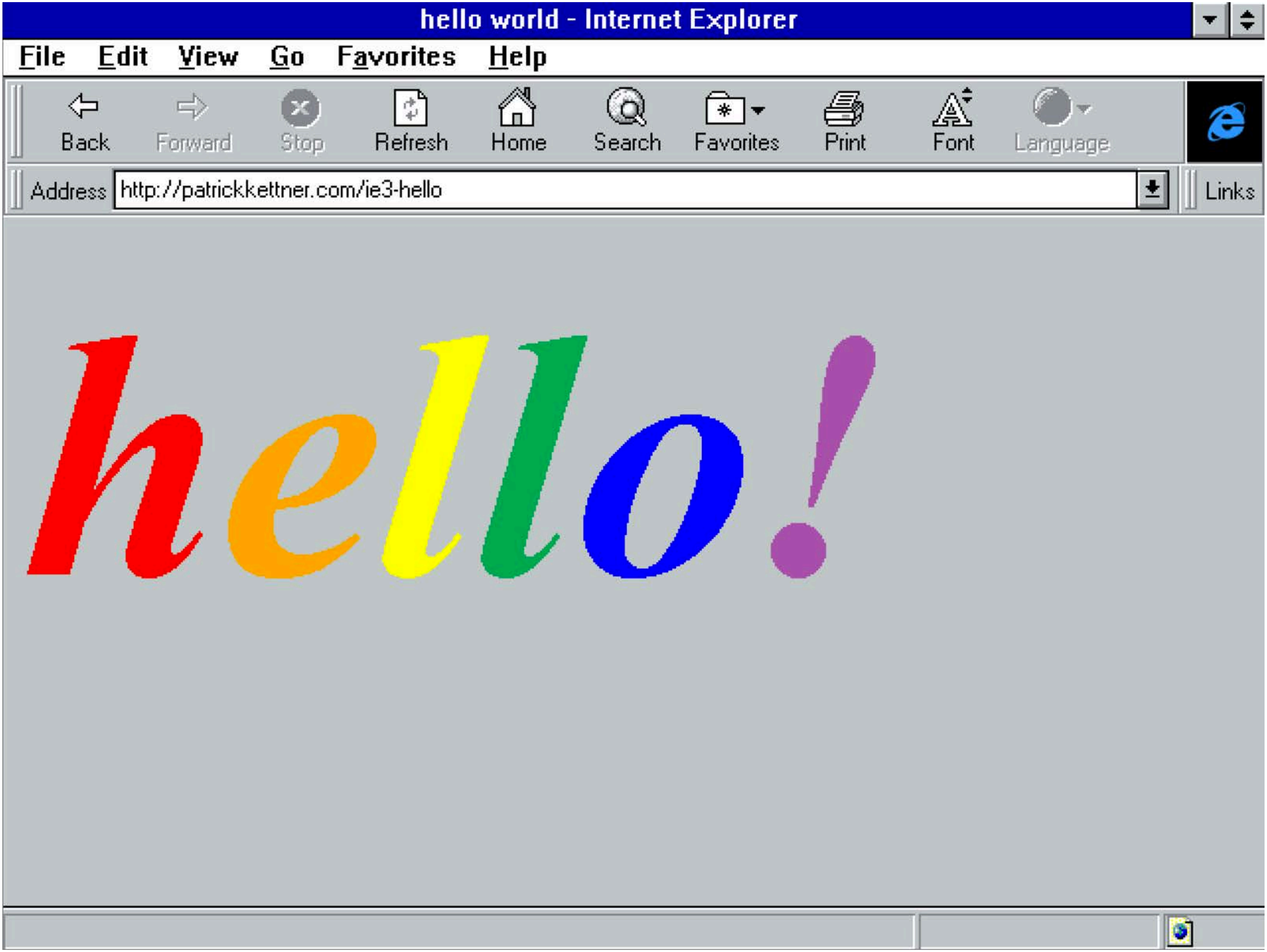
Cascading Style Sheets

DESIGNING FOR THE WEB

Håkon Wium Lie and Bert Bos

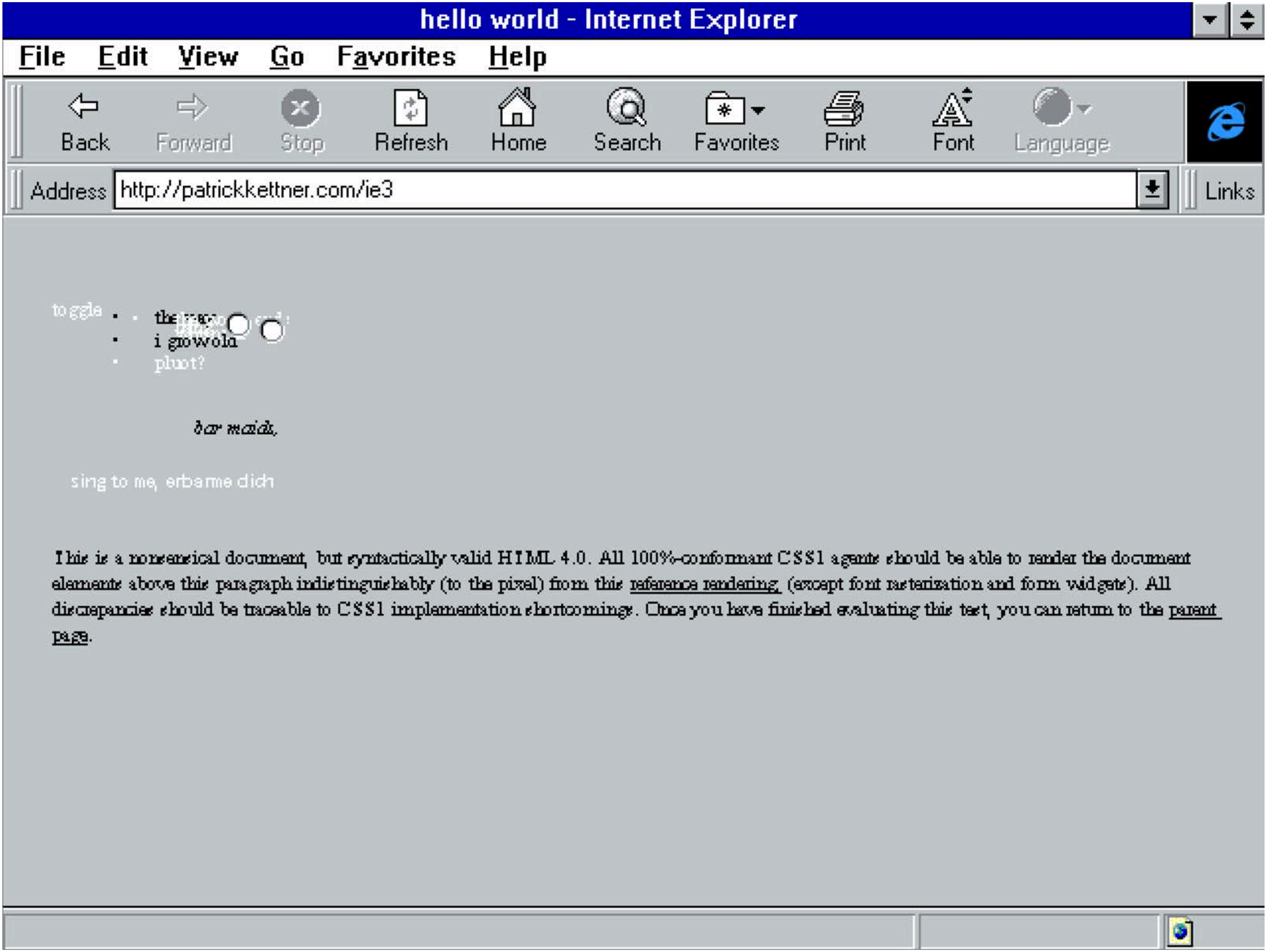
ADDISON-WESLEY

Cascading Style Sheets is a simple mechanism for adding style (eg fonts, colors, spacing) to Web documents
CSS
AT LAST!
We can set the leading and margins without using kludgy tables, convoluted FONT tags or monster GIFs
Preserving document structu… means that document… remain device indepe… and Web search … can do a be… indexing th…
Proposed by the … (W3) Consortium a… for
the first time by Microsoft – (CSS) may be th… new author-side technology that encourages in… Web design while simultaneously reducing download time
CSS SUPPORTED IN MICROSOFT INTERNET EXPLORER™
CSS SUPPORTED IN NETSCAPE NAVIGATOR™



Address http://patrickkettner.com/ie3-hello

hello!



Address http://patrickkettner.com/ie3

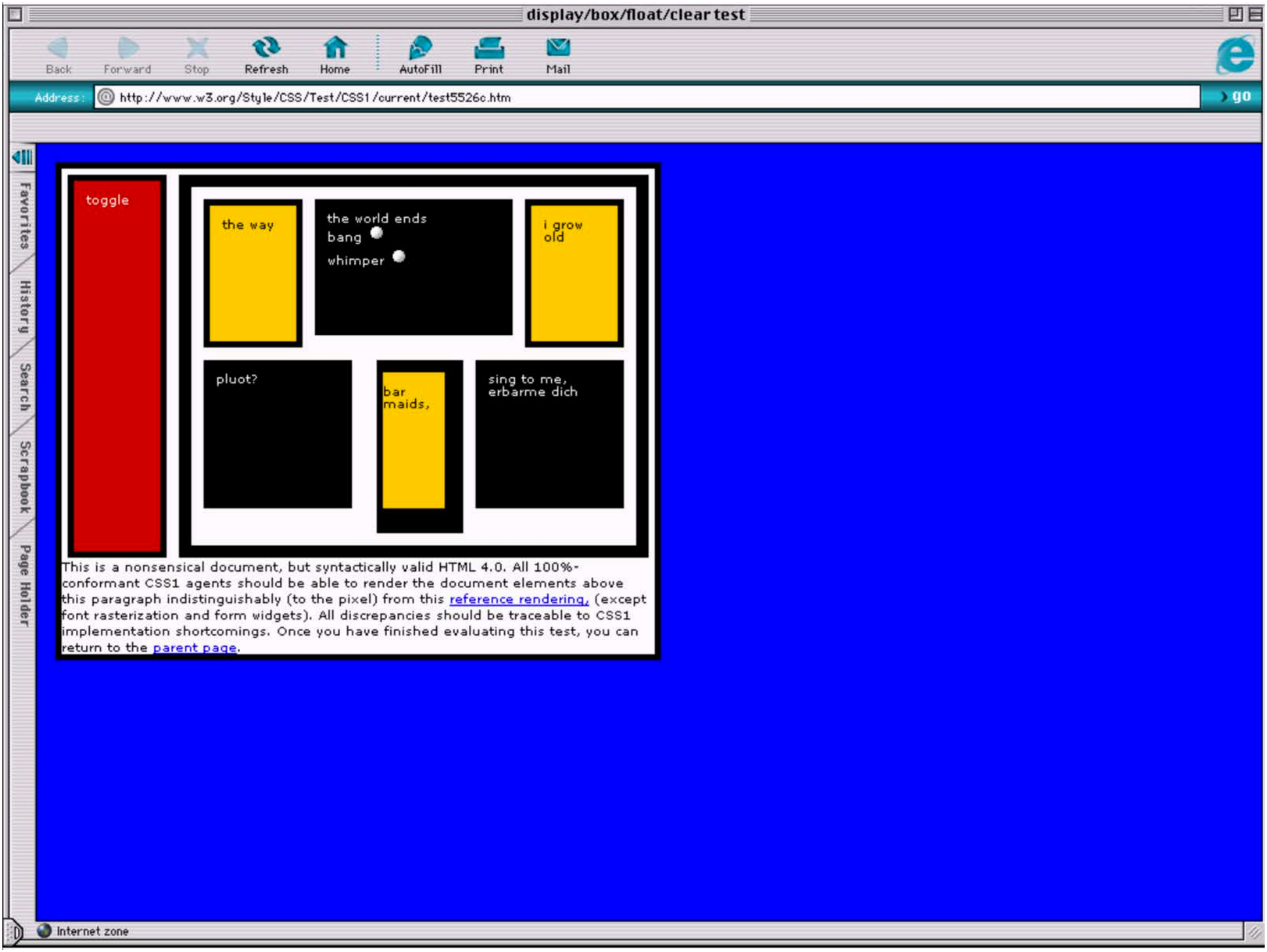
toggle

- the way [illegible]
- i grow old
- pluot?

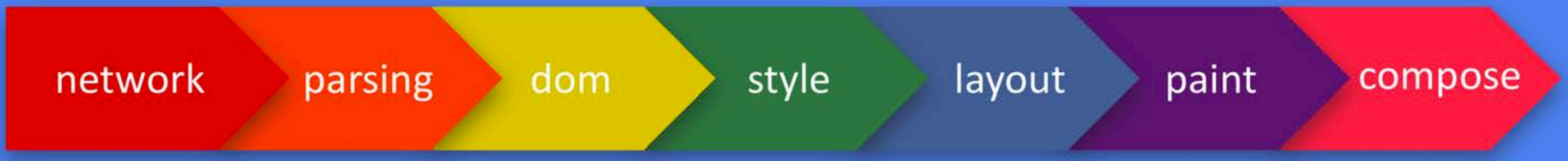
bar maids,

sing to me, erbarme dich

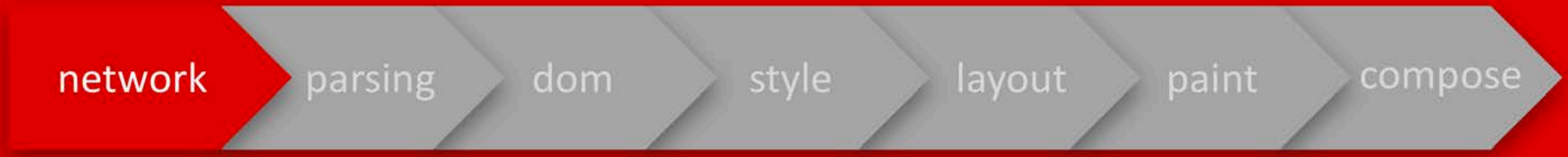
This is a nonsensical document, but syntactically valid HTML 4.0. All 100%-conformant CSS1 agents should be able to render the document elements above this paragraph indistinguishably (to the pixel) from this reference rendering (except font rasterization and form widgets). All discrepancies should be traceable to CSS1 implementation shortcomings. Once you have finished evaluating this test, you can return to the parent page.



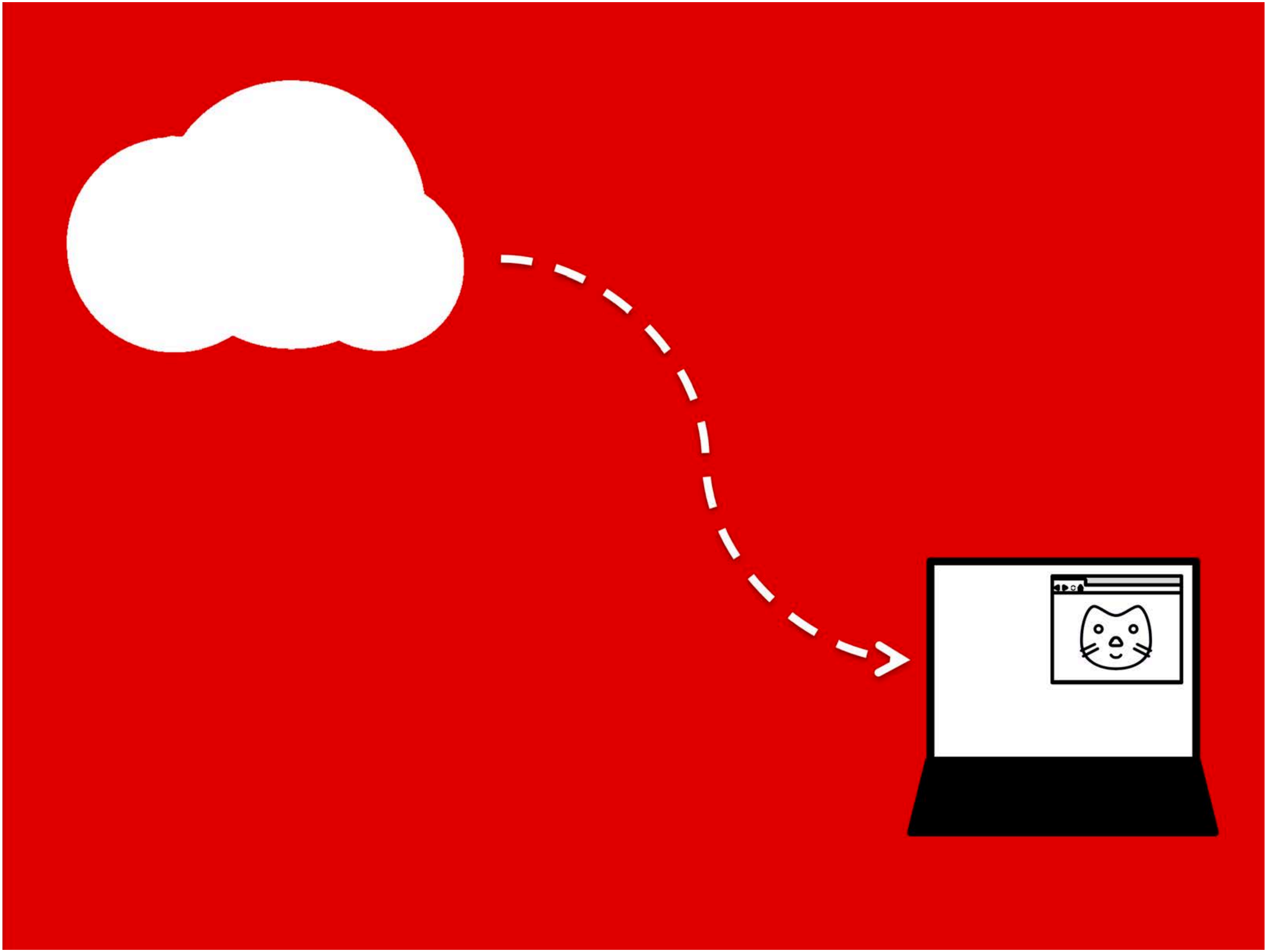
This is a nonsensical document, but syntactically valid HTML 4.0. All 100%-conformant CSS1 agents should be able to render the document elements above this paragraph indistinguishably (to the pixel) from this reference rendering, (except font rasterization and form widgets). All discrepancies should be traceable to CSS1 implementation shortcomings. Once you have finished evaluating this test, you can return to the parent page.

