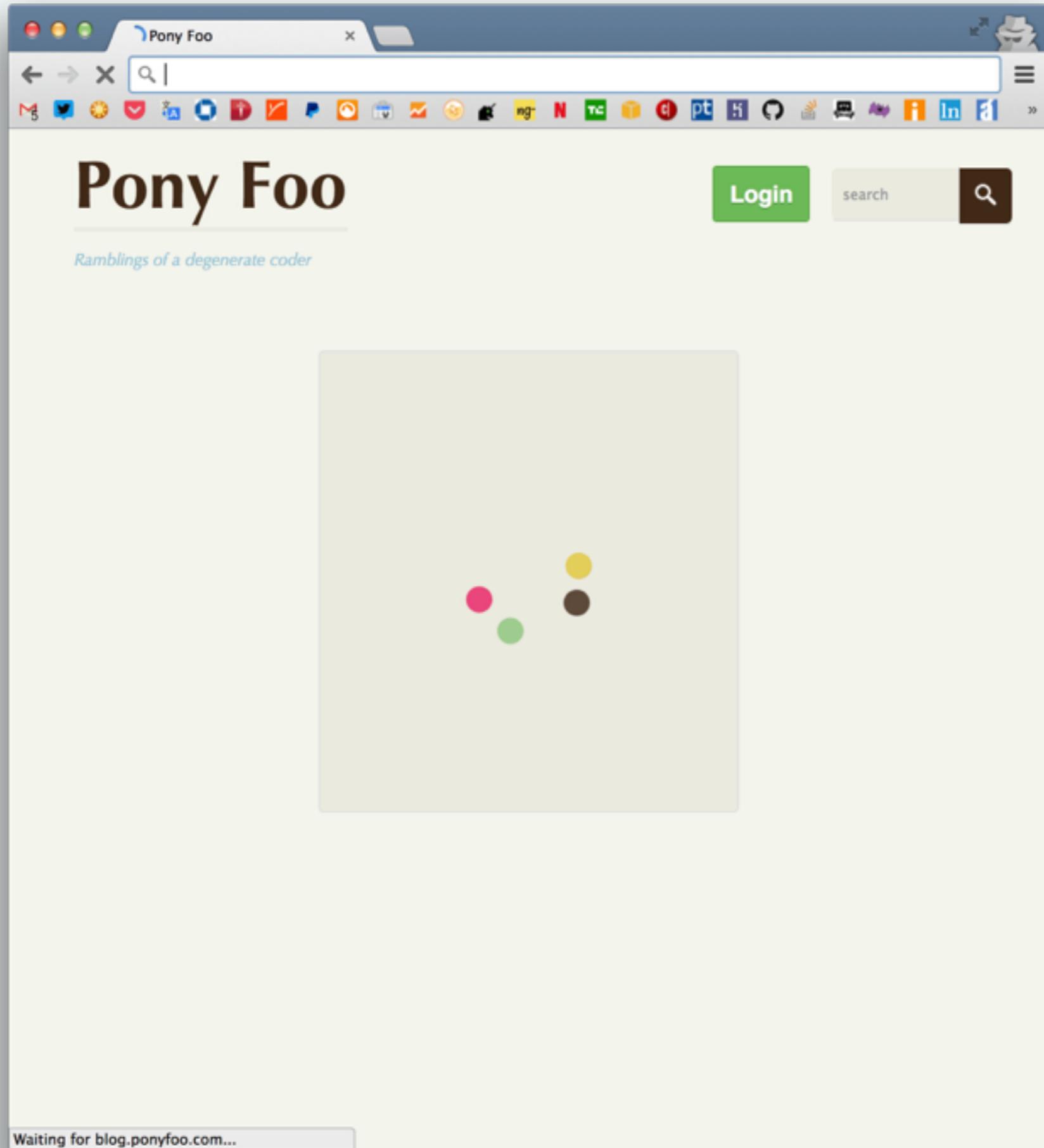




*High*  
Performance  
*in the*  
*Critical Path*



*Measure!*

*What is  
going on?*

# DevTools Audits

The screenshot shows the DevTools Audits interface for the URL <http://www.cnn.com/>. The interface has a sidebar on the left with tabs for 'Audits' (selected) and 'RESULTS'. The main area is titled 'Select audits to run' and contains three audit types: 'Select All', 'Network Utilization', and 'Web Page Performance', all of which are checked. Below these is a radio button for 'Audit Present State' and another for 'Reload Page and Audit on Load', with the latter being selected. At the bottom are 'Run' and 'Clear' buttons.

Developer Tools - <http://www.cnn.com/>

Elements Network Sources Timeline Profiles Resources Audits Console 23 4 ⚠️ ⚙️ ✎

**Audits**

RESULTS

Select audits to run

Select All

Network Utilization

Web Page Performance

Audit Present State

Reload Page and Audit on Load

Run Clear

# DevTools Audits

The screenshot shows the Chrome DevTools Audits panel. At the top, there are three colored circles (red, yellow, green) followed by the title "Developer Tools - http://www.cnn.com/". Below the title is a navigation bar with icons for search, element inspection, network monitoring, sources, timeline, profiles, resources, audits (which is selected), and console. To the right of the audit icon are status indicators: 20 errors (red), 4 warnings (yellow), and a refresh button. The main area is divided into sections: "RESULTS" on the left and "Audits" on the right. Under "RESULTS", a single item "http://www.cnn.com/ (1)" is listed. On the right, the "Audits" section is expanded, showing two main categories: "Network Utilization" and "Web Page Performance".

