QCon London

The business of front-end development.

"This is for Everyone" by Nick Webb - Flickr: DSC_3232. Licensed under CC BY 2.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:This_is_for_Everyone.jpg#mediaviewer/File:This_is_for_Everyone.jpg

Rachel Andrew

Managing Director of edgeofmyseat.com

Web developer, writer and speaker

Find me at rachelandrew.co.uk

On Twitter: @rachelandrew

grabaperch.com

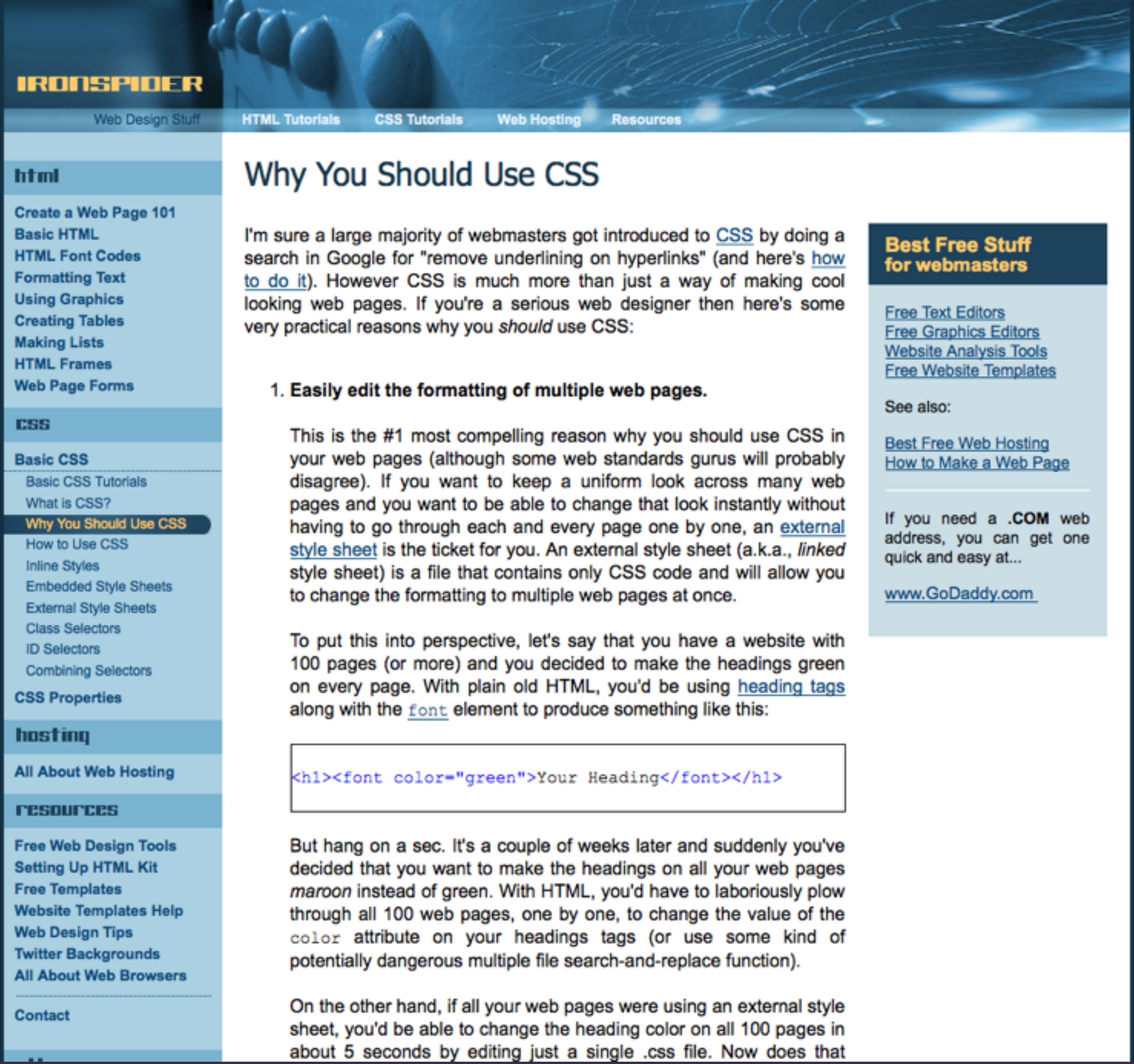
The web is an accessible medium. We can protect that, or we can break it. **I choose to protect it.**

My tasks include …

- bookkeeping
- completing baffling forms from the government
- writing Puppet Manifests
- coding PHP
- writing documentation
- writing and giving presentations & workshops
- front-end development

I ship.

NETSCAPE®



Why You Should Use CSS

I'm sure a large majority of webmasters got introduced to CSS by doing a search in Google for "remove underlining on hyperlinks" (and here's how to do it). However CSS is much more than just a way of making cool looking web pages. If you're a serious web designer then here's some very practical reasons why you *should* use CSS:

1. **Easily edit the formatting of multiple web pages.**

   This is the #1 most compelling reason why you should use CSS in your web pages (although some web standards gurus will probably disagree). If you want to keep a uniform look across many web pages and you want to be able to change that look instantly without having to go through each and every page one by one, an external style sheet is the ticket for you. An external style sheet (a.k.a., *linked* style sheet) is a file that contains only CSS code and will allow you to change the formatting to multiple web pages at once.

   To put this into perspective, let's say that you have a website with 100 pages (or more) and you decided to make the headings green on every page. With plain old HTML, you'd be using heading tags along with the `font` element to produce something like this:

```
<h1><font color="green">Your Heading</font></h1>
```

   But hang on a sec. It's a couple of weeks later and suddenly you've decided that you want to make the headings on all your web pages *maroon* instead of green. With HTML, you'd have to laboriously plow through all 100 web pages, one by one, to change the value of the `color` attribute on your headings tags (or use some kind of potentially dangerous multiple file search-and-replace function).

   On the other hand, if all your web pages were using an external style sheet, you'd be able to change the heading color on all 100 pages in about 5 seconds by editing just a single .css file. Now does that

Best Free Stuff for webmasters

Free Text Editors
Free Graphics Editors
Website Analysis Tools
Free Website Templates

See also:

Best Free Web Hosting
How to Make a Web Page

If you need a **.COM** web address, you can get one quick and easy at...

www.GoDaddy.com

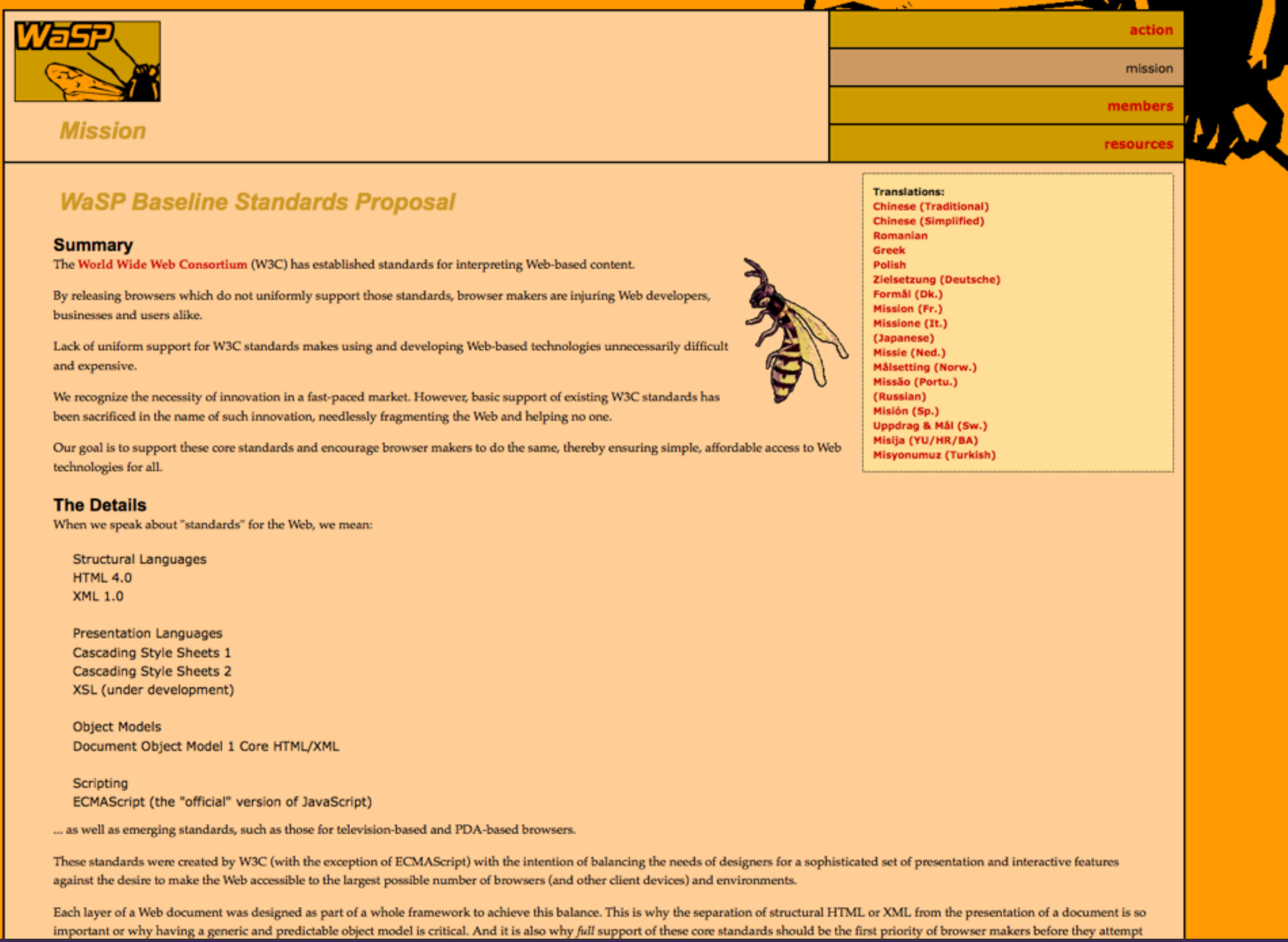
http://www.ironspider.ca/adv/basic_css/whyusecss.htm

The “Netscape Resize Fix”

```
<script type="text/javascript">
<!--
function MM_reloadPage(init) {
  if (init==true) with (navigator)
{if
((appName=="Netscape")&&(parseInt(ap
pVersion)==4)) {
    document.MM_pgW=innerWidth;
document.MM_pgH=innerHeight;
onresize=MM_reloadPage; }}
  else if (innerWidth!
=document.MM_pgW || innerHeight!
=document.MM_pgH) location.reload();
}
MM_reloadPage(true);
//-->
</script>
```

Front-end developer circa 2005?

Browser bugs expert.



