Why web performance?

Web Performance

A users journey to understand load time

Harry Mumford-Turner

Overview

1. Why are slow sites a problem
2. How fast should sites load
3. Measuring load time
4. Measuring improvement
5. What can we do
    a. Small Improvements
    b. Time Consuming Improvements

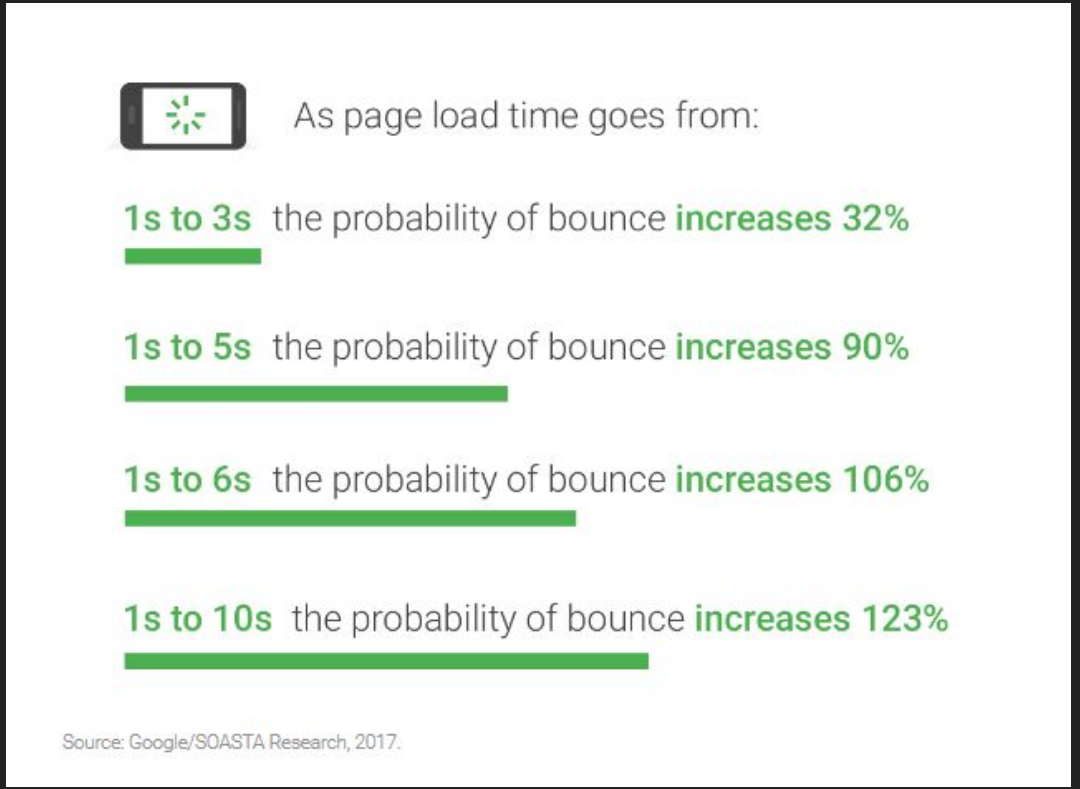
Why are slow sites a problem?

5 seconds earn 2x ad revenue than 19 seconds

Google Data from Google Analytics and DoubleClick, 2015-2016

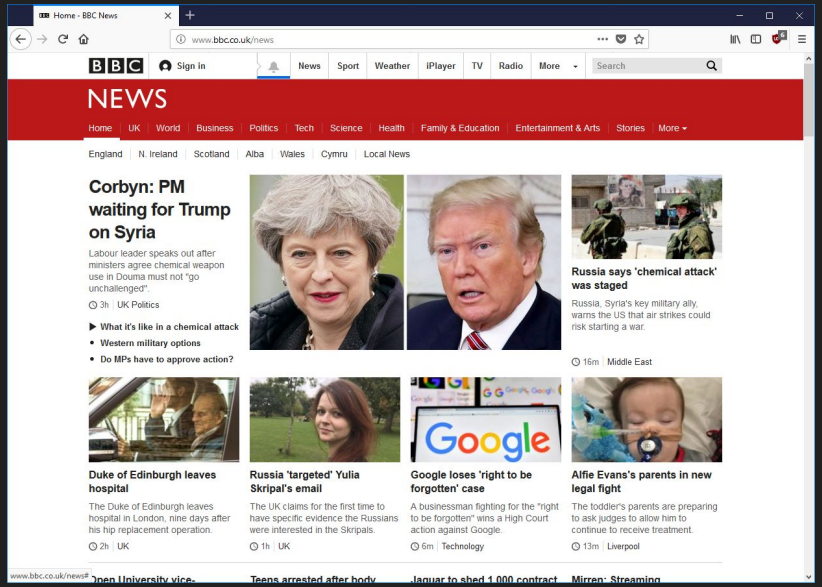
53%

Abandon after
3 seconds



https://www.soasta.com/blog/google-mobile-web-performance-study/ 2016

BBC found every additional second a page takes to load, 10% of users leave



https://www.creativebloq.com/features/how-the-bbc-builds-websites-that-scale

How fast *should* sites load?

49% EXPECT <3 SECONDS

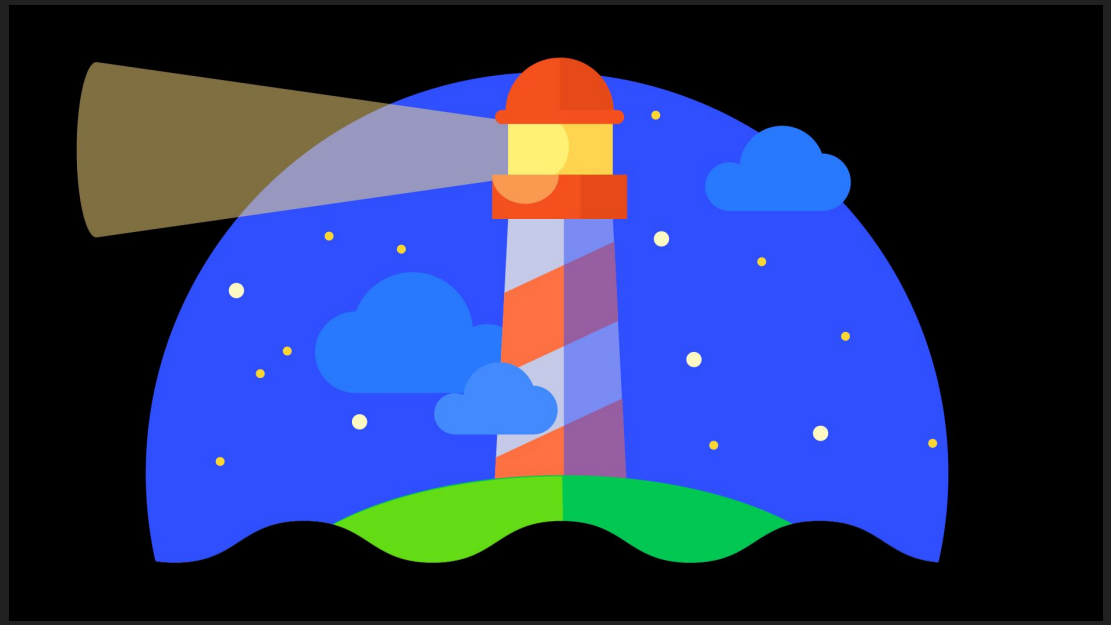
Akamai Technologies, “Consumer Web Performance Expectations Survey”, 2014

