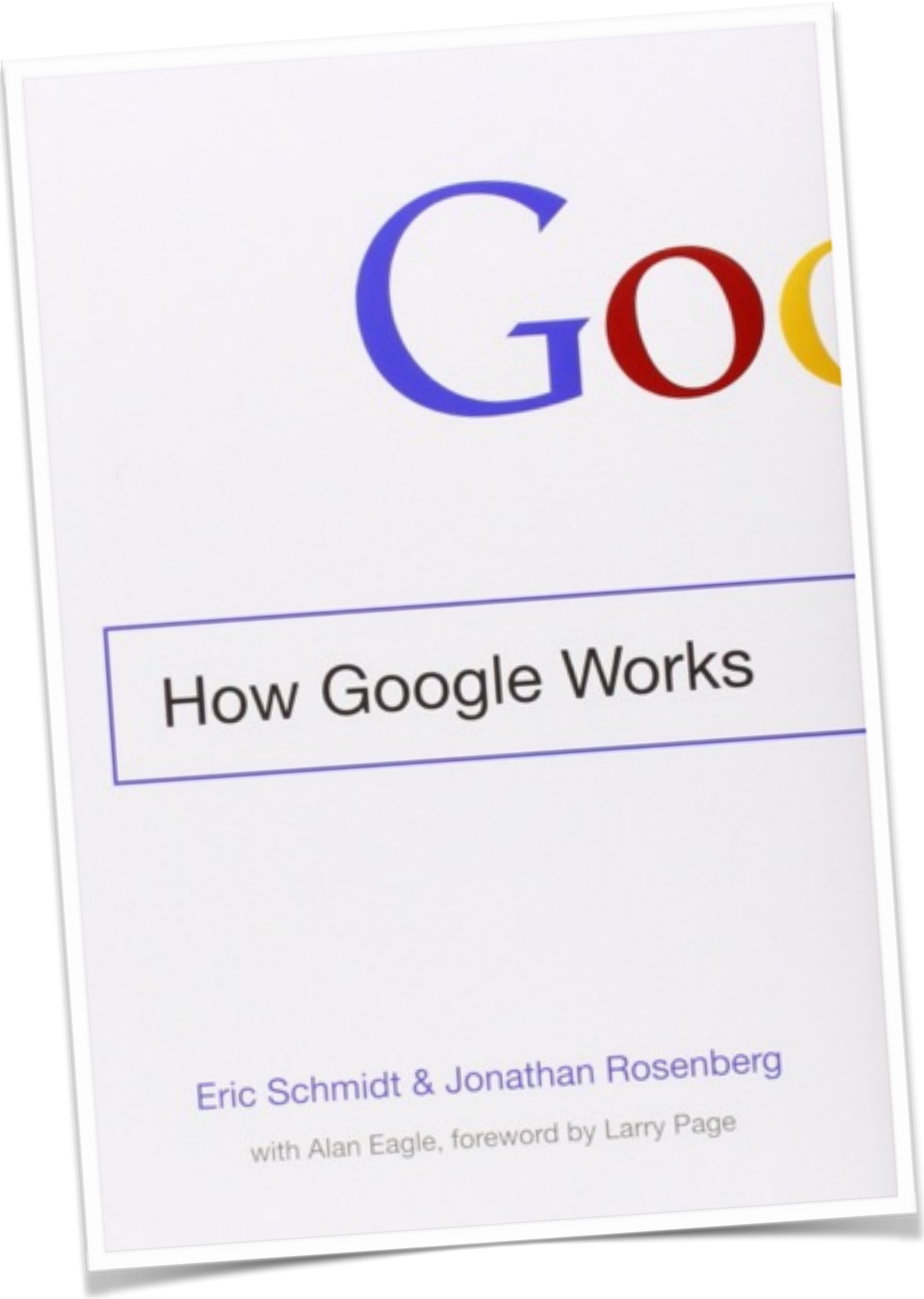
How to Take **Web Design** to the ***Next Level***

@aficiomaquinas

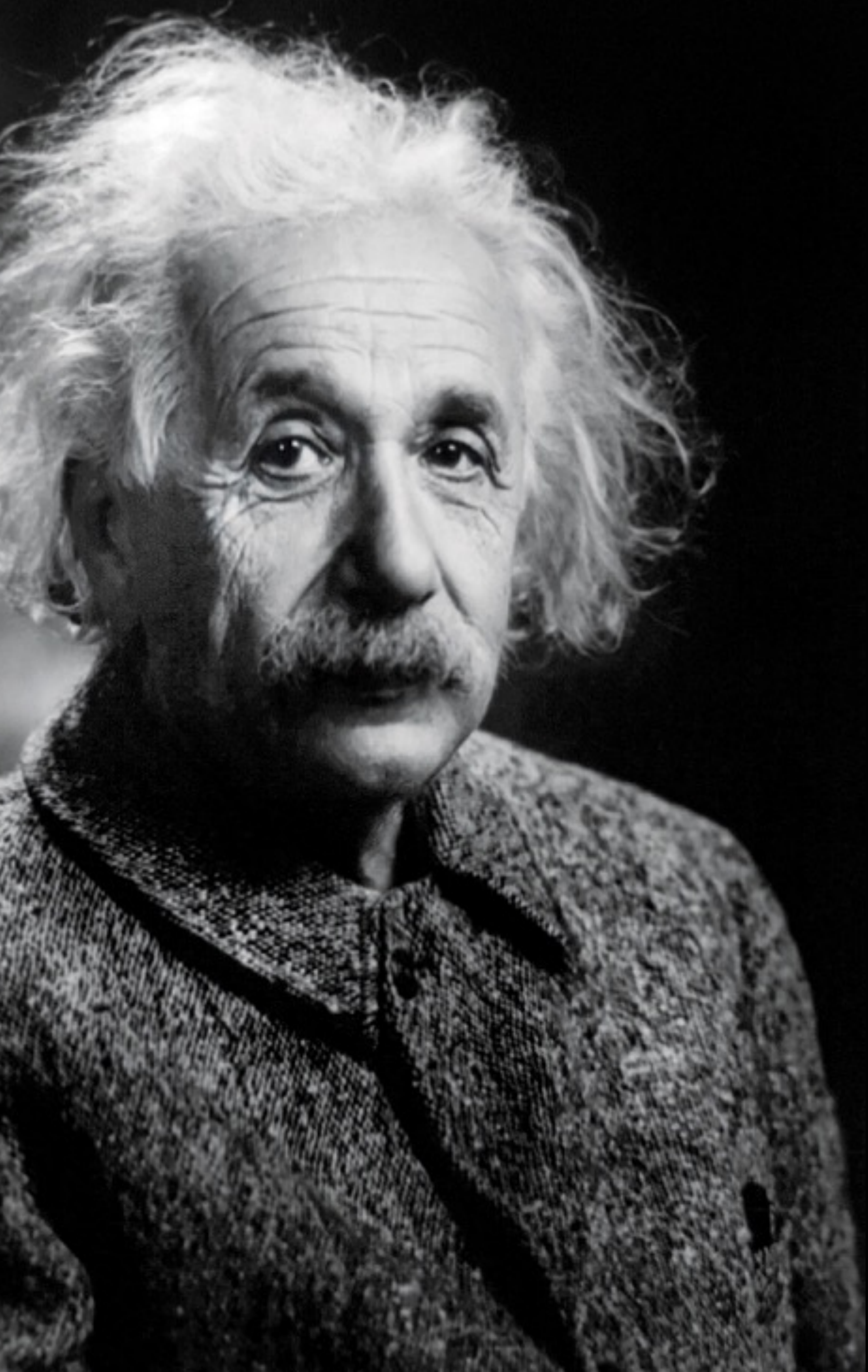


How Google Works

Eric Schmidt & Jonathan Rosenberg

with Alan Eagle, foreword by Larry Page

“Information is cheap.”



“[I do not] carry such information in my mind since it is readily available in books.

[…] The value of a college education is not the learning of many facts, but the training of the mind to think.”