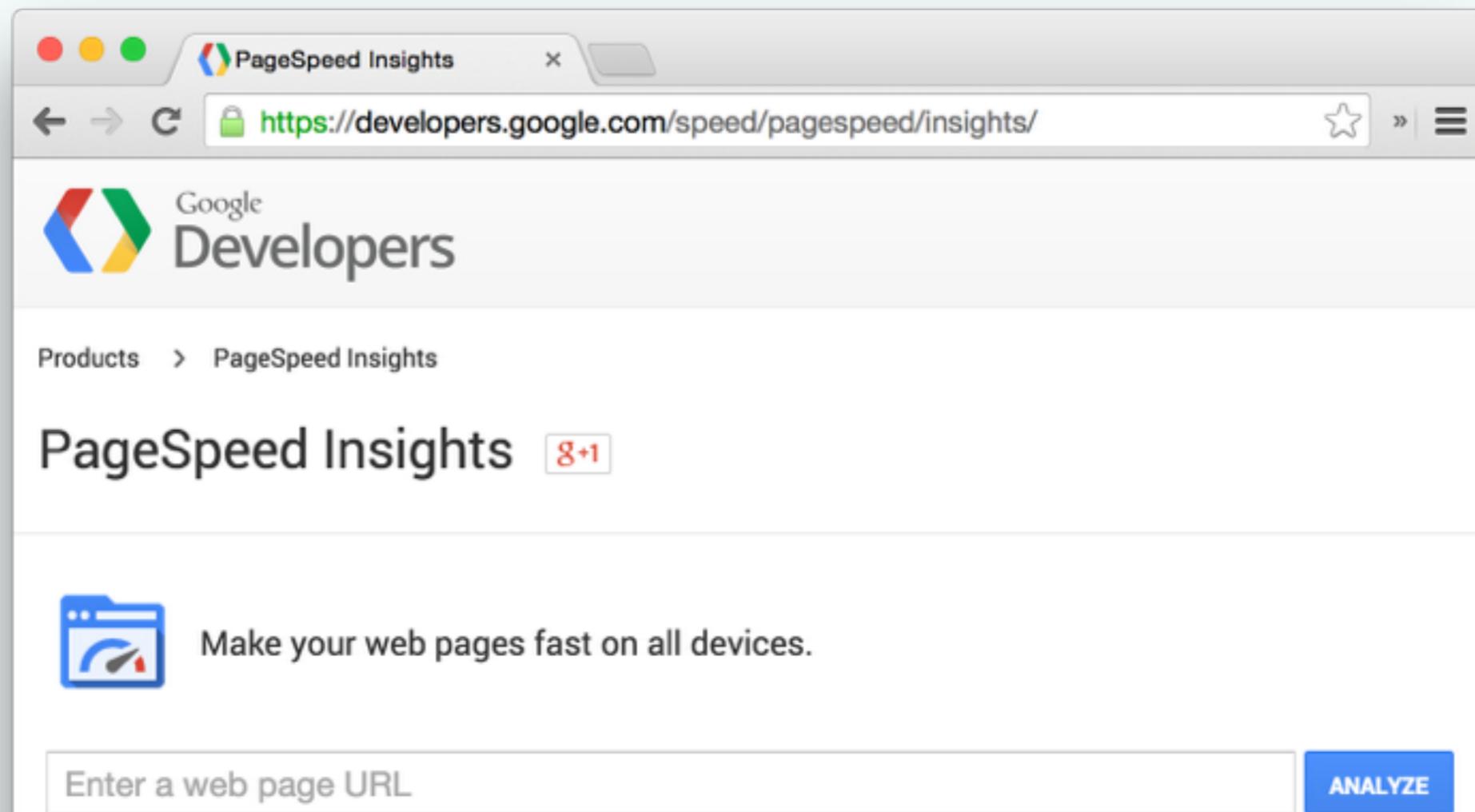
- ▼ Network Utilization
  - 🔴 ▶ Combine external JavaScript (24)
  - 🔴 ▶ Enable gzip compression (2)
  - 🔴 ▶ Leverage browser caching (87)
  - 🟡 ▶ Leverage proxy caching (46)
  - 🟡 ▶ Minimize cookie size (8)
  - 🟡 ▶ Serve static content from a cookieless domain (11)
- ▼ Web Page Performance
  - 🔴 ▶ Optimize the order of styles and scripts (8)
  - 🔴 ▶ Put CSS in the document head (4)
  - 🟡 ▶ Remove unused CSS rules (3087)
  - 🟡 ▶ Use normal CSS property names instead of vendor-prefixed ones (48)

# *DevTools Audits*

*Per-resource advice*

*Caching best practices*

# PageSpeed Insights



[developers.google.com/speed/pagespeed/insights](https://developers.google.com/speed/pagespeed/insights)

## Web Performance

Learn more about web performance tools at Google, including browser extensions.

## Give Feedback

Have comments or questions about PageSpeed Insights? Send feedback or discuss on

## About PageSpeed Insights

PageSpeed Insights analyzes the content of a web page, then generates suggestions to make

# PageSpeed Insights

The screenshot shows the Google PageSpeed Insights interface. At the top, the URL `https://developers.google.com/speed/pagespeed/insights/?url=ponyfoo.com...` is entered in the browser's address bar. Below the header, the Google Developers logo is visible. The main navigation links are "Products" and "PageSpeed Insights". The title "PageSpeed Insights" is displayed with a "8+1" badge. A search bar contains the URL `http://ponyfoo.com/`, and a blue "ANALYZE" button is to its right. Below the search bar, there are two tabs: "Mobile" (selected) and "Desktop". The main content area displays a green box with the text "96 / 100 Speed". Underneath, a yellow warning icon with an exclamation mark is followed by the text "Consider Fixing:". Two items are listed: "Leverage browser caching" and a link "Show how to fix". To the right, a smartphone icon displays a screenshot of the "Pony Foo" website with the word "Immutable" visible.

# *PageSpeed Insights*

*Mobile, desktop*

*Get a rough (1..100) score*

*Best practices*

*Practical advice*

# WebPageTest

The screenshot shows the main interface of the WebPageTest website. At the top, there's a navigation bar with links for HOME, TEST RESULT, TEST HISTORY, FORUMS, DOCUMENTATION, and ABOUT. Below the navigation, a large heading says "Test a website's performance". There are three tabs: "Analytical Review" (disabled), "Visual Comparison" (selected), and "Traceroute". A prominent input field asks "Enter a Website URL". To the right of the URL input is a large yellow "START TEST" button. Below the URL input, there are dropdown menus for "Test Location" set to "San Francisco, CA USA (IE 11, Chrome, Firefox, Safari)" and "Browser" set to "Chrome". A "Select from Map" button is also present. On the far right, it says "Provided by" followed by the WebPageTest logo. A descriptive text block below the form explains the service: "Run a free website speed test from multiple locations around the globe using real browsers (IE and Chrome) and at real consumer connection speeds. You can run simple tests or perform advanced testing including multi-step transactions, video capture, content blocking and much more. Your results will provide rich diagnostic information including resource loading waterfall charts, Page Speed optimization checks and suggestions for improvements." A small note at the bottom left says "If you have any performance/optimization questions you should join the [Google+ group](#) or [discuss Web Performance Optimization](#)".