22 seconds

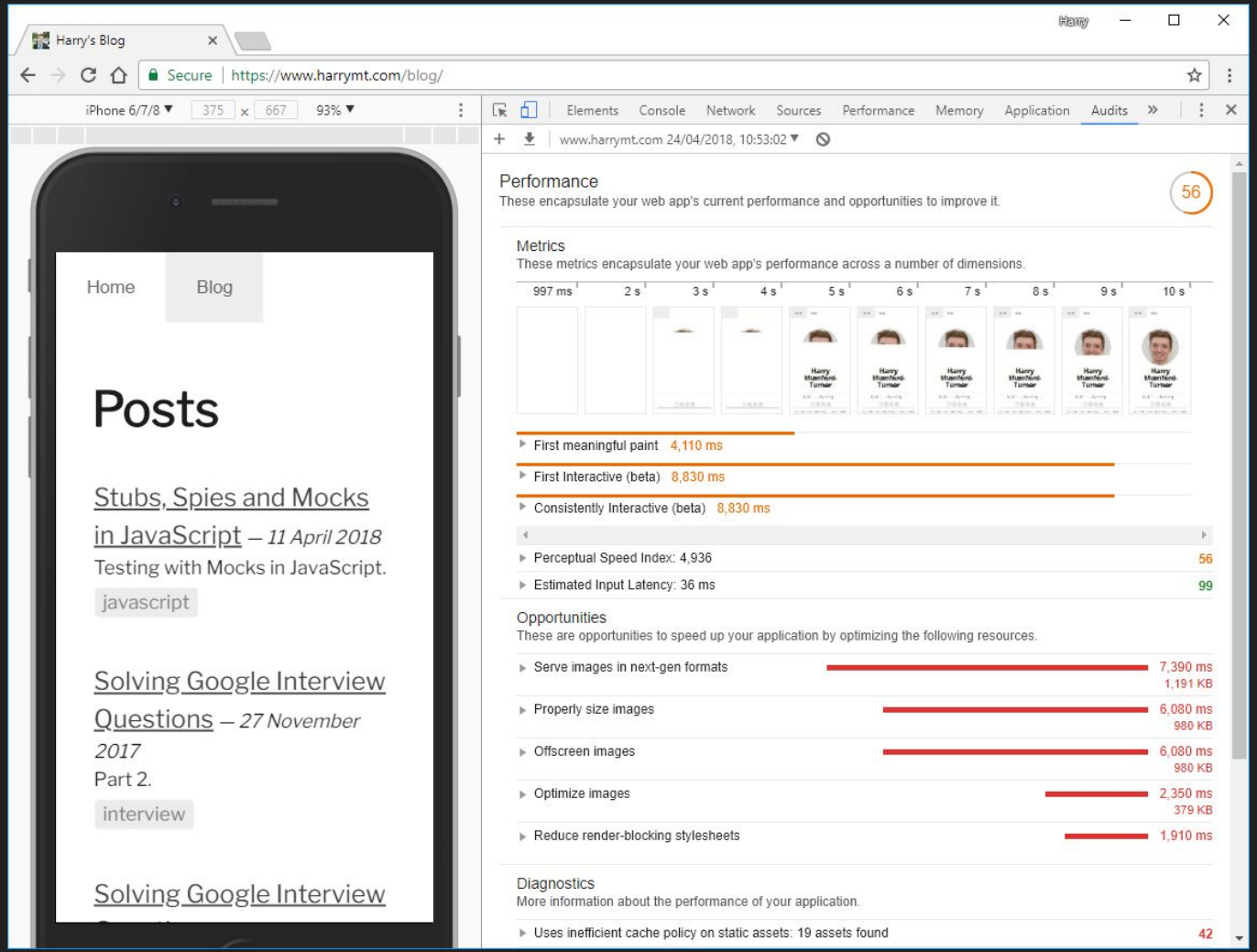
The average time it takes to fully load a mobile landing page

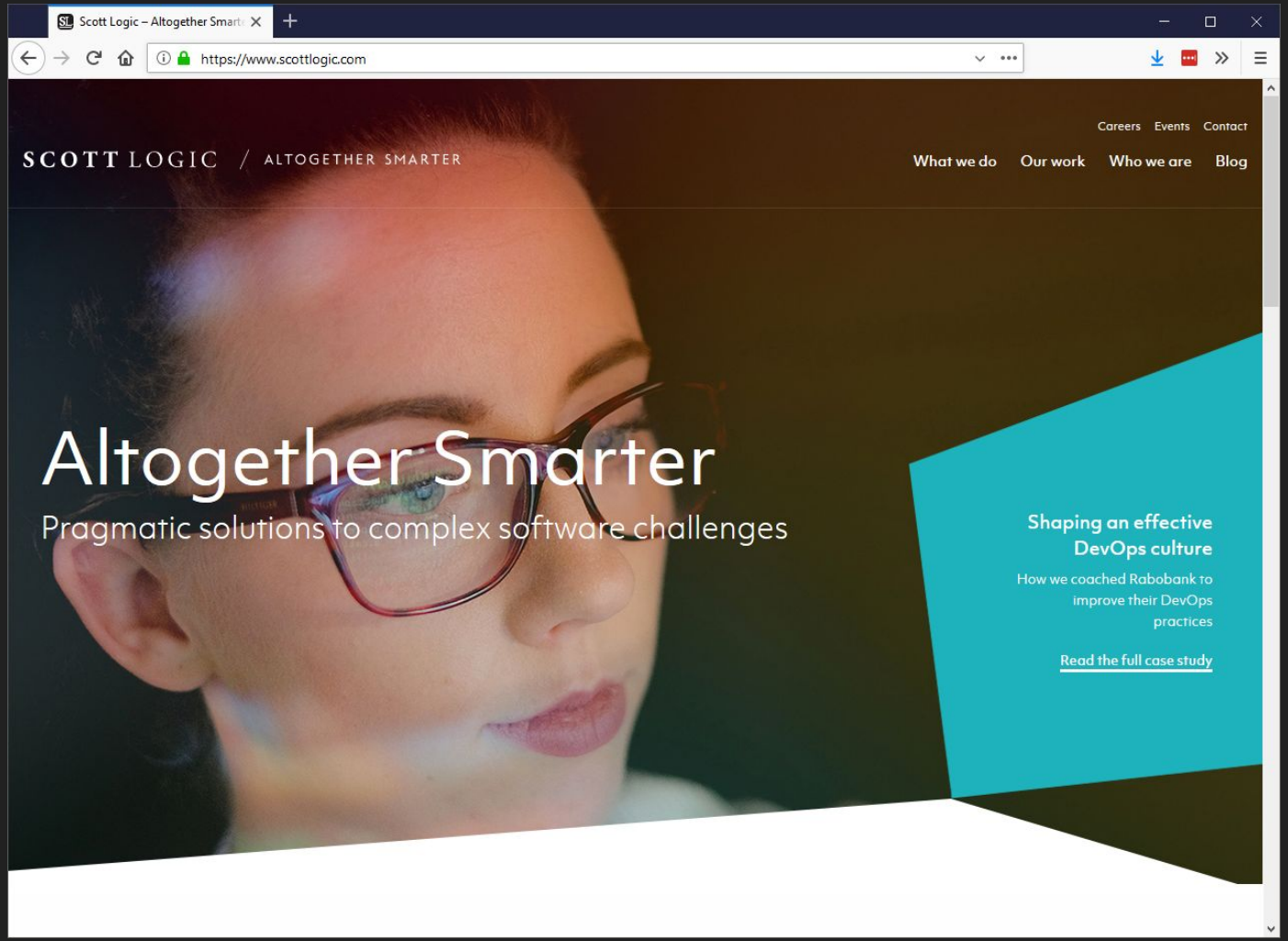
2017 Google Research with 900,000 websites

How can we measure load time?



```
$ npx lighthouse https://www.harrymt.com/
```


Harry's Blog
Secure https://www.harrymt.com/blog/
iPhone 6/7/8 375 x 667 93%
Elements Console Network Sources Performance Memory Application Audits
www.harrymt.com 24/04/2018, 10:53:02
Home Blog
Posts
Stubs, Spies and Mocks in JavaScript – 11 April 2018
Testing with Mocks in JavaScript.
javascript
Solving Google Interview Questions – 27 November 2017
Part 2.
interview
Solving Google Interview
Performance
These encapsulate your web app's current performance and opportunities to improve it.
56
Metrics
These metrics encapsulate your web app's performance across a number of dimensions.
997 ms 2 s 3 s 4 s 5 s 6 s 7 s 8 s 9 s 10 s
First meaningful paint 4,110 ms
First Interactive (beta) 8,830 ms
Consistently Interactive (beta) 8,830 ms
Perceptual Speed Index: 4,936 56
Estimated Input Latency: 36 ms 99
Opportunities
These are opportunities to speed up your application by optimizing the following resources.
Serve images in next-gen formats 7,390 ms 1,191 KB
Properly size images 6,080 ms 980 KB
Offscreen images 6,080 ms 980 KB
Optimize images 2,350 ms 379 KB
Reduce render-blocking stylesheets 1,910 ms
Diagnostics
More information about the performance of your application.
Uses inefficient cache policy on static assets: 19 assets found 42


Scott Logic – Altogether Smarter
https://www.scottlogic.com
SCOTT LOGIC / ALTOGETHER SMARTER
Careers Events Contact
What we do Our work Who we are Blog
Altogether Smarter
Pragmatic solutions to complex software challenges
Shaping an effective DevOps culture
How we coached Rabobank to improve their DevOps practices
Read the full case study