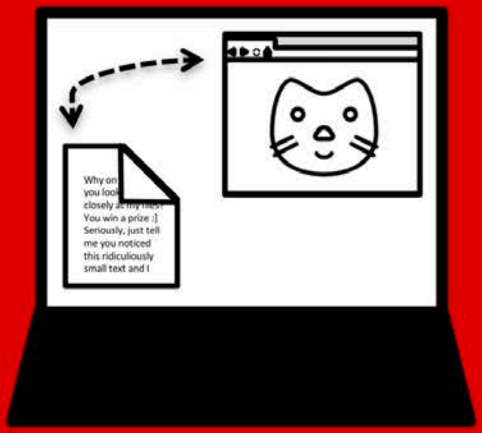


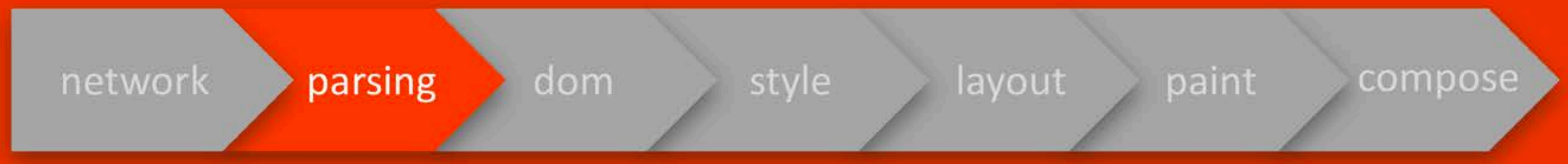
network parsing dom style layout paint compose



network parsing dom style layout paint compose




Why on
you look
closely at my files?
You win a prize :]
Seriously, just tell
me you noticed
this ridiculiously
small text and I

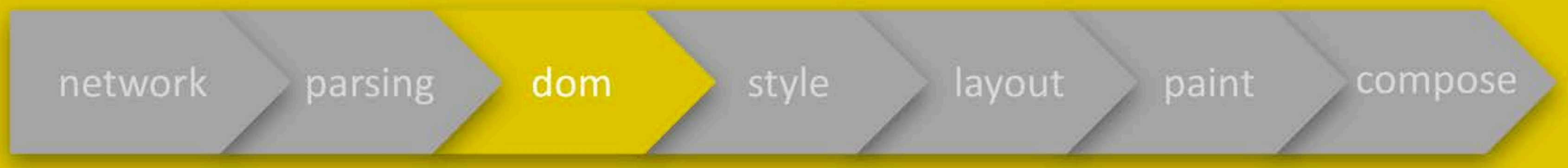


network → parsing → dom → style → layout → paint → compose

```
var sum = 1 + 2
```

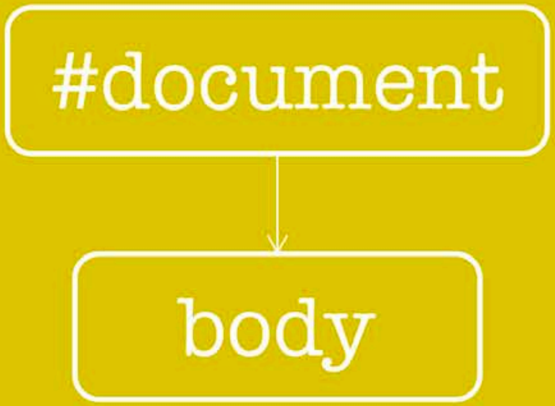
“hey, that’s a keyword!”

keyword	identifier	assignment	int	addition	int
var	sum	=	1	+	2



network parsing dom style layout paint compose

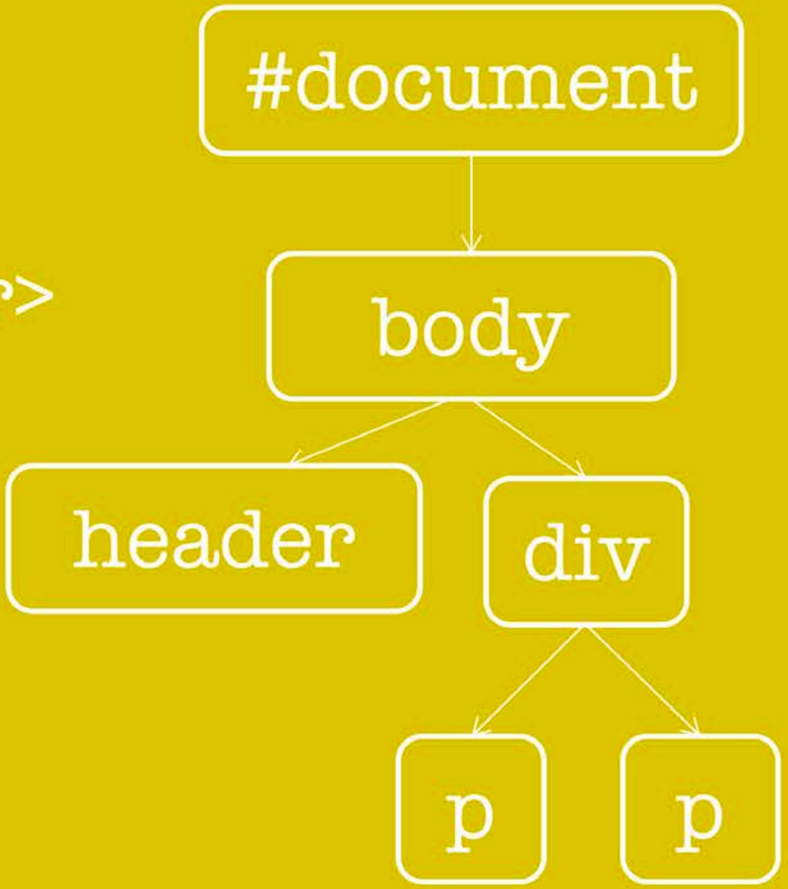
```
<!DOCTYPE html>
<body>
 <header>hello</header>
 <div>
  <p>world</p>
  <p>:]</p>
 </div>
</body>
```

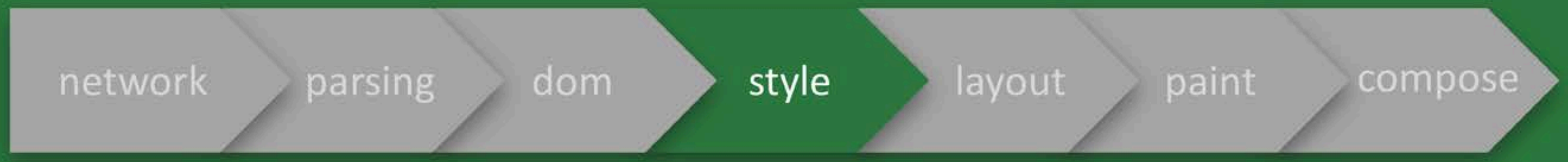


#document

body

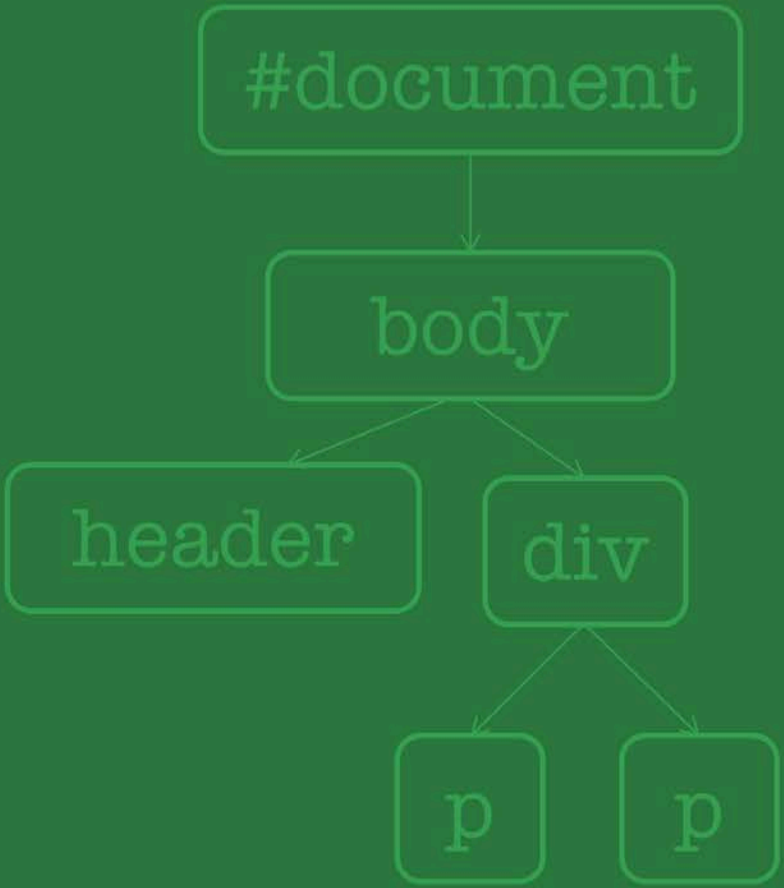
```
<!DOCTYPE html>
<body>
 <header>hello</header>
 <div>
  <p>world</p>
  <p>:]</p>
 </div>
</body>
```



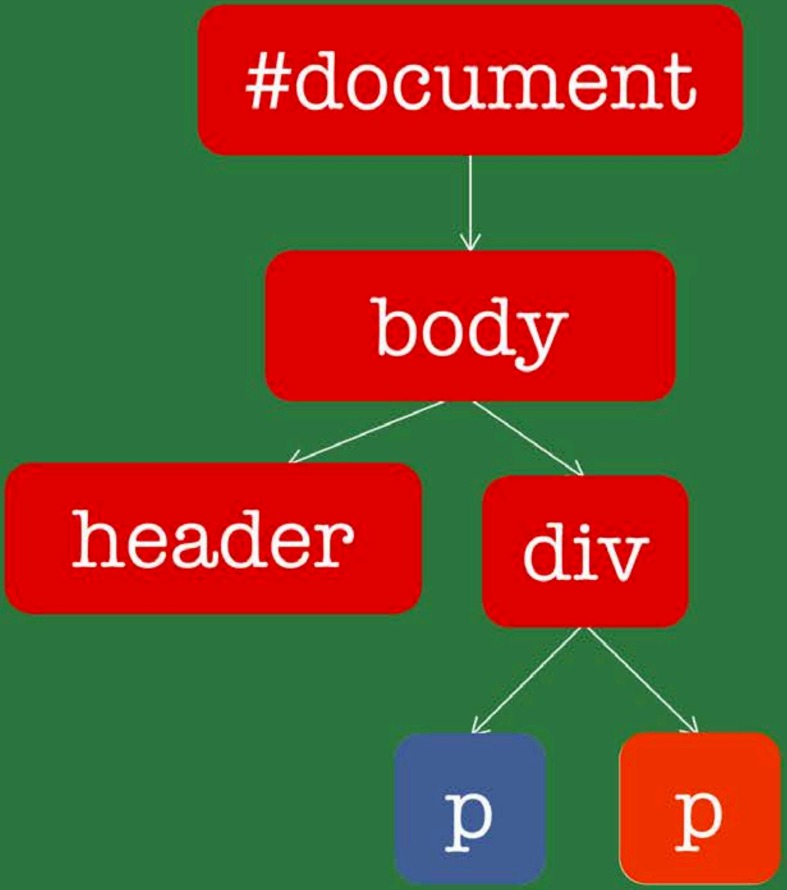


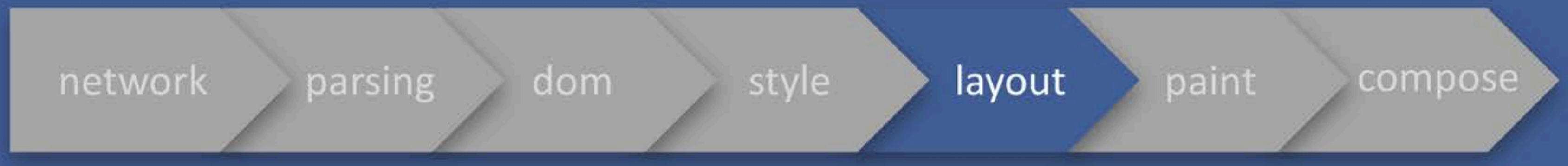
network parsing dom style layout paint compose

```
*  {
 background: red
}

div p {
 background: blue
}

p:last-of-type {
 background: orange
}
```



```
* {
 ~~background: red~~
}

div p {
 ~~background: blue~~
}

p:last-of-type {
 background: orange
}
```




network

parsing

dom

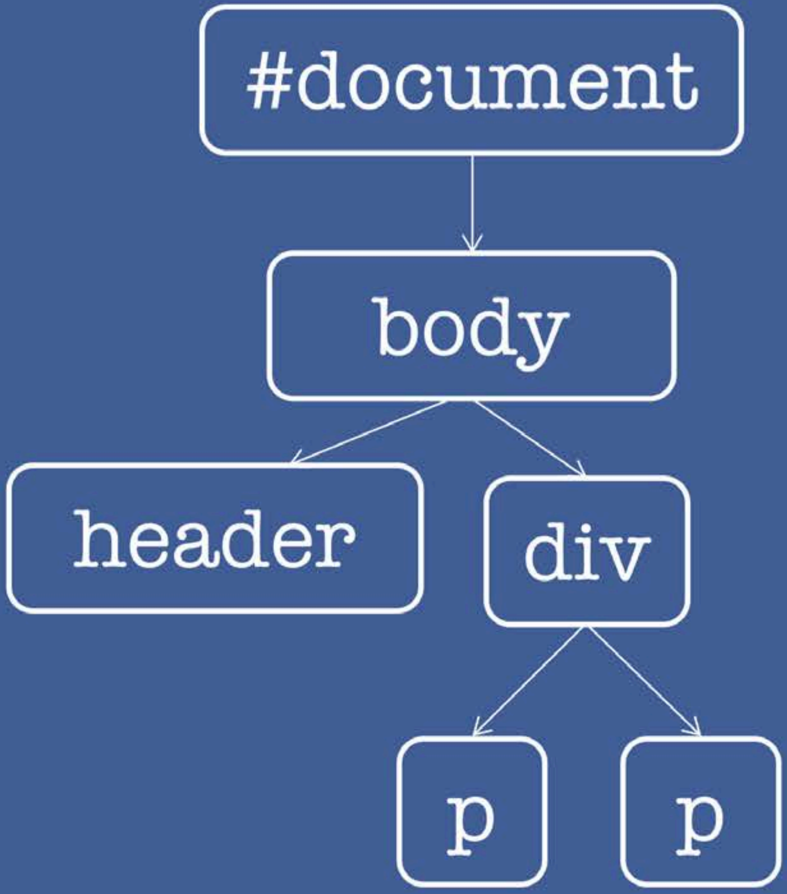
style

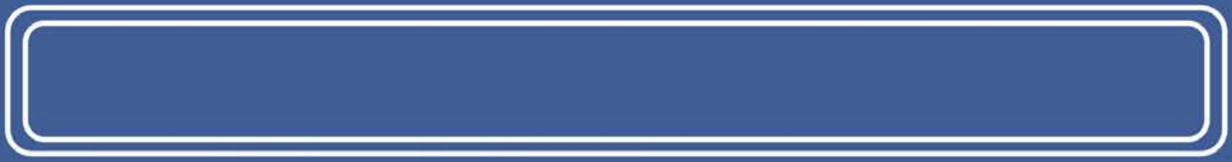
layout

paint

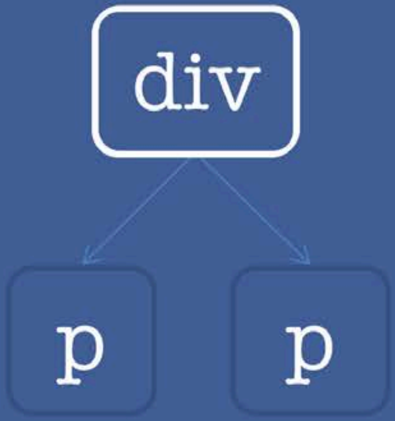
compose

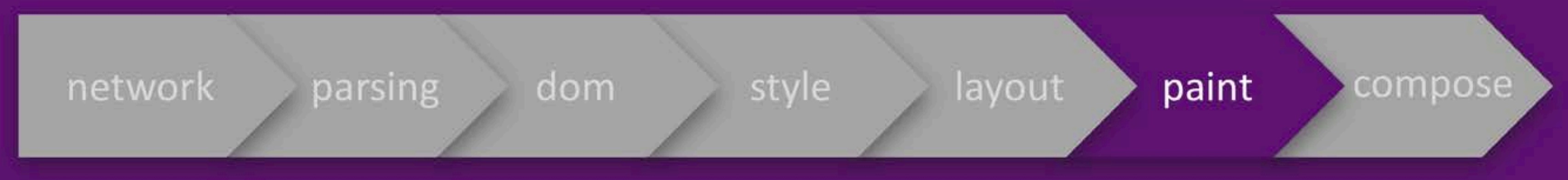
```
body{
 width: 250px
}

div {
 width: 10em;
}

p {
 width: 50%
}
```





```
div {
  width: 10em;
}

p {
  width: 50%
}
```