–Albert Einstein

“In any interesting domain, the difficulties will be novel. Which means you can't simply plow through them, because you don't know initially how hard they are; you don't know whether you're about to plow through a block of foam or granite. So you have to be resourceful. You have to keep trying new things.”

Be relentlessly resourceful.

–Paul Graham

Y Combinator Co-founder

Lazy memory allocation

+

Relentlessly-resourcefulness

Selectively Lazy

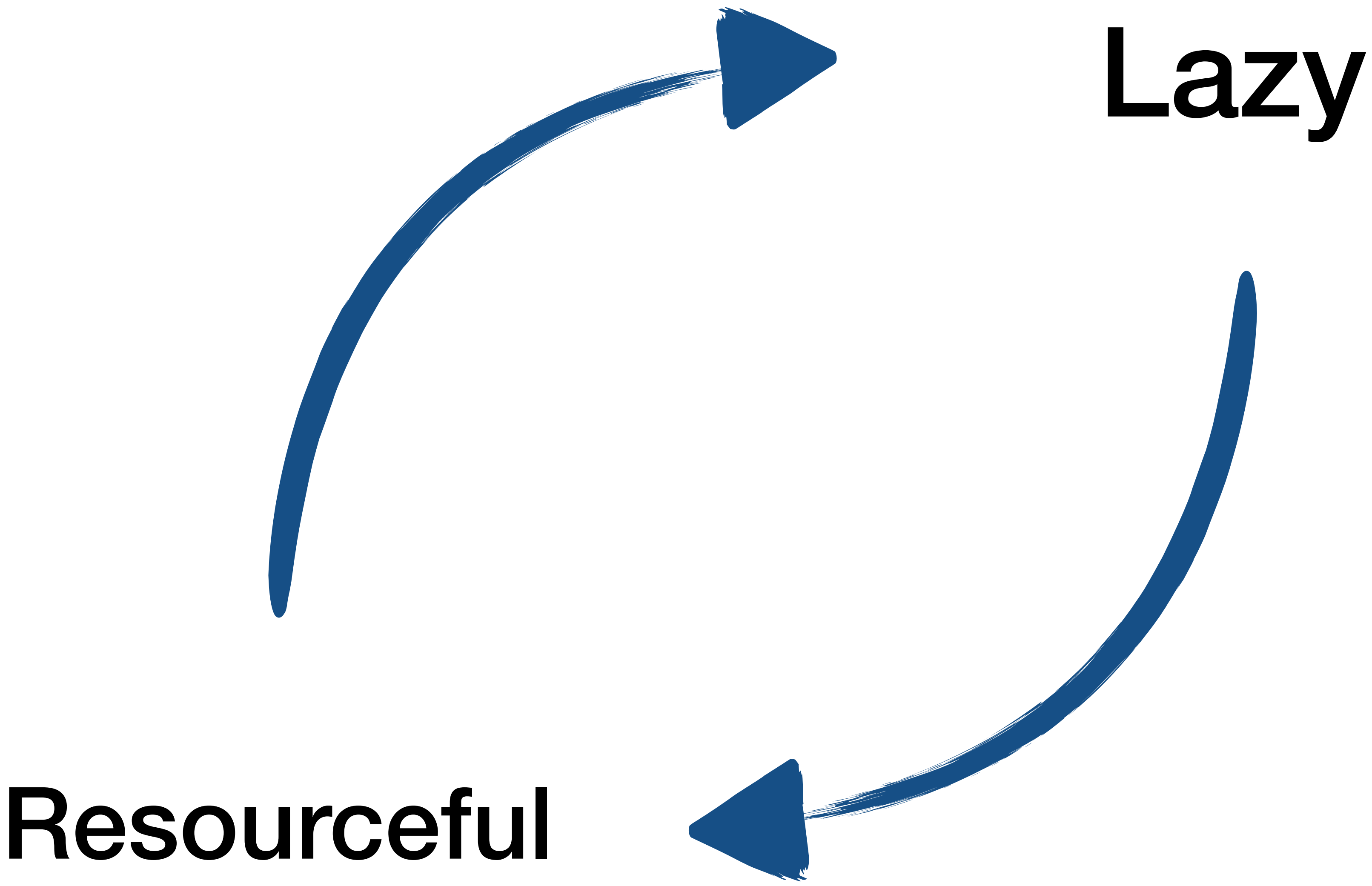
+ = *The Hacker Way*

Resourceful

The Hacker Way

- **Selectively Lazy**
  - Definir nuestros objetivos. Delimitar lo que *no* nos interesa lograr.
  - Aprender y profundizar sólo lo necesario para ser productivos y llegar al siguiente milestone rápidamente.[1]
  - Es más importante la aplicación práctica que el fundamento teórico.[2]
  - No penalizar nuestros errores, aprender de ellos.[3]
- **Resourceful**
  - Con base en nuestros objetivos, investigar las implicaciones y estructura del problema. Con esto podremos definir milestones y buscar en los lugares correctos.[4]
  - Priorizar la búsqueda sobre la experimentación *brute-force*.
  - Enfocarnos en comprender cómo y porqué funcionan las cosas, no tanto en juzgar su diseño y razón de ser.

[1] *Dentro de los límites del buen gusto.*
[2] *Comprender exhaustiva y profundamente todas las herramientas tomaría mucho tiempo.*
[3] *Inclusive, es menester re-significarlo como un proceso de mejora continua, iteración.*
[4] *Mejora con el tiempo y la experiencia.*



Lazy

Resourceful

The Status quo

HTML5 & CSS3

N
E
S
W

Priority of Constituencies

- “In case of conflict, consider users over authors over implementors over specifiers over theoretical purity.”

1
A BOOK APART
Jeremy Keith
HTML5 FOR WEB DESIGNERS
FOREWORD BY Jeffrey Zeldman

Living standard ⇒ ~~HTML6~~

Living standard ⇒ Always changing

Web ⇒ Always changing

I CAN ONLY SHOW YOU THE DOOR...

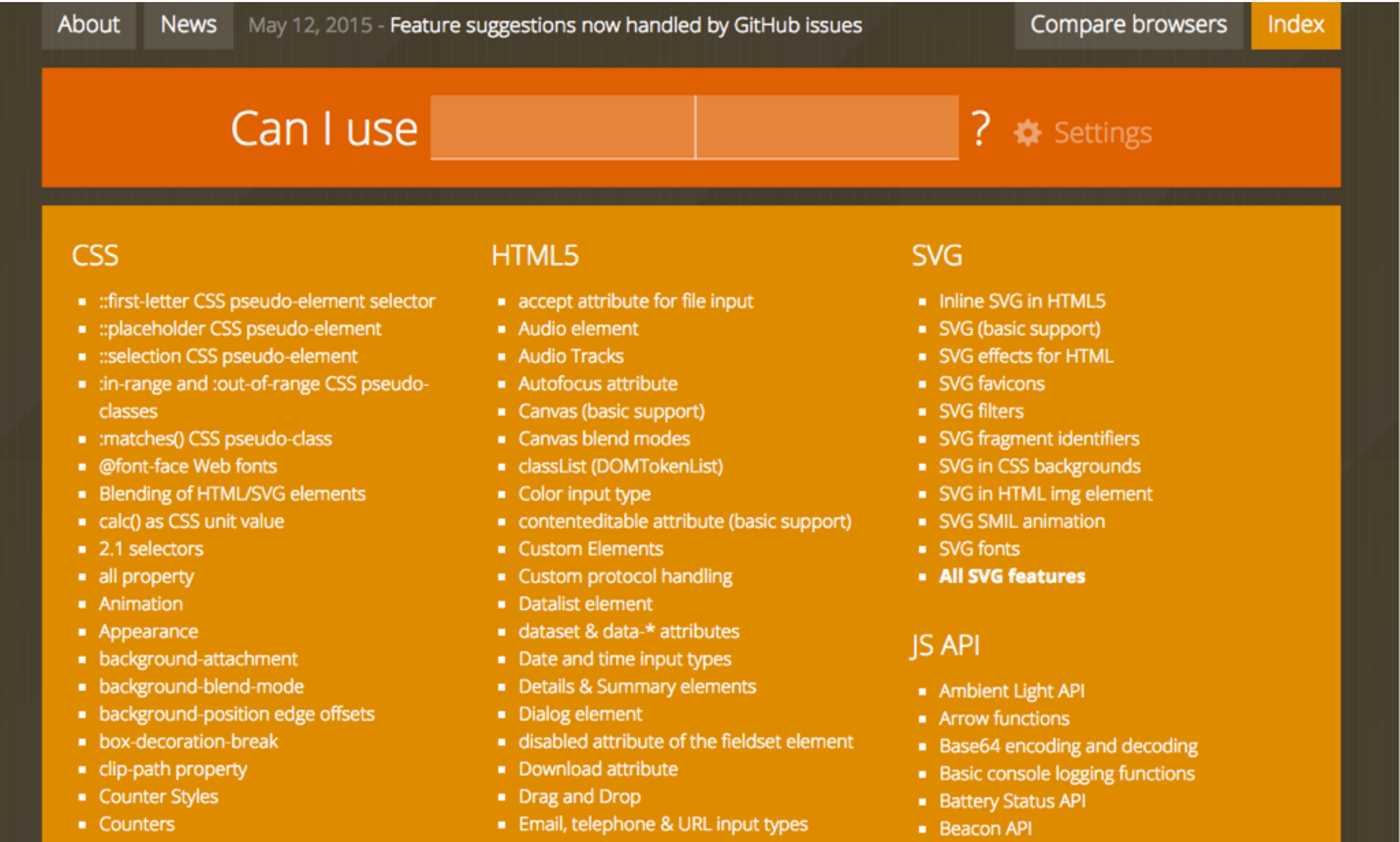
YOU'RE THE ONE THAT HAS TO WALK THROUGH IT