Lighthouse
Version: 2.9.1

Results for: https://www.scottlogic.com/
Apr 8, 2018, 6:43 PM GMT+1 • Runtime settings

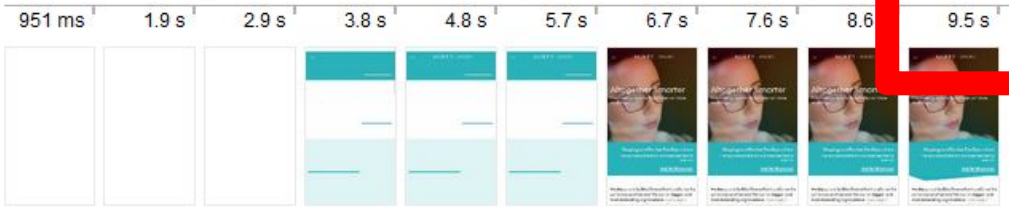
Performance	45
Progressive Web App	55
Accessibility	51
Best Practices	81
SEO	80

45 Performance | 55 Progressive Web App | 51 Accessibility | 81 Best Practices | 80 SEO

Performance
These encapsulate your web app's current performance and opportunities to improve it.

45

Metrics
These metrics encapsulate your web app's performance across a number of dimension



- First meaningful paint 6,110 ms
- First Interactive (beta) 9,350 ms
- Consistently Interactive (beta) 9,350 ms
- Perceptual Speed Index: 6,304 42

9 seconds

Load Time

www.scottlogic.com

Lighthouse

Version: 2.9.1

- Performance 45
- Progressive Web App 55
- Accessibility 51
- Best Practices 81
- SEO 80

Results for: https://www.scottlogic.com/

Apr 8, 2018, 6:43 PM GMT+1 • Runtime settings

Performance	Progressive Web App	Accessibility	Best Practices	SEO
45	55	51	81	80

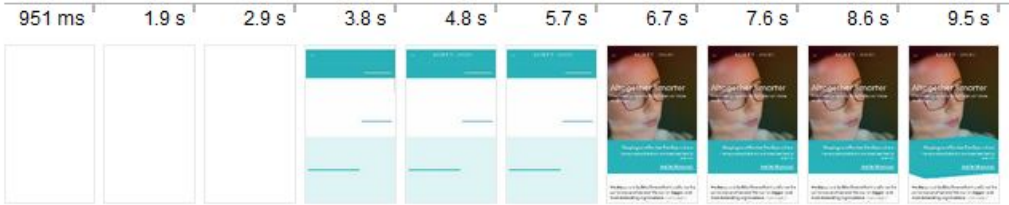
Performance

These encapsulate your web app's current performance and opportunities to improve it.

45

Metrics

These metrics encapsulate your web app's performance across a number of dimensions.



- First meaningful paint 6,110 ms
- First Interactive (beta) 9,350 ms
- Consistently Interactive (beta) 9,350 ms
- Perceptual Speed Index: 6,304 42

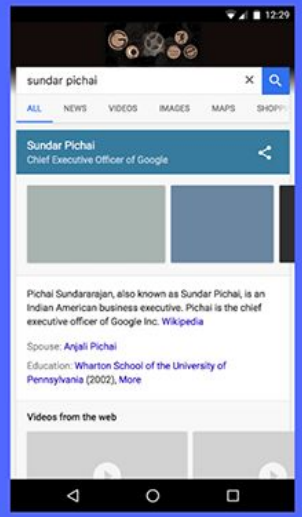

npx
npm
node_modules
# Before
$ node ./node_modules/.bin/mocha
# Now with npx:
$ npx mocha
npx
/node_modules
Establishing secure ...

User-centric performance metrics

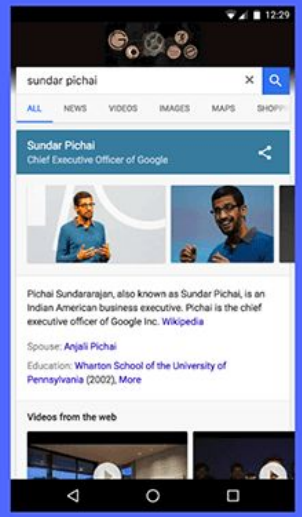
The Experience	The Metric
Is it happening?	First Paint (FP)
Is it useful?	First Meaningful Paint (FMP)
Is it usable?	Time to Interactive (TTI)
Is it delightful?	Long Tasks (technically the absence of long tasks)

First Paint
(FP)

First Contentful
Paint (FCP)



First Meaningful
Paint (FMP)



Time to
Interactive (TTI)

Lighthouse

Version: 2.9.1

- Performance 45
- Progressive Web App 55
- Accessibility 51
- Best Practices 81
- SEO 80

Results for: https://www.scottlogic.com/

Apr 8, 2018, 6:43 PM GMT+1 • Runtime settings

Performance	Progressive Web App	Accessibility	Best Practices	SEO
45	55	51	81	80

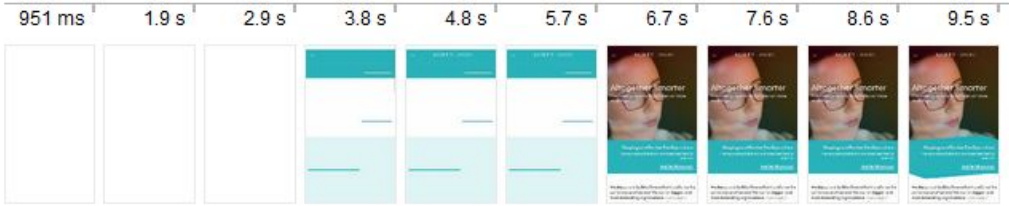
Performance

These encapsulate your web app's current performance and opportunities to improve it.

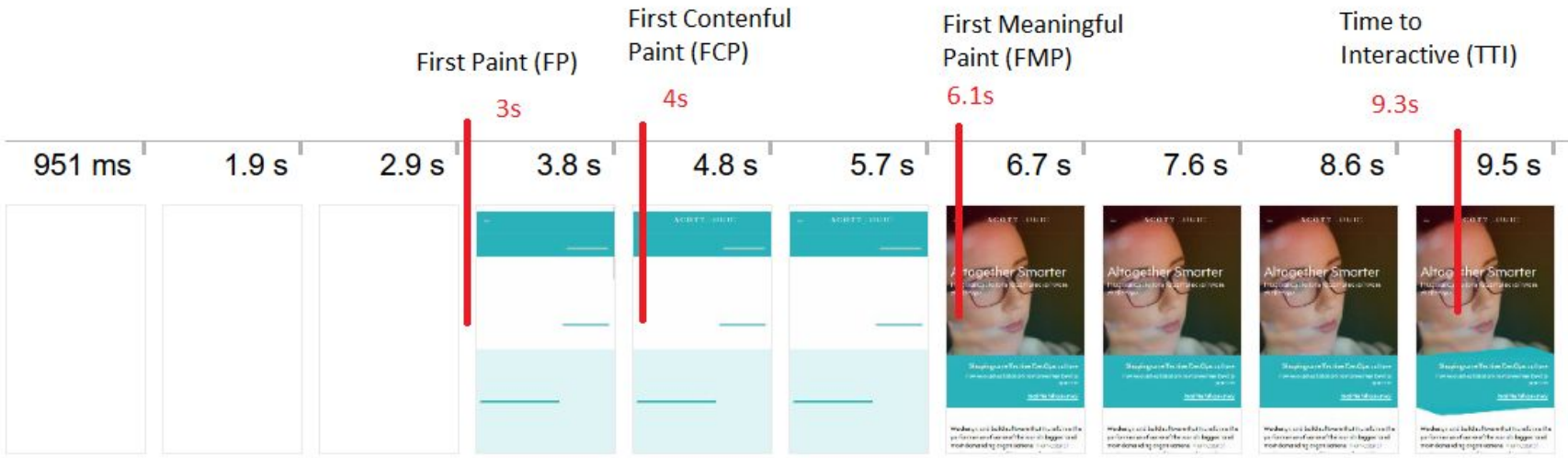
45

Metrics

These metrics encapsulate your web app's performance across a number of dimensions.