[webpagetest.org](http://webpagetest.org)

## Recent Industry Blog Posts

New Features: Multiple Interactive Waterfalls and HAR Importing

smush.it is dead, long live smushing

f8 2015

Zoomaf Joins ATDC Select

## Recent Discussions

how to do performance test in this website?

Fastest way to redirect users based on Geolocation

Running scripts with RESTful APIs

Private Instances - configurations

# WebPageTest

WebPagetest Test Result - [www.webpagetest.org/result/150408\\_E5\\_12AJ/](#)

HOME TEST RESULT TEST HISTORY FORUMS DOCUMENTATION ABOUT Need help improving?

## Web Page Performance Test for ponyfoo.com

From: San Francisco, CA USA - Chrome - Cable  
4/8/2015, 2:16:03 PM

First Byte Time Keep-alive Enabled Compress Transfer Compress Images Cache static content Effective use of CDN

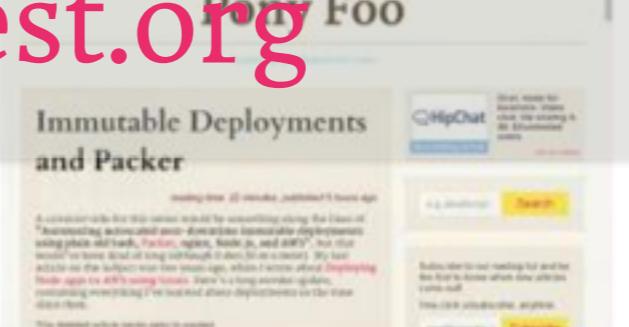
A A A C F X

[Summary](#) [Details](#) [Performance Review](#) [Content Breakdown](#) [Domains](#) [Screen Shot](#)

Tester: i-f949b832 [Raw page data](#) - [Raw object data](#)  
[Re-run the test](#) [Export HTTP Archive \(.har\)](#) [See in ShowSlow](#) [View Test Log](#)

	Document Complete					Fully Loaded						
	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View	2.041s	0.396s	0.694s	729	390	2.041s	16	105 KB	2.493s	22	363 KB	\$---
Repeat View	1.477s	0.597s	0.805s	832	389	1.477s	4	33 KB	1.690s	6	62 KB	

Waterfall Screen Shot Video

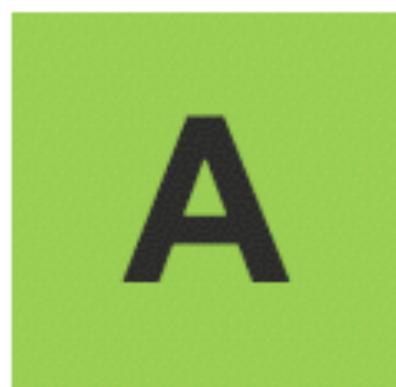
First View (2.041s)  

Immutable Deployments and Packer [Filmstrip View](#) [Watch Video](#)

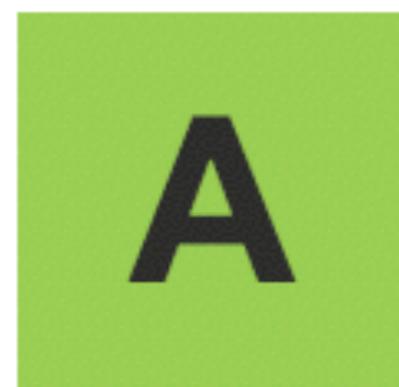
# WebPageTest

PageSpeed 1.12 Score: 88/100

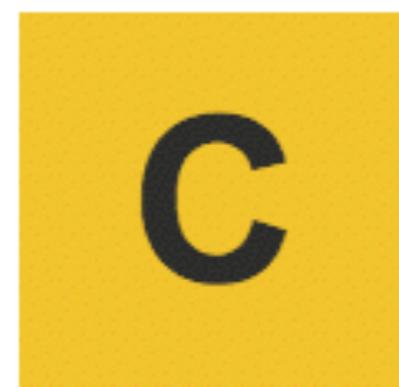
Need help improving?



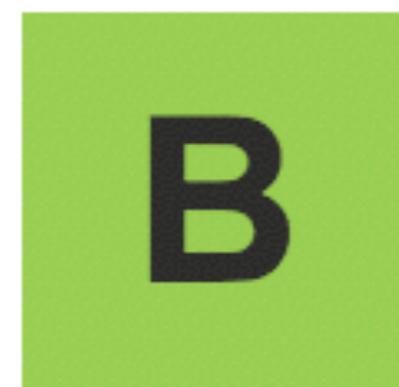
First Byte  
Time



Keep-alive  
Enabled



Compress  
Transfer



Compress  
Images



Cache  
static  
content

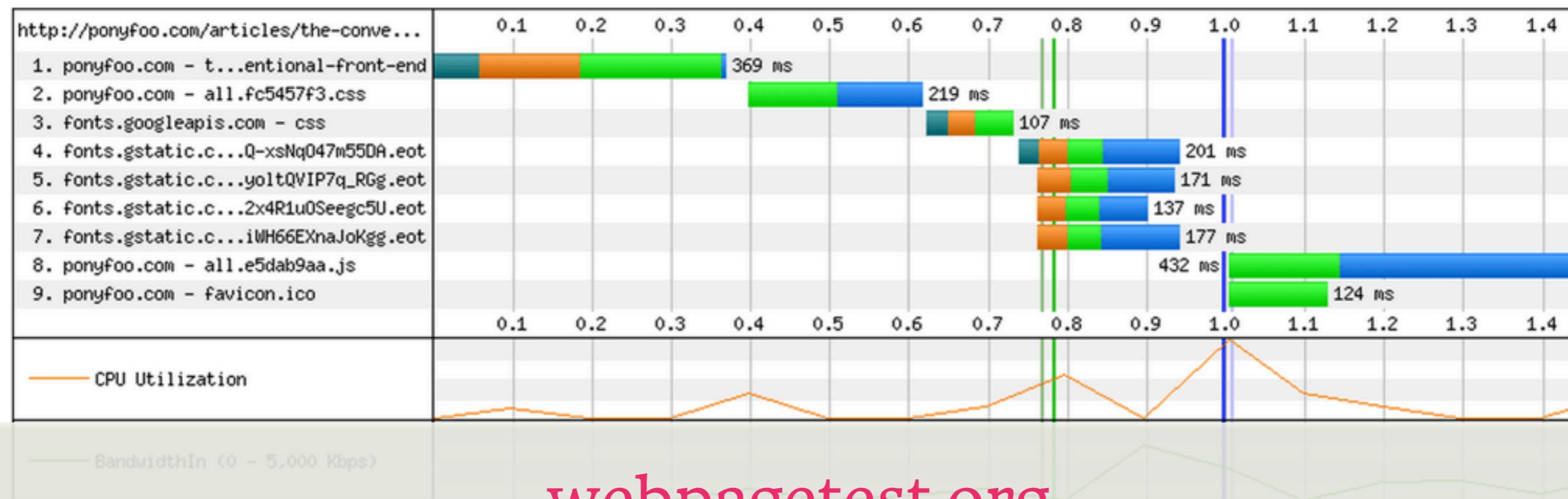
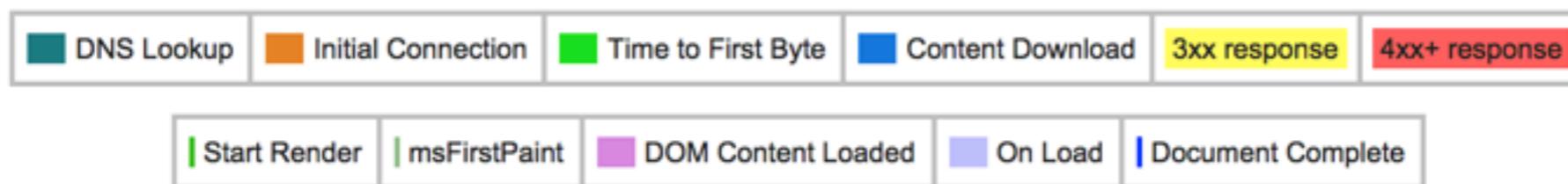