Mission

WaSP Baseline Standards Proposal

Translations:
Chinese (Traditional)
Chinese (Simplified)
Romanian
Greek
Polish
Zielsetzung (Deutsche)
Formål (Dk.)
Mission (Fr.)
Missione (It.)
(Japanese)
Missie (Ned.)
Målsetting (Norw.)
Missão (Portu.)
(Russian)
Misión (Sp.)
Uppdrag & Mål (Sw.)
Misija (YU/HR/BA)
Misyonumuz (Turkish)

Summary

The World Wide Web Consortium (W3C) has established standards for interpreting Web-based content.

By releasing browsers which do not uniformly support those standards, browser makers are injuring Web developers, businesses and users alike.

Lack of uniform support for W3C standards makes using and developing Web-based technologies unnecessarily difficult and expensive.

We recognize the necessity of innovation in a fast-paced market. However, basic support of existing W3C standards has been sacrificed in the name of such innovation, needlessly fragmenting the Web and helping no one.

Our goal is to support these core standards and encourage browser makers to do the same, thereby ensuring simple, affordable access to Web technologies for all.

The Details

When we speak about "standards" for the Web, we mean:

Structural Languages
HTML 4.0
XML 1.0

Presentation Languages
Cascading Style Sheets 1
Cascading Style Sheets 2
XSL (under development)

Object Models
Document Object Model 1 Core HTML/XML

Scripting
ECMAScript (the "official" version of JavaScript)

... as well as emerging standards, such as those for television-based and PDA-based browsers.

These standards were created by W3C (with the exception of ECMAScript) with the intention of balancing the needs of designers for a sophisticated set of presentation and interactive features against the desire to make the Web accessible to the largest possible number of browsers (and other client devices) and environments.

Each layer of a Web document was designed as part of a whole framework to achieve this balance. This is why the separation of structural HTML or XML from the presentation of a document is so important or why having a generic and predictable object model is critical. And it is also why *full* support of these core standards should be the first priority of browser makers before they attempt

http://archive.webstandards.org/mission.html

“Thanks to the hard work of countless WaSP members and supporters (like you), Tim Berners-Lee’s vision of the web as an open, accessible, and universal community is largely the reality. While there is still work to be done, the sting of the WaSP is no longer necessary. And so it is time for us to close down The Web Standards Project.”

http://www.webstandards.org/2013/03/01/our-work-here-is-done/

Browser vendors are implementing standard things in a standard way.

Innovation happens through the standards process.

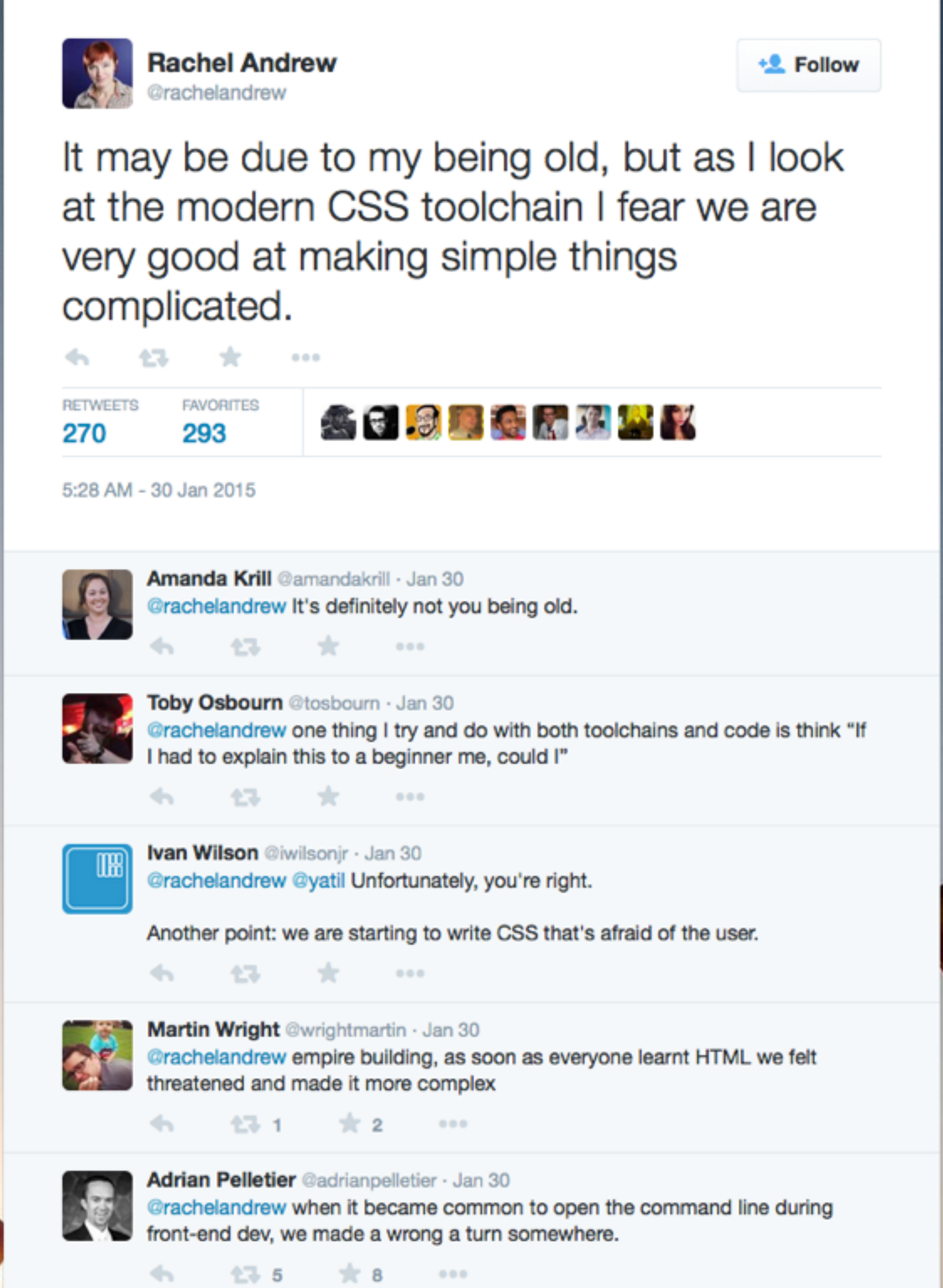
Showstopping browser bugs when doing straightforward things in modern browsers are rare.

“Studies show that a todo list is the most complex JavaScript app you can build before a newer, better framework is invented.”

http://www.allenpike.com/2015/javascript-framework-fatigue/

We’re creating complexity.

Hiding the simple languages of the web behind tooling and process.



Rachel Andrew @rachelandrew

It may be due to my being old, but as I look at the modern CSS toolchain I fear we are very good at making simple things complicated.

RETWEETS 270 FAVORITES 293

5:28 AM - 30 Jan 2015

Amanda Krill @amandakrill · Jan 30
@rachelandrew It's definitely not you being old.

Toby Osbourn @tosbourn · Jan 30
@rachelandrew one thing I try and do with both toolchains and code is think "If I had to explain this to a beginner me, could I"

Ivan Wilson @iwilsonjr · Jan 30
@rachelandrew @yatil Unfortunately, you're right.

Another point: we are starting to write CSS that's afraid of the user.

Martin Wright @wrightmartin · Jan 30
@rachelandrew empire building, as soon as everyone learnt HTML we felt threatened and made it more complex

Adrian Pelletier @adrianpelletier · Jan 30
@rachelandrew when it became common to open the command line during front-end dev, we made a wrong a turn somewhere.

Replacing divs ...

```
<div class="header">
   <h1>My website</h1>
   <div class="nav">

   </div>
</div>

<div class="article">

</div>

<div class="sidebar">

</div>

<div class="footer">

</div>
```

… with new semantic elements.

```
<header>
   <h1>My website</h1>
   <nav>

   </nav>
</header>

<article>

</article>

<aside>

</aside>

<footer>

</footer>
```

Web Video Text Tracks Format (WebVTT)

```
WEBVTT

1
00:00:22.230 --> 00:00:24.606
This is the first subtitle.

2
00:00:30.739 --> 00:00:34.074
This is the second.

3
00:00:34.159 --> 00:00:35.743
Third
```

Web Video Text Tracks Format (WebVTT)

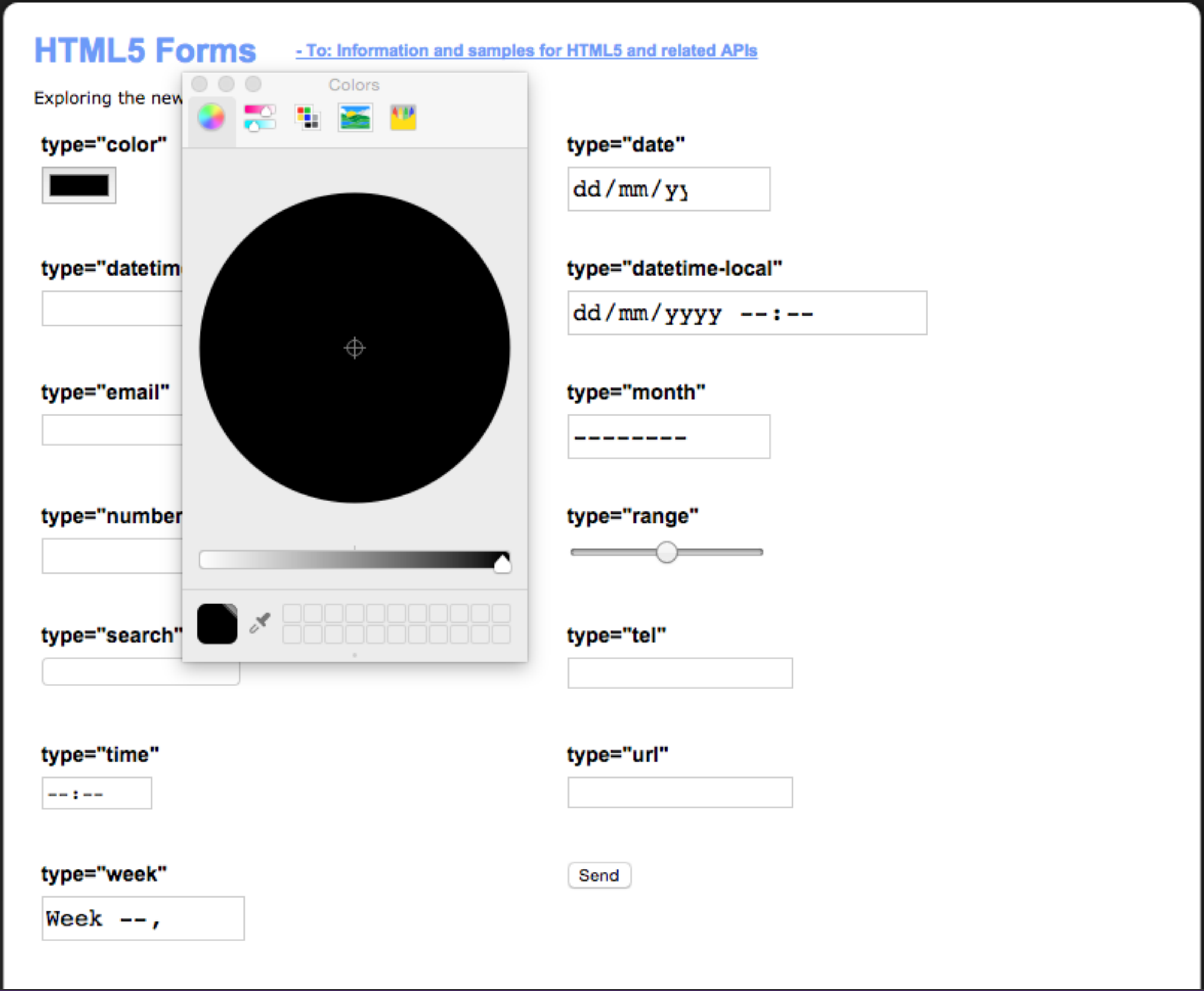
```
WEBVTT

1
00:00:22.230 --> 00:00:24.606
This is the first subtitle.

2
00:00:30.739 --> 00:00:34.074
This is the second.

3
00:00:34.159 --> 00:00:35.743
Third
```

https://developer.mozilla.org/en/docs/Web/API/Web_Video_Text_Tracks_Format



HTML5 Forms - To: Information and samples for HTML5 and related APIs

Exploring the new

type="color"

type="date"

dd/mm/yy

type="datetim

type="datetime-local"

dd/mm/yyyy --:--

type="email"

type="month"