- First meaningful paint 6,110 ms
- First Interactive (beta) 9,350 ms
- Consistently Interactive (beta) 9,350 ms
- Perceptual Speed Index: 6,304 42



- First meaningful paint 6,110 ms
- First Interactive (beta) 9,350 ms
- Consistently Interactive (beta) 9,350 ms

6 seconds

First Meaningful Paint (FMP)

www.scottlogic.com

6 s nds

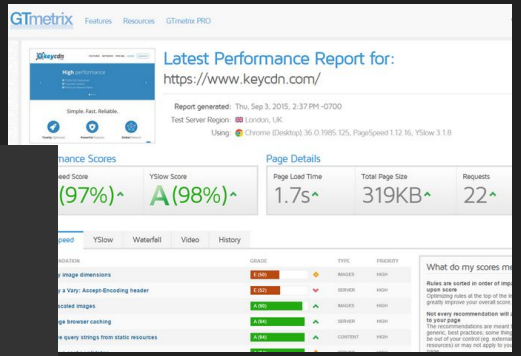
Meaningful Paint

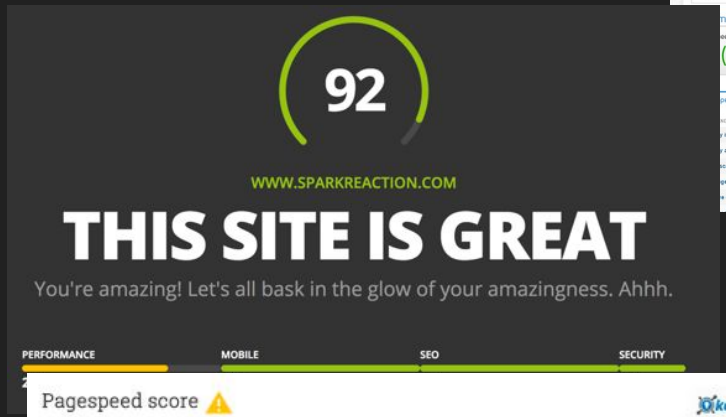
www.scottlogic.com

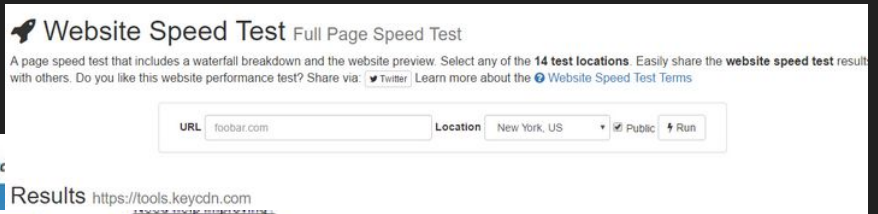
“Our site loads in X.XX seconds”

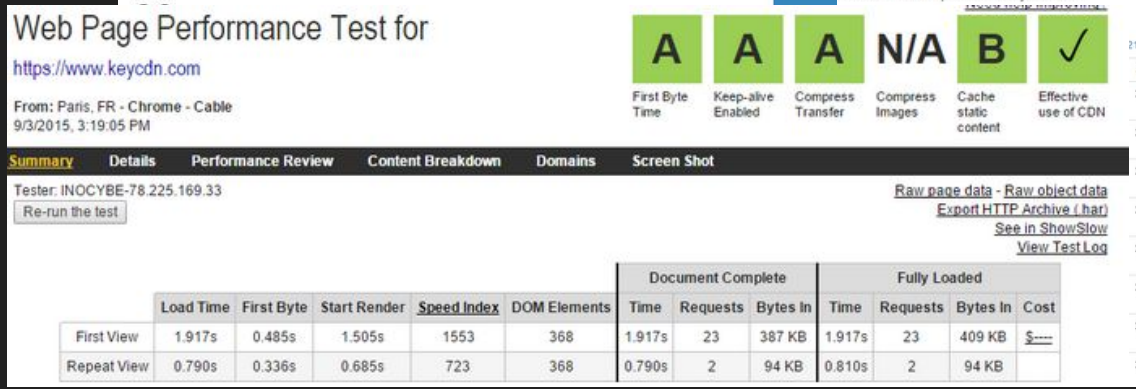
How can we measure improvements?

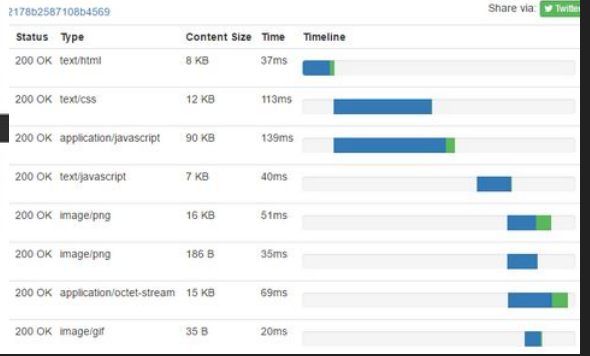
There are lots of free tools available

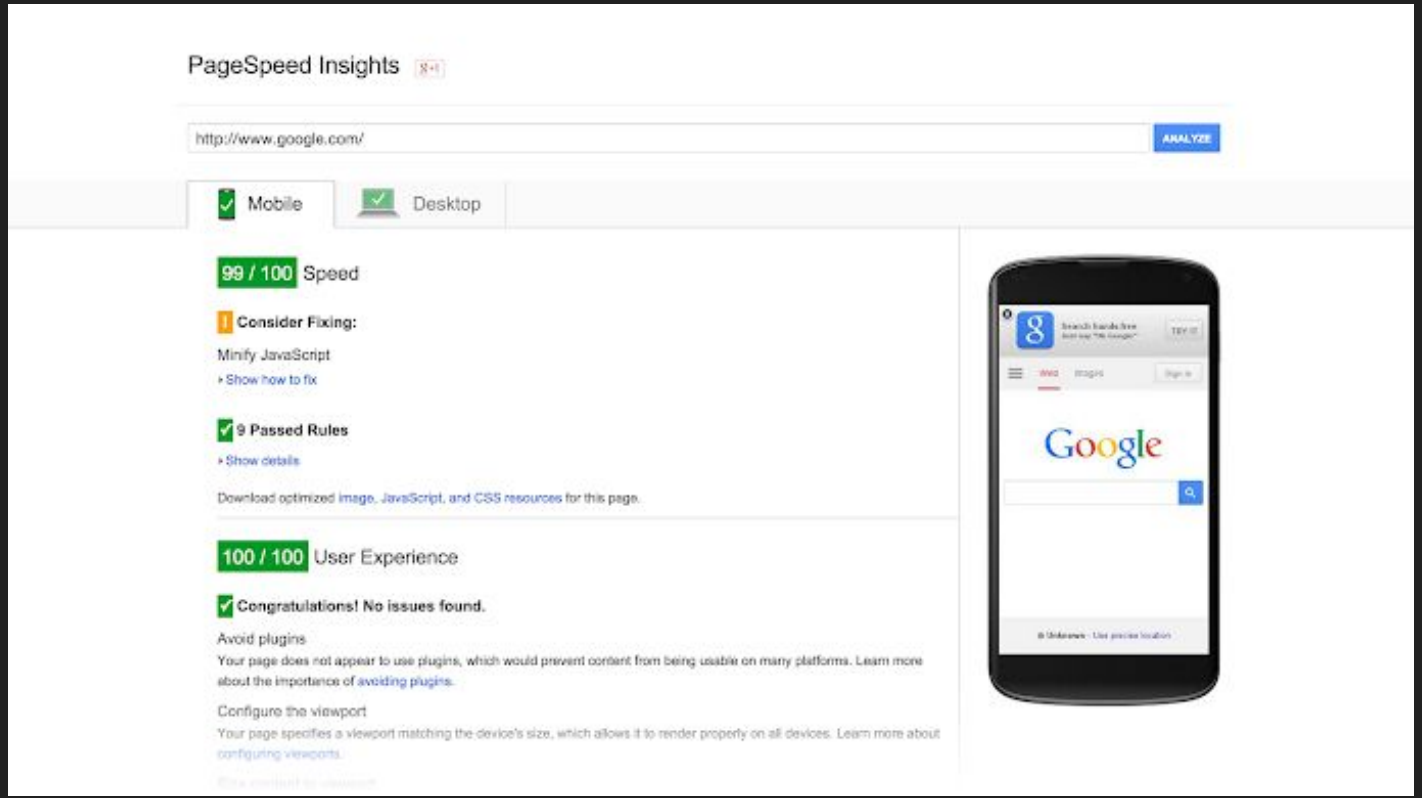











PageSpeed Insights
http://www.google.com/
ANALYZE
Mobile
Desktop
99 / 100 Speed
Consider Fixing:
Minify JavaScript
Show how to fix
9 Passed Rules
Show details
Download optimized image, JavaScript, and CSS resources for this page.
100 / 100 User Experience
Congratulations! No issues found.
Avoid plugins
Your page does not appear to use plugins, which would prevent content from being usable on many platforms. Learn more about the importance of avoiding plugins.
Configure the viewport
Your page specifies a viewport matching the device's size, which allows it to render properly on all devices. Learn more about configuring viewports.
Google