network

parsing

dom

style

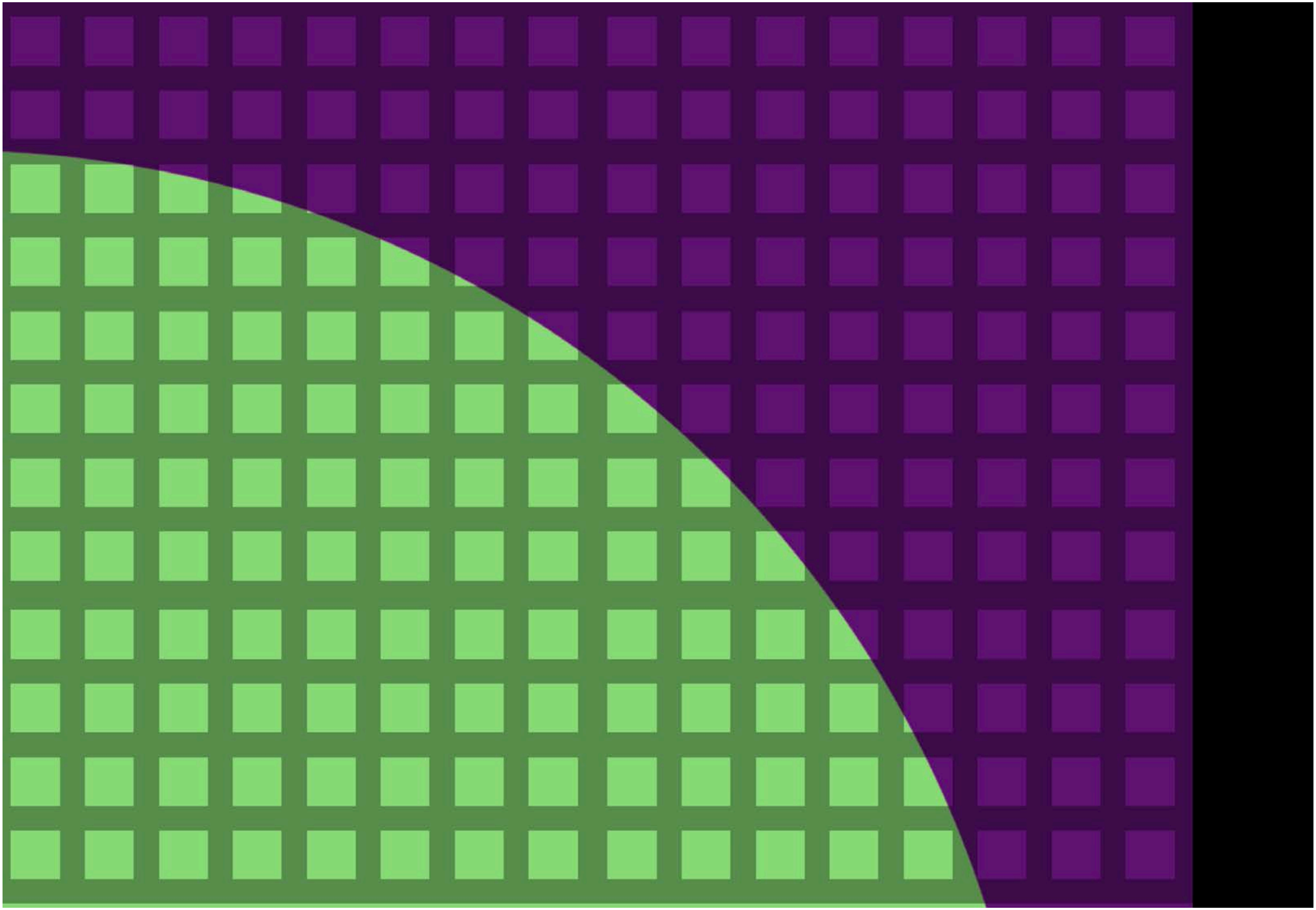
layout

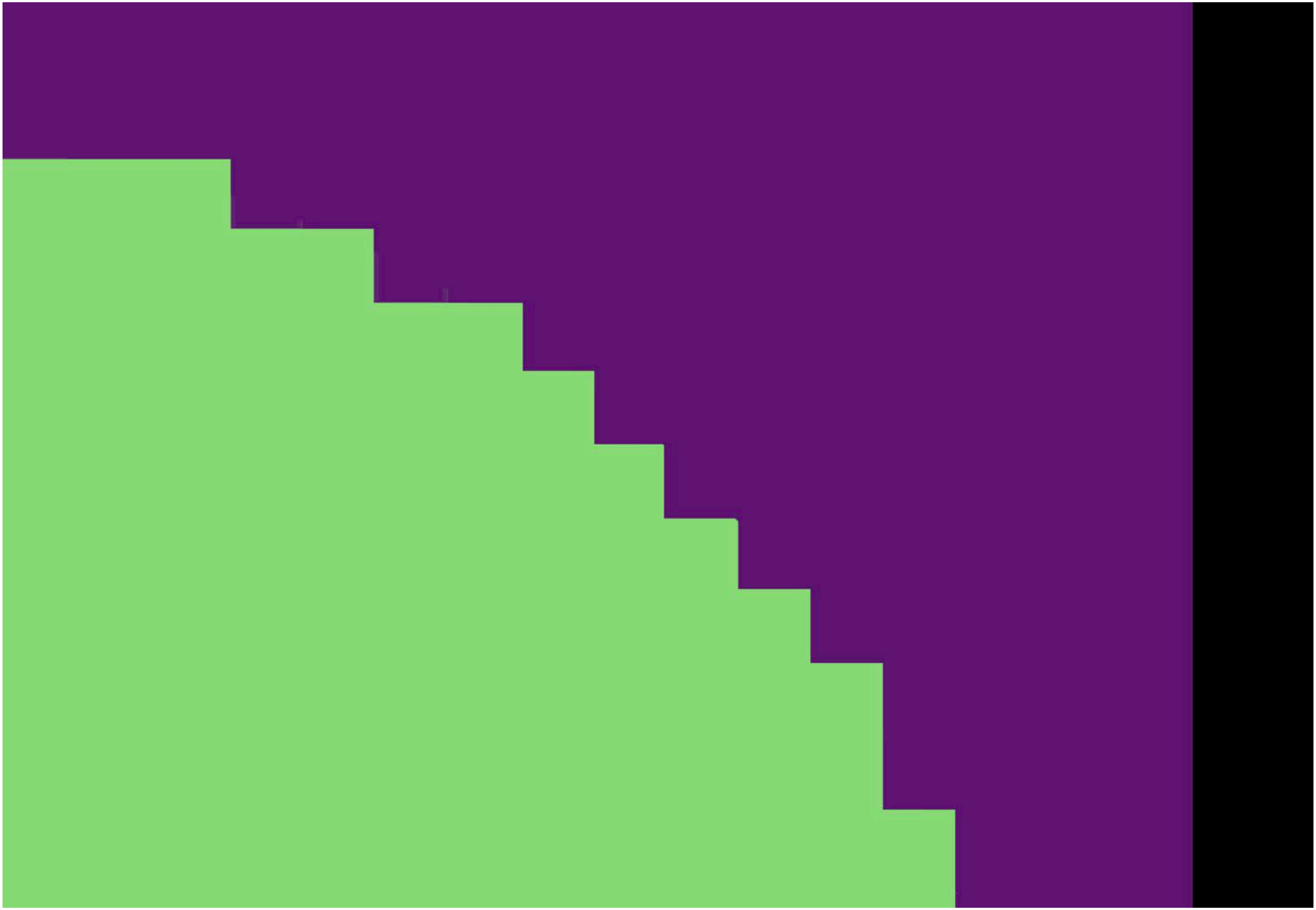
paint

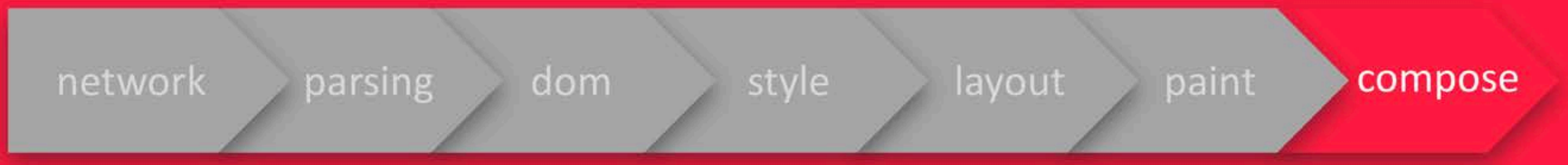
compose

```
div {
 left: -192
 width: 384px;
 height: 384px;
 border-raidus: 50%;
 background: rgb(142, 222, 123);
}
```

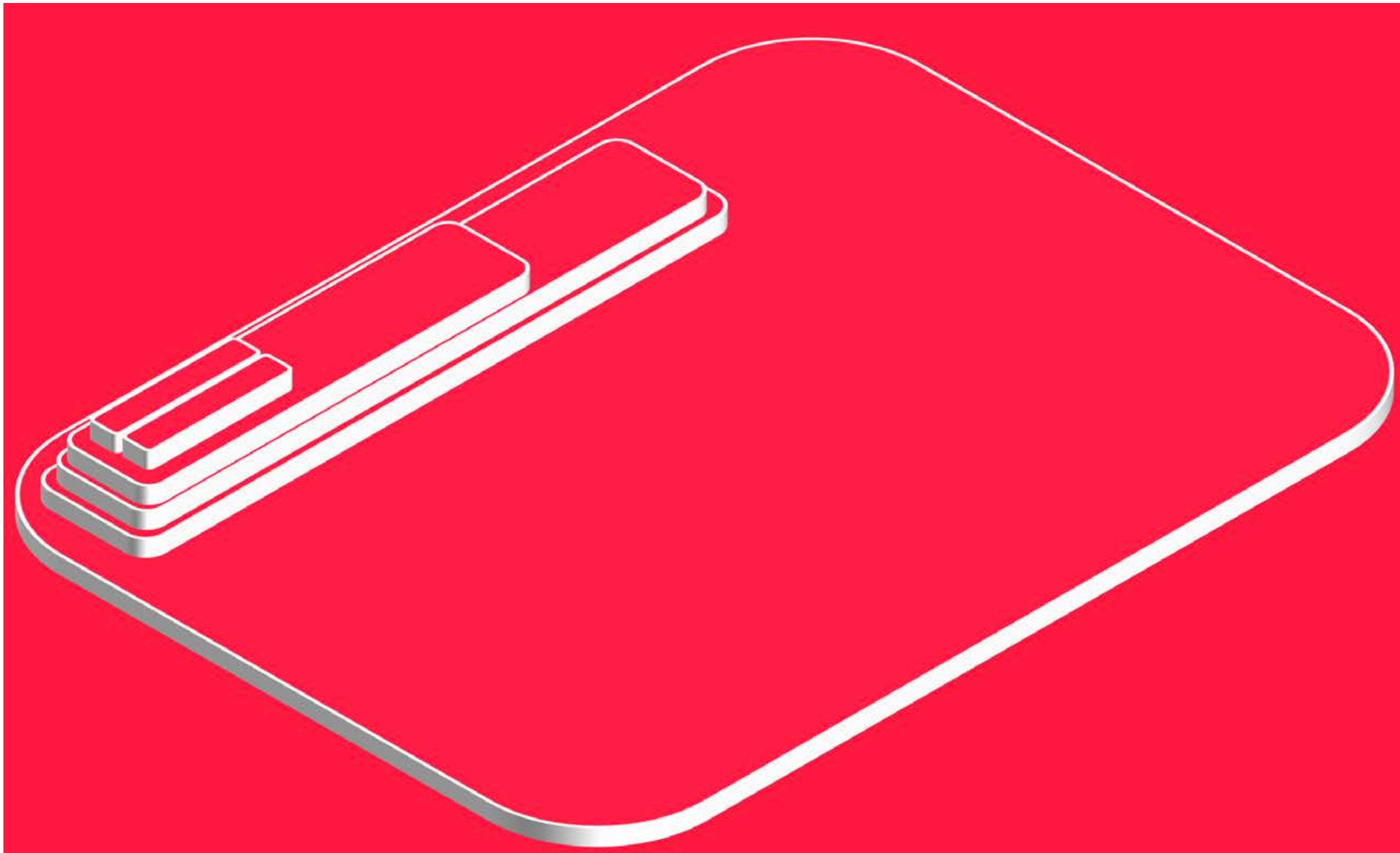
0101
011111011
0101101111010
1110101010010
101001010010111010
10100101001010110101001101010