----

type="number

type="range"

type="search"

type="tel"

type="time"

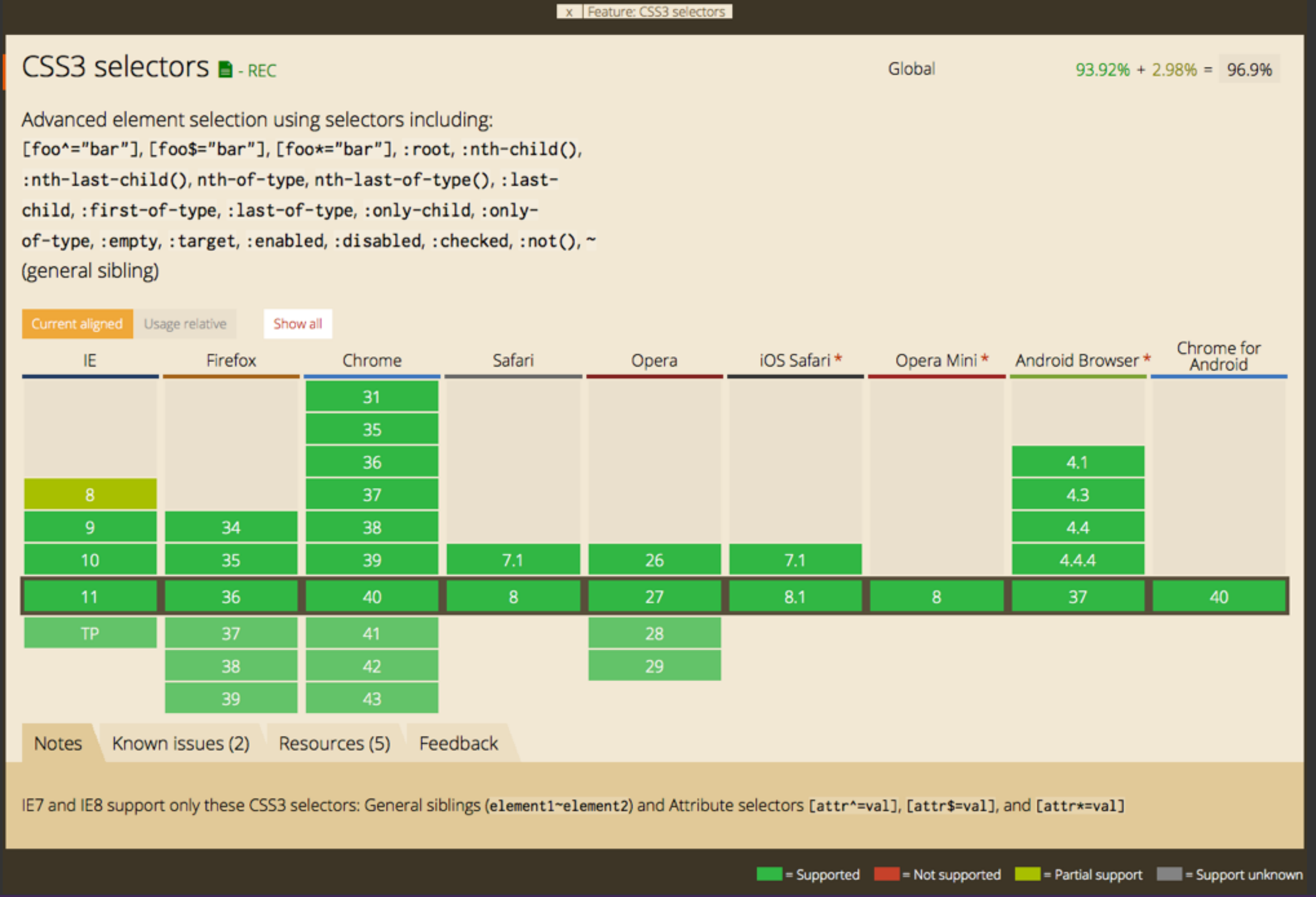
--:--

type="url"

type="week"

Week --,

http://robertnyman.com/html5/forms/input-types.html



x | Feature: CSS3 selectors

CSS3 selectors - REC

Global 93.92% + 2.98% = 96.9%

Advanced element selection using selectors including: `[foo^="bar"]`, `[foo$="bar"]`, `[foo*="bar"]`, `:root`, `:nth-child()`, `:nth-last-child()`, `nth-of-type`, `nth-last-of-type()`, `:last-child`, `:first-of-type`, `:last-of-type`, `:only-child`, `:only-of-type`, `:empty`, `:target`, `:enabled`, `:disabled`, `:checked`, `:not()`, `~` (general sibling)

IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		31						
		35						
		36					4.1	
8		37					4.3	
9	34	38					4.4	
10	35	39	7.1	26	7.1		4.4.4	
11	36	40	8	27	8.1	8	37	40
TP	37	41		28				
	38	42		29				
	39	43						

IE7 and IE8 support only these CSS3 selectors: General siblings (`element1~element2`) and Attribute selectors `[attr^=val]`, `[attr$=val]`, and `[attr*=val]`

🟩 = Supported 🟥 = Not supported 🟨 = Partial support ⬜ = Support unknown

http://caniuse.com/#feat=css-sel3

Time-dimensional Pseudo-classes

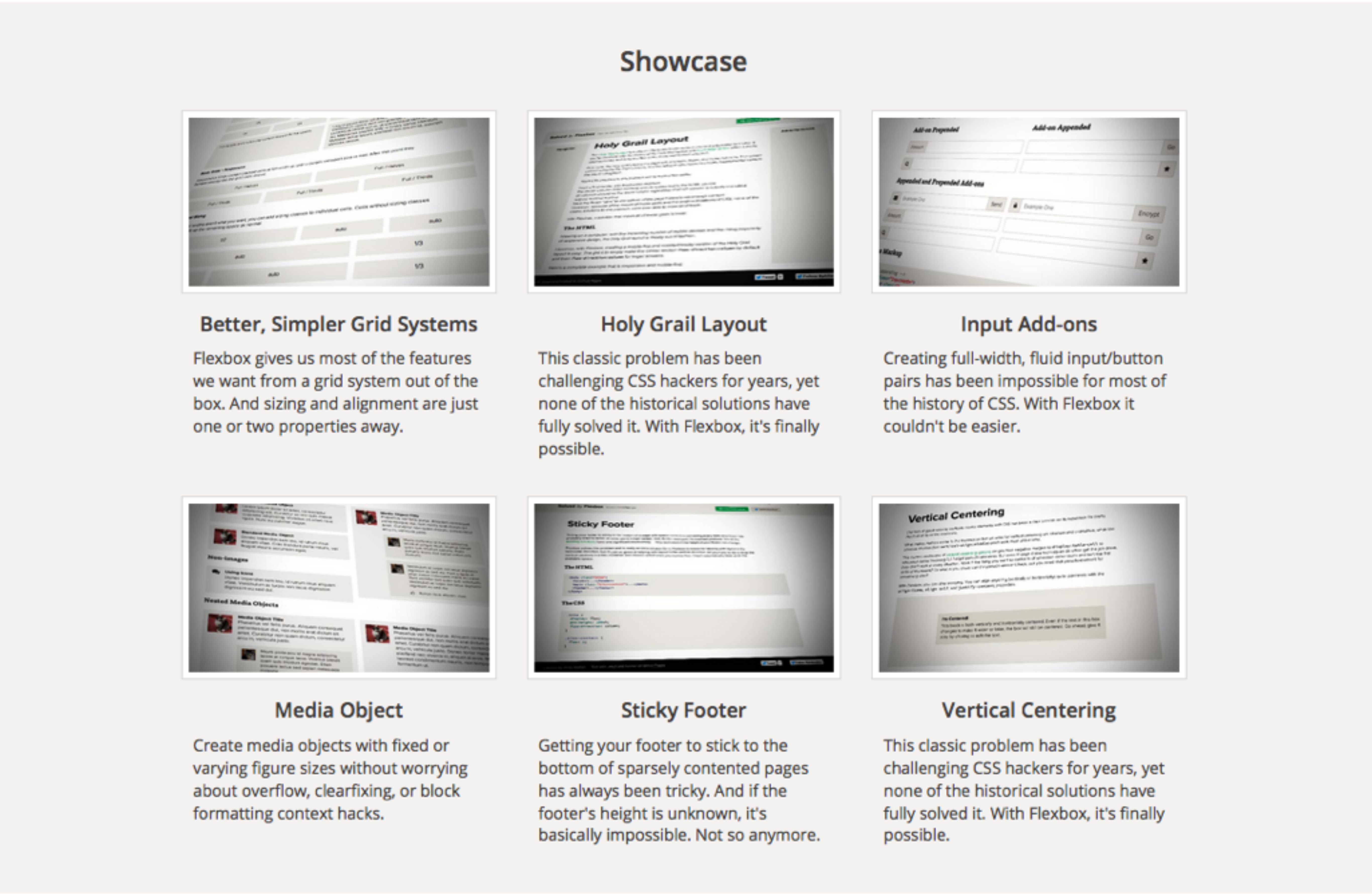
:current

:past

:future

```
:current(p, li, dt, dd) {
  background: yellow;
}

:past(p, li, dt, dd) {
  background: transparent;
  color: #999999;
}
```



Showcase

Better, Simpler Grid Systems

Flexbox gives us most of the features we want from a grid system out of the box. And sizing and alignment are just one or two properties away.