Effective  
use of CDN

[webpagetest.org](http://webpagetest.org)

# WebPageTest

## Waterfall View



webpagetest.org

# WebPageTest

Full Optimization Checklist

	Keep-Alive	GZip	Compress Img	Progressive	Cache Static	CDN Detected
http://ponyfoo.com	100%	100%	78%	0%	64%	50%
1: ponyfoo.com - /	✓	✓			✗	
2: ponyfoo.com - all.82d1d6be.css	✓	✓			✓	✗
3: cdn.carbonads.com - carbon.js	✓	✓			⚠	✓
4: fonts.googleapis.com - css		✓			✗	✓
5: srv.carbonads.net - C6AILKT.json						
6: fonts.gstatic.com...K15tYj8yhly0.woff2	✓				✓	
7: fonts.gstatic.com...RcAd0FWWLdxQ.woff2	✓				✓	
8: fonts.gstatic.com...xsNq047m550A.woff2	✓				✓	
9: fonts.gstatic.com...r5-oayXS0efg.woff2	✓				✓	
10: assets.servedby...sellads.com - 20881	✓				✓	✓
11: ponyfoo.com - /	✓				✗	✗
12: ponyfoo.com - all.cb9b1cae.js	✓	✓			✓	✗
13: ponyfoo.com - favicon.ico	✓	✓			✓	✗
14: ponyfoo.com - opensearch.xml	✓				✓	✗
15: static.getclicky.com - js	✓				✓	✓
16: www.gravatar.com...b4427541e46cb8.png			⚠		✗	✓
17: www.google-anal....com - analytics.js	✓	✓		✗	⚠	✓
18: platform.twitter.com - widgets.js	✓	✓			✗	✗
19: www.google-analutice.com - collect	✓					
20: in.getclicky.com - in.php		✓				

webpagetest.org

# WebPageTest



Makes it easy to spot **FOIT**  
Calculates **SpeedIndex**

SpeedIndex takes the visual progress of the visible page loading and computes an overall score for how quickly the content painted

# *WebPageTest*

*Inspect every request*

*Analyze TCP traffic*

*Identify bottlenecks*

*Visualize progress*

*Automate!*

Measure early.

Measure often.

 Mobile

Desktop

52 / 100 Speed

 Should Fix:

Optimize images

▶ Show how to fix

Eliminate render-blocking JavaScript and CSS

▶ Show how to fix

 Consider Fixing:

Enable compression

▶ Show how to fix

Prioritize visible content

▶ Show how to fix

Leverage browser caching

▶ Show how to fix

# npm install psi-g

```
nico@CommandCenter: ~/dev/psi (zsh)
~/dev/psi (zsh) % 1
~/dev/psi
» psi http://ponyfoo.com

URL: http://ponyfoo.com/
Score: 88
Strategy: desktop

Number Resources 18
Number Hosts 10
Total Request 1.8 kB
Number Static Resources 11
Html Response 289.46 kB
Css Response 31.31 kB
Image Response 295.96 kB
Javascript Response 330.56 kB
Other Response 80.1 kB
Number Js Resources 5
Number Css Resources 2

Avoid Landing Page Redirects 0
Enable Gzip Compression 0
Leverage Browser Caching 1.5
Main Resource Server Response Time 0
Minify Css
Minify HTML
Minify Java Script
Minimize Render Blo
Optimize Images
Prioritize Visible Content

var gulp = require('gulp');
var psi = require('psi');
gulp.task('mobile', function (cb) {
  psi({
    nokey: 'true',
    url: 'http://ponyfoo.com',
    strategy: 'mobile',
  }, cb);
});
```

nico@CommandCenter: ~/dev (zsh)

~/dev (zsh) #1

~/dev

» webpagetest test ponyfoo.com

{

```
"statusCode": 200,  
"statusText": "Ok",  
"data": {  
    "testId": "150408_BK_13YW",  
    "ownKey": "ccccd9e7ed3aff4c666c960eb199070f32f0acb9",  
    "jsonUrl": "http://www.webpagetest.org/jsonResult.php?test=150408_BK_13YW",  
    "xmlUrl": "http://www.webpagetest.org/xmlResult/150408_BK_13YW",  
    "userUrl": "http://www.webpagetest.org/result/150408_BK_13YW/",  
    "summaryCSV": "http://www.webpagetest.org/result/150408_BK_13YW/page_data.csv",  
    "detailCSV": "http://www.webpagetest.org/result/150408_BK_13YW/requests.csv"  
}
```

E

~/dev

» curl http://www.webpagetest.org/jsonResult.php?test=150408\_ND\_13SC | underscore process data.data.runs[1].firstView.breakdown

