

A BASELINE FOR WEB PERFORMANCE WITH PHANTOMJS

@WESLEYHALES

DO YOU AUTOMATE BROWSER PERF?

You might occasionally test your sites using Firebug, Chrome DevTools, PageSpeed, YSlow, etc..

Abiding by these rules...

Make Fewer HTTP Requests

Use a Content Delivery Network

Add an Expires Header

Gzip Components

Put Stylesheets at the Top

Put Scripts at the Bottom

Avoid CSS Expressions

Make JavaScript and CSS External

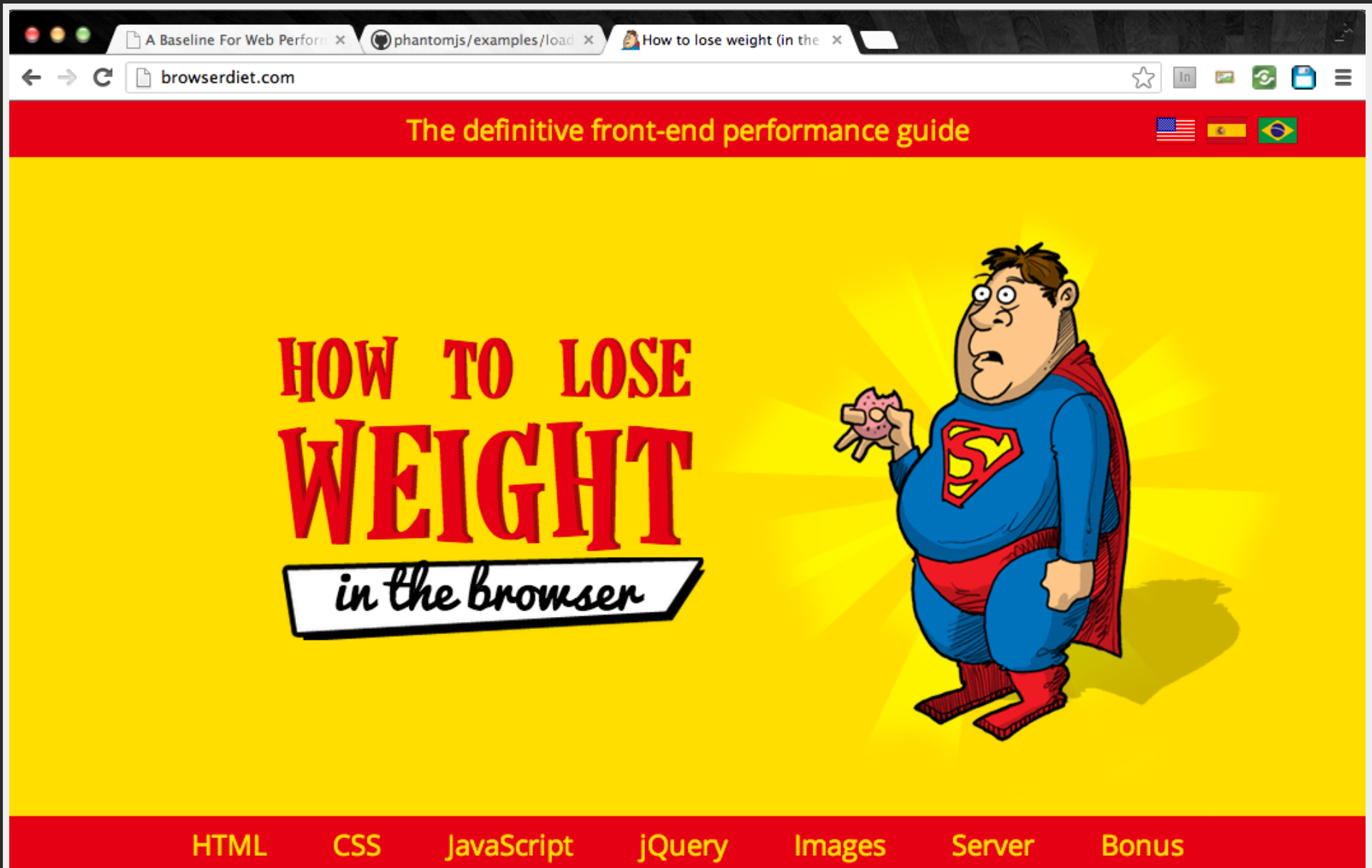
Reduce DNS Lookups

Minify JavaScript

Remove Duplicate Scripts

Make AJAX Cacheable

Check out browserdiet.com



The screenshot shows a web browser with three tabs: "A Baseline For Web Perform...", "phantomjs/examples/load...", and "How to lose weight (in the...". The address bar shows "browserdiet.com". The website has a red header with the text "The definitive front-end performance guide" and flags for the USA, Spain, and Brazil. The main content area has a yellow background with the text "HOW TO LOSE WEIGHT" in large red letters, followed by "in the browser" in a white box with a black border. To the right is a cartoon illustration of a very overweight Superman holding a donut. At the bottom is a red navigation bar with links: HTML, CSS, JavaScript, jQuery, Images, Server, and Bonus.