Holy Grail Layout

This classic problem has been challenging CSS hackers for years, yet none of the historical solutions have fully solved it. With Flexbox, it's finally possible.

Input Add-ons

Creating full-width, fluid input/button pairs has been impossible for most of the history of CSS. With Flexbox it couldn't be easier.

Media Object

Create media objects with fixed or varying figure sizes without worrying about overflow, clearfixing, or block formatting context hacks.

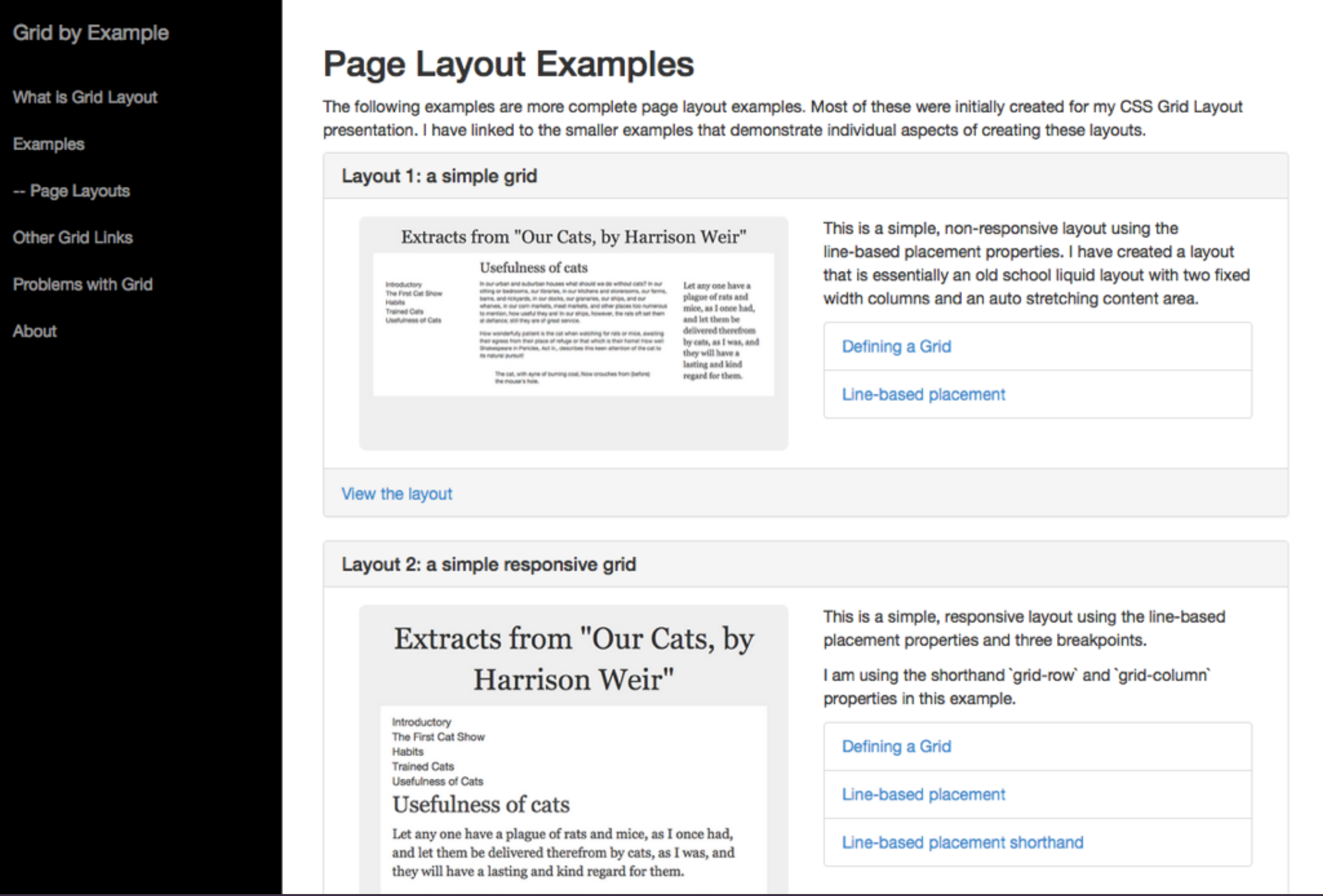
Sticky Footer

Getting your footer to stick to the bottom of sparsely contented pages has always been tricky. And if the footer's height is unknown, it's basically impossible. Not so anymore.

Vertical Centering

This classic problem has been challenging CSS hackers for years, yet none of the historical solutions have fully solved it. With Flexbox, it's finally possible.

http://philipwalton.github.io/solved-by-flexbox/



Grid by Example

What is Grid Layout

Examples

-- Page Layouts

Other Grid Links

Problems with Grid

About

Page Layout Examples

The following examples are more complete page layout examples. Most of these were initially created for my CSS Grid Layout presentation. I have linked to the smaller examples that demonstrate individual aspects of creating these layouts.

Layout 1: a simple grid

Extracts from "Our Cats, by Harrison Weir"

Usefulness of cats

Introductory
The First Cat Show
Habits
Trained Cats
Usefulness of Cats

In our urban and suburban houses what should we do without cats? In our sitting or bedrooms, our libraries, in our kitchens and storerooms, our farms, barns, and rickyards, in our docks, our granaries, our ships, and our wharves, in our corn markets, meat markets, and other places too numerous to mention, how useful they are! In our ships, however, the rats oft set them at defiance; still they are of great service.

How wonderfully patient is the cat when watching for rats or mice, awaiting their egress from their place of refuge or that which is their home! How well Shakespeare in Pericles, Act iii., describes this keen attention of the cat to its natural pursuit!

The cat, with eyne of burning coal, Now crouches from (before) the mouse's hole.

Let any one have a plague of rats and mice, as I once had, and let them be delivered therefrom by cats, as I was, and they will have a lasting and kind regard for them.

This is a simple, non-responsive layout using the line-based placement properties. I have created a layout that is essentially an old school liquid layout with two fixed width columns and an auto stretching content area.

Defining a Grid

Line-based placement

View the layout

Layout 2: a simple responsive grid

Extracts from "Our Cats, by Harrison Weir"

Introductory
The First Cat Show
Habits
Trained Cats
Usefulness of Cats

Usefulness of cats

Let any one have a plague of rats and mice, as I once had, and let them be delivered therefrom by cats, as I was, and they will have a lasting and kind regard for them.

This is a simple, responsive layout using the line-based placement properties and three breakpoints.

I am using the shorthand `grid-row` and `grid-column` properties in this example.