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current	
Dload	Upload	Total	Spent	Left	Speed			
100	161k	0	161k	0	0	55844	0	--:--:-- 0:00:02 --:--:-- 55830

{

```
"html": { "color": [130, 181, 252], "bytes": 12273, "requests": 2 },  
"js": { "color": [254, 197, 132], "bytes": 158503, "requests": 9 },  
"css": { "color": [178, 234, 148], "bytes": 17621, "requests": 2 },  
"image": { "color": [196, 154, 232], "bytes": 38748, "requests": 4 },  
"flash": { "color": [45, 183, 193], "bytes": 0, "requests": 0 },  
"font": { "color": [255, 82, 62], "bytes": 65752, "requests": 4 },  
"other": { "color": [196, 196, 196], "bytes": 780, "requests": 1 }  
}
```

npm install webpagetest-api underscore-cli

```
nico@CommandCenter: ~/dev/yslow (zsh)
~/dev/yslow (zsh) #1
```

```
~/dev/yslow
» grunt yslow
Running "yslow:pages" (yslow) task
>> Testing 1 URLs, this might take a few moments...
>>
>> -----
>> Test 1: http://ponyfoo.com/articles/critical-path-p
>> -----
>> Requests: 10 [PASS]
>> YSlow score: 97/100 [PASS]
```

```
nico@CommandCenter: ~/dev/ponyfoo (zsh) #1
~/dev/ponyfoo (zsh) #1
```

```
master ~/dev/ponyfoo
» phantomjs yslow.js --info basic --f
version: 3.1.8
size: 361.5K (361506 bytes)
overall score: A (94)
```

url: http://ponyfoo.com/

# of requests: 11  
ruleset: ydefault

page load time: 2993

chrome-extension://ninejjcohidippngpapii

Home Grade Components Statistics

**Grade A** Overall performance score 96 Ruleset applied: YSlow(V2) URL: http://ponyfoo.com/articles/critical-path-p

[ALL \(23\)](#) FILTER BY: [CONTENT \(6\)](#) | [COOKIE \(2\)](#) | [CSS \(6\)](#) | [IMAGES \(2\)](#) | [JAVASCRIPT \(1\)](#)

A Make fewer HTTP requests

D Use a Content Delivery Network (CDN)

A Avoid empty src or href

B Add Expires headers

A Compress components with gzip

A Put CSS at top

A Put JavaScript at bottom

A Avoid CSS expressions

n/a Make JavaScript and CSS external

B Reduce DNS lookups

A Minify JavaScript and CSS

A Avoid URL redirects

A Remove duplicate JavaScript and CSS

A Configure entity tags (ETags)

A Make AJAX cacheable

A Use GET for AJAX requests

A Reduce the number of DOM elements

A Avoid HTTP 404 (Not Found) error

A Reduce cookie size

B Use cookie-free domains

A Avoid Alphabetic Leaver Error

A Do not scale images in HTML

A Make favicon small and cacheable

Grade A on Make fewer HTTP requests

Decreasing the number of components required to render the page, resulting in faster page loads. Components include: combine multiple CSS files into one.

[»Read More](#)

Copyright © 2014 Yahoo! Inc. All rights reserved.

# Budgets!

Enforce a ***performance budget***

Track impact of ***every commit***

## What should I track?

# *Milestone Timings*

Load time, time to interact, "time to first tweet"

## *SpeedIndex*

Average time at which parts of a page are displayed

## *Quantity based metrics*

Request count, page weight, image weight...

## *Rule based metrics*

YSlow grade, PageSpeed score, etc.

```
perfbudget: {  
  default: {  
    options: {  
      url: 'http://  
      key: 'API_KEY'  
      runs: 5,  
      budget: {  
        visualCompl  
        SpeedIndex:  
      }  
    }  
  }  
}
```

```
nico@CommandCenter: ~/dev/perfbudget (zsh)  
..ev/perfbudget (zsh) ⌘1  
~/dev/perfbudget  
» grunt perfbudget  
Running "perfbudget:default" (perfbudget) task  
->  
-> Test for http://ponyfoo.com PASSED  
->  
-> render: 395 [PASS]. Budget is 1000  
-> SpeedIndex: 432 [PASS]. Budget is 1000  
-> Summary: http://www.webpagetest.org/result/150408_08_18BR/  
  
Done, without errors.  
~/dev/perfbudget  
» |
```

npm install grunt-perfbudget --save-dev

# *Beyond minification*

Or, what can be done  
to improve performance?

# *Web Stack*

TCP

HTTP(s)

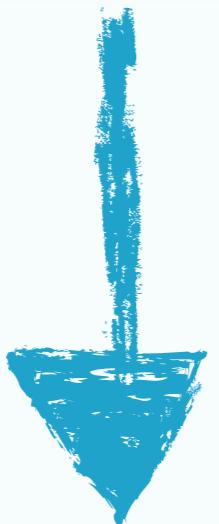
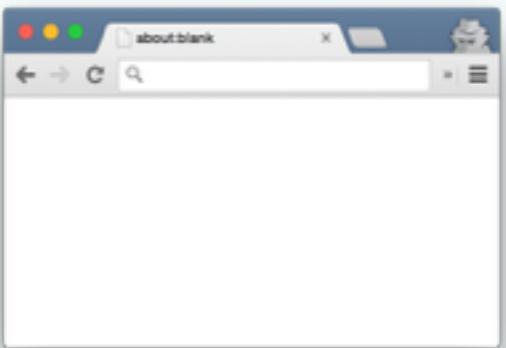
HTML

CSS

Fonts

Images

JavaScript



# *Networking*

But *not* the kind  
you do at conferences

*What Every Web Developer Should Know About  
Networking and Browser Performance*



*High Performance*  
**Browser  
Networking**

O'REILLY®

Ilya Grigorik

# *Optimizing TCP*

## Increase initial TCP cwnd size

Sender-side limit for in-flight data (**cwnd**)

More data in first TCP roundtrip

Accelerates connection ramp up

## Disable slow-start restart (SSR)

Long-lived, bursty TCP connections can't afford SSR

Improve HTTP performance by disabling it

```
sysctl -w net.ipv4.tcp_slow_start_after_idle = 0
```

# *Optimizing HTTP*

Make less requests!

Turn on keep-alive

GZip all the text

Add Expires and ETag headers

Use a CDN

# HTTP/2!

## SPDY?

Yes.