1. The right method
2. The right tools & technologies
3. The right workflow
4. The right application of design principles

Method

- Leverage new features
- Be semantically correct
- Graceful Degradation vs Progressive Enhancement
- Think responsive
- Think about interaction with different devices (mobile first?)
- Think about performance
- Think about UX

caniuse.com



May 12, 2015 - Feature suggestions now handled by GitHub issues

CSS

- ::first-letter CSS pseudo-element selector
- ::placeholder CSS pseudo-element
- ::selection CSS pseudo-element
- :in-range and :out-of-range CSS pseudo-classes
- :matches() CSS pseudo-class
- @font-face Web fonts
- Blending of HTML/SVG elements
- calc() as CSS unit value
- 2.1 selectors
- all property
- Animation
- Appearance
- background-attachment
- background-blend-mode
- background-position edge offsets
- box-decoration-break
- clip-path property
- Counter Styles
- Counters

HTML5

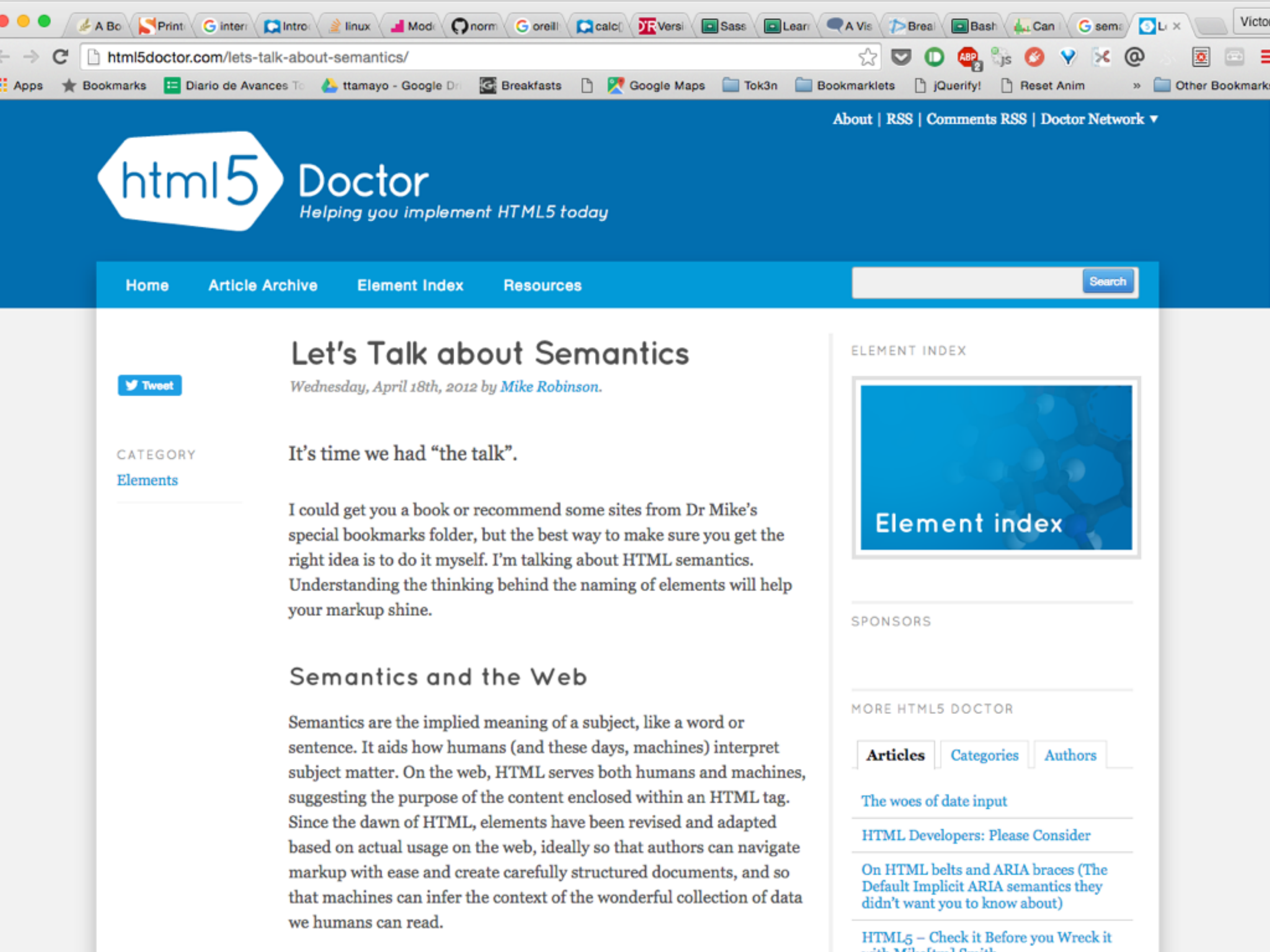
- accept attribute for file input
- Audio element
- Audio Tracks
- Autofocus attribute
- Canvas (basic support)
- Canvas blend modes
- classList (DOMTokenList)
- Color input type
- contenteditable attribute (basic support)
- Custom Elements
- Custom protocol handling
- Datalist element
- dataset & data-* attributes
- Date and time input types
- Details & Summary elements
- Dialog element
- disabled attribute of the fieldset element
- Download attribute
- Drag and Drop
- Email, telephone & URL input types

SVG

- Inline SVG in HTML5
- SVG (basic support)
- SVG effects for HTML
- SVG favicons
- SVG filters
- SVG fragment identifiers
- SVG in CSS backgrounds
- SVG in HTML img element
- SVG SMIL animation
- SVG fonts
- **All SVG features**

JS API

- Ambient Light API
- Arrow functions
- Base64 encoding and decoding
- Basic console logging functions
- Battery Status API
- Beacon API



Let's Talk about Semantics

Wednesday, April 18th, 2012 by Mike Robinson.