The definitive front-end performance guide

HOW TO LOSE WEIGHT

in the browser

HTML CSS JavaScript jQuery Images Server Bonus



And what if we got together a bunch of experts who work on large sites to create a

How can we abide to these rules (or metrics you care about) and monitor changes over time?

@WESLEYHALES

- Developer at Apigee
- **Atlanta HTML5**
- **loadreport.js**
- **slidfast.js**
- **this article**

The screenshot shows the HTML5 Rocks website. The top navigation bar is orange with links: HOME, POSTS & TUTORIALS, HTML5 FEATURES, SLIDES, RESOURCES, WHY HTML5?, WHO WE ARE, CONTRIBUTE, and SEARCH. The main header area has 'HTML5 ROCKS TUTORIALS' on the left and the article title 'HTML5 Techniques for Optimizing Mobile Performance' on the right. Below the title, it says 'By Wesley Hales' and 'Published Sept. 19, 2011'. There is a small profile picture of Wesley Hales and an HTML5 logo. Below the title, it says 'SUPPORTED BROWSERS:' followed by icons for Chrome, Firefox, Safari, and Opera. Below that, it says '25 Comments and 0 Reactions'. On the left side, there is a 'Table of Contents' section with links to Introduction, Hardware acceleration, Page transitions, Sliding, Flipping, Rotating, Debugging hardware acceleration, Behind the Scenes: Fetching and Caching, Network type detection, handling, and profiling, and Conclusion. The main content area has an 'Introduction' section with text about spinning refreshes and page transitions, and a 'Hardware acceleration' section with text about GPUs.

HOME POSTS & TUTORIALS HTML5 FEATURES SLIDES RESOURCES WHY HTML5? WHO WE ARE CONTRIBUTE SEARCH

HTML5 ROCKS TUTORIALS

HTML5 Techniques for Optimizing Mobile Performance

By Wesley Hales
Published Sept. 19, 2011

SUPPORTED BROWSERS:

25 Comments and 0 Reactions

Table of Contents

- Introduction
- Hardware acceleration
- Page transitions
 - Sliding
 - Flipping
 - Rotating
- Debugging hardware acceleration
- Behind the Scenes: Fetching and Caching
- Network type detection, handling, and profiling
- Conclusion

Introduction

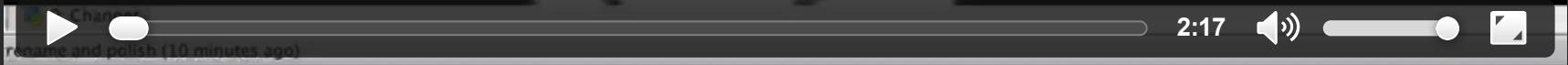
Spinning refreshes, choppy page transitions, and periodic delays in tap events are just a few headaches in today's mobile web environments. Developers are trying to get as close to native as they possibly can, but are often derailed by hacks, resets, and rigid frameworks.

In this article, we will discuss the bare minimum of what it takes to create a mobile HTML5 app. The main point is to unmask the hidden complexities which today's mobile framework hide. You will see a minimalistic approach (using core HTML5 APIs) and basic fundamentals will empower you to write your own framework or contribute to the one you currently use.

Hardware acceleration

Normally, GPUs handle detailed 3D modeling or CAD diagrams, but in this case, we want to

Why should you be concerned with measuring client-side performance?



Many, many reasons, but user expectations, SEO, and \$\$ usually trump all.

The good parts...

**CNN.COM REMOVED JQUERY AND
PROTOACULOUS FROM CONCATENATION
AND ADDED TO CDN**

IMPROVED SITE PERFORMANCE BY 17%

REDUCED ALMOST 1TB OF BANDWIDTH PER DAY

The hard parts...

Do you use/understand gzip, cdn, cache headers, and the like?