# *Why?*

*Non-blocking Multiplexing*

*One Connection per Origin*

*Header Compression*

*Proactive Server Push*

*The Web*

Needs you

#perfatters

# *Optimizing* HTML

Render server-side!

*Become a Single Page App later*

Defer non-critical asset loading

Keep it accessible with aria

# *Optimizing CSS*

Inline critical CSS

Remove unused styles

Avoid m. Be responsive

Concatenate and minify

Follow a style guide. Seriously.

# *Optimizing Fonts*

Load asynchronously

Use a fallback while fonts load

Prevent FOIT using JavaScript

Use fewer fonts, avoid repaints

Cache them aggressively

# *Optimizing* Images

Minify and shrink

Defer images below the fold

Create spritesheets with tools

Try inlining tiny dynamic images

Use CSS for simple icons

# *Optimizing JavaScript*

You can live without it

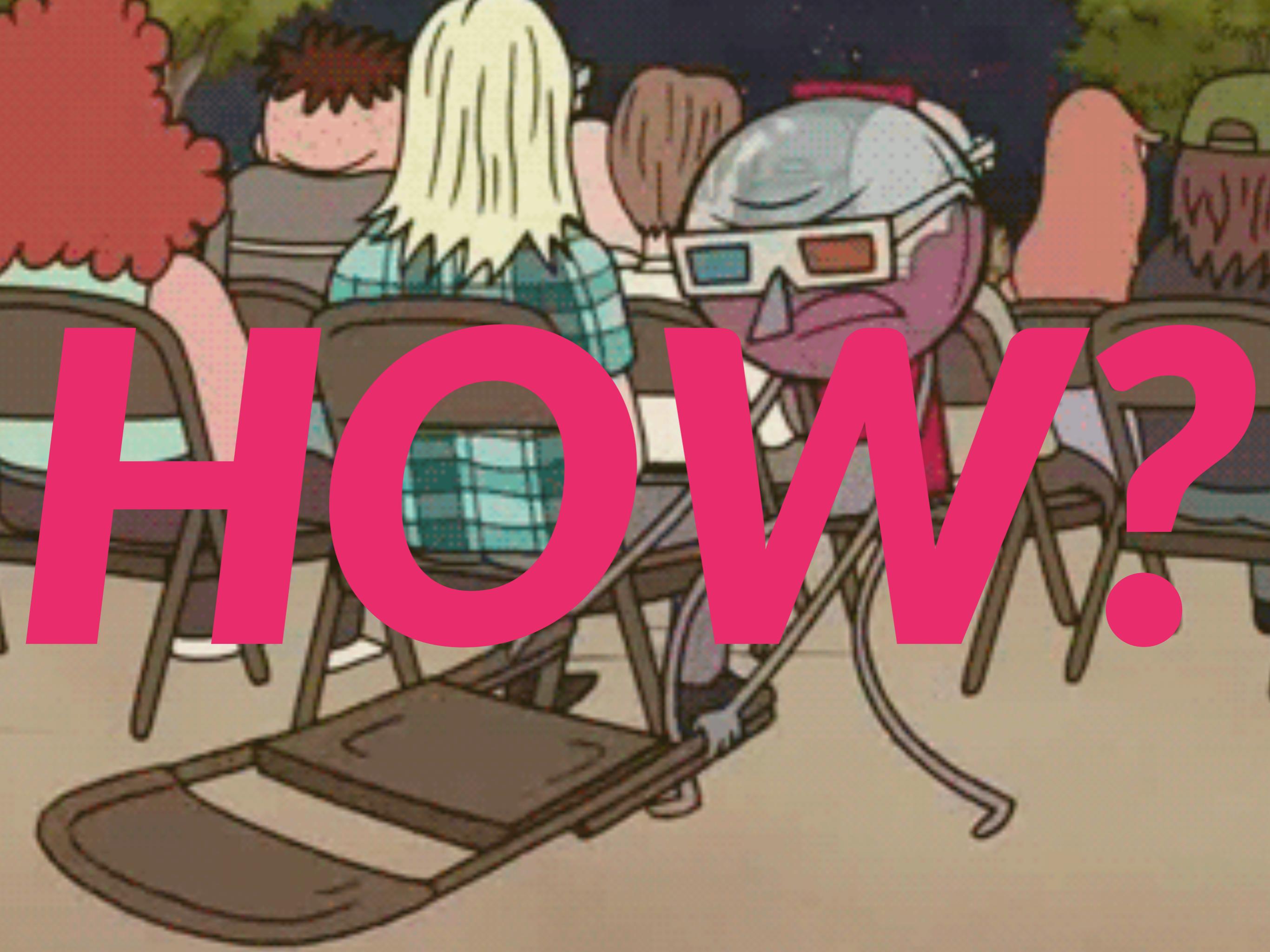
Defer *all* of it

Use small modules

Use asset hashing

Cache vendor scripts separately

**HOW?**





*Reverse proxy*  
*Static asset caching*  
*Supports SPDY*  
*GZip compression*

# *Set up a CDN!*

**fastly**



**CLOUDFLARE®**



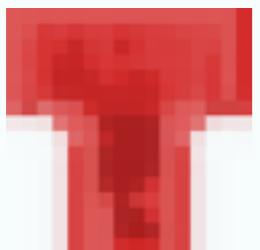
# *Shared Rendering*



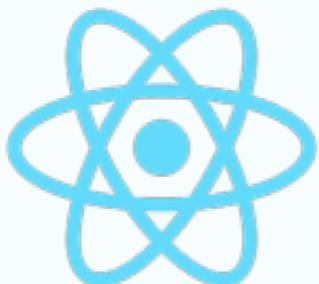
Needs Rendr



Presumably in 2.0



Native support



Native support

# Defer Assets

```
<script src='/js/all.js' async></script>
```

```
<noscript>
| <link rel='stylesheet' type='text/css' href='http://fonts.googleapis.com/css?family=Merriweather:400,700'>
</noscript>
```

```
~function (document) {
  var elem = document.createElement('link');
  var head = document.getElementsByTagName('head')[0];
  elem.rel = 'stylesheet';
  elem.href = 'http://fonts.googleapis.com/css?family=Merriweather:400,700';
  elem.media = 'only x';
  head.appendChild(elem);
  setTimeout(function () {
    elem.media = 'all';
  });
}(document);
```