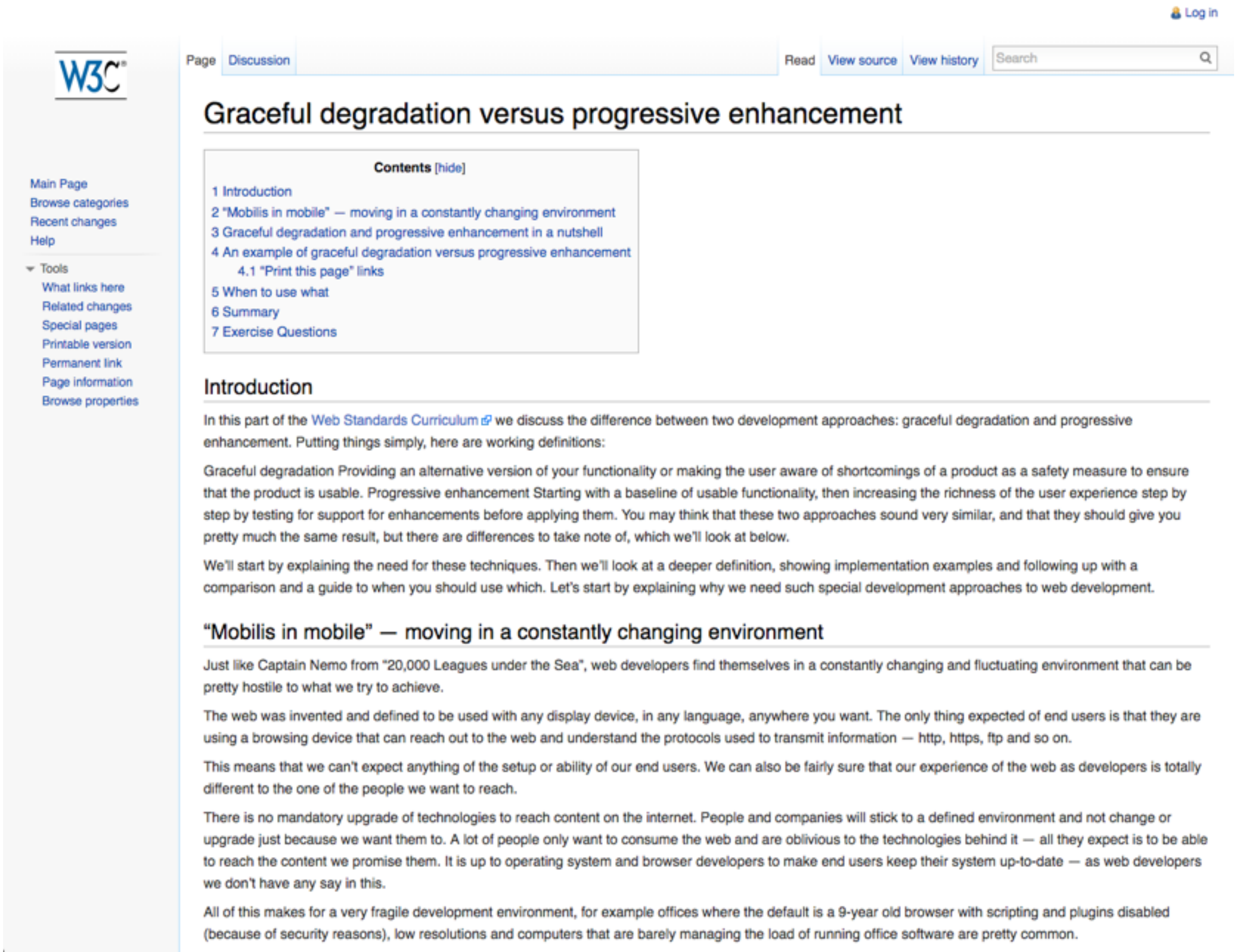
It's time we had "the talk".

I could get you a book or recommend some sites from Dr Mike's special bookmarks folder, but the best way to make sure you get the right idea is to do it myself. I'm talking about HTML semantics. Understanding the thinking behind the naming of elements will help your markup shine.

Semantics and the Web

Semantics are the implied meaning of a subject, like a word or sentence. It aids how humans (and these days, machines) interpret subject matter. On the web, HTML serves both humans and machines, suggesting the purpose of the content enclosed within an HTML tag. Since the dawn of HTML, elements have been revised and adapted based on actual usage on the web, ideally so that authors can navigate markup with ease and create carefully structured documents, and so that machines can infer the context of the wonderful collection of data we humans can read.

https://www.w3.org/wiki/Graceful_degradation_versus_progressive_enhancement



Graceful degradation versus progressive enhancement

Contents [hide]

Introduction

In this part of the Web Standards Curriculum we discuss the difference between two development approaches: graceful degradation and progressive enhancement. Putting things simply, here are working definitions:

Graceful degradation Providing an alternative version of your functionality or making the user aware of shortcomings of a product as a safety measure to ensure that the product is usable. Progressive enhancement Starting with a baseline of usable functionality, then increasing the richness of the user experience step by step by testing for support for enhancements before applying them. You may think that these two approaches sound very similar, and that they should give you pretty much the same result, but there are differences to take note of, which we'll look at below.

We'll start by explaining the need for these techniques. Then we'll look at a deeper definition, showing implementation examples and following up with a comparison and a guide to when you should use which. Let's start by explaining why we need such special development approaches to web development.

"Mobilis in mobile" — moving in a constantly changing environment

Just like Captain Nemo from "20,000 Leagues under the Sea", web developers find themselves in a constantly changing and fluctuating environment that can be pretty hostile to what we try to achieve.

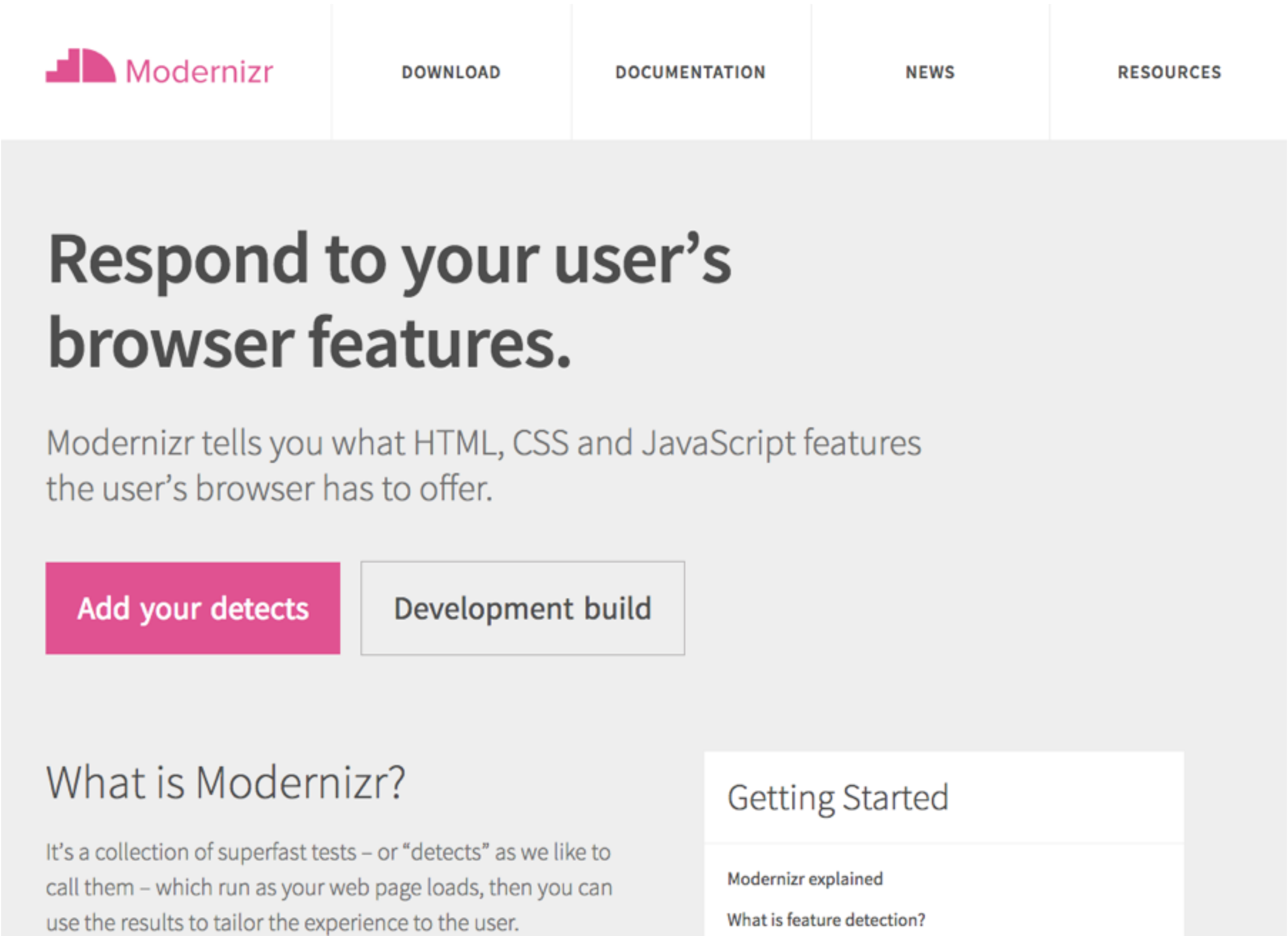
The web was invented and defined to be used with any display device, in any language, anywhere you want. The only thing expected of end users is that they are using a browsing device that can reach out to the web and understand the protocols used to transmit information — http, https, ftp and so on.