Defining a Grid

Line-based placement

Line-based placement shorthand

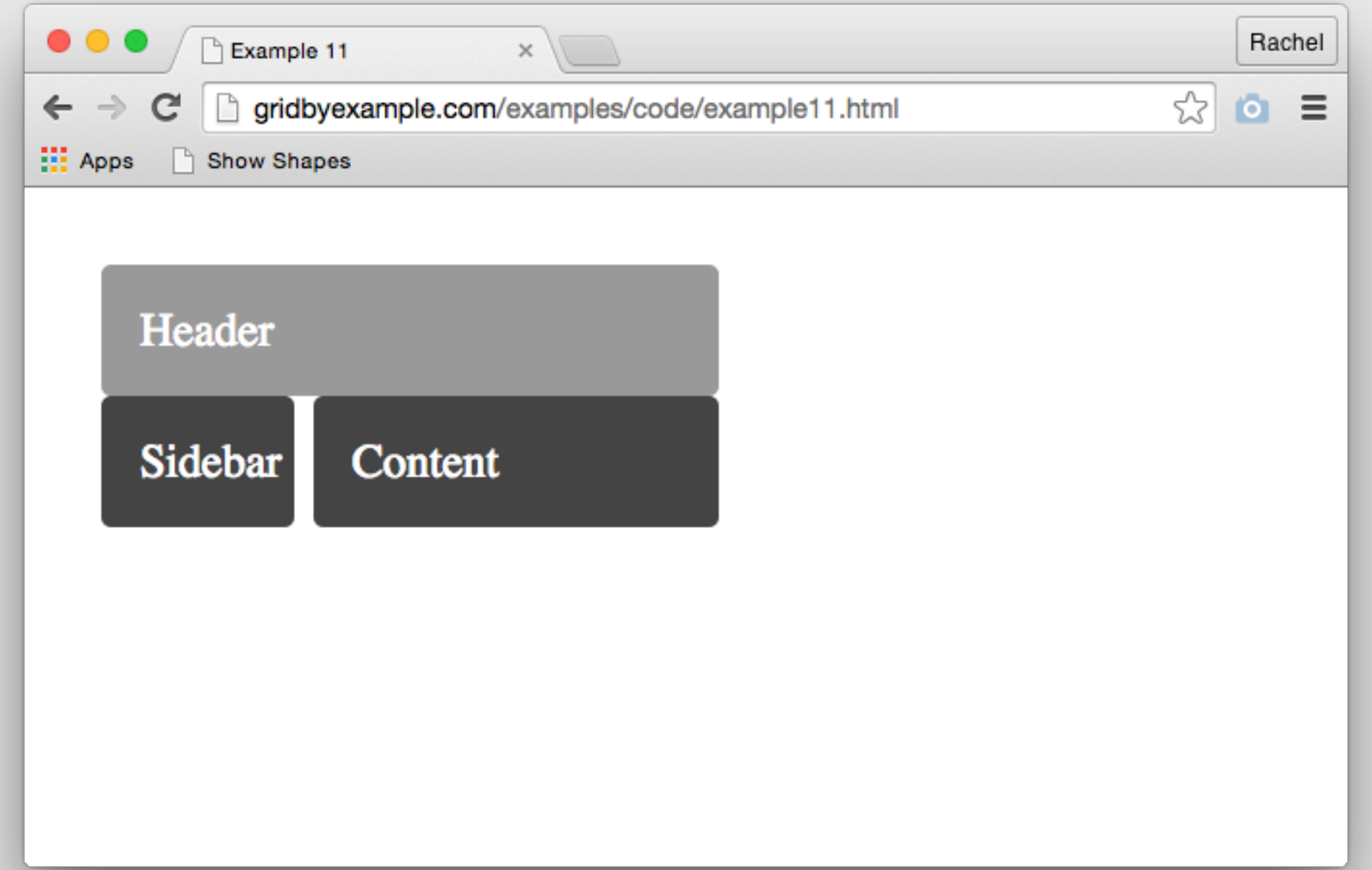
gridbyexample.com

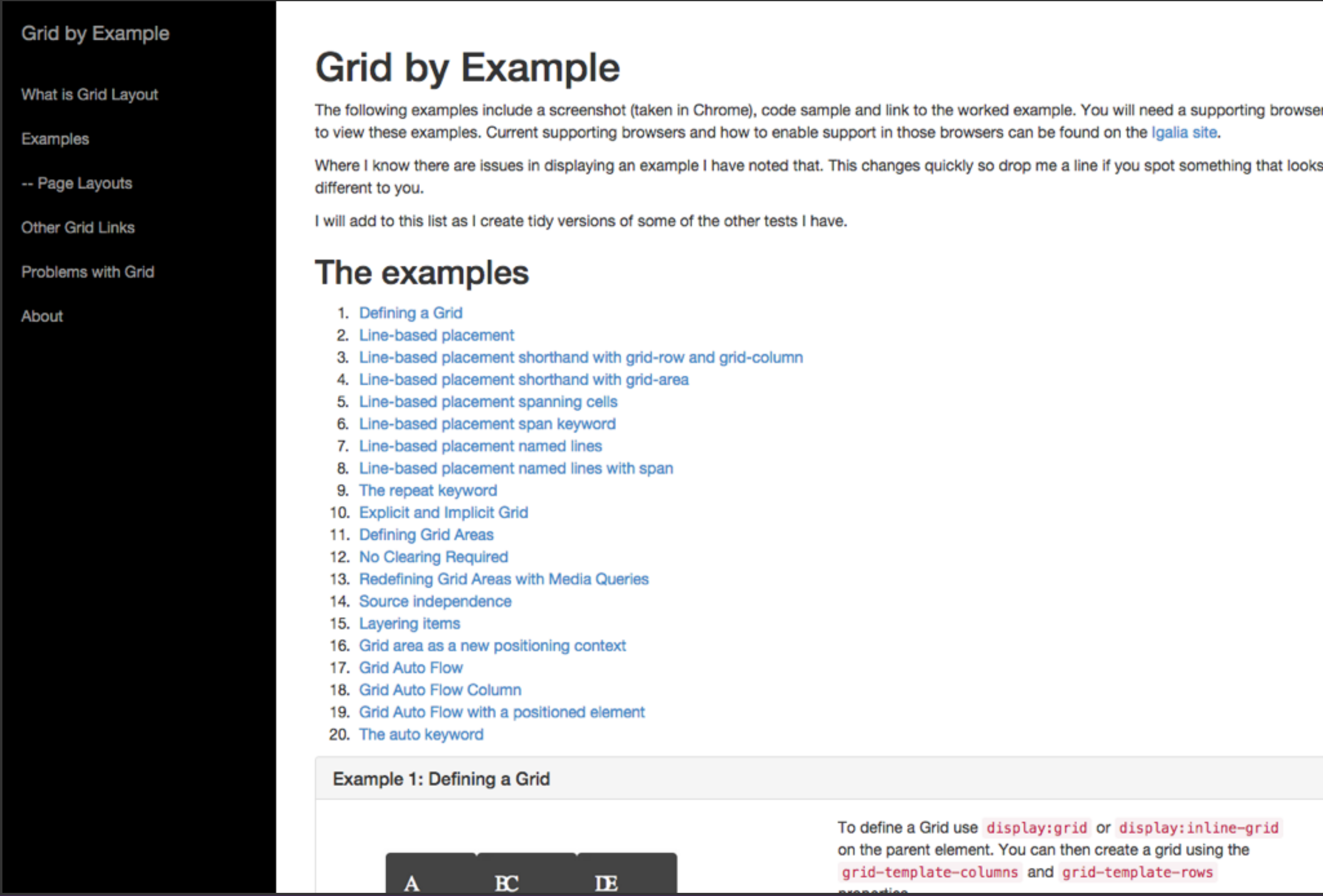
CSS Grid Layout

```html
<div class="wrapper">
  <div class="header">Header</div>
  <div class="sidebar">Sidebar</div>
  <div class="content">Content</div>
</div>
```

CSS Grid Layout

```
.sidebar {
  grid-area: sidebar;
}
.content {
  grid-area: content;
}
.header {
  grid-area: header;
}
.wrapper {
  display: grid;
  grid-template-columns:
    100px 10px 100px 10px 100px;
  grid-template-rows: auto;
  grid-template-areas:
   "header header header header header"
   "sidebar . content content content";
}
```


Example 11
Rachel
gridbyexample.com/examples/code/example11.html
Apps
Show Shapes
Header
Sidebar
Content



Grid by Example

The following examples include a screenshot (taken in Chrome), code sample and link to the worked example. You will need a supporting browse to view these examples. Current supporting browsers and how to enable support in those browsers can be found on the Igalia site.

Where I know there are issues in displaying an example I have noted that. This changes quickly so drop me a line if you spot something that looks different to you.

I will add to this list as I create tidy versions of some of the other tests I have.

The examples

1. Defining a Grid
2. Line-based placement
3. Line-based placement shorthand with grid-row and grid-column
4. Line-based placement shorthand with grid-area
5. Line-based placement spanning cells
6. Line-based placement span keyword
7. Line-based placement named lines
8. Line-based placement named lines with span
9. The repeat keyword
10. Explicit and Implicit Grid
11. Defining Grid Areas
12. No Clearing Required
13. Redefining Grid Areas with Media Queries
14. Source independence
15. Layering items
16. Grid area as a new positioning context
17. Grid Auto Flow
18. Grid Auto Flow Column
19. Grid Auto Flow with a positioned element
20. The auto keyword

Example 1: Defining a Grid

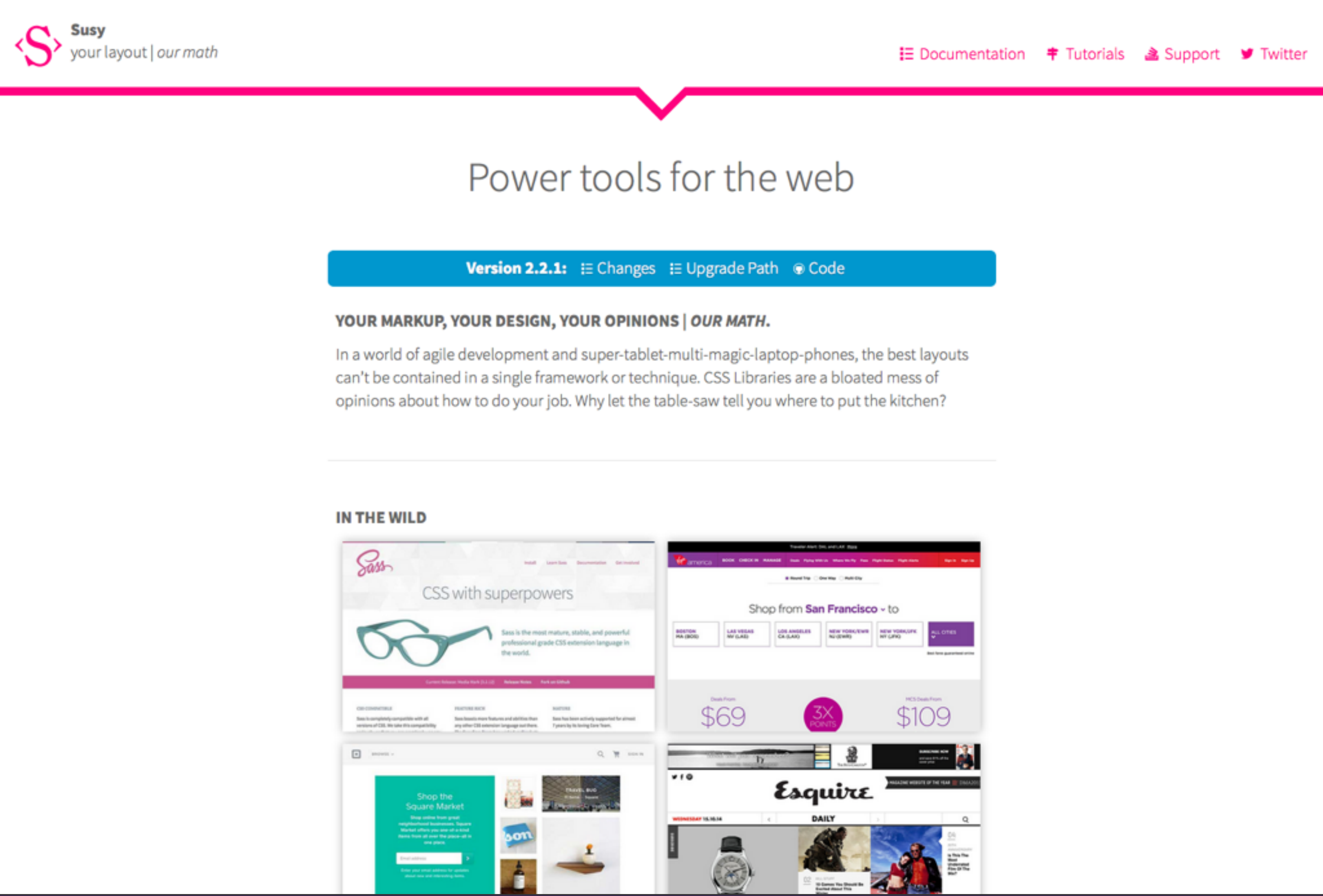
To define a Grid use `display:grid` or `display:inline-grid` on the parent element. You can then create a grid using the `grid-template-columns` and `grid-template-rows`

gridbyexample.com

“I’ll take a look if you create a Sass Mixin ...”

via my inbox.