# Remove Unused CSS

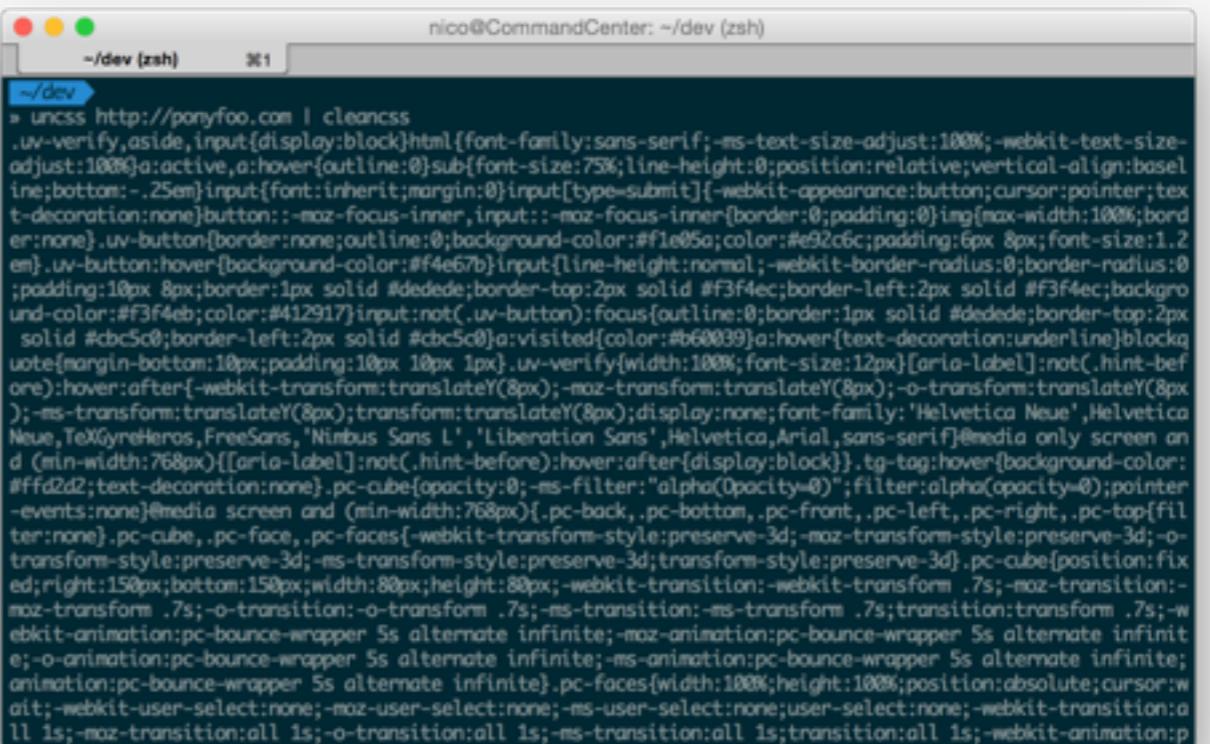
**Useful when using  
Bootstrap or other  
CSS frameworks**

```
var gulp = require('gulp');
var uncss = require('gulp-uncss');

gulp.task('default', function() {
  return gulp.src('all.css')
    .pipe(uncss({
      html: ['http://ponyfoo.com']
    }))
    .pipe(gulp.dest('./build'));
});
```

```
uncss: {
  dist: {
    files: {
      'dist/css/tidy.css': ['app/in']
    }
  }
}
```

**Only what's needed!**



# Inline Critical CSS

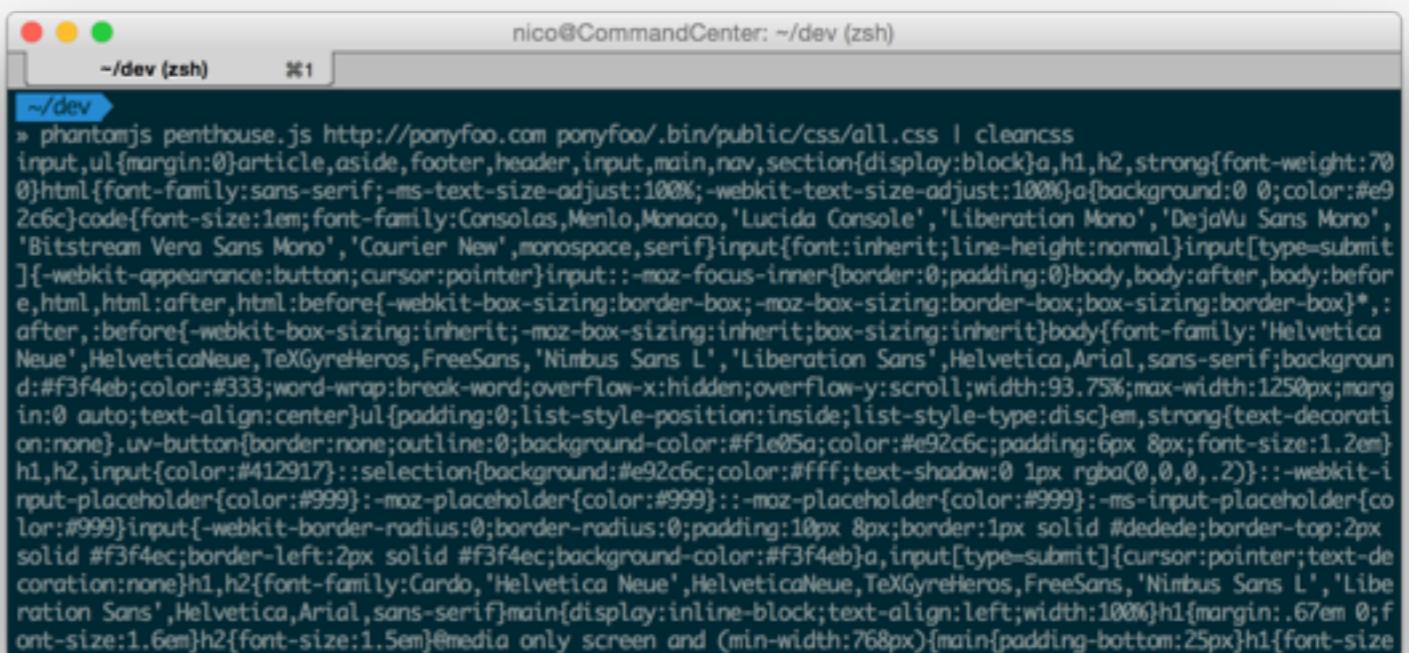
```
var penthouse = require('penthouse');
var path = require('path');

penthouse({
  url : 'http://ponyfoo.com',
  css : 'build/css/all.css',
}, function(err, critical) {
  console.log(critical);
});
```