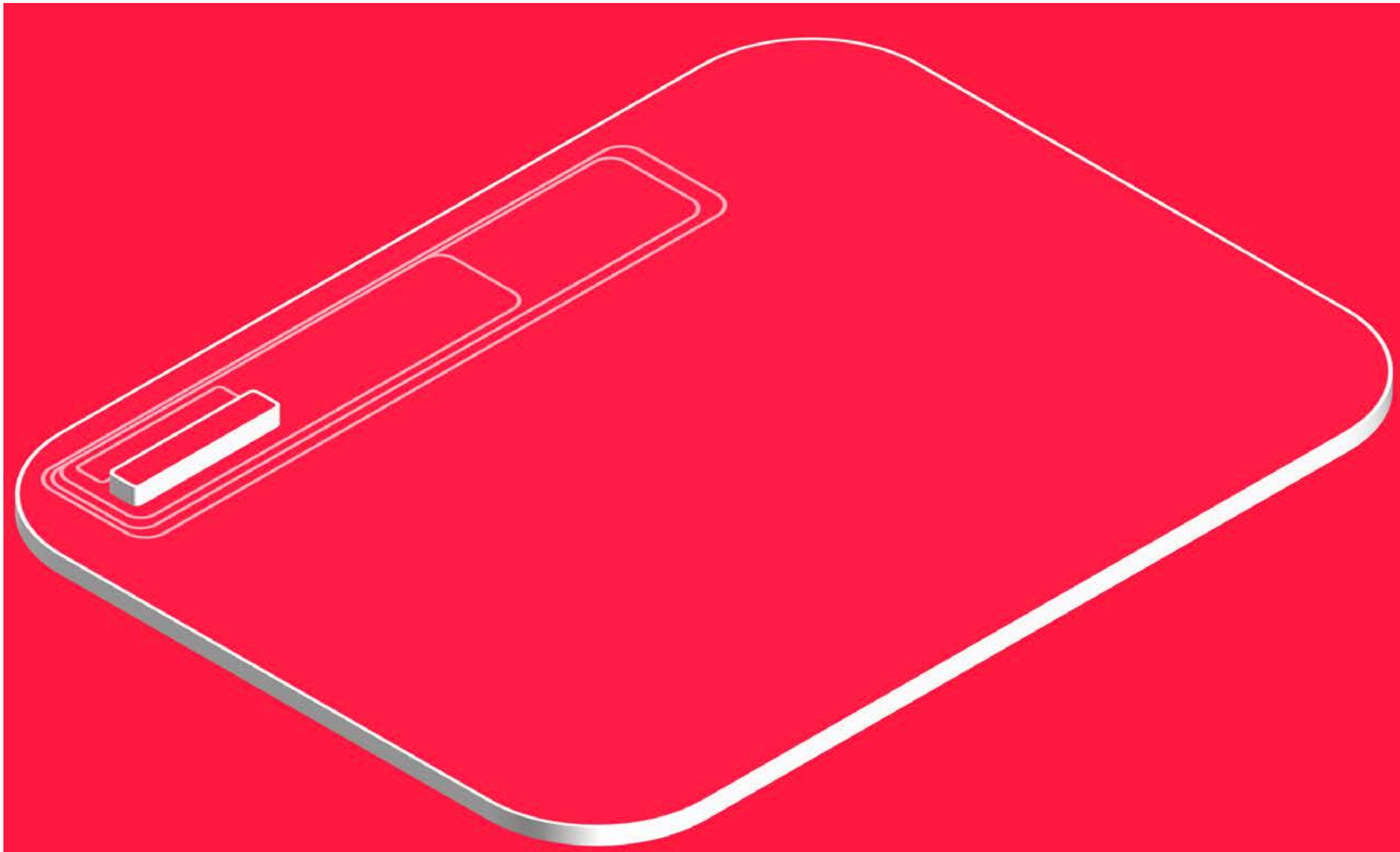




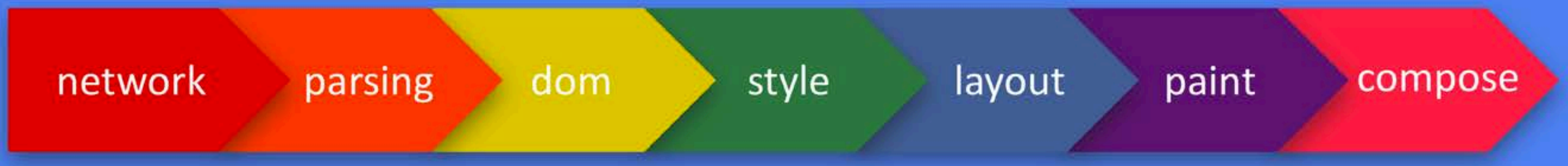


network parsing dom style layout paint compose

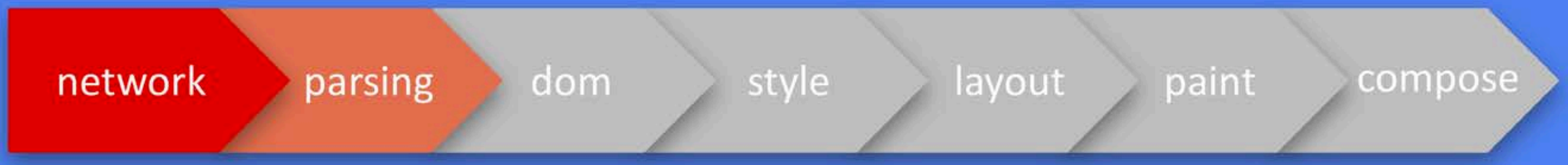


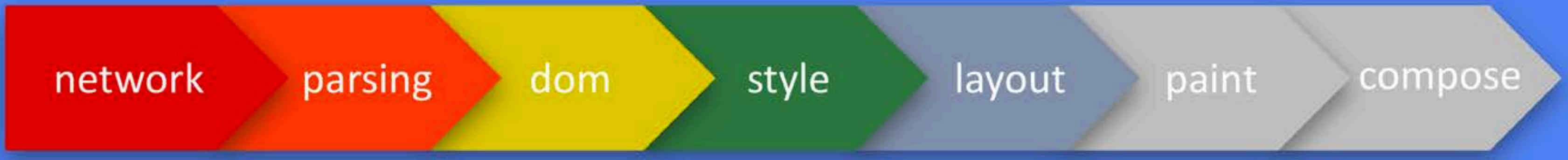


```
p:last-of-type {
  transform: translate3d(0,0,0)
}
```

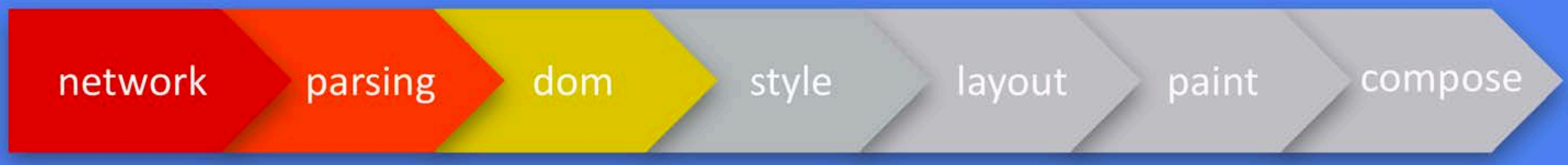


network parsing dom style layout paint compose

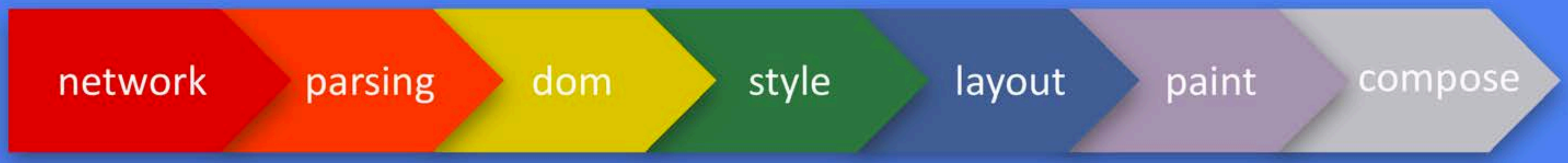

network
parsing
dom
style
layout
paint
compose



network parsing dom style layout paint compose



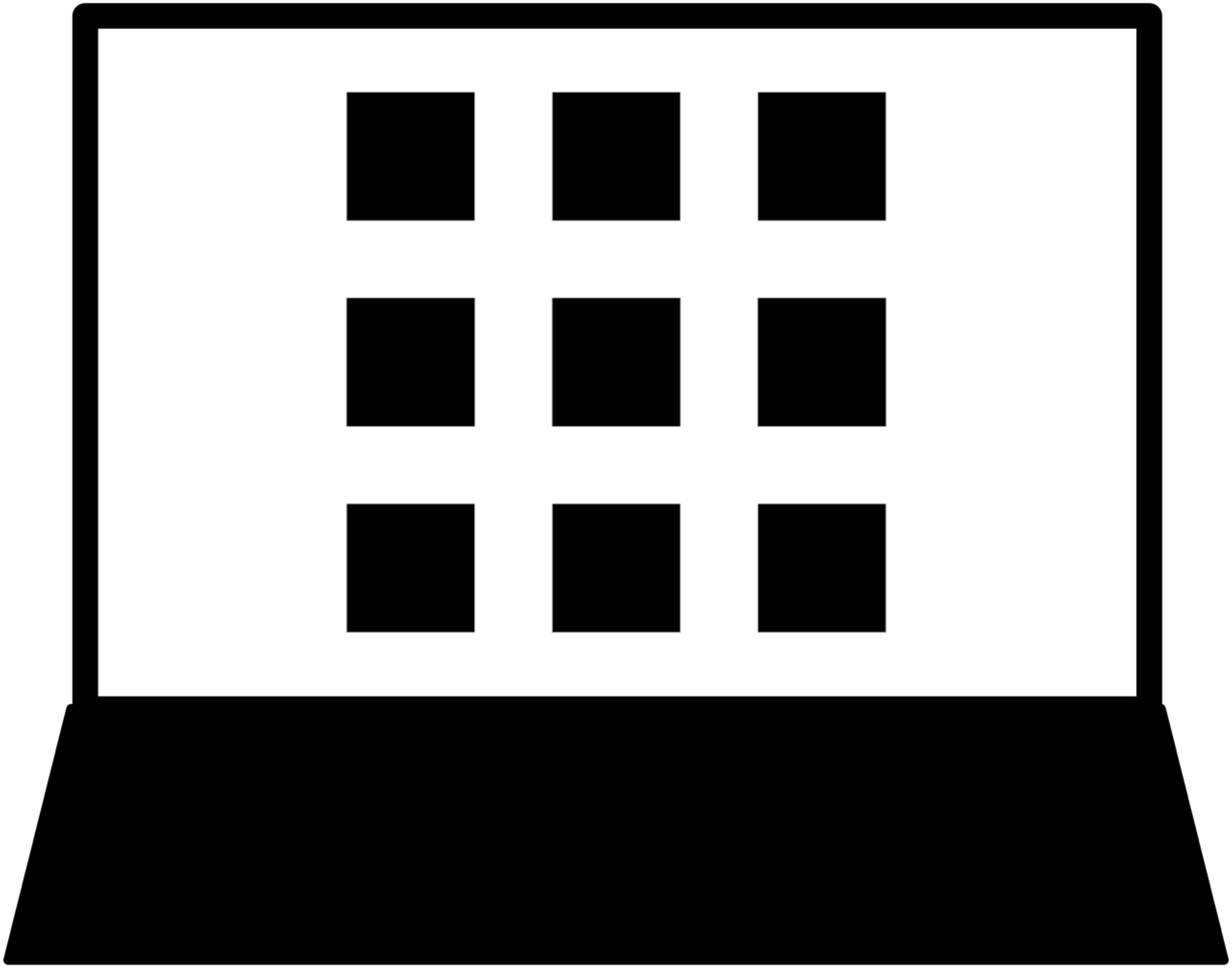
JS

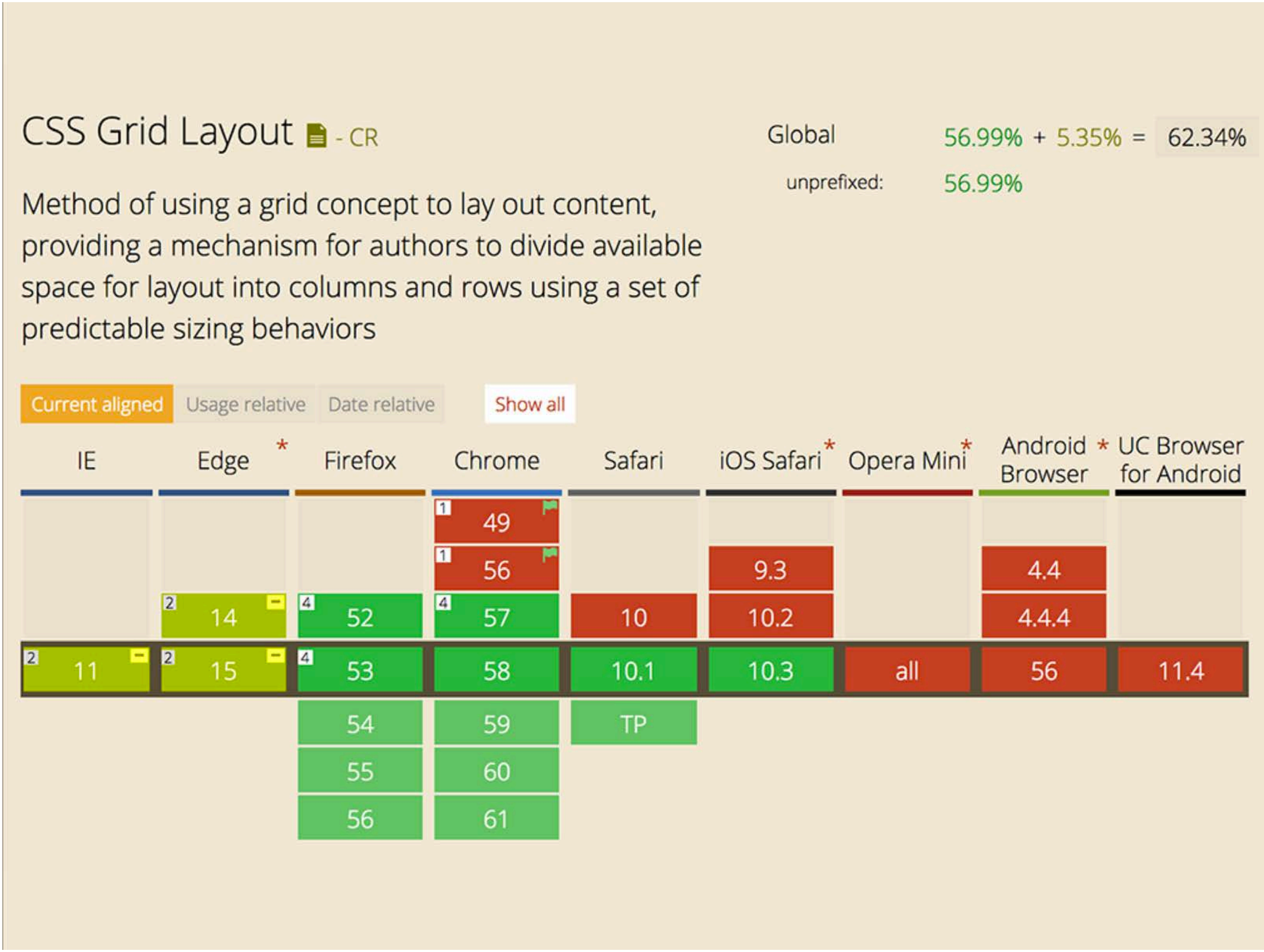


network parsing dom style layout paint compose

JS

css grid





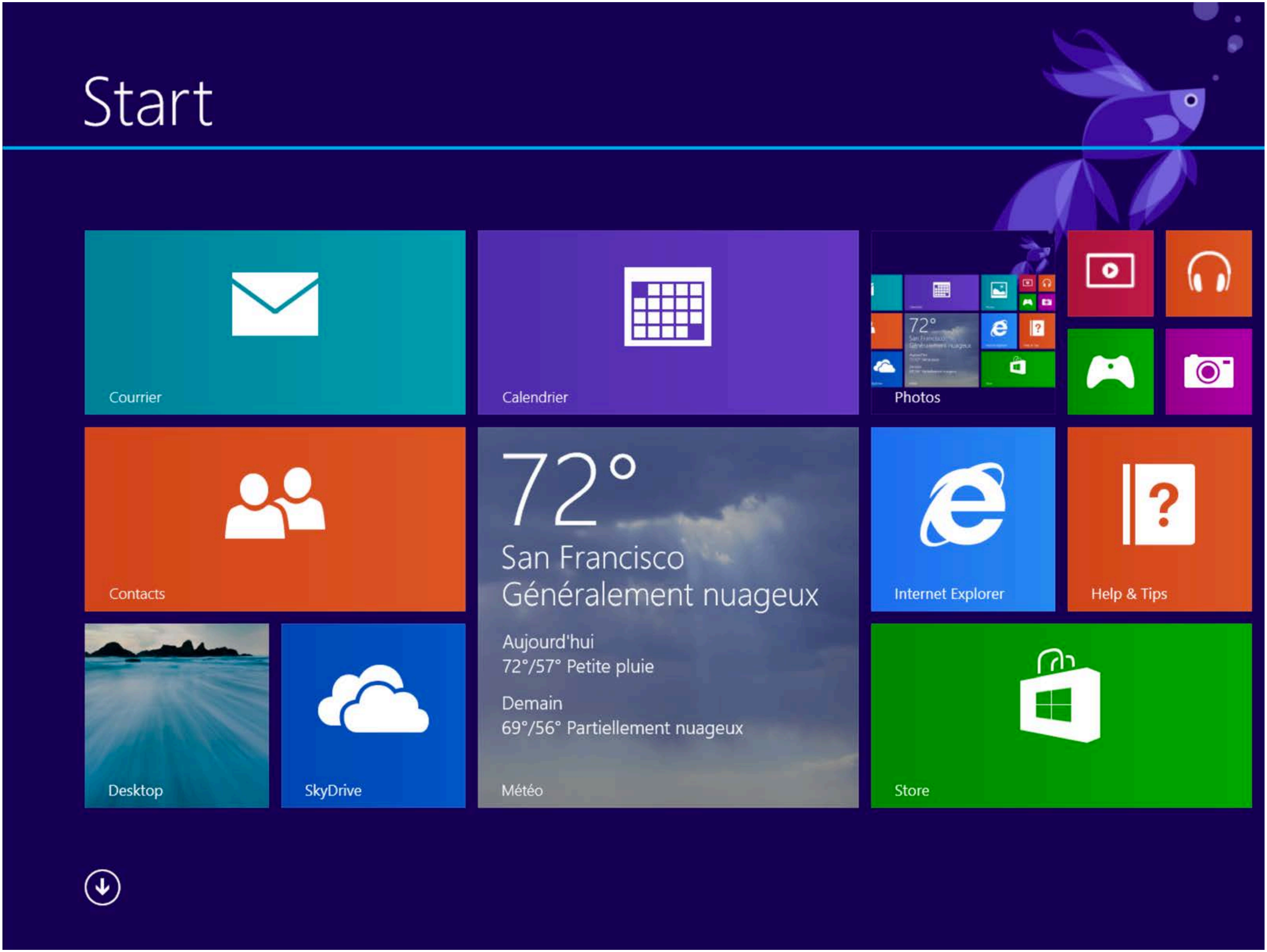
CSS Grid Layout - CR

Global 56.99% + 5.35% = 62.34%

unprefixed: 56.99%

Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for layout into columns and rows using a set of predictable sizing behaviors

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Android Browser *	UC Browser for Android
			[1] 49					
			[1] 56		9.3		4.4	
	[2] 14	[4] 52	[4] 57	10	10.2		4.4.4	
[2] 11	[2] 15	[4] 53	58	10.1	10.3	all	56	11.4
		54	59	TP				
		55	60					
		56	61					