Emerging specifications like Grid remove the need for some of the pre-processing



Susy
your layout | *our math*

Documentation Tutorials Support Twitter

Power tools for the web

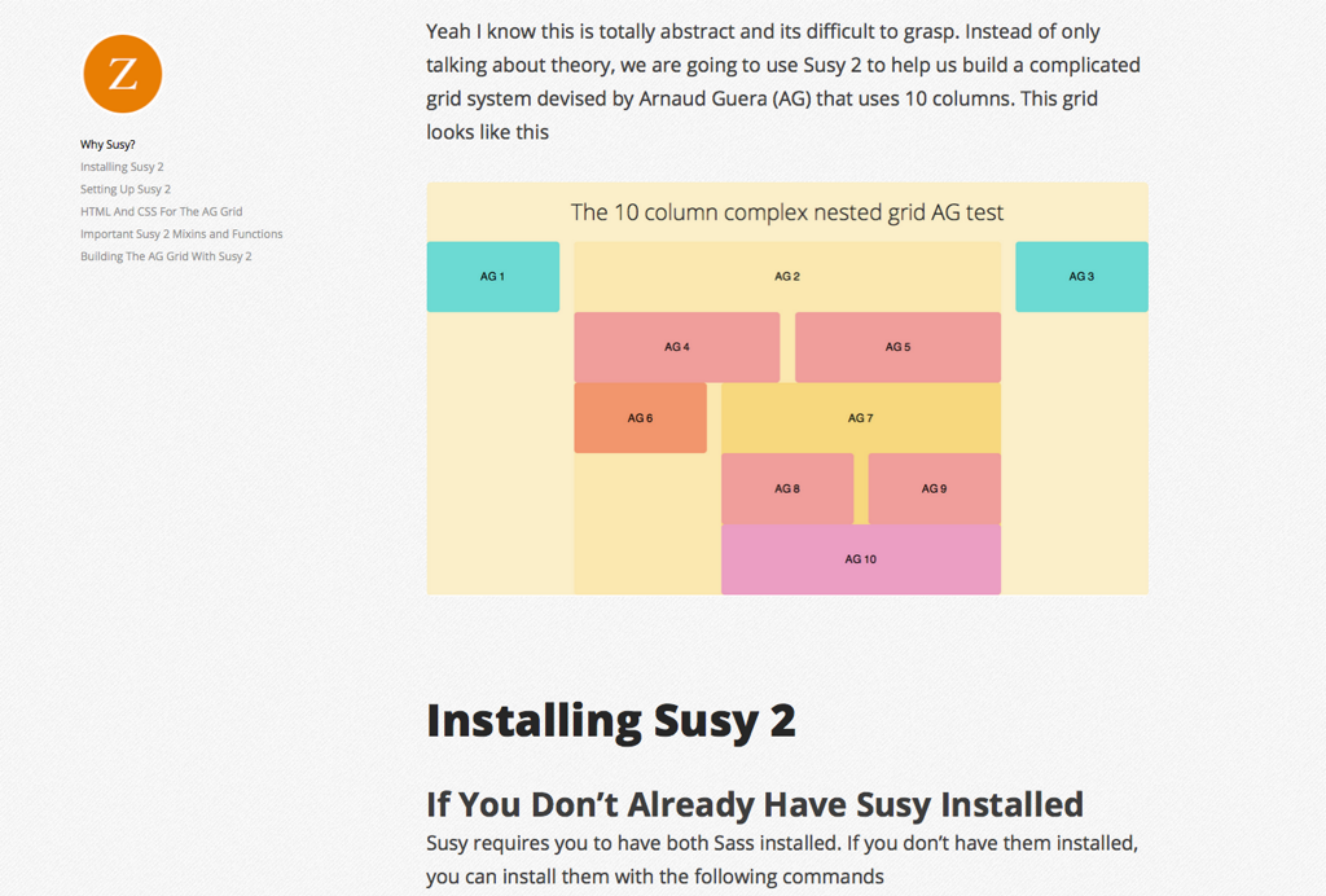
Version 2.2.1: Changes Upgrade Path Code

YOUR MARKUP, YOUR DESIGN, YOUR OPINIONS | *OUR MATH.*

In a world of agile development and super-tablet-multi-magic-laptop-phones, the best layouts can't be contained in a single framework or technique. CSS Libraries are a bloated mess of opinions about how to do your job. Why let the table-saw tell you where to put the kitchen?

IN THE WILD

http://susy.oddbird.net/



Yeah I know this is totally abstract and its difficult to grasp. Instead of only talking about theory, we are going to use Susy 2 to help us build a complicated grid system devised by Arnaud Guera (AG) that uses 10 columns. This grid looks like this

The 10 column complex nested grid AG test

Installing Susy 2

If You Don't Already Have Susy Installed

Susy requires you to have both Sass installed. If you don't have them installed, you can install them with the following commands

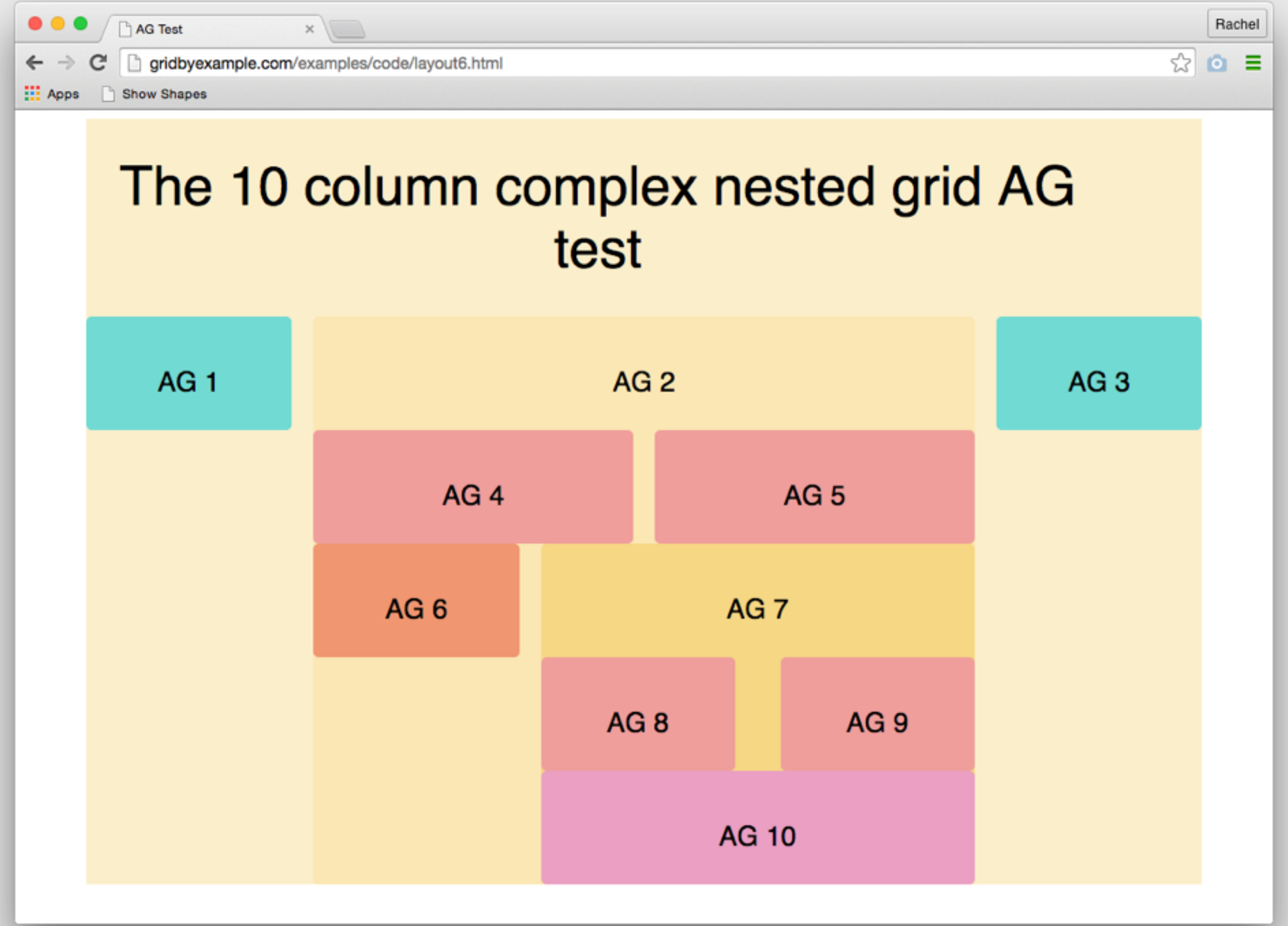
http://www.zell-weekeat.com/susy2-tutorial/

Using the Susy Mixins.

```
.ag1 {
  @include span(2 of 10);
}

.ag2 {
  @include span(6 of 10);
  @include clearfix;
}

.ag3 {
  @include span(2 of 10 last);
}
```



The 10 column complex nested grid AG test

Grid Layout lets you place elements on the Grid without calculations.

```
/* declare a grid and set up a 10 column grid
with gutters */
.container {
    width: 90%;
    margin: 0 auto 0 auto;
    display: grid;
    grid-template-columns:  (col ) 4.25fr
repeat(9, (gutter) 1fr (col) 4.25fr ) (gutter);
    grid-template-rows: auto repeat(5, 100px);
}

/* boxes positioned like so */
/* heading in row 1 full width */
h1 {
    grid-column: col / span col 10;
    grid-row: 1 / 2;
}

/* left hand sidebar */
.ag1 {
    grid-column: col / span gutter 2;
    grid-row: 2 / 3;
}
```

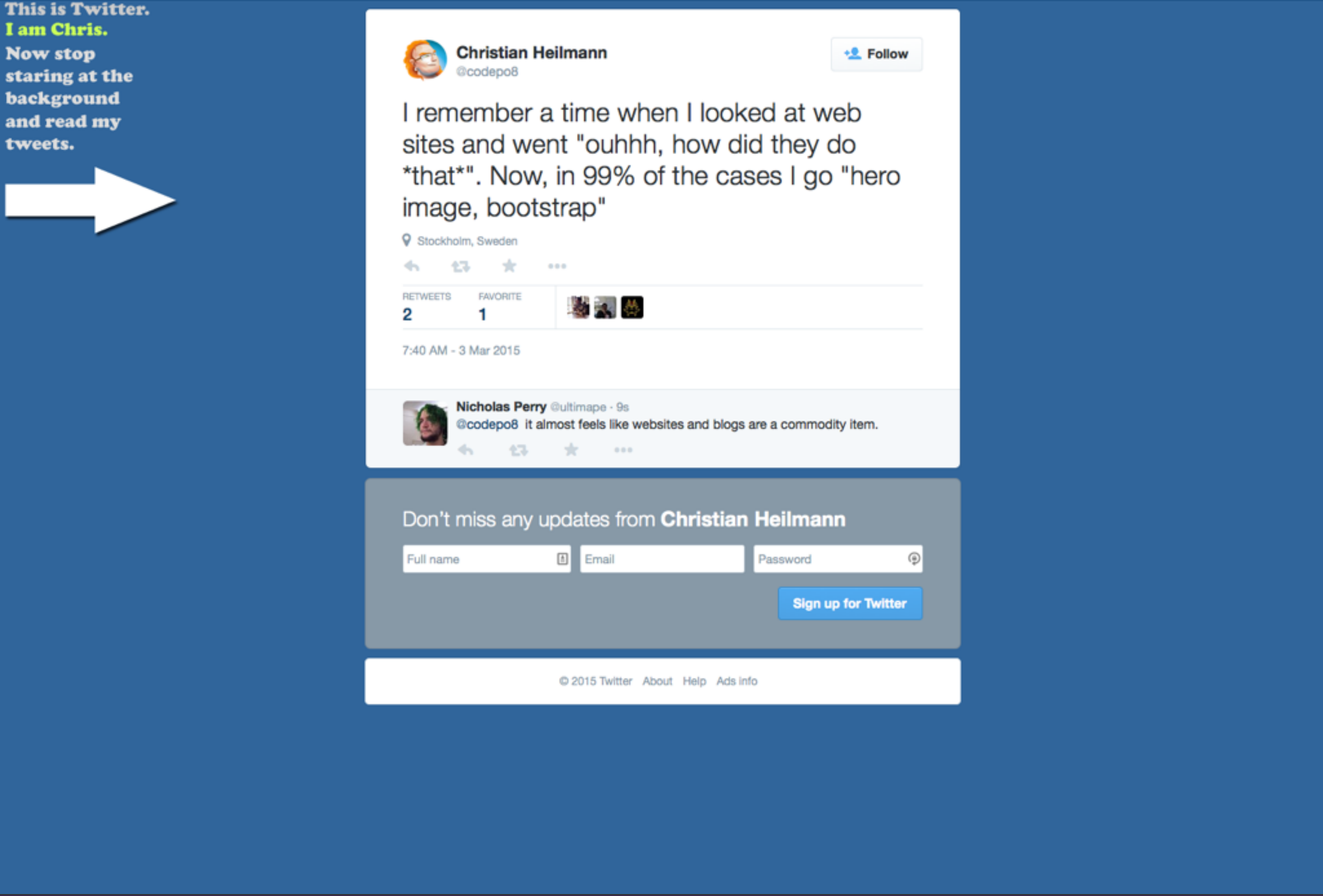
Web designers and developers should be excited by specifications like grid. **This is the future.**

By leaning on frameworks, are we masking the issues?