PageSpeed Insights

https://scottlogic.com/ ANALYZE

Mobile | Desktop

Speed

Fast

0.8s FCP **0.9s** DCL

Optimization

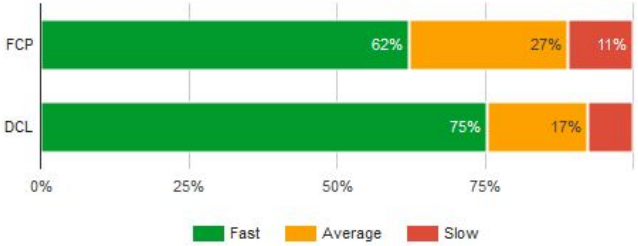
Low

57 / 100

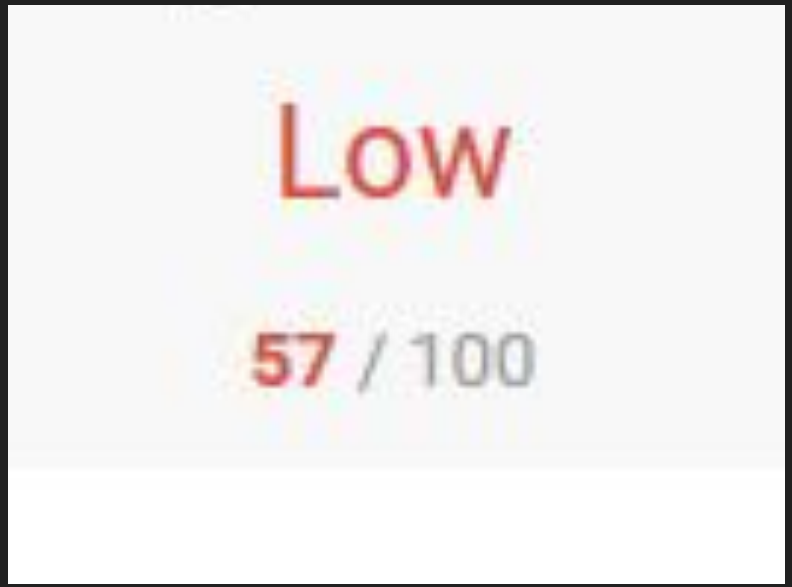
Data from the Chrome User Experience report indicates this page's median FCP (0.8s) and DCL (0.9s) ranks it in the fastest third of all pages. Although the page could be more optimized it is probably unnecessary. Learn more.

Report for: https://www.scottlogic.com/

Page Load Distributions



Desktop



PageSpeed Insights

https://scottlogic.com/ ANALYZE

Mobile | Desktop

Speed	Optimization
Unavailable	Low
	47 / 100

Data about the real-world performance of this page was unavailable. PageSpeed Insights was still able to analyze this page to find potential optimizations. Applying these optimizations may improve the speed of this page. Please investigate the recommendations below. Learn more.

Report for: https://www.scottlogic.com/

Page Stats

PSI estimates this page requires 8 additional round trips to load render blocking resources and 1.5 MB to fully render. The median page requires 4 render blocking round trips and 1.2 MB. Fewer round trips and bytes results in faster pages.

Optimization Suggestions

Eliminate render-blocking JavaScript and CSS in above-the-fold content

▸Show how to fix

Leverage browser caching

▸Show how to fix

Minify JavaScript

Mobile

pingdom.com

Summary

Performance grade	Load time	Faster than
B 84	520 ms	97 % of tested sites

Page size	Requests	Tested from
1.3 MB	37	Stockholm on Apr 8 at 19:51

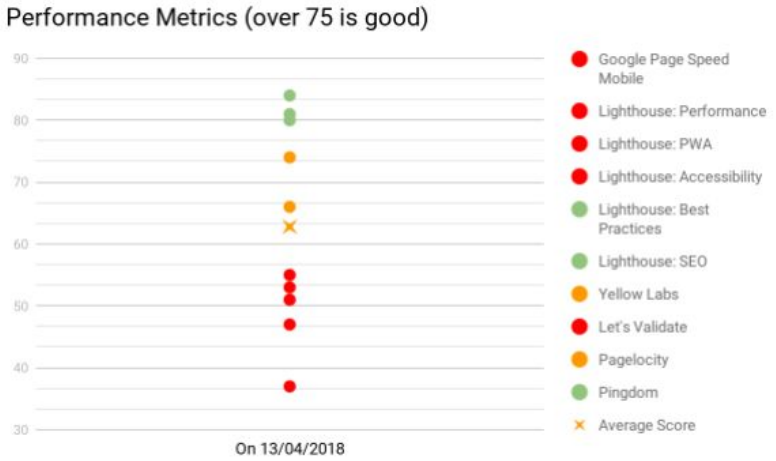
pingdom

Performance insights

GRADE	SUGGESTION
F 46	Remove query strings from static resources
E 59	Minimize redirects
B 85	Serve static content from a cookieless domain
B 89	Leverage browser caching
A 96	Specify a cache validator
A 96	Specify a Vary: Accept-Encoding header
A 100	Avoid bad requests
A 100	Minimize request size

Scottlogic.com Performance Review

Several optimisations can be made to www.scottlogic.com to improve, page load speed, accessibility and SEO.