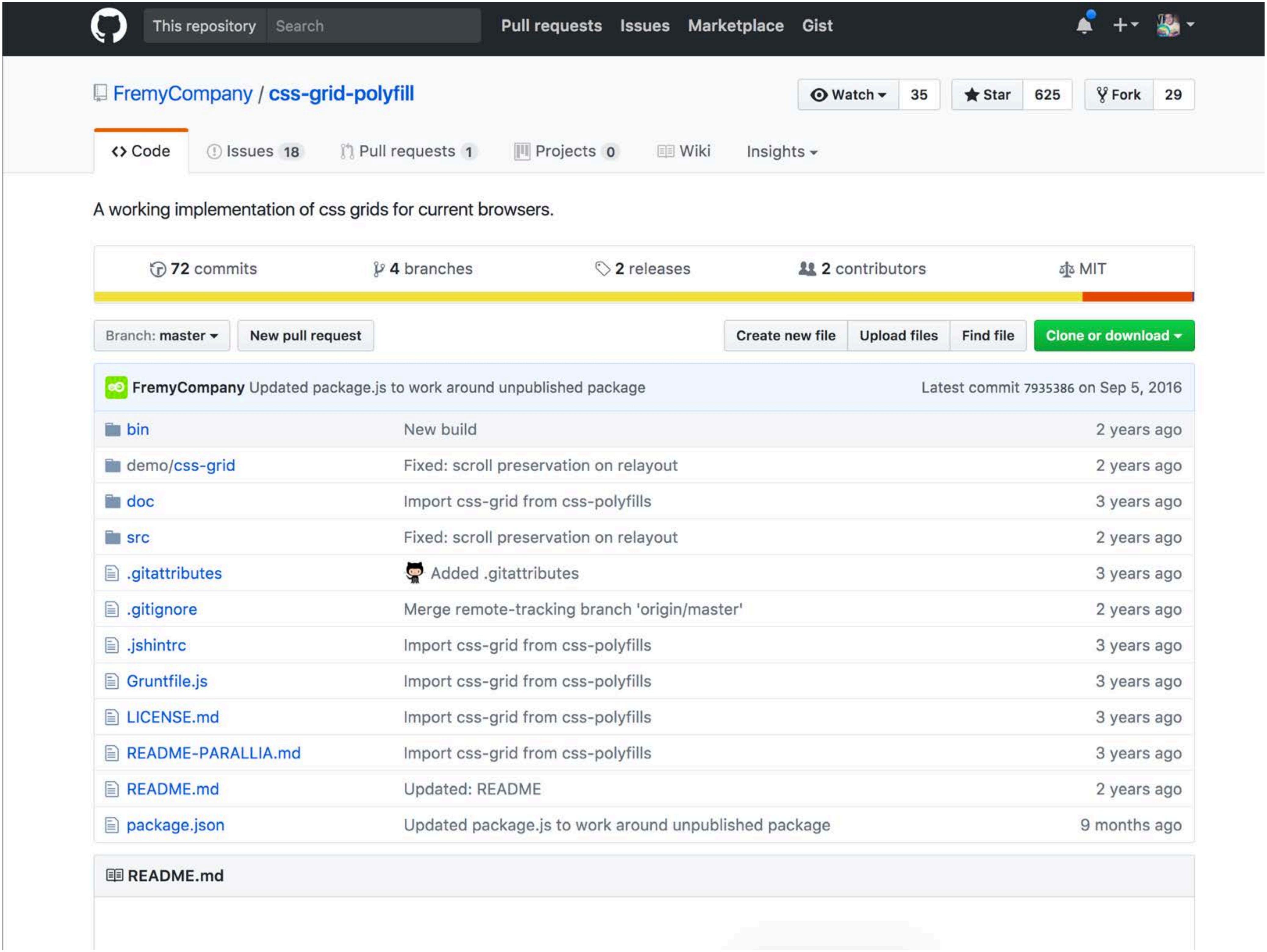
72°
San Francisco
Généralement nuageux

Aujourd'hui
72°/57° Petite pluie

Demain
69°/56° Partiellement nuageux

5

years



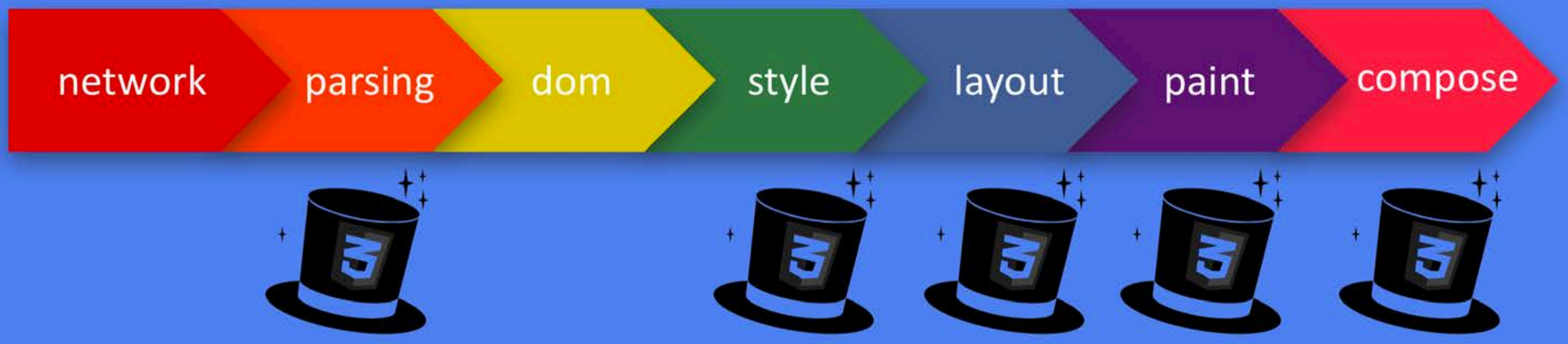
FremyCompany / css-grid-polyfill

A working implementation of css grids for current browsers.

FremyCompany Updated package.js to work around unpublished package — Latest commit 7935386 on Sep 5, 2016

bin	New build	2 years ago
demo/css-grid	Fixed: scroll preservation on relayout	2 years ago
doc	Import css-grid from css-polyfills	3 years ago
src	Fixed: scroll preservation on relayout	2 years ago
.gitattributes	Added .gitattributes	3 years ago
.gitignore	Merge remote-tracking branch 'origin/master'	2 years ago
.jshintrc	Import css-grid from css-polyfills	3 years ago
Gruntfile.js	Import css-grid from css-polyfills	3 years ago
LICENSE.md	Import css-grid from css-polyfills	3 years ago
README-PARALLIA.md	Import css-grid from css-polyfills	3 years ago
README.md	Updated: README	2 years ago
package.json	Updated package.js to work around unpublished package	9 months ago

README.md



JS

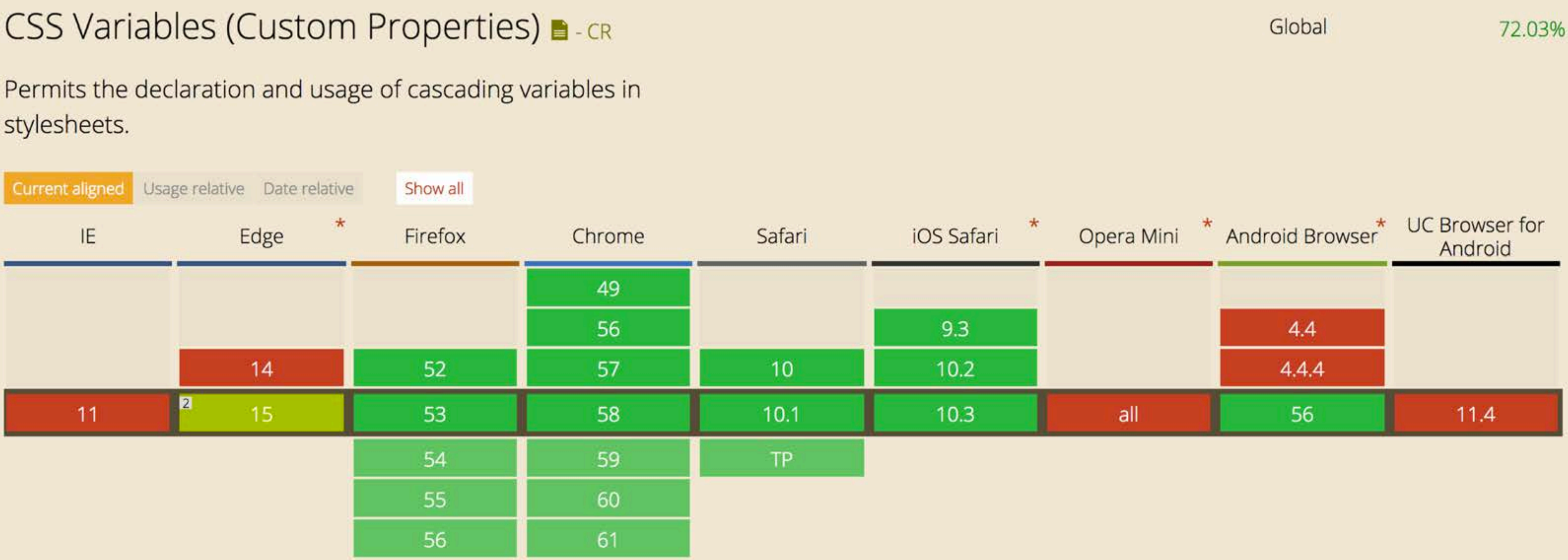
CSS Properties and Values API
CSSOM

PaintWorklet
LayoutWorklet
AnimationWorklet

Properties and Values API

CSS Custom Properties

CSS Custom Properties



CSS Variables (Custom Properties) - CR

Global 72.03%

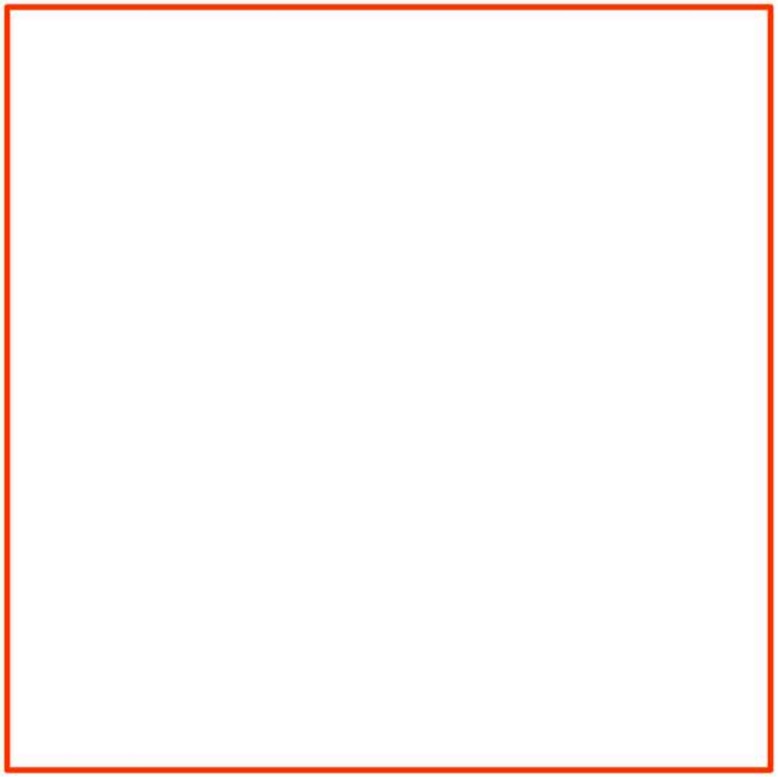
Permits the declaration and usage of cascading variables in stylesheets.

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Android Browser *	UC Browser for Android
			49					
			56		9.3		4.4	
	14	52	57	10	10.2		4.4.4	
11	[2] 15	53	58	10.1	10.3	all	56	11.4
		54	59	TP				
		55	60					
		56	61					

```css
:root {
  --distance: 400px;
}

div {
  width: var(--distance);
  height: var(--distance);
  background: cornflowerblue;
  transition: width: 1s, height: 1s;
}

div:hover {
  --distance: 150px;
}
```

```css
:root {
  --distance: 400px;
}

div {
  width: var(--distance);
  height: var(--distance);
  background: cornflowerblue;
  transition: width: 1s, height: 1s;
}

div:hover {
  --distance: 150px;
}
```

```
:root {
  --distance: 400px;
}

div {
  width: var(--distance);
  height: var(--distance);
  background: cornflowerblue;
  transition: width: 1s, height: 1s;
}

div:hover {
  --distance: 150px;
}
```

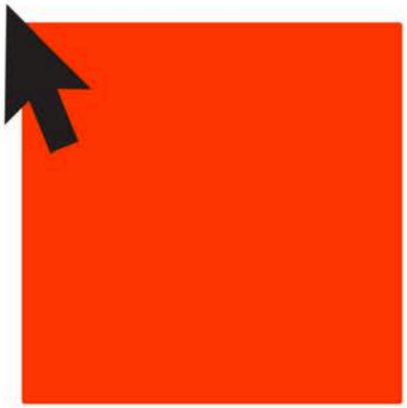


```
:root {
  --distance: 400px;
}

div {
  width: var(--distance);
  height: var(--distance);
  background: cornflowerblue;
  transition: width: 1s, height: 1s;
}

div:hover {
  --distance: 150px;
}
```

```css
:root {
  --distance: 400px;
}

div {
  width: var(--distance);
  height: var(--distance);
  background: cornflowerblue;
  transition: width: 1s, height: 1s;
}

div:hover {
  --distance: 150px;
}
```

```
:root {
  --distance: 400px;
}

div {
  width: var(--distance);
  height: var(--distance);
  background: cornflowerblue;
  transition: --distance: 1s;
}

div:hover {
  --distance: 150px;
}
```

```css
:root {
  --distance: 400px;
}

div {
  width: var(--distance);
  height: var(--distance);
  background: cornflowerblue;
  transition: --distance: 1s;
}

div:hover {
  --distance: 150px;
}
```



```
:root {
  --distance: 400px;
}

div {
  width: var(--distance);
  height: var(--distance);
  background: cornflowerbl
  transition: --distance: 1s;
}

div:hover {
  --distance: 150px;
}
```

You Can’t Animate Strings

Properties and Values API

```
CSS.registerProperty({
  name: "--distance",
  syntax: "<length>",
  initialValue: ’0'
})
```

```
:root {
  --distance: 400px;
}

div {
  width: var(--distance);
  height: var(--distance);
  background: cornflowerblue;
  transition: --distance: 1s;
}

div:hover {
  --distance: 150px;
}
```

```
CSS.registerProperty({
  name: "--distance",
  syntax: "<length>",
  initialValue: ’0'
})
```

```
:root {
  --distance: 400px;
}

div {
  width: var(--distance);
  height: var(--distance);
  background: cornflowerblue;
  transition: --distance: 1s;
}

div:hover {
  --distance: 150px;
}
```

required: any string

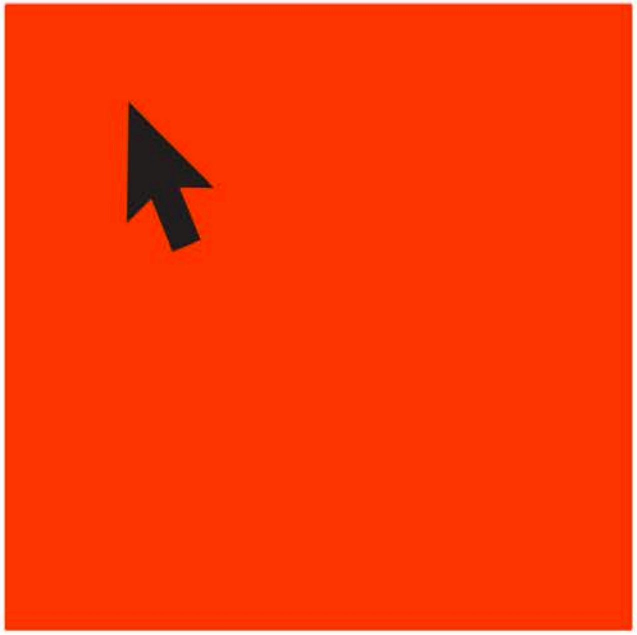
```
CSS.registerProperty({
  name: "--distance",
  syntax: "<length>",
  initialValue: ’0'
})
```

```
:root {
  --distance: 400px;
}

div {
  width: var(--distance);
  height: var(--distance);
  background: cornflowerblue;
  transition: --distance: 1s;
}

div:hover {
  --distance: 150px;
}
```

optional: <length>
<number>
<percentage>
<color>
<image>
<url>
<integer>
<angle>
<time>
<resolution>
<ident>
*

```
CSS.registerProperty({
  name: "--distance",
  syntax: "<length>",
  initialValue: ’0'
})
```

```
:root {
  --distance: 400px;
}

div {
  width: var(--distance);
  height: var(--distance);
  background: cornflowerblue;
  transition: --distance: 1s;
}

div:hover {
  --distance: 150px;
}
```