Working **with** the specifications we can contribute to improving them

Sheer frustration drove much of the Web Standards movement.

My fear is that due to our reliance on frameworks we will stop pushing for better solutions.



This is Twitter.
I am Chris.
Now stop staring at the background and read my tweets.

Christian Heilmann
@codepo8

I remember a time when I looked at web sites and went "ouhhh, how did they do *that*". Now, in 99% of the cases I go "hero image, bootstrap"

Stockholm, Sweden

RETWEETS	FAVORITE
2	1

7:40 AM - 3 Mar 2015

Nicholas Perry @ultimape · 9s
@codepo8 it almost feels like websites and blogs are a commodity item.

Don't miss any updates from **Christian Heilmann**

© 2015 Twitter About Help Ads info

https://twitter.com/codepo8/status/572783496550359040

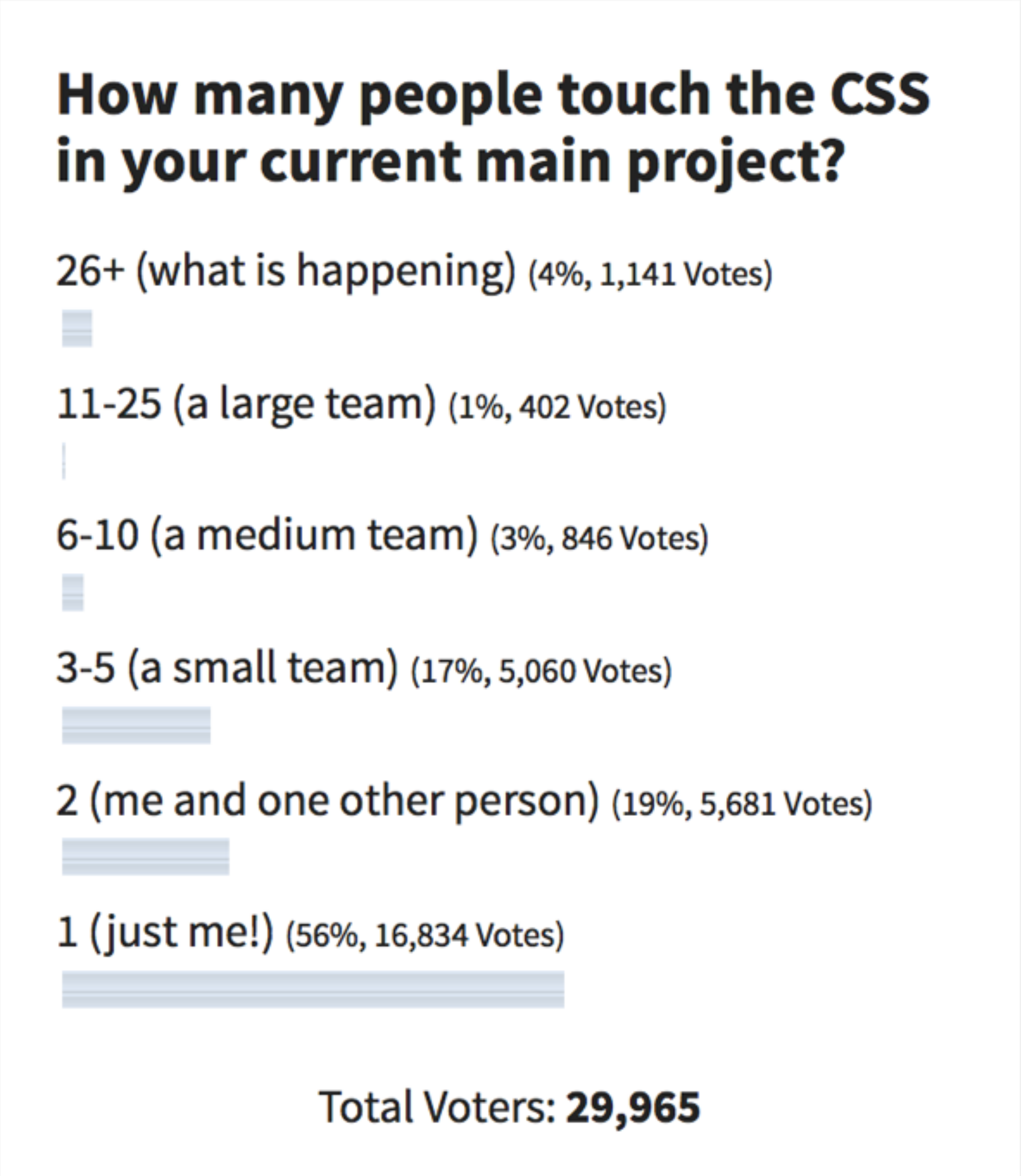
There are always compromises. They shouldn’t be the same for every project.

Standardising on tools should not be at the expense of learning HTML, CSS and JavaScript.

Use your tools and frameworks lightly. Be ready to put them aside when they don’t suit a project.

Don’t become an expert in one brand of hammer. Become a master carpenter. Develop timeless skills.

Considerations when choosing tools and processes.



How many people touch the CSS in your current main project?

26+ (what is happening) (4%, 1,141 Votes)

11-25 (a large team) (1%, 402 Votes)

6-10 (a medium team) (3%, 846 Votes)

3-5 (a small team) (17%, 5,060 Votes)

2 (me and one other person) (19%, 5,681 Votes)

1 (just me!) (56%, 16,834 Votes)

Total Voters: **29,965**

https://css-tricks.com

Is it responsible to use a brand new framework on a site you will complete then hand over?

Large teams and in-house projects often require more process than projects built by one or two people.

Who is the audience?

- Internal or External?
- Can we make any assumptions about technology used to access?

What browsers and devices are currently used to access the site?

What time do we have available?

Whose time are we saving?

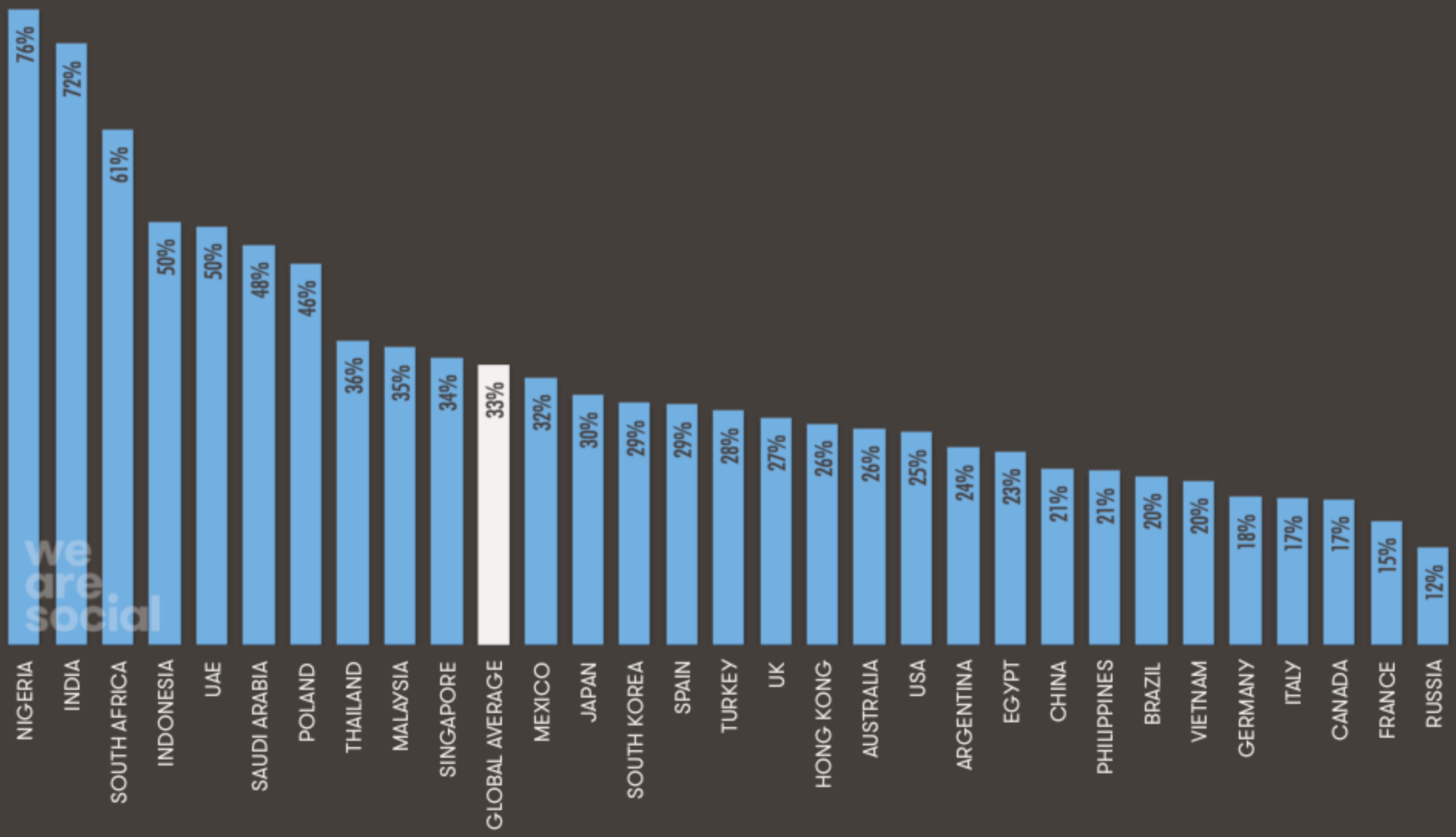
“When I look around, I see our community spending a lot of time coming up with new tools and techniques to make our jobs easier. To ship faster. And it’s not that I’m against efficiency, but I think we need to consider the implications of our decisions. And if one of those implications is making our users suffer—or potentially suffer—in order to make our lives easier, I think we need to consider their needs above our own.”

http://aaron-gustafson.com/notebook/who-should-pay/

JAN 2015

MOBILE'S SHARE OF WEB TRAFFIC

PERCENTAGE OF TOTAL WEB PAGES SERVED TO MOBILE PHONES



We Are Social: http://wearesocial.net/blog/2015/01/digital-social-mobile-worldwide-2015/

Will this tool …

- Save me time?
- Cause accessibility issues?
- Slow the site down on mobile?
- Limit the user agents that will be able to use the core experience?