No prob, let's review...

The hard parts of monitoring client-side performance

DEVELOPER STANDPOINT

- Caching (headers, localStorage, appcache)
- When to gzip (don't gzip stuff that's natively compressed)
- When to pull from CDN, local, or break apart (single HTTP reqs aren't always a win)

SOME PERFORMANCE ISSUES REQUIRE MORE THOUGHT

AUTOMATION STANDPOINT

- Server(Browser) Location
- Browser Type
- Connection Speed
- CPU Activity

SERVER(BROWSER) LOCATION

Will you be running tests on your LAN, Cloud, or elsewhere?

CONNECTION SPEED

Speeds and latencies should be as consistent as possible.

```
Wesleys-MacBook-Pro:html5devconf wesleyhales$ phantomjs loadspeed.js http://www.wesleyhales.com
Page title is wesleyhales.com: Blog
Loading time 12471 msec
```

BROWSER TYPE

DIY options are limited

CPU ACTIVITY

Affects all tests. Need as few (or consistent) processes running - that won't cause a spike during test.

AVAILABLE TOOLS AND SERVICES

CLIENT PERFORMANCE MEASUREMENT TOOLS

- Firebug
- Developer Tools

Browser tabs: A x, we x, loc x, Hc x, Im x, WF x, Ch x, we x

Address bar: wesleyhales.com

wesleyhales.com

Elements Resources Network Sources Timeline Profiles Audits Console Intent

Name Path	Me...	Statu Text	Type	Initiator	Size Conte	Time Laten	Timeline	1.87 s	2.80 s	3.73 s
AvenirNextLTPr /type	GET	200 OK	ap...	wesleyhale Parser	65.7 B / 65.2 B	958 n / 605 n				
header-bg2.gif /images/skin/re	GET	200 OK	im...	wesleyhale Parser	10.9 B / 10.5 B	929 n / 547 n				
header-bg-pat /images/skin/re	GET	200 OK	im...	wesleyhale Parser	1.5 Ki / 1.1 Ki	928 n / 546 n				
avatar.php?defa www.gravatar.cc	GET	301 Moved	tex...	recent_cor Script	692 B / 0 B	923 n / 541 n				
_utm.gif?utmww www.google-ani	GET	200 OK	im...	qa.js:60 Script	527 B / 35 B	555 n / 555 n				
retweet.png si0.twimg.com/i	GET	200 OK	im...	site.js:13 Script	1.6 Ki / 1.1 Ki	552 n / 551 n				
reply.png si0.twimg.com/i	GET	200 OK	im...	site.js:13 Script	1.7 Ki / 1.1 Ki	540 n / 539 n				
f3ada405ce890 www.gravatar.cc	GET	302 Moved	tex...	http://ww Redirect	602 B / 0 B	582 n / 582 n				
noavatar92.png i1.wp.com/medi	GET	200 OK	im...	http://ww Redirect	5.5 Ki / 5.0 Ki	230 n / 228 n				

36 requests | 395 KB transferred | 3.73 s (onload: 3.73 s, DOMContentLoaded: 2.92 s)

Icons: All Documents Stylesheets Images Scripts XHR Fonts We 2

CLIENT PERFORMANCE MEASUREMENT SERVICES

- [webpagetest.org](https://www.webpagetest.org)

CLIENT PERFORMANCE MEASUREMENT AUTOMATION

- With a headless browser

A headless web browser is a browser without a graphical user interface.

There are many others like HtmlUnit (Java), EnvJS (JavaScript), Ghost (Python), WebDriver (for all)...

PHANTOMJS

Fork me on GitHub



PhantomJS

[SOURCE CODE](#)

[DOCUMENTATION](#)

[API](#)

[EXAMPLES](#)

[FAQ](#)

Full web stack No browser required

PhantomJS is a headless WebKit scriptable with a JavaScript API. It has **fast** and **native** support for various web standards: DOM handling, CSS selector, JSON, Canvas, and SVG.

[Download v1.9](#)

[Get started](#)

Simple Javascript example

```
console.log('Loading a web page');
var page = require('webpage').create();
var url = 'http://www.phantomjs.org/';
page.open(url, function (status) {
    //Page is loaded!
    phantom.exit();
});
```

Community:



[Read the release notes](#)



[Join the mailing list](#)



[Report bugs](#)

PhantomJS is an optimal solution for

HEADLESS WEBSITE TESTING

Run functional tests with frameworks such as Jasmine, QUnit, Mocha, Capybara, WebDriver, and many others.