This means that we can't expect anything of the setup or ability of our end users. We can also be fairly sure that our experience of the web as developers is totally different to the one of the people we want to reach.

There is no mandatory upgrade of technologies to reach content on the internet. People and companies will stick to a defined environment and not change or upgrade just because we want them to. A lot of people only want to consume the web and are oblivious to the technologies behind it — all they expect is to be able to reach the content we promise them. It is up to operating system and browser developers to make end users keep their system up-to-date — as web developers we don't have any say in this.

All of this makes for a very fragile development environment, for example offices where the default is a 9-year old browser with scripting and plugins disabled (because of security reasons), low resolutions and computers that are barely managing the load of running office software are pretty common.

https://modernizr.com/



Respond to your user's browser features.

Modernizr tells you what HTML, CSS and JavaScript features the user's browser has to offer.

What is Modernizr?

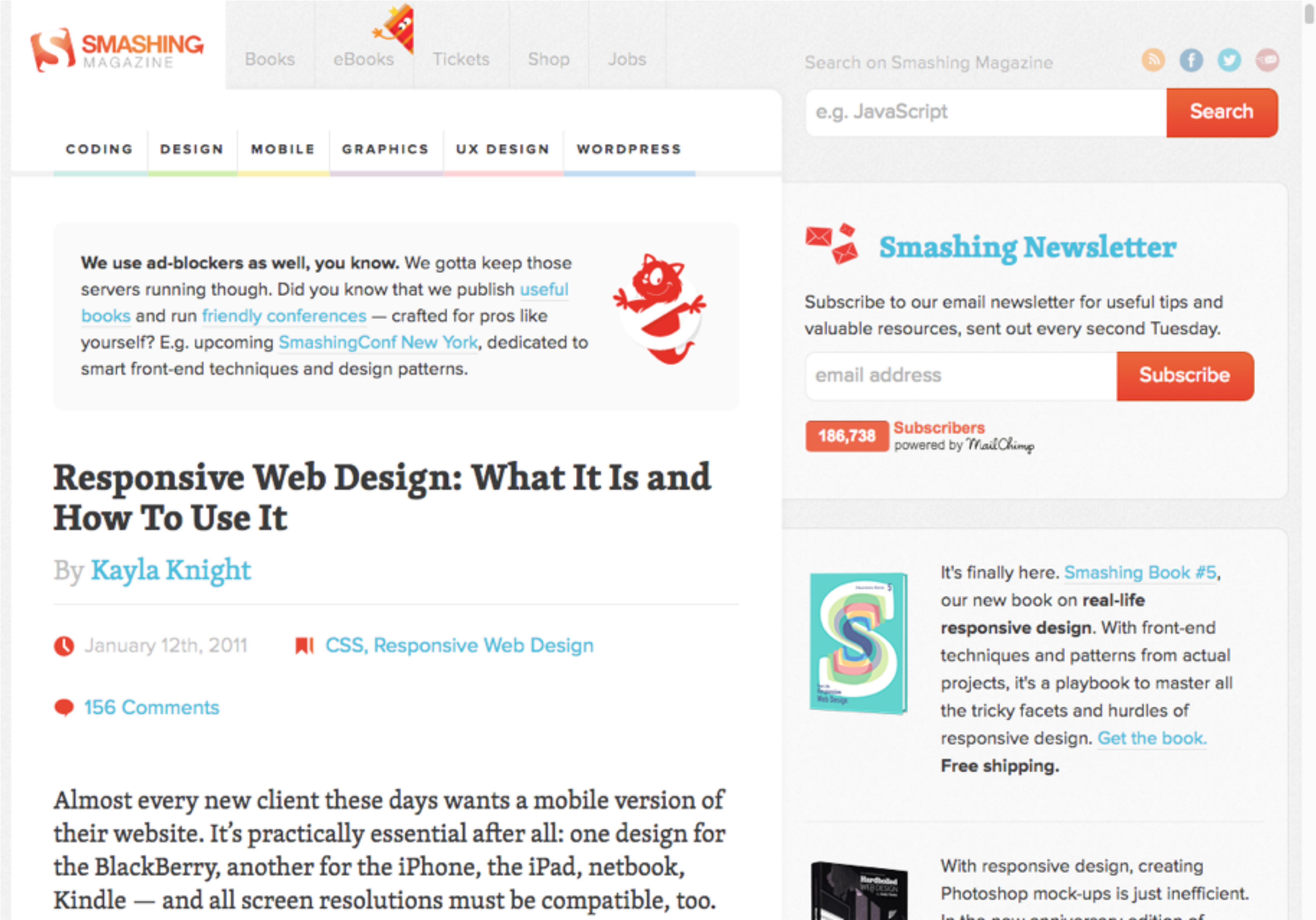
It's a collection of superfast tests – or "detects" as we like to call them – which run as your web page loads, then you can use the results to tailor the experience to the user.

Getting Started

Modernizr explained

What is feature detection?

https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/



We use ad-blockers as well, you know. We gotta keep those servers running though. Did you know that we publish useful books and run friendly conferences — crafted for pros like yourself? E.g. upcoming SmashingConf New York, dedicated to smart front-end techniques and design patterns.

Responsive Web Design: What It Is and How To Use It

By Kayla Knight

January 12th, 2011 CSS, Responsive Web Design

156 Comments

Almost every new client these days wants a mobile version of their website. It's practically essential after all: one design for the BlackBerry, another for the iPhone, the iPad, netbook, Kindle — and all screen resolutions must be compatible, too.