**Remove  
roadblocks from  
the critical path**

```
<style>
  .inline-styles-go-here {
  }
</style>
```

**Only what's seen!**



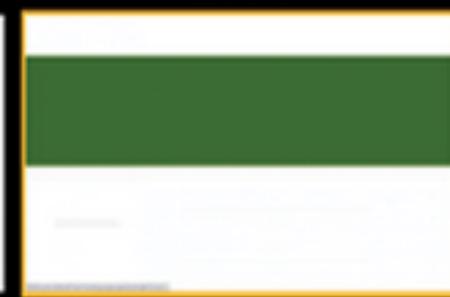
The terminal window shows the command being run: `* phantomjs penthouse.js http://ponyfoo.com ponyfoo/.bin/public/css/all.css | cleancss`. The output displays a large amount of CSS code, which is the result of inlining critical styles. The CSS includes various rules for HTML elements like `input`, `ul`, `article`, `aside`, `footer`, `header`, `input`, `main`, `nav`, `section`, and `strong`. It also includes rules for `h1` and `h2` headings, and `input[type=submit]` buttons. The CSS uses a mix of sans-serif and monospace fonts, and includes vendor-specific properties like `-webkit-appearance:button` and `-moz-focus-inner`.

# Use a font loader

npm install fontfaceonload

Prevent **FOIT!**

1.6s



1.7s



1.8s



1.9s

...800ms...



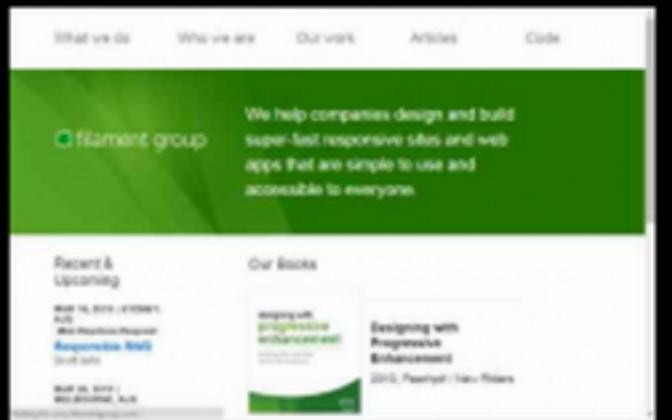
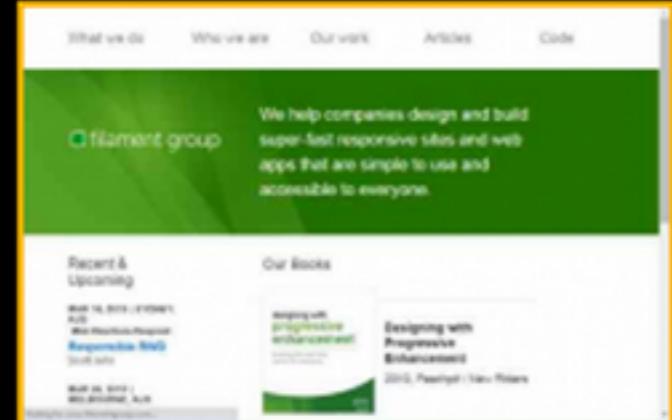
```
FontFaceOnload("My Custom Font Family", {  
  success: function() {  
    document.documentElement.className += " my-custom-font-family";  
  }  
});
```

0.3s

0.4s

0.5s

0.6s



# Optimize Images

```
var Imagemin = require('imagemin');
```

```
var imagemin = new Imagemin()  
.use(Imagemin.jpegtran({progressive: true}));
```

```
$ imagemin --help
```

Usage

```
$ imagemin <file> <directory>  
$ imagemin <directory> <output>  
$ imagemin <file> > <output>  
$ cat <file> | imagemin > <output>
```

Up to 90%  
*in savings!*

```
imagemin: {  
  release: {  
    files: [{  
      expand: true,  
      src: 'build/img/**/*.{jpg}'  
    }],  
    options: {  
      progressive: true  
    }  
  }  
}
```

# Create Spritesheets

```
var spritesmith = require('spritesmith');
var sprites = ['fork.png', 'github.png', 'twitter.png'];
spritesmith({ src: sprites }, function (err, result) {
  result.image; // binary string representation of image
  result.coordinates; // maps filename to {x, y, width, height} of image
  result.properties; // spritesheet metadata {width, height}
});
```

```
grunt.initConfig({
  sprite: {
    icons: {
      src: 'public/img/icons/*.png',
      destImg: 'build/img/icons.png',
      destCSS: 'build/css/icons.css'
    }
  }
});
```

**Save bandwidth  
by keeping the  
noise to a  
minimum**

# *Inline Images*

**Save a request  
by inlining a  
base64 image**

```
npm install datauri  
datauri ponyfoo.png
```

```
var Datauri = require('datauri'),  
dUri      = Datauri('test/myfile.png');  
  
console.log(dUri); //=> "data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAA..."
```

**Careful: Data URLs are slow on mobile!**

# Use a module system



```
var $ = require('dominus');
var taunus = require('taunus');
var moment = require('moment');
```



```
// lib/math.js
export function sum(x, y) {
  return x + y;
}
export var pi = 3.141593;
```

```
// app.js
import * as math from "lib/math";
alert("2π = " + math.sum(math.pi, math.pi));
```

```
// otherApp.js
import {sum, pi} from "lib/math";
alert("2π = " + sum(pi, pi));
```



INTERACT

*In Summary...*

Measure  
your  
metrics

*In Summary...*

Automate  
your  
metrics

*In Summary...*

Define  
your  
budget

*In Summary...*

Enforce

your  
budget

*In Summary...*

Prioritize  
your  
content

*In Summary...*

Optimize  
your  
content

*In Summary...*

Respect  
your  
humans

*In Summary...*

Respect  
our  
web

*bevacqua.io/bf*

*Automation  
Performance  
Modularity  
Asynchronicity  
MVC  
Testing  
REST API  
~100 Code Samples*



**JavaScript  
Application Design**

A Build First approach

Nicolas G. Bevacqua



4 Mayo

@BeerJSBA

Puerta Roja

San Telmo

Buenos Aires



*Thanks!*

@nzgb  
ponyfoo.com