SCREEN CAPTURE

Programmatically capture web contents, including SVG and Canvas. Create web site screenshots with thumbnail

PAGE AUTOMATION

Access and manipulate webpages with the standard DOM API, or with usual libraries like jQuery. [Learn more](#)

NETWORK MONITORING

Monitor page loading and export as standard HAR files. Automate performance analysis using YSlow and Jenkins. [Learn more](#)

TYPICAL HEADLESS BROWSER USAGE

- JavaScript Testing
- Navigation and submitting forms
- Clicking links
- Web Scraping
- ...

PHANTOMJS GIVES US ACCESS TO

- Cookie and Local Storage related APIs
- SSL support
- DOM readystate
- Screen capture
- generate HAR files
- A ton more...

SIMPLE PHANTOMJS DEMO

- simple

LOADREPORT.JS (SERVICE DEMO)

Measures things like Document Readystate, Resource load times and size, trouble makers, birdseye view for loading (in WebKit)

http://www.cnn.com

noncached

JSON

Show Me Some Charts!

Url	Dom Readystate Interactive	Window Onload	Dom Readystate Complete	Elapsed Load Time	Number Of Resources	Total Resources Time	Slowest Resource	Largest Resource	Total Resources Size	Non Reporting Resources	Date	Time	Errors
http://www.cnn.com	723	0	0	960	157	6158	Show it	Show it	1594.237	15	16/7/2012	23:1:48	0 JavaScript Errors
http://www.cnn.com	689	0	0	988	157	6611	Show it	Show it	1684.322	15	16/7/2012	23:1:51	0 JavaScript Errors
http://www.cnn.com	674	0	0	967	160	6662	Show it	Show it	2083.736	16	16/7/2012	23:1:54	0 JavaScript Errors
http://www.cnn.com	819	0	0	1042	162	6123	Show it	Show it	1904.889	11	16/7/2012	23:1:58	0 JavaScript Errors
http://www.cnn.com	664	0	0	932	151	6850	Show it	Show it	1692.332	11	16/7/2012	23:2:1	0 JavaScript Errors

LOADREPORT.JS SCRIPT DEMOS

- Loadreport.js (**Basic**, Filmstrip, **Angular**, **Ember**)
- Speedreport.js (**Basic**)

LOADREPORT.JS SERVER (FLUXUI)

Reporting is single threaded to maximize median accuracy

http://www.cnn.com

noncached

[JSON](#)

[Show Me Some Charts!](#)

Url	Dom Readystate Interactive	Window Onload	Dom Readystate Complete	Elapsed Load Time	Number Of Resources	Total Resources Time	Slowest Resource	Largest Resource	Total Resources Size	Non Reporting Resources	Date	Time	Errors
http://www.cnn.com	723	0	0	960	157	6158	Show it	Show it	1594.237	15	16/7/2012	23:1:48	0 JavaScript Errors
http://www.cnn.com	689	0	0	988	157	6611	Show it	Show it	1684.322	15	16/7/2012	23:1:51	0 JavaScript Errors
http://www.cnn.com	674	0	0	967	160	6662	Show it	Show it	2083.736	16	16/7/2012	23:1:54	0 JavaScript Errors
http://www.cnn.com	819	0	0	1042	162	6123	Show it	Show it	1904.889	11	16/7/2012	23:1:58	0 JavaScript Errors
http://www.cnn.com	664	0	0	932	151	6850	Show it	Show it	1692.332	11	16/7/2012	23:2:1	0 JavaScript Errors

EXTRAS

- [phantomas.js](#)
- [Yslow](#)

Getting a baseline

Remember this...

- Make Fewer HTTP Requests
- Use a Content Delivery Network
- Add an Expires Header
- Gzip Components
- Put Stylesheets at the Top
- Put Scripts at the Bottom
- Avoid CSS Expressions
- Make JavaScript and CSS External
- Reduce DNS Lookups
- Minify JavaScript
- Remove Duplicate Scripts
- Make AJAX Cacheable

From an opensource perspective, we can use Travis CI

- github OAuth token
- encryption with travis
- pushing your site with an automated travisci build
- [Setup info here](#)

From the enterprise side, we can run the scripts internally with no special setup.

- Bamboo
- Jenkins
- Barebone Linux system without X11 is not a problem for PhantomJS.

SUMMARY

- For simplicity, use phantomJS and build a script that works for you.
- For more advanced, host your own instance of webpagetest.org