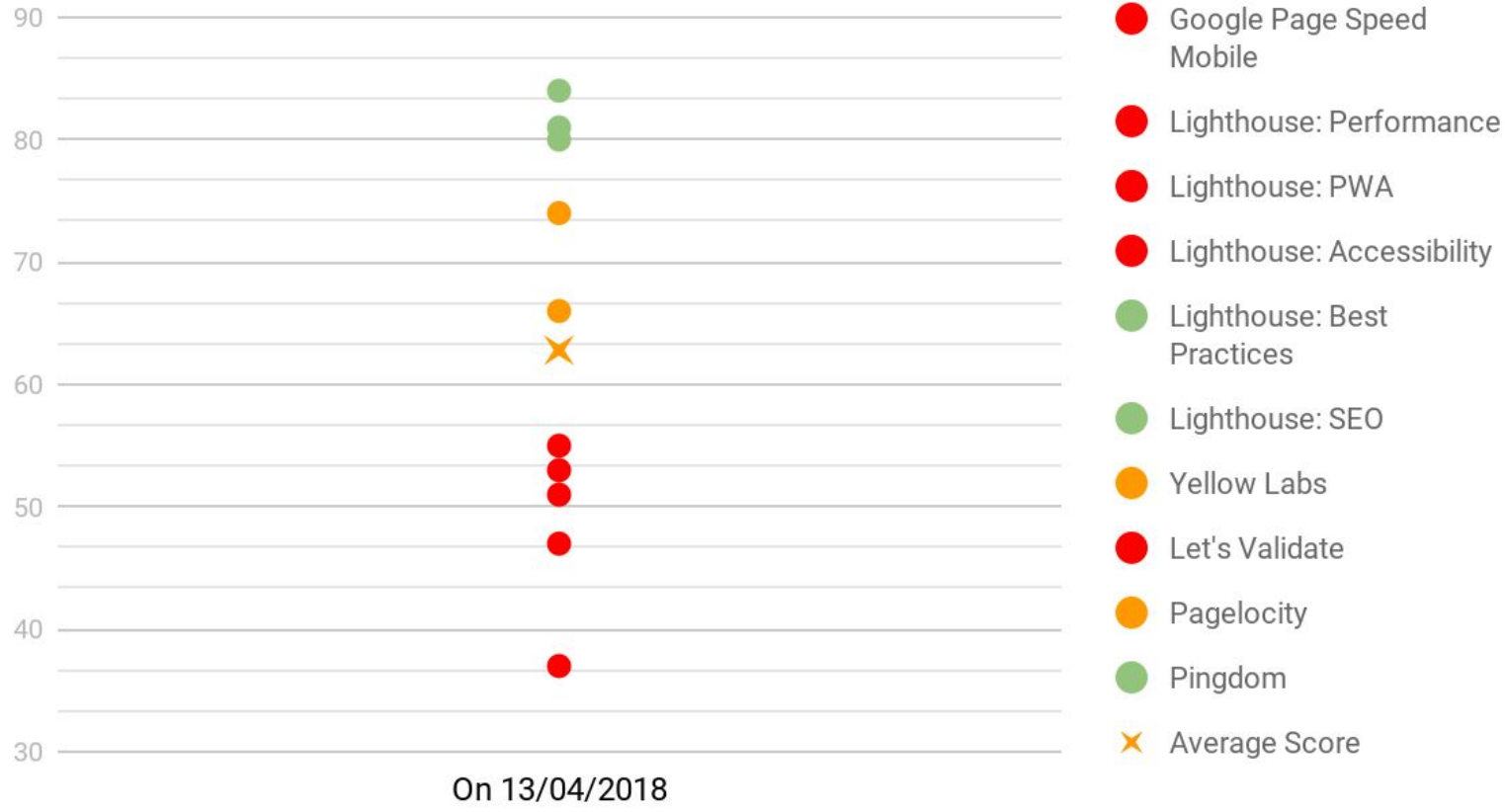


Metrics

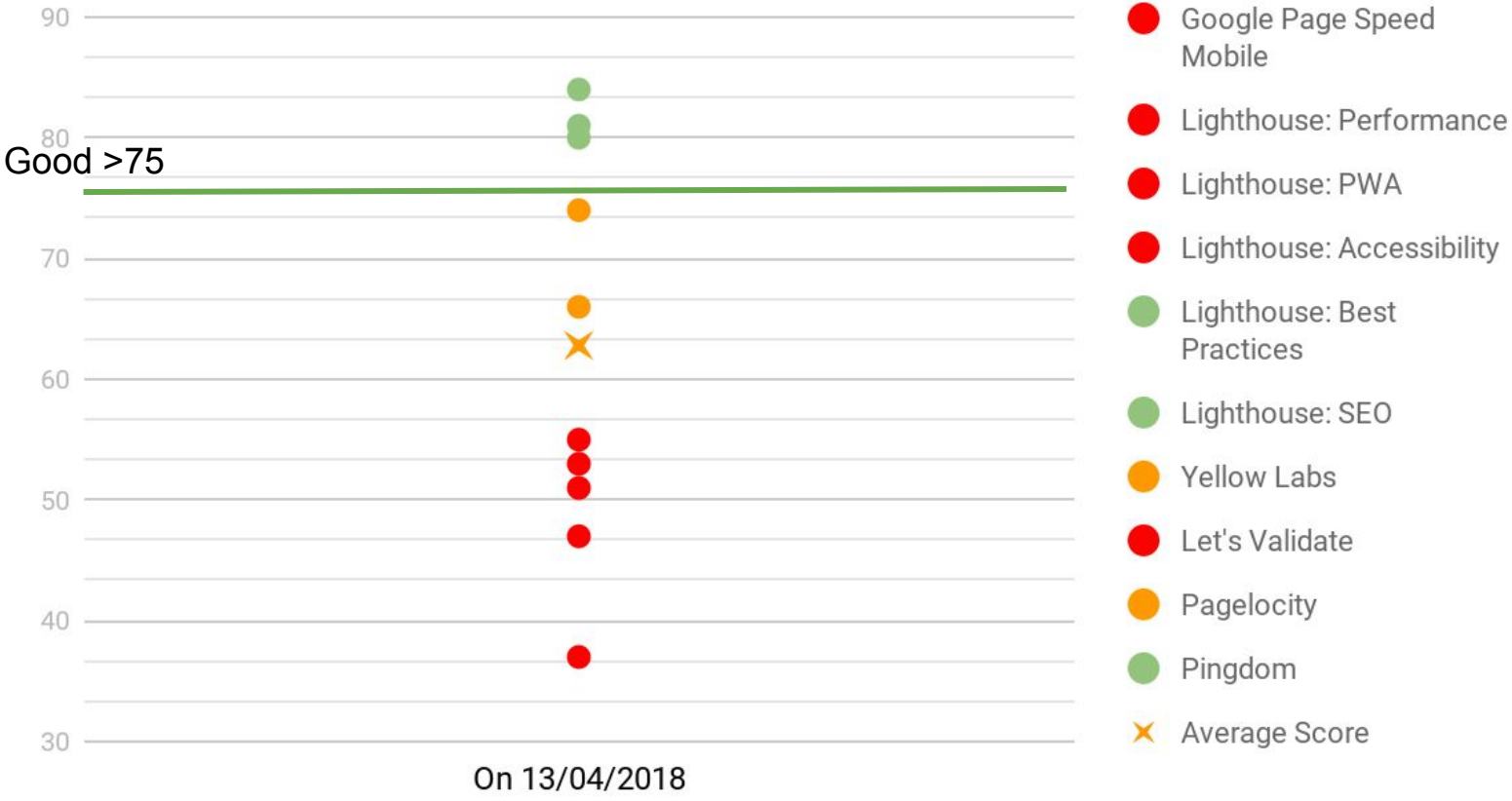
Ideally we want scores close to 100/100, under 75 is bad and affects Google Pagerank.

Tool	Score
Google Page Speed	47/100 (mobile)
Lighthouse: Performance	53/100

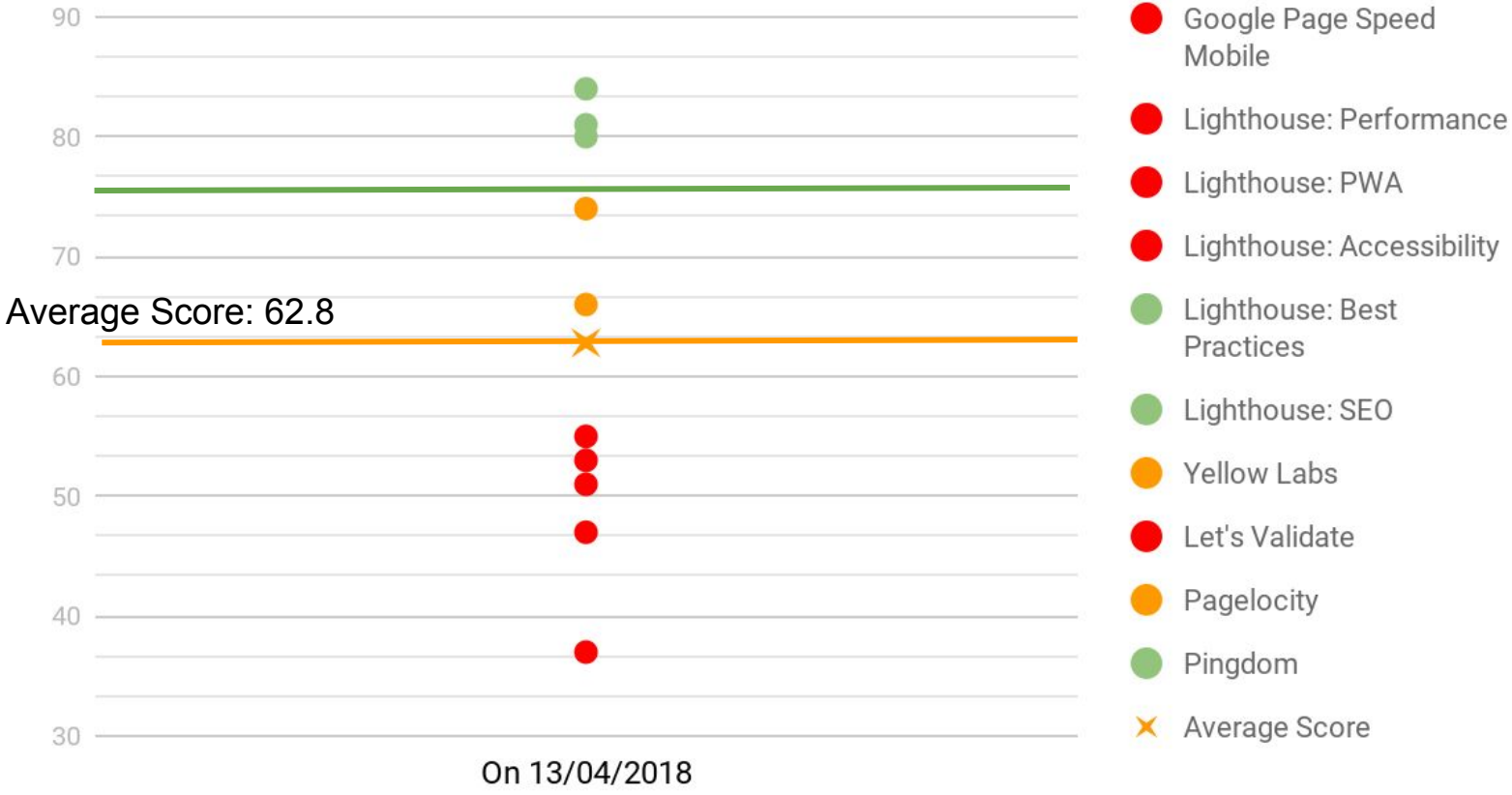
Performance Metrics (over 75 is good)



Performance Metrics (over 75 is good)



Performance Metrics (over 75 is good)



63/100

Current average performance metric for
www.scottlogic.com

What can we do?