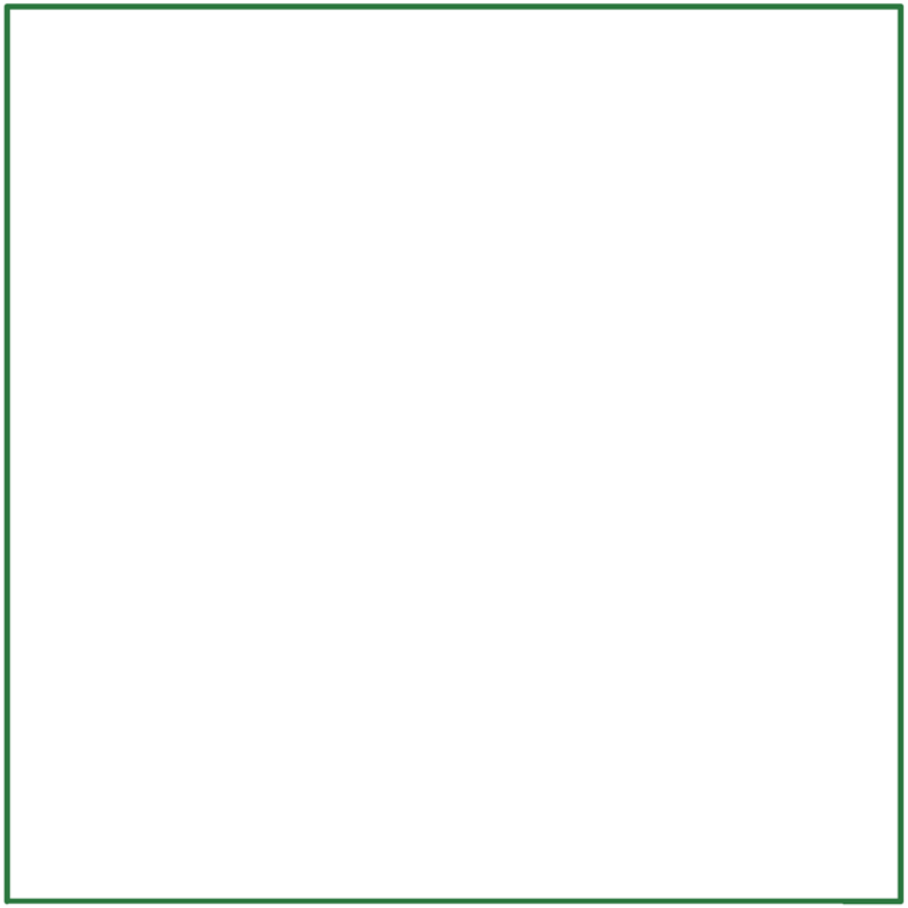
```
CSS.registerProperty({
  name: "--distance",
  syntax: "<length>",
  initialValue: ’0'
})
```

```
:root {
  --distance: 400px;
}

div {
  width: var(--distance);
  height: var(--distance);
  background: cornflowerblue;
  transition: --distance: 1s;
}

div:hover {
  --distance: 150px;
}
```

CSSOM

CSS Object Model

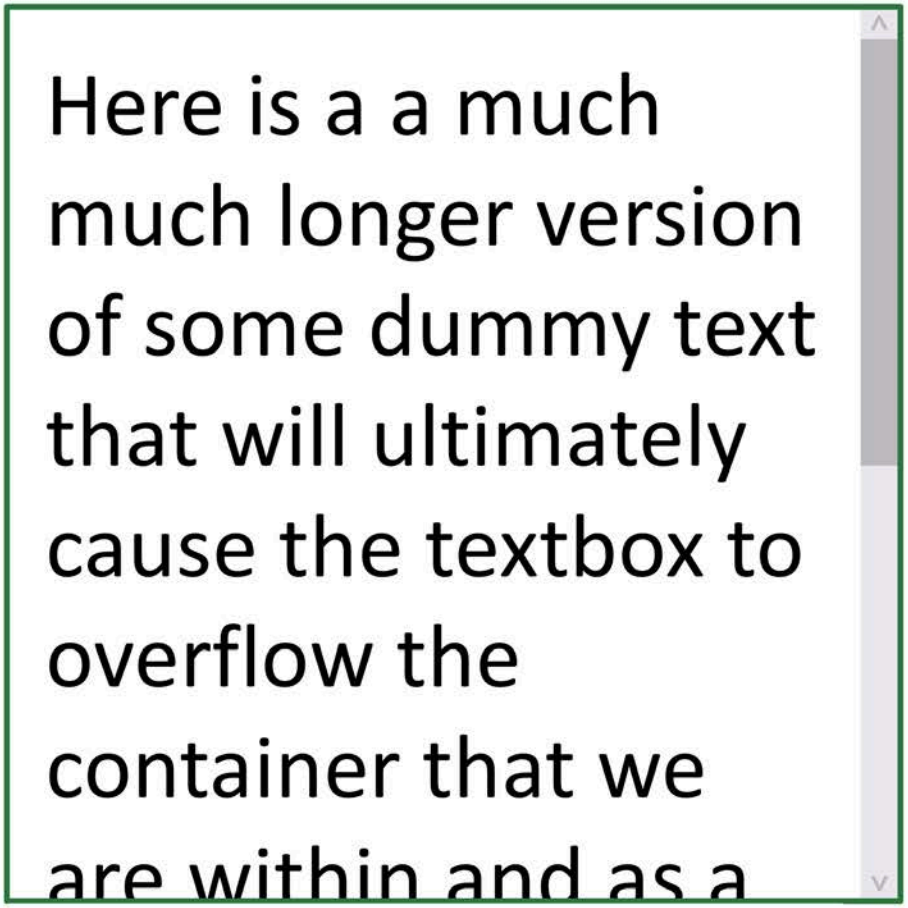
CSS Typed Object Model



Here is a some dummy text

Here is a some dummy text

```
window.getComputedStyle('#elem').width == '200px';
```



Here is a a much much longer version of some dummy text that will ultimately cause the textbox to overflow the container that we are within and as a

```
window.getComputedStyle('#elem').width == ’187px';
```

```
function getInnerWidth (elem) {
  var style  = window.getComputedStyle(elem);
  var width  = elem.offsetWidth;

  var right  = parseFloat(style.paddingRight);
  var left   = parseFloat(style.paddingLeft);
  var bRight = parseFloat(style.borderRightWidth);
  var bLeft  = parseFloat(style.borderLeftWidth);

  return width - right - left - bRight - bLeft;
};
```

```
function getInnerWidth (elem) {
  var style  = window.getComputedStyle(elem);
  var width  = elem.offsetWidth;

  var right  = parseFloat(style.paddingRight);
  var left   = parseFloat(style.paddingLeft);
  var bRight = parseFloat(style.borderRightWidth);
  var bLeft  = parseFloat(style.borderLeftWidth);

  return width - right - left - bRight - bLeft;
};
```

```
function getInnerWidth (elem) {
  var style  = window.getComputedStyleMap(elem);
  var width  = elem.offsetWidth;

  var right  = style.paddingRight.value;
  var left   = style.paddingLeft.value;
  var bRight = style.borderRightWidth.value;
  var bLeft  = style.borderLeftWidth.value;

  return width - right - left - bRight - bLeft;
};
```

```
function getInnerWidth (elem) {
  var style  = window.getComputedStyleMap(elem);
  var width  = elem.offsetWidth;

  var right  = style.paddingRight.value;
  var left   = style.paddingLeft.value;
  var bRight = style.borderRightWidth.value;
  var bLeft  = style.borderLeftWidth.value;

  return width - right - left - bRight - bLeft;
};
```

```
function getInnerWidth (elem) {
  var style  = window.getComputedStyleMap(elem);
  var width  = elem.offsetWidth;

  var right  = style.paddingRight.value;
  var left   = style.paddingLeft.value;
  var bRight = style.borderRightWidth.value;
  var bLeft  = style.borderLeftWidth.value;

  return width - right - left - bRight - bLeft;
};
```

```
getComputedStyleMap(elem) // to query any current value

elem.styleMap             // for styles that have been
                          // set via CSS explicitly
```

```
var style = document.querySelector('#elem').styleMap

style.get('width').value == '200'
style.get('width').unit == 'px'
```

```
var style = document.querySelector('#elem').styleMap

style.get('width').value == '200'
style.get('width').unit == ’vmax'
```

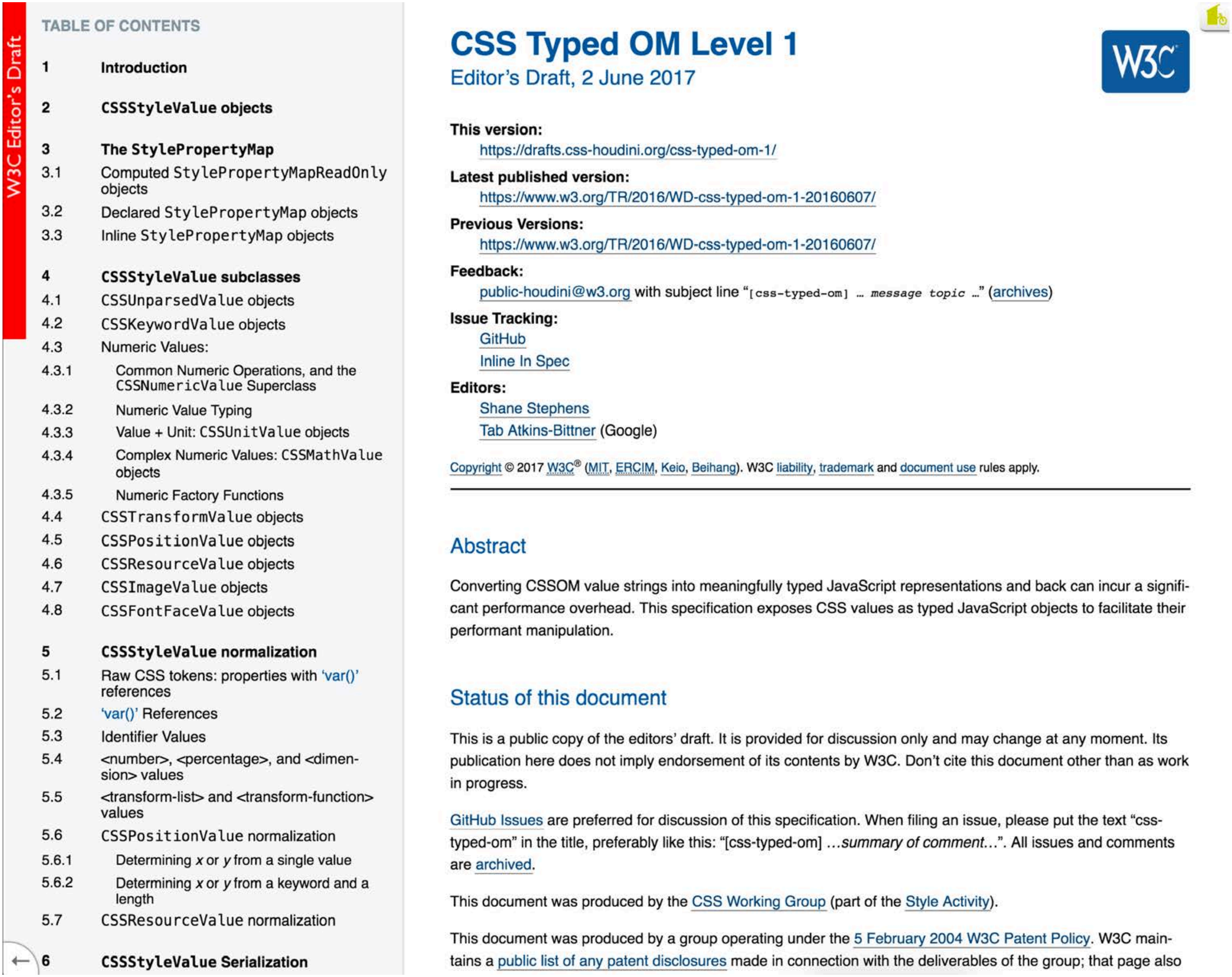
```
var style = document.querySelector('#elem').styleMap

style.set('width', new CSSUnitValue(50, 'em'))
```

```
#elem {
  background-position: center bottom 10px;
}


var style = document.querySelector('#elem').styleMap

var x = style.get(’background-position').x
  x.value == 50
  x.unit == ’percent'
  // CSSUnitValue(50, "percent")
```

```
#elem {
  background-position: center bottom 10px;
}


var style = document.querySelector('#elem').styleMap

var x = style.get(’background-position').x
  x.value == 50
  x.unit == ’percent'
  // CSSUnitValue(50, "percent")

var y = style.get(’background-position').y
  x.px == -10
  x.percent == 100
  x.unit == undefined

  // CSSCalcValue({percent: 100, px: -10})
```

it’s about efficiency,
not ease of use

```
CSSUnparsedValue
CSSKeywordValue
CSSUnitValue
CSSCalcValue
CSSTransformValue
CSSPositionValue
CSSImageValue
CSSFontFaceValue
```



W3C Editor's Draft

TABLE OF CONTENTS

CSS Typed OM Level 1

Editor's Draft, 2 June 2017

This version:
https://drafts.css-houdini.org/css-typed-om-1/

Latest published version:
https://www.w3.org/TR/2016/WD-css-typed-om-1-20160607/

Previous Versions:
https://www.w3.org/TR/2016/WD-css-typed-om-1-20160607/

Feedback:
public-houdini@w3.org with subject line "`[css-typed-om] … message topic …`" (archives)

Issue Tracking:
GitHub
Inline In Spec

Editors:
Shane Stephens
Tab Atkins-Bittner (Google)

Abstract

Converting CSSOM value strings into meaningfully typed JavaScript representations and back can incur a significant performance overhead. This specification exposes CSS values as typed JavaScript objects to facilitate their performant manipulation.

Status of this document

This is a public copy of the editors' draft. It is provided for discussion only and may change at any moment. Its publication here does not imply endorsement of its contents by W3C. Don't cite this document other than as work in progress.

GitHub Issues are preferred for discussion of this specification. When filing an issue, please put the text "css-typed-om" in the title, preferably like this: "[css-typed-om] *…summary of comment…*". All issues and comments are archived.

This document was produced by the CSS Working Group (part of the Style Activity).

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also

Paint

Layout

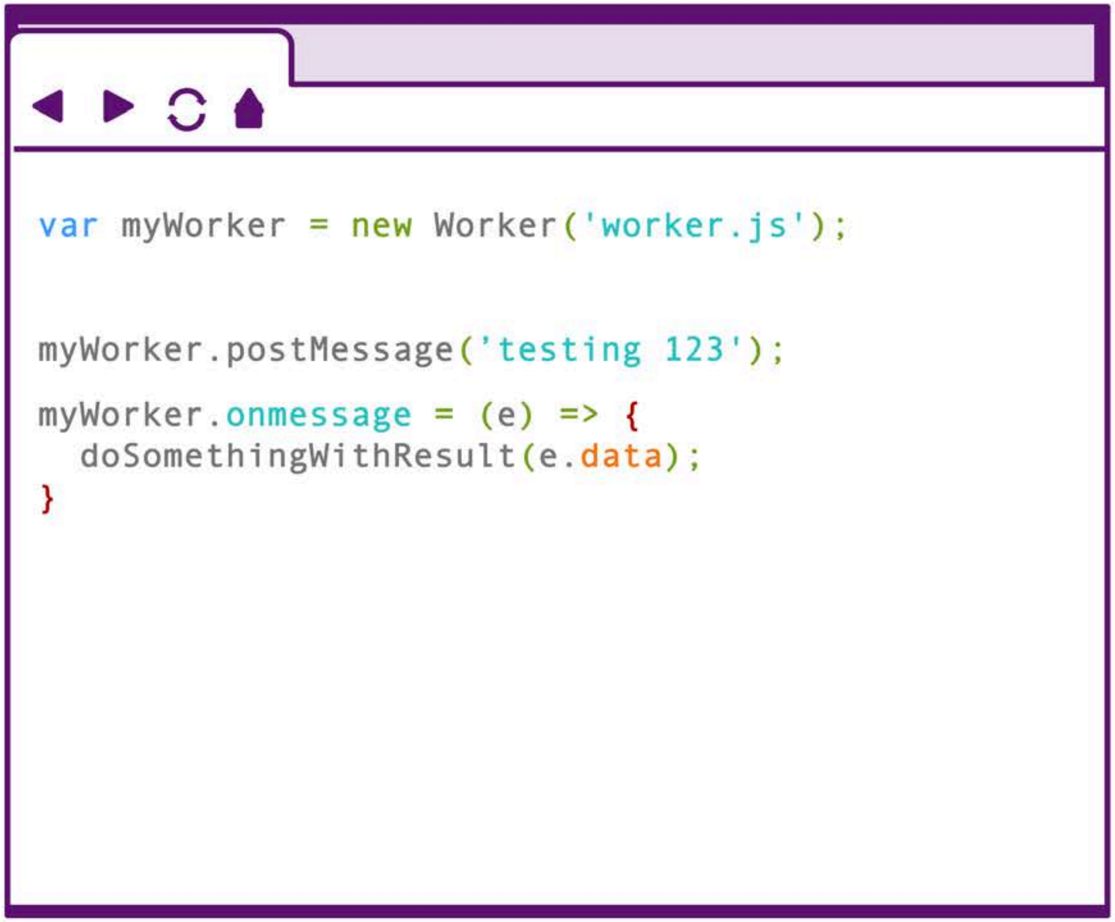
Animation

Worklets

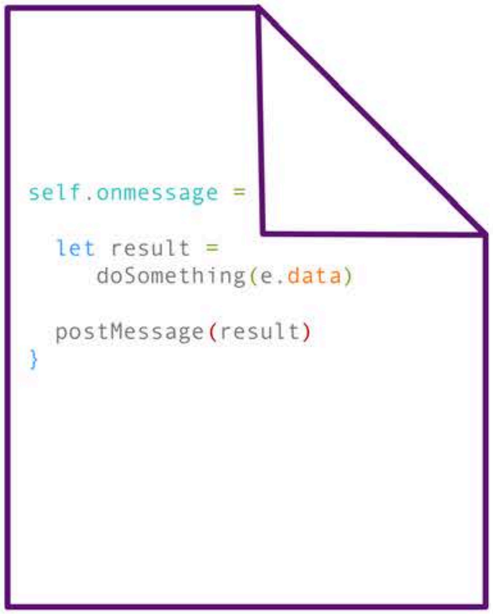
WebWorkers

```
commit 90b52e847359ae902d3f7ce7bc511cadfbc29ea8
Author: Alexey Proskuryakov <ap@webkit.org>
Date:   Thu Nov 6 2008 07:04:47 +0000

Implement Worker global object
```