Smashing Newsletter

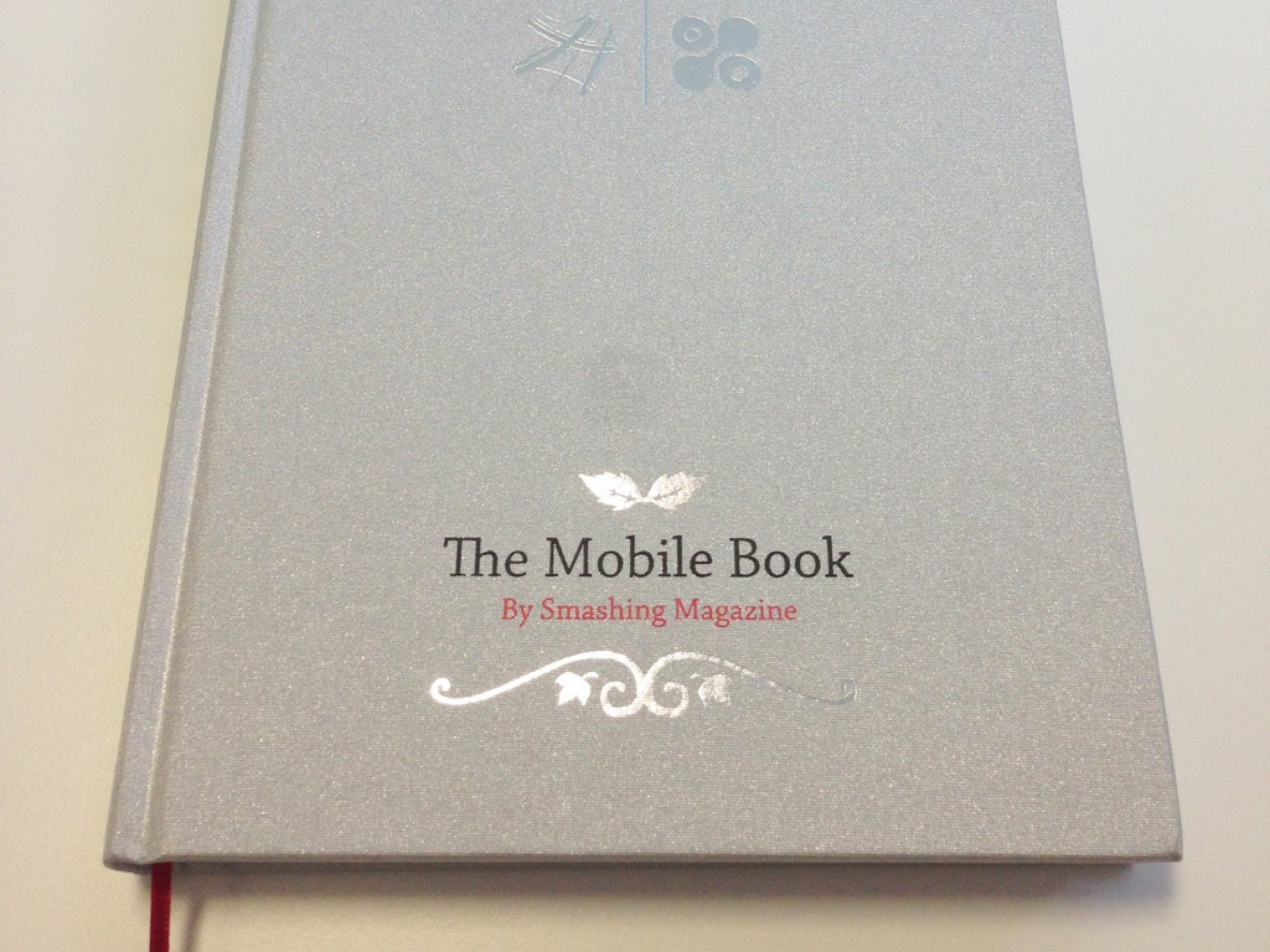
Subscribe to our email newsletter for useful tips and valuable resources, sent out every second Tuesday.

186,738 Subscribers powered by MailChimp

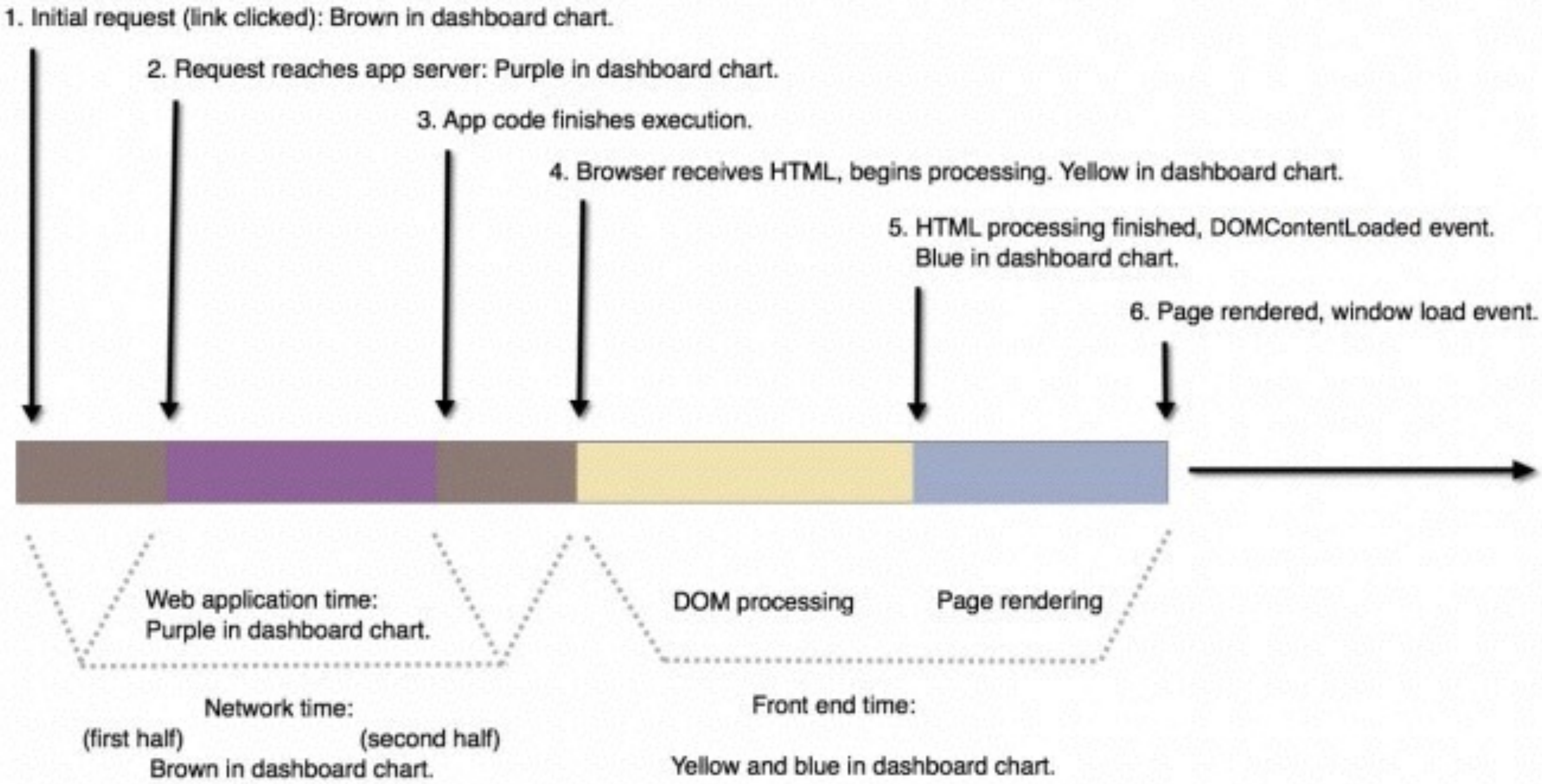
It's finally here. Smashing Book #5, our new book on **real-life responsive design**. With front-end techniques and patterns from actual projects, it's a playbook to master all the tricky facets and hurdles of responsive design. Get the book. **Free shipping.**

With responsive design, creating Photoshop mock-ups is just inefficient.

Real-Life
Responsive
Web Design
...nsive Web Design


The Mobile Book
By Smashing Magazine

Page load timeline



1. Initial request (link clicked): Brown in dashboard chart.

2. Request reaches app server: Purple in dashboard chart.

3. App code finishes execution.

4. Browser receives HTML, begins processing. Yellow in dashboard chart.

5. HTML processing finished, DOMContentLoaded event.
Blue in dashboard chart.

6. Page rendered, window load event.

Web application time:
Purple in dashboard chart.