It’s only temporary …

"This is for Everyone" by Nick Webb - Flickr: DSC_3232. Licensed under CC BY 2.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:This_is_for_Everyone.jpg#mediaviewer/File:This_is_for_Everyone.jpg

This is for everyone

Progressive enhancement

> *“a robust site or application in the more traditional sense minimises its dependencies. The minimum dependency for a web site should be an internet connection and the ability to parse HTML.”*

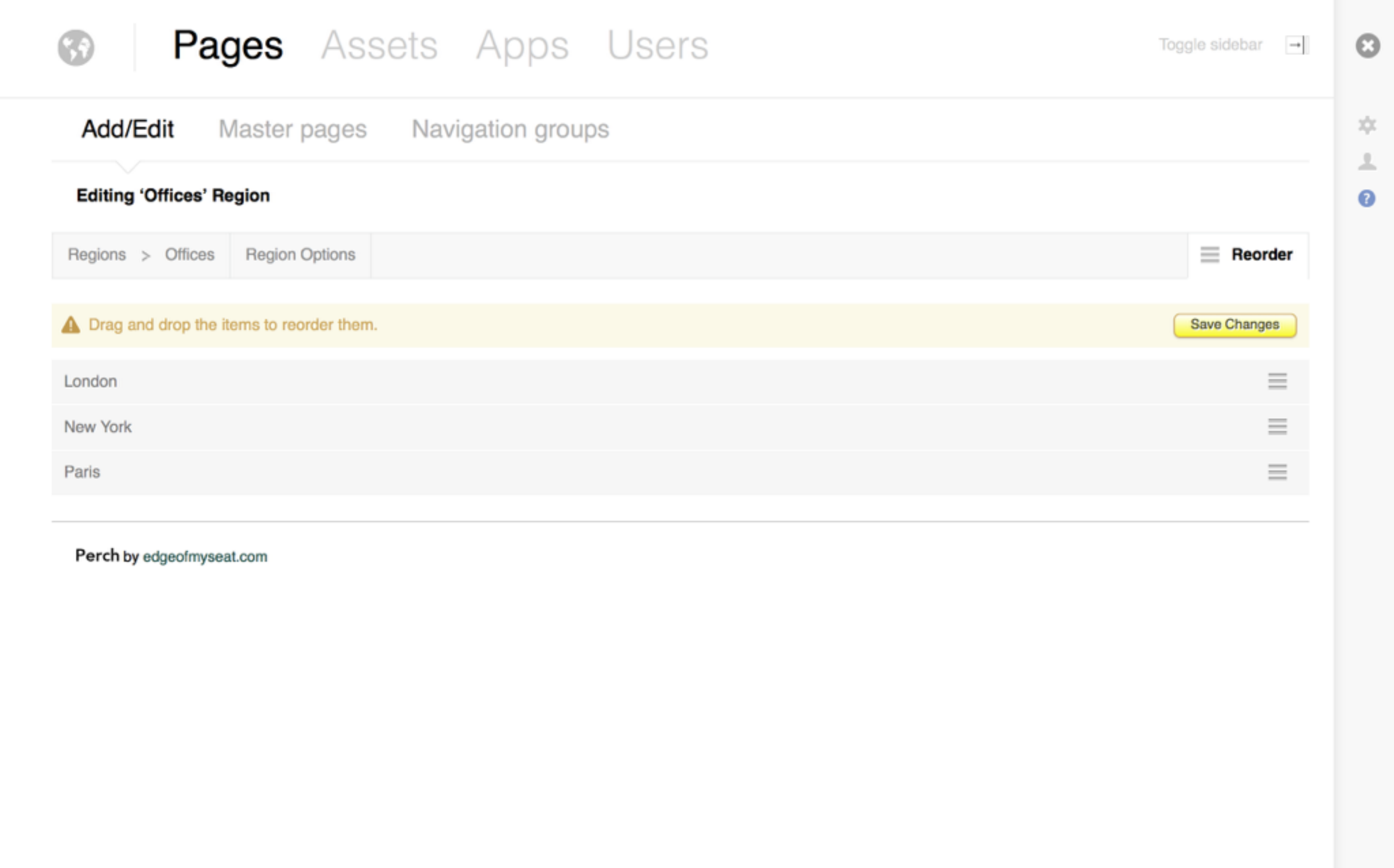
http://www.bbc.co.uk/guidelines/futuremedia/accessibility/html/progressive-enhancement.shtml

Start with the **core experience**

What is the minimum that I need to ship? How can I ensure that minimum protects the core experience for everyone?

We ship. We iterate.

grabaperch.com



Editing ‘Offices’ Region

Drag and drop the items to reorder them.

London

New York

Paris

How to integrate third party code

Not Invented Here

“Are you afraid to write code? Does the thought linger in your brain that somewhere out there somebody has already done this? Do you find yourself trapped in an analysis cycle where nothing is getting done? Is your product mutating to accommodate third party components? If yes, then perhaps you are suffering from invented-here syndrome.”

http://mortoray.com/2015/02/25/invented-here-syndrome/

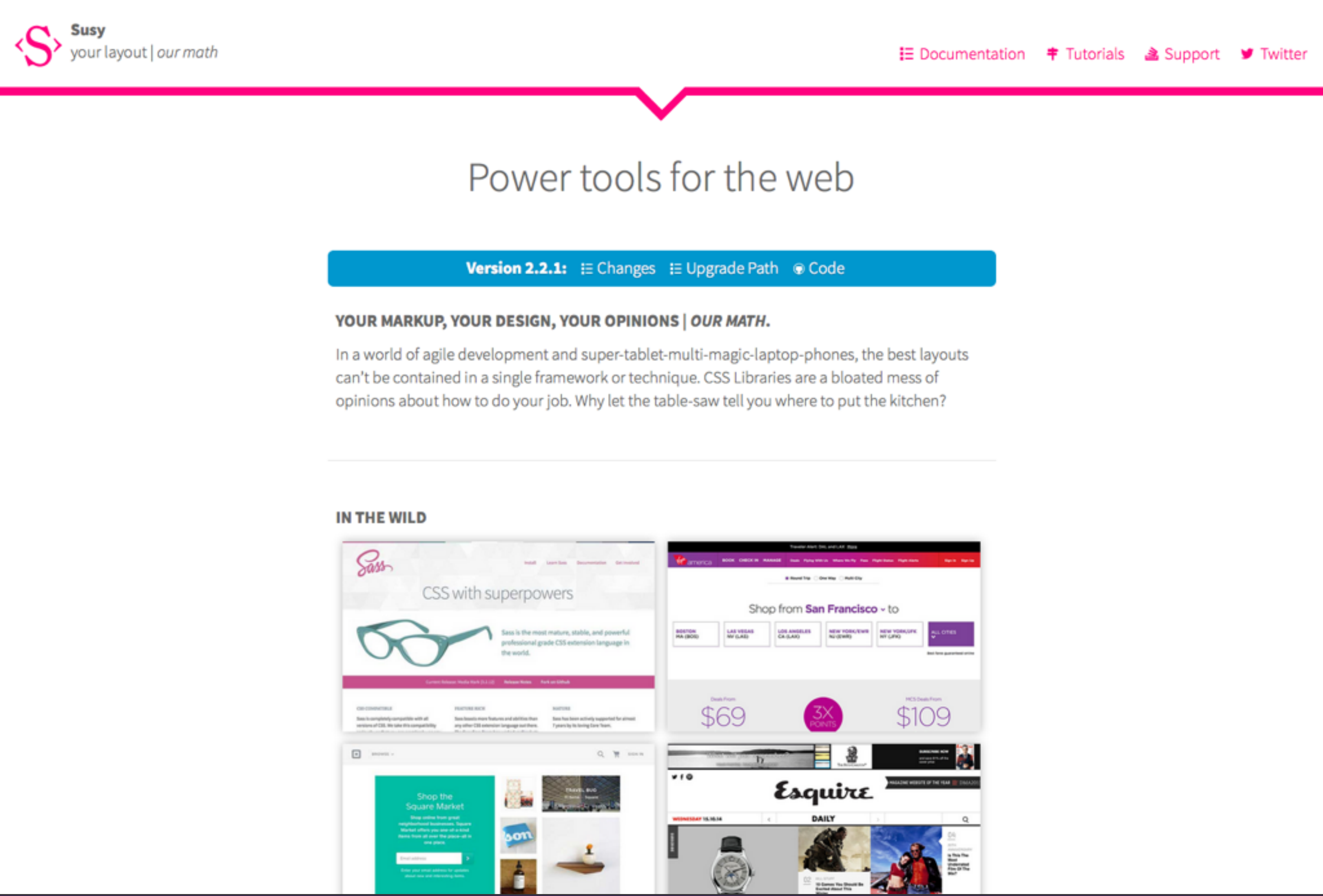
Avoid turning shortcuts and third party code into dependencies.

Dependency Inversion

“High level modules should not depend upon low-level modules. Both should depend upon abstractions.

Abstractions should never depend upon details. Details should depend upon abstractions.”

http://www.objectmentor.com/resources/articles/dip.pdf



Power tools for the web

YOUR MARKUP, YOUR DESIGN, YOUR OPINIONS | *OUR MATH*.

In a world of agile development and super-tablet-multi-magic-laptop-phones, the best layouts can't be contained in a single framework or technique. CSS Libraries are a bloated mess of opinions about how to do your job. Why let the table-saw tell you where to put the kitchen?

IN THE WILD

http://susy.oddbird.net/

Progressively enhanced UI

- JavaScript implementation based on the regular HTML5 Video element
- Static maps that become draggable and zoomable - avoiding creating a dependency on one maps provider or library
- Ordering items via a form input - that become drag and drop if the user has JavaScript

You can’t do everything.

You can do something.

“A 100% pure progressively-enhanced website may not be practical on every single project you will ever encounter. While that sort of purity can exist, it’s unlikely in many business scenarios. Budgets, timelines: these things exist. Progressive enhancement isn’t a zero sum game; it’s a continuum, just like the Web.”

http://sixtwothree.org/posts/the-practical-case-for-progressive-enhancement

- Learn (and teach!) core skills. HTML, CSS, JavaScript
- Maintain an interest in emerging specifications
- Take care that you are not clinging to outdated or unhelpful abstractions
- We are no longer browser bug wranglers, instead we should be experts in performance especially as the web becomes ever more mobile

- Choose your tools and frameworks on a case by case basis
- Understand the compromises
- Don’t reinvent wheels …
- … but beware “invented here syndrome”
- Use progressive enhancement to protect the core experience while shipping quickly, build from there.

We get to create products that
people see, touch & interact with.

“We don’t stop playing because we grow old; we grow old because we stop playing”

George Bernard Shaw

Thank you!

Rachel Andrew

@rachelandrew

http://rachelandrew.co.uk/presentations/business-of-front-end