What can we do?

Small Improvements

5a

Combine Code

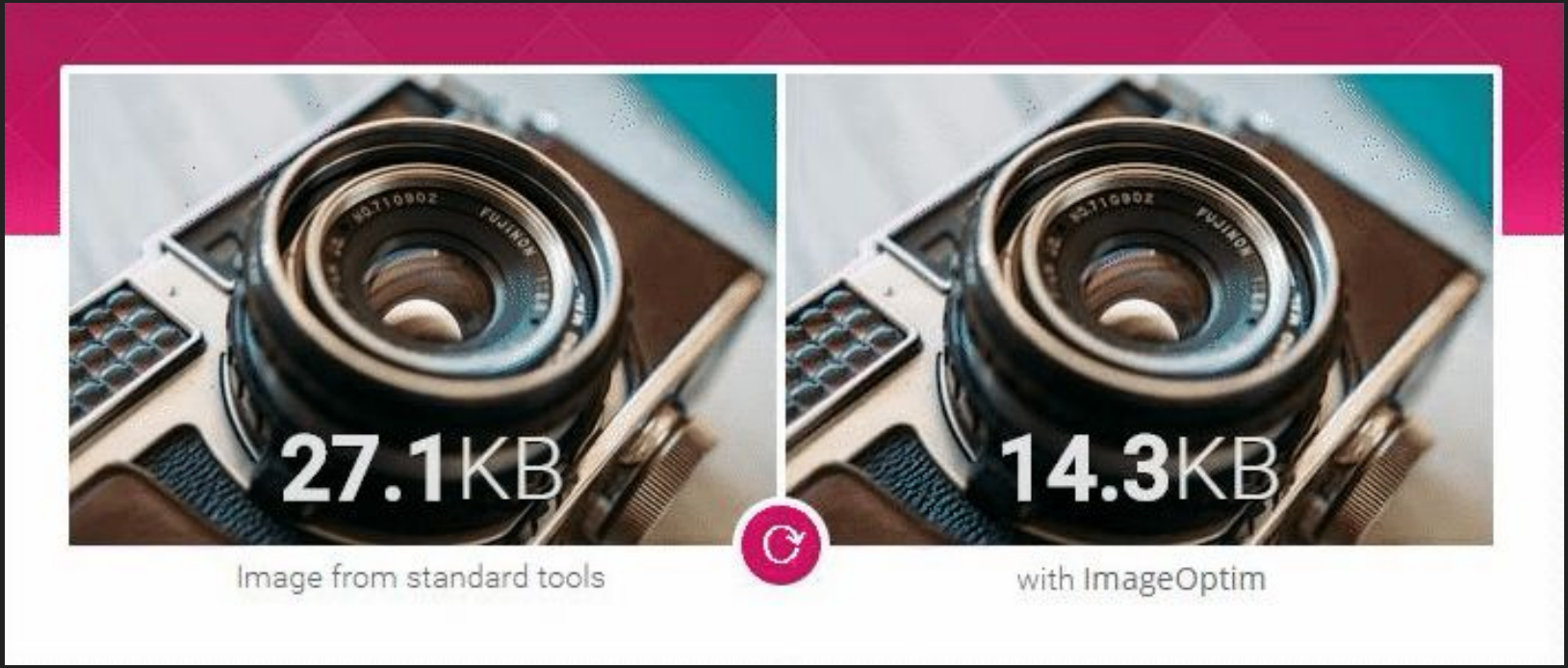
Combine all CSS into a single file

Combine all JS into a single file

Compress

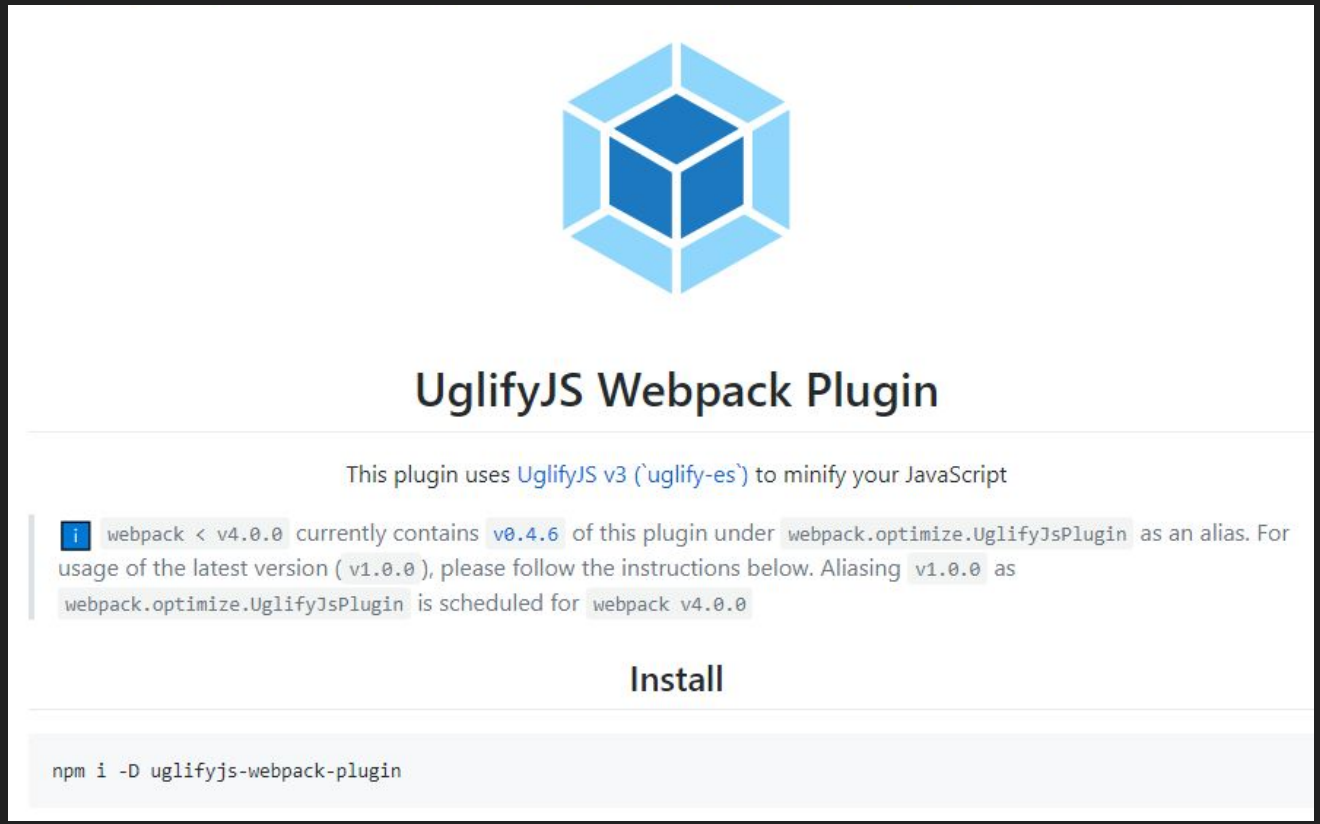
Reduce page weight using these three methods

1. Image optimisation



https://www.npmjs.com/package/imagemin-webpack-plugin

2. File optimisation



UglifyJS Webpack Plugin

This plugin uses UglifyJS v3 (`uglify-es`) to minify your JavaScript

> `webpack < v4.0.0` currently contains `v0.4.6` of this plugin under `webpack.optimize.UglifyJsPlugin` as an alias. For usage of the latest version (`v1.0.0`), please follow the instructions below. Aliasing `v1.0.0` as `webpack.optimize.UglifyJsPlugin` is scheduled for `webpack v4.0.0`

Install

```
npm i -D uglifyjs-webpack-plugin
```

3. GZIP Compression

gzthermal: pseudo thermal view of Gzip/Deflate compression efficiency

Remove Code

Avoid code duplication

Several jQuery loaded

F

0/100

Value: 2

Have 1 or less to get the 100/100 score.

jQuery is a heavy library. You should **never** load jQuery more than once on the same page.