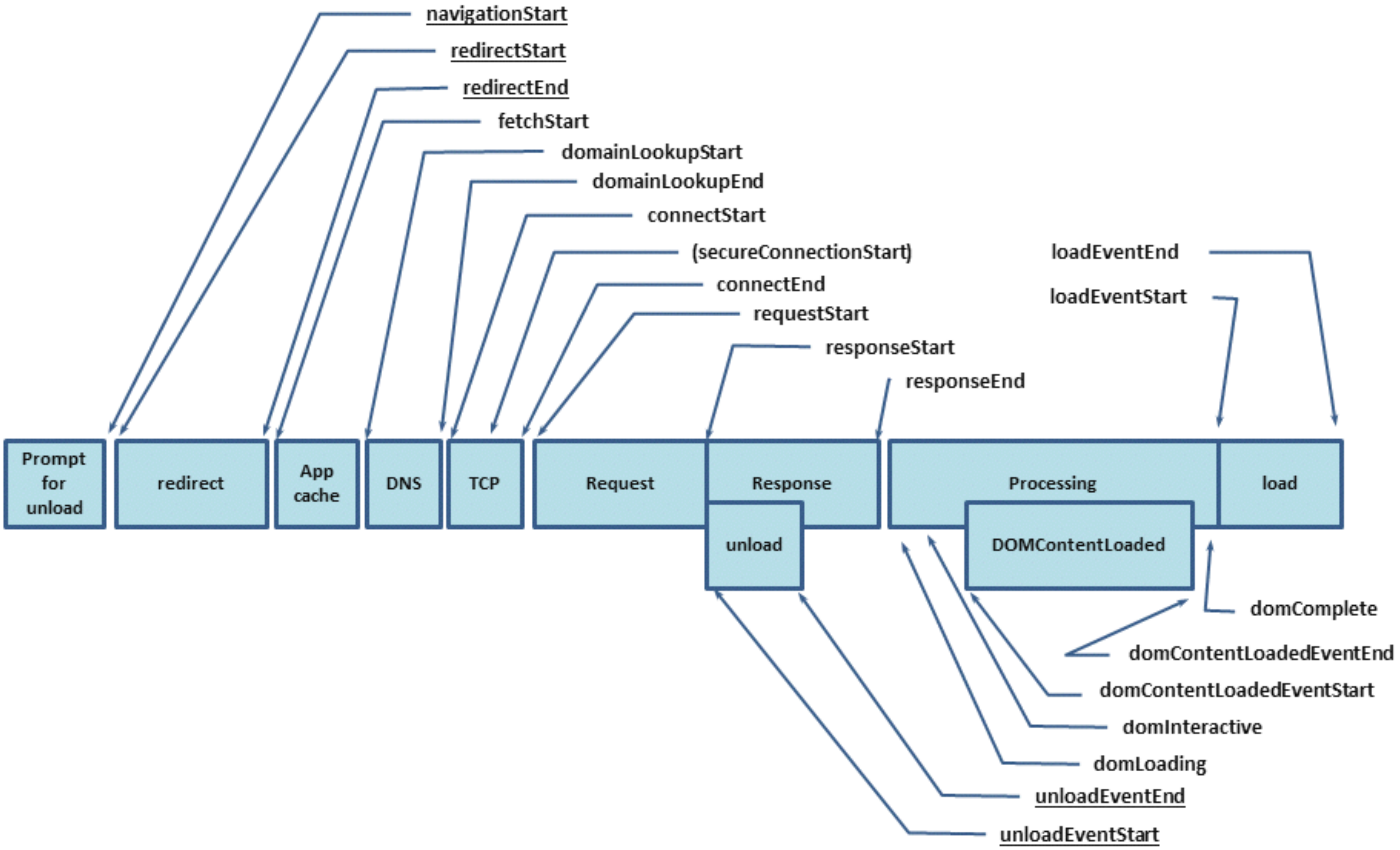
DOM processing

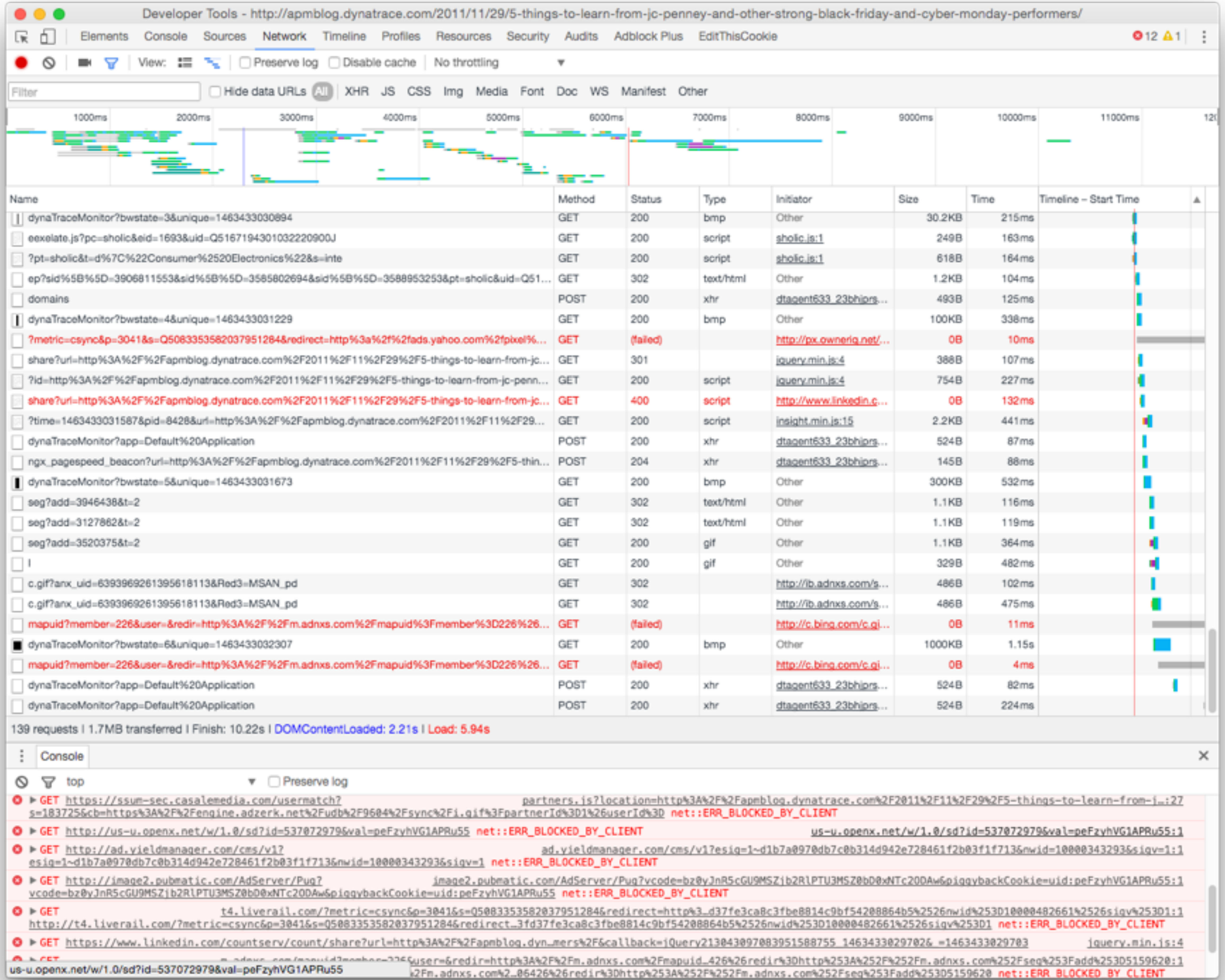
Page rendering

Network time:
(first half)
(second half)
Brown in dashboard chart.