WebWorkers



```
var myWorker = new Worker('worker.js');


myWorker.postMessage('testing 123');

myWorker.onmessage = (e) => {
  doSomethingWithResult(e.data);
}
```



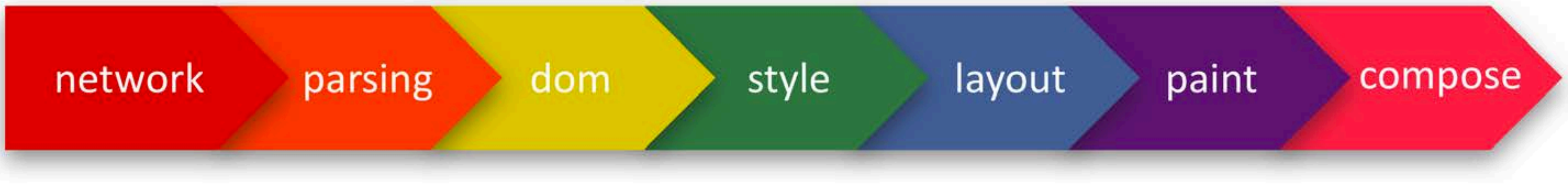
```
self.onmessage =

  let result =
     doSomething(e.data)

  postMessage(result)
}
```

WebWorkers

individual threads
event based

Worklets

API based
thread agnostic



network

parsing

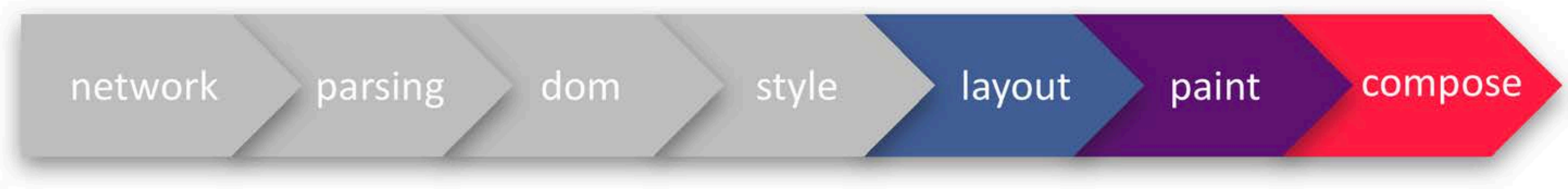
dom

style

layout

paint

compose



network parsing dom style layout paint compose

layoutWorklet

CSS Layout API

```
<!doctype html>
<div id="myElement">
  <div id="child1"></div>
  <div id="child2"></div>
</div>

<style>
  #myElement {
    width: 500px;
    height: 500px;
    display: layout('masonry')
  }
</style>

<script>
  window.layoutWorklet.addModule('masonry.js')
</script>
```

```
// masonry.js
registerLayout('masonry', class extends Layout {

  *layout(space, children, styleMap) {
    var inlineSize = resolveInlineSize(space, styleMap)
    var bordersAndPadding =
resolveBordersAndPadding(varraintSpace, styleMap)
    var scrollbarSize = resolveScrollbarSize(varraintSpace,
styleMap)

    var availableInlineSize = inlineSize -
      bordersAndPadding.inlineStart -
      bordersAndPadding.inlineEnd -
      scrollbarSize.inline;

    var availableBlockSize = resolveBlockSize(varraintSpace,
styleMap) -
      bordersAndPadding.blockStart -
      bordersAndPadding.blockEnd -
      scrollbarSize.block;
...
```

```
    return {
      inlineSize: inlineSize,
      blockSize: blockSize,
      inlineOverflowSize: inlineOverflowSize,
      blockOverflowSize: blockOverflowSize,
      childFragments: childFragments
    };
});
```

yuck

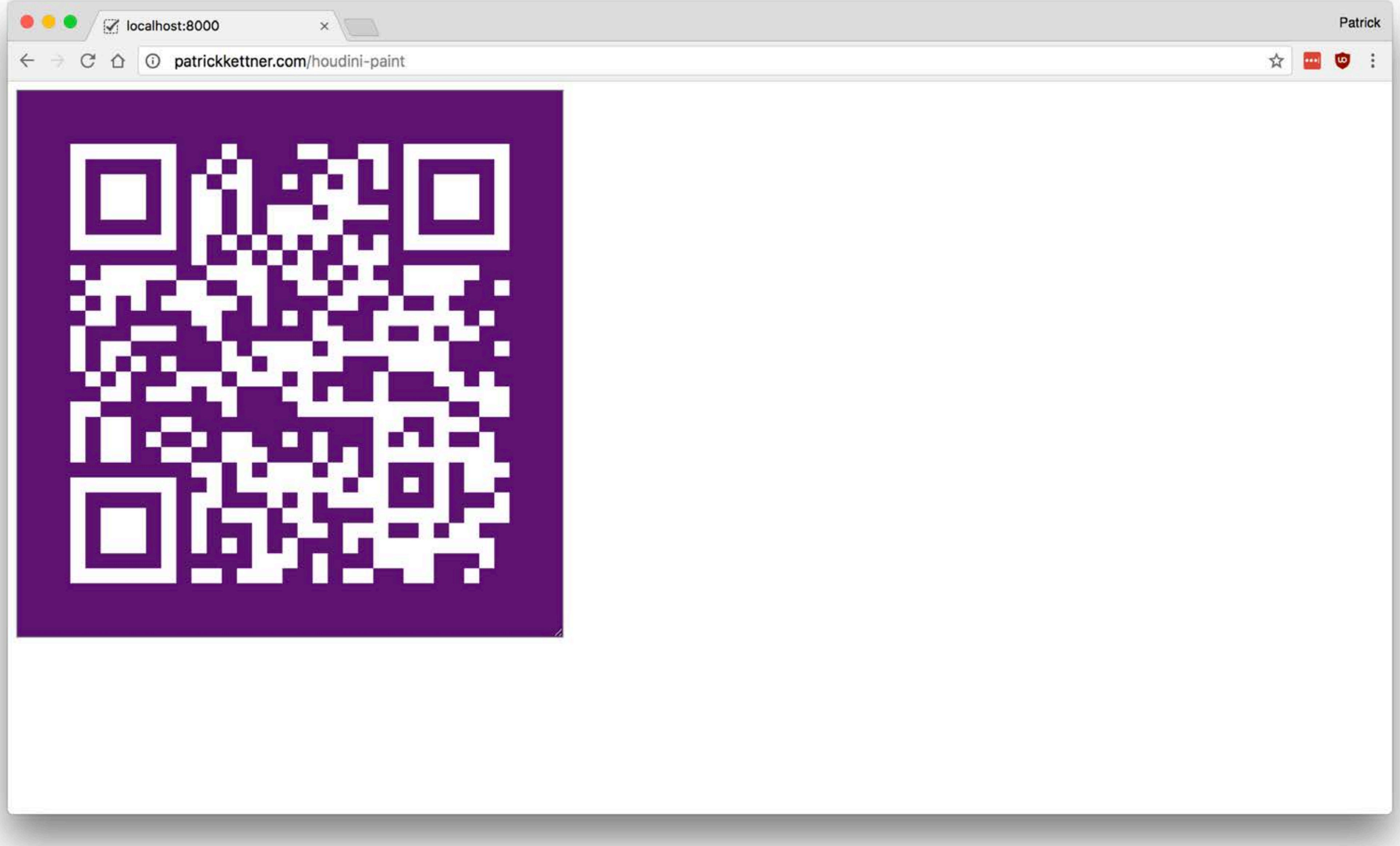
it’s about efficiency, not ease of use

paintWorklet

CSS Paint API

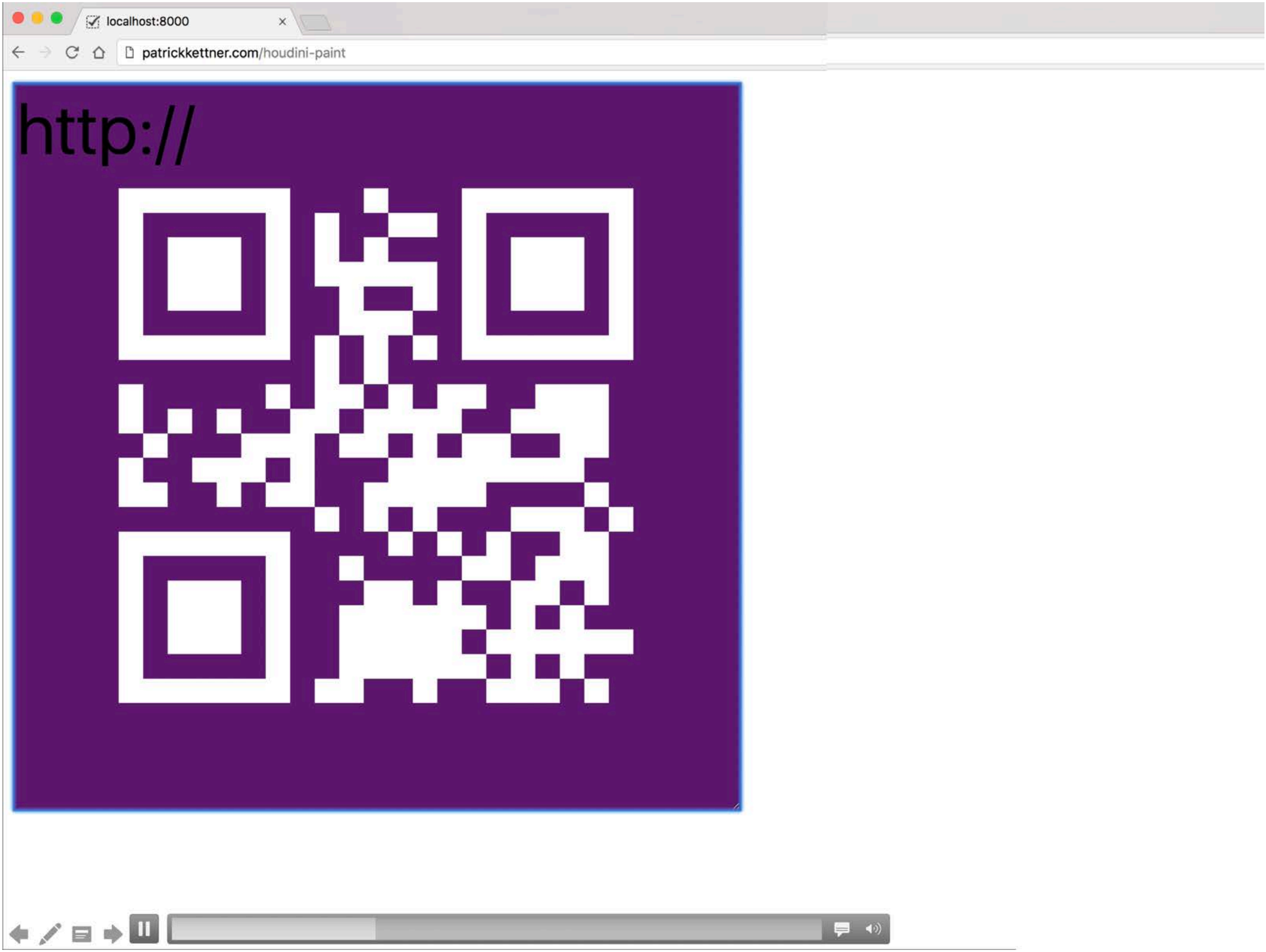
```
<!doctype html>
<textarea id="myElement"></textarea>

<style>
  #myElement {
    background-color: #5d1e6f;
    background-image: paint(qr);
    --qr-url: https://patrickkettner.com;
    width: 500px;
    height: 500px;
  }
</style>


<script>
  paintWorklet.addModule('qr.js')
</script>
```

```
// qr.js

// slightly modified version of
// https://github.com/yyx990803/QR.js
var QR=function() {…}

registerPaint('qr', class {
  static get inputProperties() { return [ '--qr-url' ]; }

  paint(ctx, geom, properties) {
    let url = properties.get('--qr-url').value
    const minSize = Math.min(geom.width, geom.height);

    if (url) {
      QR.draw(url, ctx, minSize, 2)
    }
  }
});
```

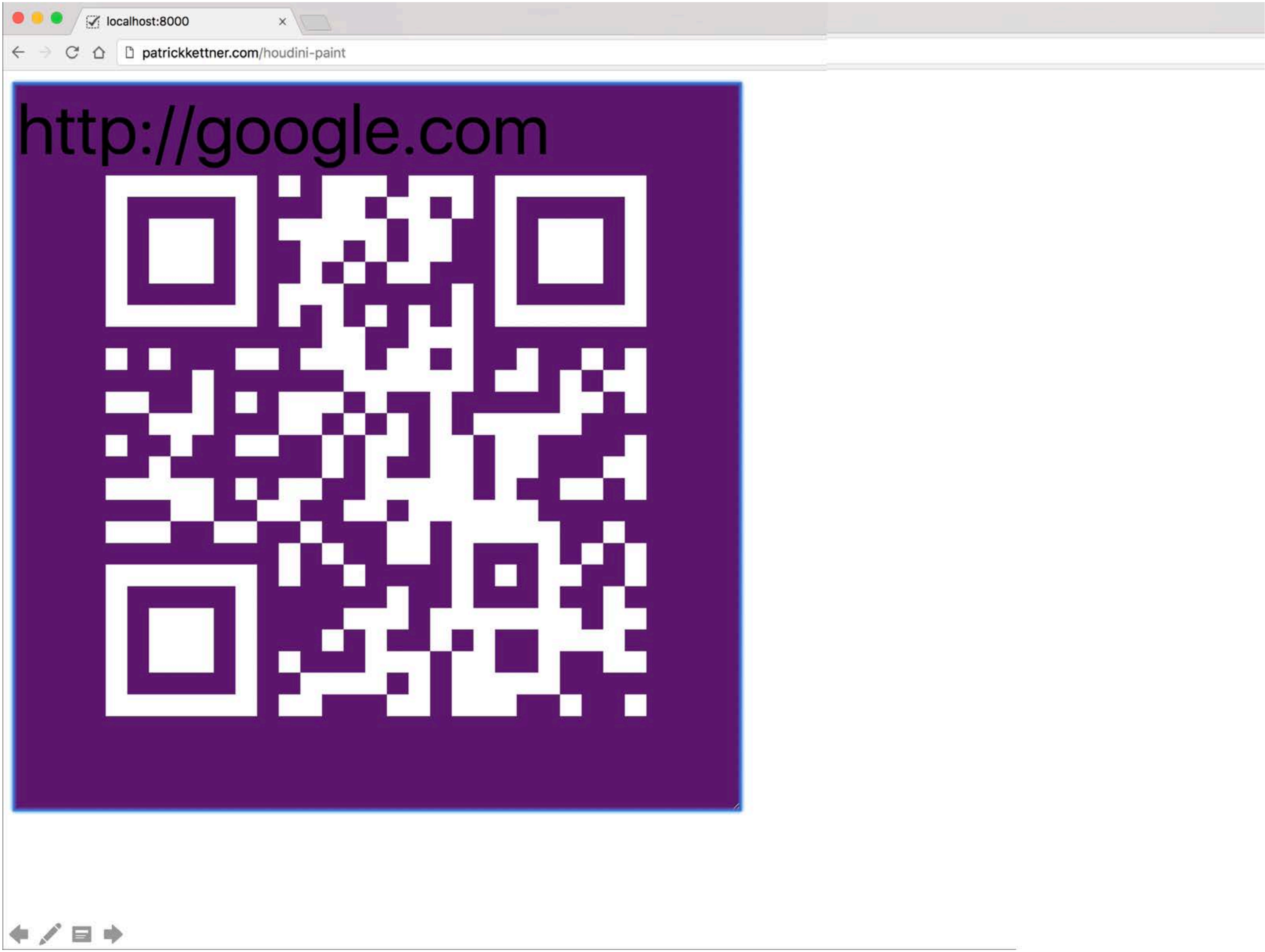


localhost:8000
Patrick
patrickkettner.com/houdini-paint

```
<!doctype html>
<textarea id="myElement"></textarea>

<style>
  #myElement {
    background-color: #5d1e6f;
    background-image: paint(qr);
    --qr-url: https://patrickkettner.com;
    width: 500px;
    height: 500px;
  }
</style>

<script>
 const txt = document.querySelector('textarea')
 txt.addEventListener('input', (e) => {
   txt.style.setProperty('--qr-url', txt.value)
 });

  paintWorklet.addModule('qr.js')
</script>
```