Warning

This rule reached the abnormality threshold, which means there is a real problem you should care about.

2 offenders

v1.12.4

v3.3.1

Avoid Render Blocking

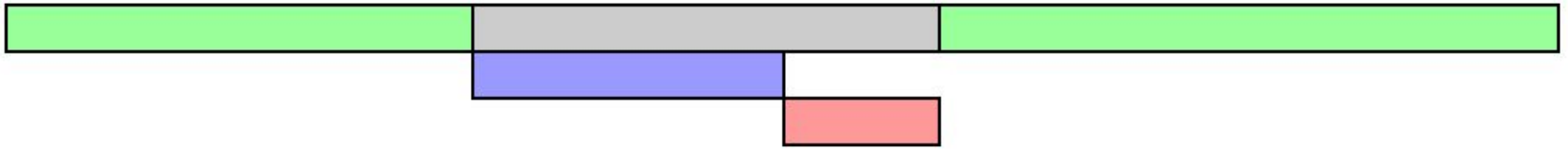
JavaScript files in <head> block rendering

```
<head>
  <meta charset="utf-8">
  <title>JS and CSS preload example</title>

  <meta  ... >


  <link rel="stylesheet" href="style.css">
</head>

<body>
  <h1>bouncing balls</h1>
  <canvas></canvas>

  <script src="main.js"></script>
</body>
```

```
<head>
  <meta charset="utf-8">
  <title>JS and CSS preload example</title>

  <link rel="preload" href="style.css" as="style">
  <link rel="preload" href="main.js" as="script">

  <link rel="stylesheet" href="style.css">
</head>

<body>
  <h1>bouncing balls</h1>
  <canvas></canvas>

  <script src="main.js"></script>
</body>
```

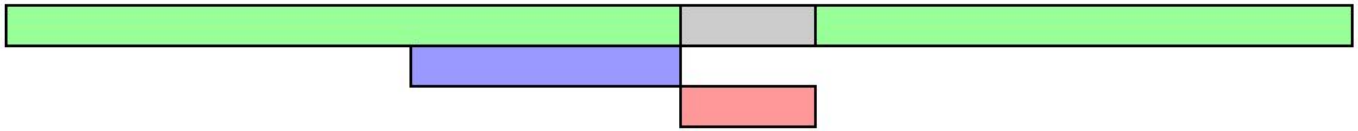
<script>

Let's start by defining what **<script>** without any attributes does. The HTML file will be parsed until the script file is hit, at that point parsing will stop and a request will be made to fetch the file (if it's external). The script will then be executed before parsing is resumed.



<script async>

async downloads the file during HTML parsing and will pause the HTML parser to execute it when it has finished downloading.



<script defer>

defer downloads the file during HTML parsing and will only execute it after the parser has completed. **defer** scripts are also guaranteed to execute in the order that they appear in the document.

Limit Web Fonts

Webfonts are beautiful, but heavy

Add your embed code

Copy the code below, and paste it into the pages on **bac.com** where the fonts and settings for your **abc** kit will be used. Make sure it goes into the `<head>` tag.

DEFAULT @IMPORT ADVANCED

```
<script>
  (function(d) {
    var config = {
      kitId: 'gqz5cam',
      scriptTimeout: 3000,
      async: true
```

The advanced embed code loads asynchronously. It provides better control over rendering, but it requires custom CSS to hide the flash of unstyled text. Learn more .

Continue

What can we do?

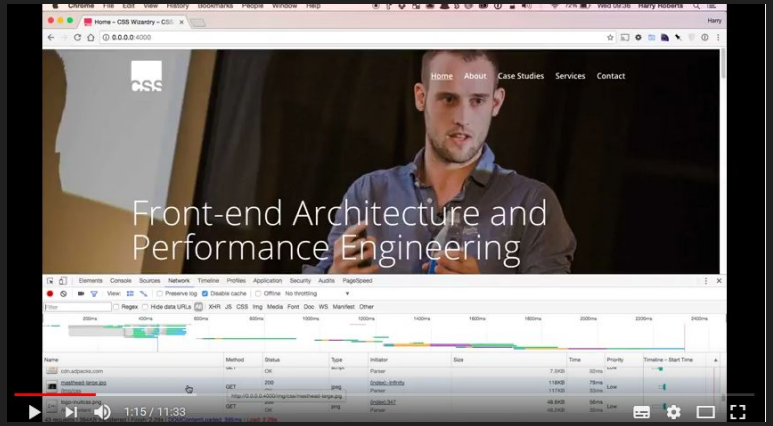
Time Consuming Improvements

5b

Reduce Code

Above-the-Fold Performance

Improve Above-the-Fold performance



https://www.youtube.com/watch?v=sKlA7U5ZKVs

Critical Path

Automatically generate critical CSS and embed it <style> in <head>

Critical extracts & inlines critical-path (above-the-fold) CSS from HTML

Critical

Install

```
$ npm install --save critical
```

Split Code Per Page

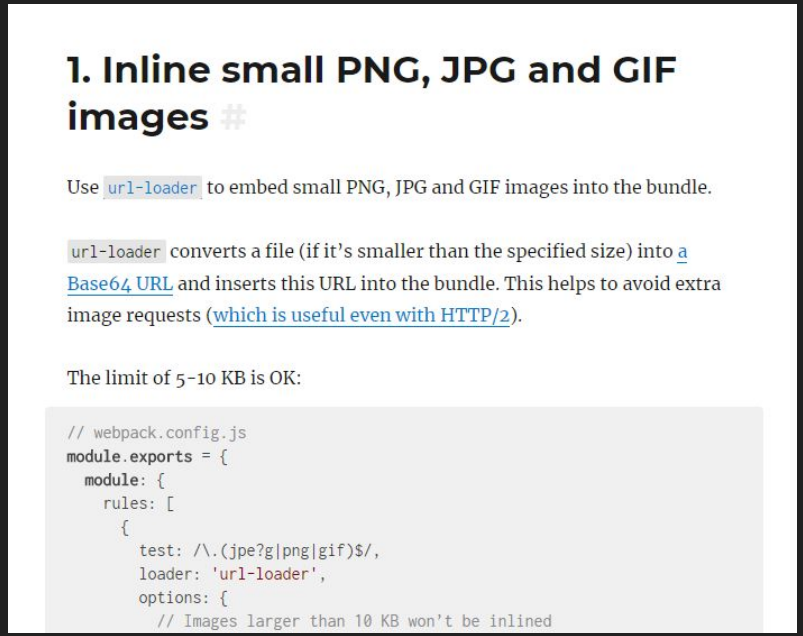
Limit the amount of scripts downloaded to only what is needed by the current page.

https://webpack.js.org/guides/code-splitting/

Lazy Load Offscreen Images

Defer the loading of offscreen images until they're needed.

Inline Small Images



1. Inline small PNG, JPG and GIF images

Use `url-loader` to embed small PNG, JPG and GIF images into the bundle.

`url-loader` converts a file (if it's smaller than the specified size) into a Base64 URL and inserts this URL into the bundle. This helps to avoid extra image requests (which is useful even with HTTP/2).

The limit of 5-10 KB is OK:

```
// webpack.config.js
module.exports = {
  module: {
    rules: [
      {
        test: /\.(jpe?g|png|gif)$/,
        loader: 'url-loader',
        options: {
          // Images larger than 10 KB won't be inlined
```

https://www.npmjs.com/package/url-loader

https://iamakulov.com/notes/optimize-images-webpack/#1-inline-small-png-jpg-and-gif-images

Optimise CSS

Inline, don’t use @import merge or separate link, dont use complex selectors use BEM methodology instead

And there is much, much more!

Summary

1. Slow sites lose business and lose money.
2. Difficult to use load time as a metric, find your own.
3. Gather metrics and add them to your build pipeline.
4. Keep Measuring as you improve.
5. Lots of quick wins and performance improvements.

Questions

Tools used in this presentation

https://github.com/harrymt/web-performance-talk

The First Website I Built

https://www.harrymt.com/blog/2017/11/15/my-first-website.html