Front end time:

Yellow and blue in dashboard chart.


navigationStart
redirectStart
redirectEnd
fetchStart
domainLookupStart
domainLookupEnd
connectStart
(secureConnectionStart)
connectEnd
requestStart
responseStart
responseEnd
loadEventEnd
loadEventStart
Prompt for unload
redirect
App cache
DNS
TCP
Request
Response
Processing
load
unload
DOMContentLoaded
domComplete
domContentLoadedEventEnd
domContentLoadedEventStart
domInteractive
domLoading
unloadEventEnd
unloadEventStart



Name	Method	Status	Type	Initiator	Size	Time	Timeline – Start Time
dynaTraceMonitor?bwstate=3&unique=1463433030894	GET	200	bmp	Other	30.2KB	215ms	
eexelate.js?pc=sholic&eid=1693&uid=Q5167194301032220900J	GET	200	script	sholic.js:1	249B	163ms	
?pt=sholic&t=d%7C%22Consumer%2520Electronics%22&s=inte	GET	200	script	sholic.js:1	618B	164ms	
ep?sid%5B%5D=3906811553&sid%5B%5D=3585802694&sid%5B%5D=3588953253&pt=sholic&uid=Q51…	GET	302	text/html	Other	1.2KB	104ms	
domains	POST	200	xhr	dtagent633_23bhiprs…	493B	125ms	
dynaTraceMonitor?bwstate=4&unique=1463433031229	GET	200	bmp	Other	100KB	338ms	
?metric=csync&p=3041&s=Q5083353582037951284&redirect=http%3a%2f%2fads.yahoo.com%2fpixel%…	GET	(failed)		http://px.owneriq.net/…	0B	10ms	
share?url=http%3A%2F%2Fapmblog.dynatrace.com%2F2011%2F11%2F29%2F5-things-to-learn-from-jc…	GET	301		jquery.min.js:4	388B	107ms	
?id=http%3A%2F%2Fapmblog.dynatrace.com%2F2011%2F11%2F29%2F5-things-to-learn-from-jc-penn…	GET	200	script	jquery.min.js:4	754B	227ms	
share?url=http%3A%2F%2Fapmblog.dynatrace.com%2F2011%2F11%2F29%2F5-things-to-learn-from-jc…	GET	400	script	http://www.linkedin.c…	0B	132ms	
?time=1463433031587&pid=8428&url=http%3A%2F%2Fapmblog.dynatrace.com%2F2011%2F11%2F29…	GET	200	script	insight.min.js:15	2.2KB	441ms	
dynaTraceMonitor?app=Default%20Application	POST	200	xhr	dtagent633_23bhiprs…	524B	87ms	
ngx_pagespeed_beacon?url=http%3A%2F%2Fapmblog.dynatrace.com%2F2011%2F11%2F29%2F5-thin…	POST	204	xhr	dtagent633_23bhiprs…	145B	88ms	
dynaTraceMonitor?bwstate=5&unique=1463433031673	GET	200	bmp	Other	300KB	532ms	
seg?add=394643&t=2	GET	302	text/html	Other	1.1KB	116ms	
seg?add=3127862&t=2	GET	302	text/html	Other	1.1KB	119ms	
seg?add=3520375&t=2	GET	200	gif	Other	1.1KB	364ms	
l	GET	200	gif	Other	329B	482ms	
c.gif?anx_uid=6393969261395618113&Red3=MSAN_pd	GET	302		http://ib.adnxs.com/s…	486B	102ms	
c.gif?anx_uid=6393969261395618113&Red3=MSAN_pd	GET	302		http://ib.adnxs.com/s…	486B	475ms	
mapuid?member=226&user=&redir=http%3A%2F%2Fm.adnxs.com%2Fmapuid%3Fmember%3D226%26…	GET	(failed)		http://c.bing.com/c.gi…	0B	11ms	
dynaTraceMonitor?bwstate=6&unique=1463433032307	GET	200	bmp	Other	1000KB	1.15s	
mapuid?member=226&user=&redir=http%3A%2F%2Fm.adnxs.com%2Fmapuid%3Fmember%3D226%26…	GET	(failed)		http://c.bing.com/c.gi…	0B	4ms	
dynaTraceMonitor?app=Default%20Application	POST	200	xhr	dtagent633_23bhiprs…	524B	82ms	
dynaTraceMonitor?app=Default%20Application	POST	200	xhr	dtagent633_23bhiprs…	524B	224ms	

139 requests | 1.7MB transferred | Finish: 10.22s | DOMContentLoaded: 2.21s | Load: 5.94s

```
▶ GET https://ssum-sec.casalemedia.com/usermatch?                partners.js?location=http%3A%2F%2Fapmblog.dynatrace.com%2F2011%2F11%2F29%2F5-things-to-learn-from-j…:27
s=183725&cb=https%3A%2F%2Fengine.adzerk.net%2Fudb%2F9604%2Fsync%2Fi.gif%3FpartnerId%3D1%26userId%3D net::ERR_BLOCKED_BY_CLIENT
▶ GET http://us-u.openx.net/w/1.0/sd?id=537072979&val=peFzyhVG1APRu55 net::ERR_BLOCKED_BY_CLIENT         us-u.openx.net/w/1.0/sd?id=537072979&val=peFzyhVG1APRu55:1
▶ GET http://ad.yieldmanager.com/cms/v1?                ad.yieldmanager.com/cms/v1?esig=1~d1b7a0970db7c0b314d942e728461f2b03f1f713&nwid=10000343293&sigv=1:1
esig=1~d1b7a0970db7c0b314d942e728461f2b03f1f713&nwid=10000343293&sigv=1 net::ERR_BLOCKED_BY_CLIENT
▶ GET http://image2.pubmatic.com/AdServer/Pug?          image2.pubmatic.com/AdServer/Pug?vcode=bz0yJnR5cGU9MSZjb2RlPTU3MSZ0bD0xNTc2ODAw&piggybackCookie=uid:peFzyhVG1APRu55:1
vcode=bz0yJnR5cGU9MSZjb2RlPTU3MSZ0bD0xNTc2ODAw&piggybackCookie=uid:peFzyhVG1APRu55 net::ERR_BLOCKED_BY_CLIENT
▶ GET                      t4.liverail.com/?metric=csync&p=3041&s=Q5083353582037951284&redirect=http%3…d37fe3ca8c3fbe8814c9bf54208864b5%2526nwid%253D10000482661%2526sigv%253D1:1
http://t4.liverail.com/?metric=csync&p=3041&s=Q5083353582037951284&redirect…3fd37fe3ca8c3fbe8814c9bf54208864b5%2526nwid%253D10000482661%2526sigv%253D1 net::ERR_BLOCKED_BY_CLIENT
▶ GET https://www.linkedin.com/countserv/count/share?url=http%3A%2F%2Fapmblog.dyn…mers%2F&callback=jQuery213043097083951588755_1463433029702&_=1463433029703        jquery.min.js:4
▶ GET                      m.adnxs.com/mapuid?member=226&user=&redir=http%3A%2F%2Fm.adnxs.com%2Fmapuid…426%26redir%3Dhttp%253A%252F%252Fm.adnxs.com%252Fseg%253Fadd%253D5159620:1
us-u.openx.net/w/1.0/sd?id=537072979&val=peFzyhVG1APRu55  %2Fm.adnxs.com%2…06426%26redir%3Dhttp%253A%252F%252Fm.adnxs.com%252Fseg%253Fadd%253D5159620 net::ERR_BLOCKED_BY_CLIENT
```

Tools & technologies

- Bash & terminal
- SASS, Compass, Bourbon
- Libraries & frameworks
    - Cross-compatibility, graceful degradation
    - Charts, canvas, validation
- Fonts & typography

CLI

- Complex, precise tasks
- Interface and connect things
- Automation
- GUIs are actually programs running on a CLI

```
find . -mtime -60 -name '*.txt' -exec sed -i.bak 's/fish/carrot/g' {}
\;
```

- Search the current directory structure
- for all files modified within the last 60 days,
- called '*.txt'
- replace the word 'fish' with 'carrot' in all of them.
- leaving a backup copy suffixed .bak.

Source: http://unix.stackexchange.com/a/199711

Workflow

- Versioning: git, semantic versioning
- Structure, conventions and methodology: BEM, SMACSS, OOCSS, file structure
- Package & dependencies: npm, bundler
- Licenses & legal considerations
- Advanced streamlining: Grunt.js, Gulp, Webpack
- Jedi: test-driven development

Design principles

- Apply semantics to
  - Color theory
  - Proportions
- Standards are king
  - Leverage the user’s intuition
  - If you think you are innovating, you are probably doing things wrong