localhost:8000
patrickkettner.com/houdini-paint
http://

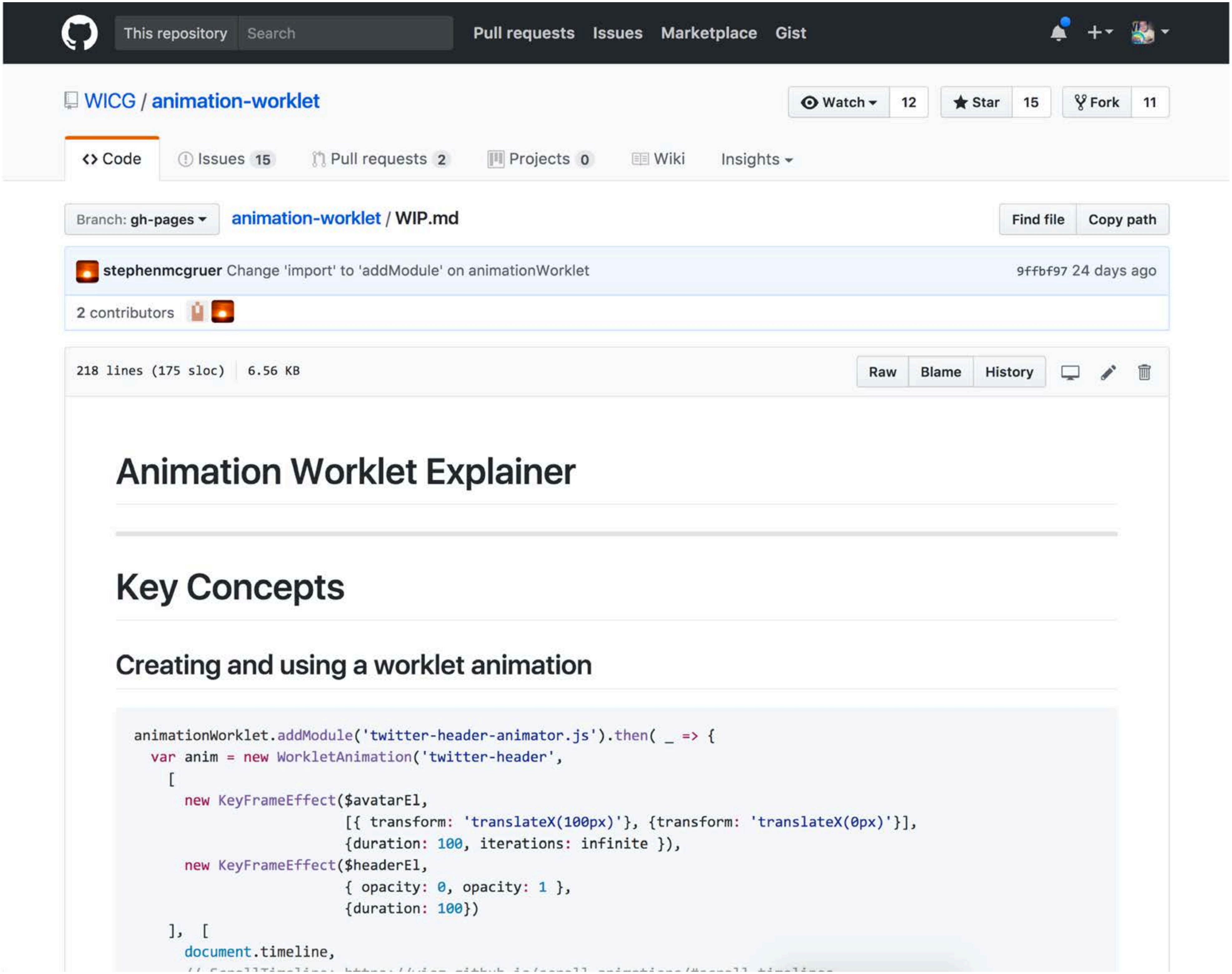

localhost:8000
patrickkettner.com/houdini-paint
http://google.com

animationWorklet

compositorWorklet?

¯_(ツ)_/¯

```
commit a72d0e6b4af29e8825baee923b80dd59578f2e6f
From: Stephen McGruer <smcgruer@chromium.org
Date: Fri, 26 May 2017 11:11:33 -0400

index.bs version of the spec is out of date
```



animation-worklet / WIP.md

stephenmcgruer Change 'import' to 'addModule' on animationWorklet 9ffbf97 24 days ago

2 contributors

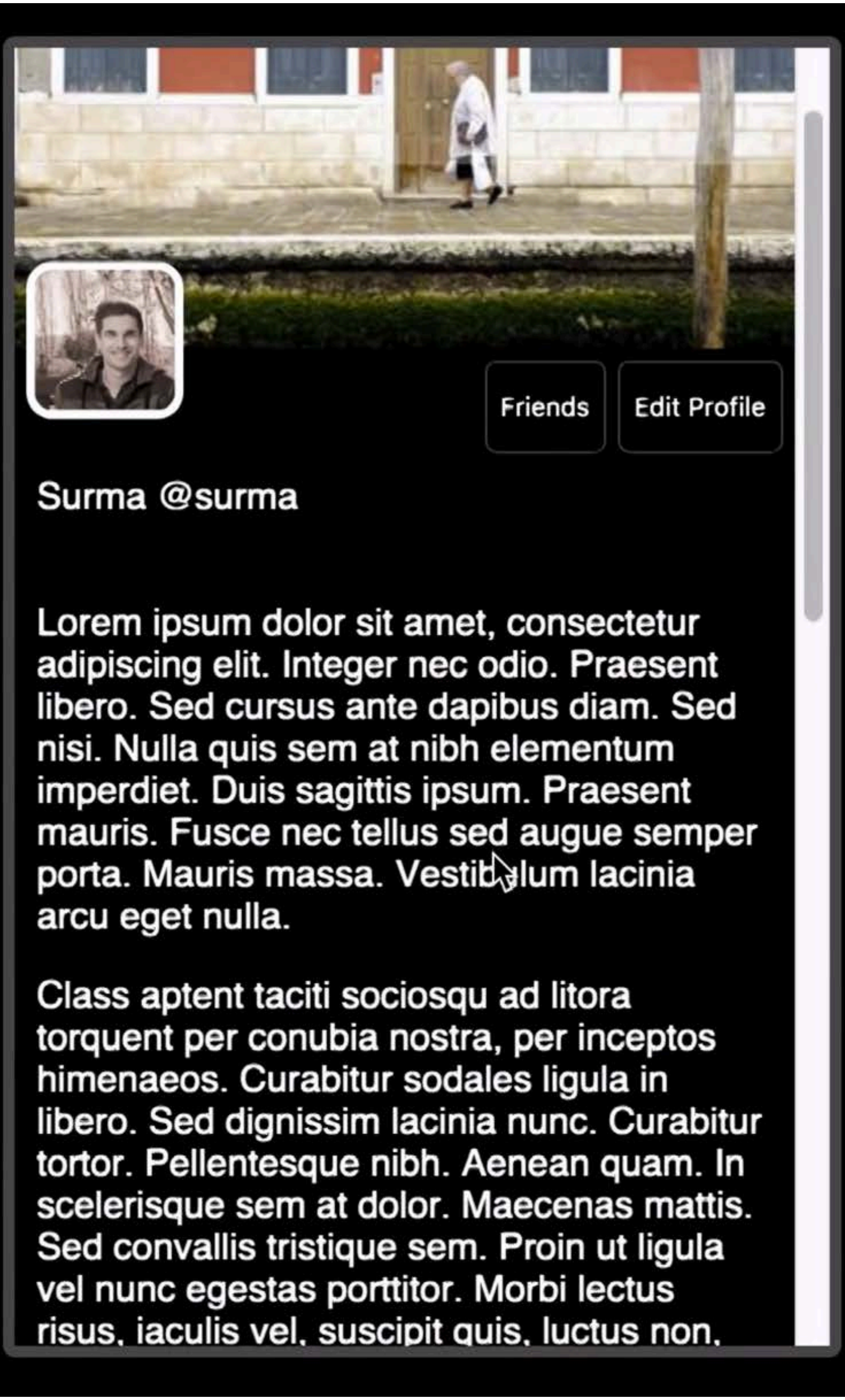
218 lines (175 sloc) 6.56 KB

Animation Worklet Explainer

Key Concepts

Creating and using a worklet animation

```
animationWorklet.addModule('twitter-header-animator.js').then( _ => {
  var anim = new WorkletAnimation('twitter-header',
    [
      new KeyFrameEffect($avatarEl,
                         [{ transform: 'translateX(100px)'}, {transform: 'translateX(0px)'}],
                         {duration: 100, iterations: infinite }),
      new KeyFrameEffect($headerEl,
                         { opacity: 0, opacity: 1 },
                         {duration: 100})
    ],  [
      document.timeline,
      // ScrollTimeline: https://wicg.github.io/scroll-animations/#scroll-timelines
```



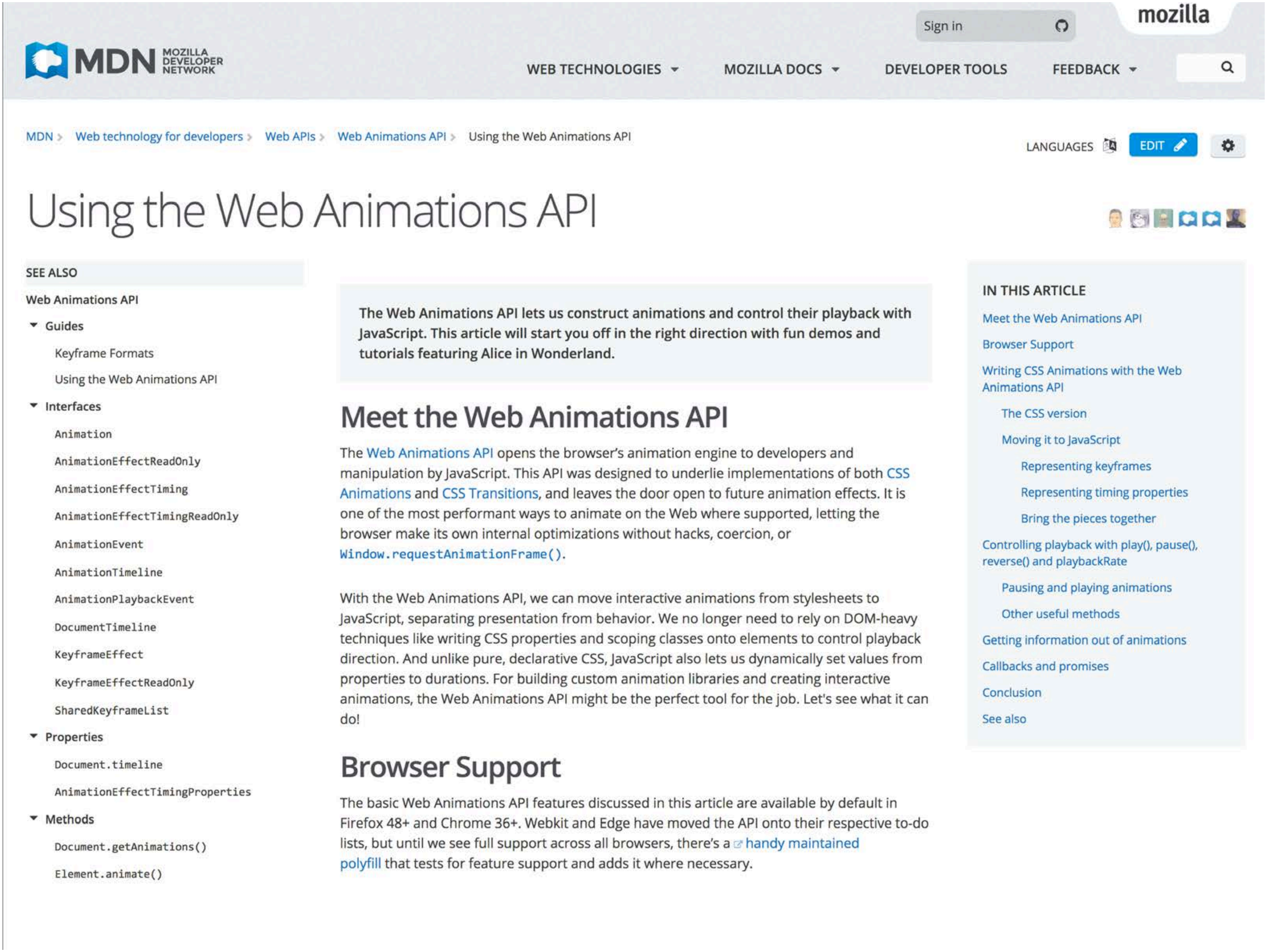
Surma @surma

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla.

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Curabitur sodales ligula in libero. Sed dignissim lacinia nunc. Curabitur tortor. Pellentesque nibh. Aenean quam. In scelerisque sem at dolor. Maecenas mattis. Sed convallis tristique sem. Proin ut ligula vel nunc egestas porttitor. Morbi lectus risus, iaculis vel, suscipit quis, luctus non,

```
animationWorklet.addModule('twitter-header-animator.js')
  .then( _ => {
    var anim = new WorkletAnimation('twitter-header', [
      new KeyFrameEffect($avatarEl, [{
          transform: 'translateX(100px)’
      }, {transform: 'translateX(0px)'}], {
          duration: 100,
          iterations: infinite }),
          new KeyFrameEffect($headerEl,
            { opacity: 0, opacity: 1 },
            {duration: 100})], [
             document.timeline,

          new ScrollingTimeline(scrollingElement,
            {timeRange: 100})
    ],{some_awesome_data:42}
  );});
```

```
registerAnimator('twitter-header', class {
  animate(timelines, outputEffects) {
    const time = timelines[1].currentTime;

    outputEffects[0].localTime = time;
    outputEffects[1].localTime = Math.min(1, time * 10);
  }
});
```



MDN › Web technology for developers › Web APIs › Web Animations API › Using the Web Animations API

Using the Web Animations API

SEE ALSO

Web Animations API

- Guides
  - Keyframe Formats
  - Using the Web Animations API
- Interfaces
  - `Animation`
  - `AnimationEffectReadOnly`
  - `AnimationEffectTiming`
  - `AnimationEffectTimingReadOnly`
  - `AnimationEvent`
  - `AnimationTimeline`
  - `AnimationPlaybackEvent`
  - `DocumentTimeline`
  - `KeyframeEffect`
  - `KeyframeEffectReadOnly`
  - `SharedKeyframeList`
- Properties
  - `Document.timeline`
  - `AnimationEffectTimingProperties`
- Methods
  - `Document.getAnimations()`
  - `Element.animate()`

The Web Animations API lets us construct animations and control their playback with JavaScript. This article will start you off in the right direction with fun demos and tutorials featuring Alice in Wonderland.

Meet the Web Animations API

The Web Animations API opens the browser's animation engine to developers and manipulation by JavaScript. This API was designed to underlie implementations of both CSS Animations and CSS Transitions, and leaves the door open to future animation effects. It is one of the most performant ways to animate on the Web where supported, letting the browser make its own internal optimizations without hacks, coercion, or `Window.requestAnimationFrame()`.

With the Web Animations API, we can move interactive animations from stylesheets to JavaScript, separating presentation from behavior. We no longer need to rely on DOM-heavy techniques like writing CSS properties and scoping classes onto elements to control playback direction. And unlike pure, declarative CSS, JavaScript also lets us dynamically set values from properties to durations. For building custom animation libraries and creating interactive animations, the Web Animations API might be the perfect tool for the job. Let's see what it can do!

Browser Support

The basic Web Animations API features discussed in this article are available by default in Firefox 48+ and Chrome 36+. Webkit and Edge have moved the API onto their respective to-do lists, but until we see full support across all browsers, there's a handy maintained polyfill that tests for feature support and adds it where necessary.

IN THIS ARTICLE

it’s about efficiency,
not ease of use

start playing with nightlies

give feedback!

https://aka.ms/EdgeUserVoice

thanks!

patrickkettner

f patrickkettner

patrickkettner

in patrickkettner

✉ patrickkettner@microsoft.com