yorkshire **twist**
professional software solutions

Building resilient websites

How to make web stuff that doesn't suck

resilience or re·sil·ien·cy

[ri-**zil**-y*uh* ns, -**zil**-ee-*uh* ns *or* ri-**zil**-y*uh* n-see, -**zil**-ee-*uh* n-see] SHOW IPA 🔊

SEE SYNONYMS FOR *resilience* ON THESAURUS.COM

noun

1   the power or ability to return to the original form, position, etc., after being bent, compressed, or stretched; elasticity.

2   ability to recover readily from illness, depression, adversity, or the like; buoyancy.

Website resilience

The ability for a site
to encounter unexpected conditions,
yet not fail the user.

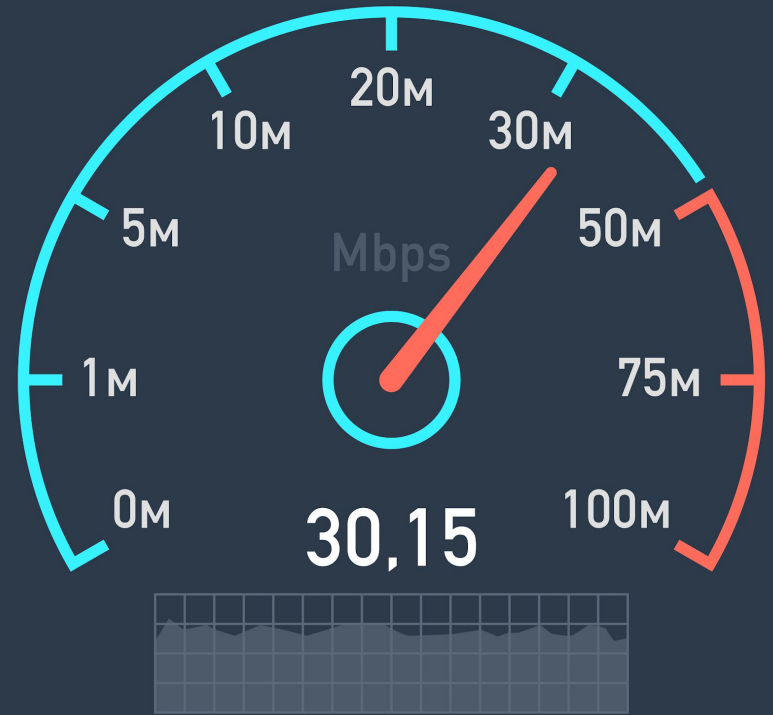# Web development

With great power
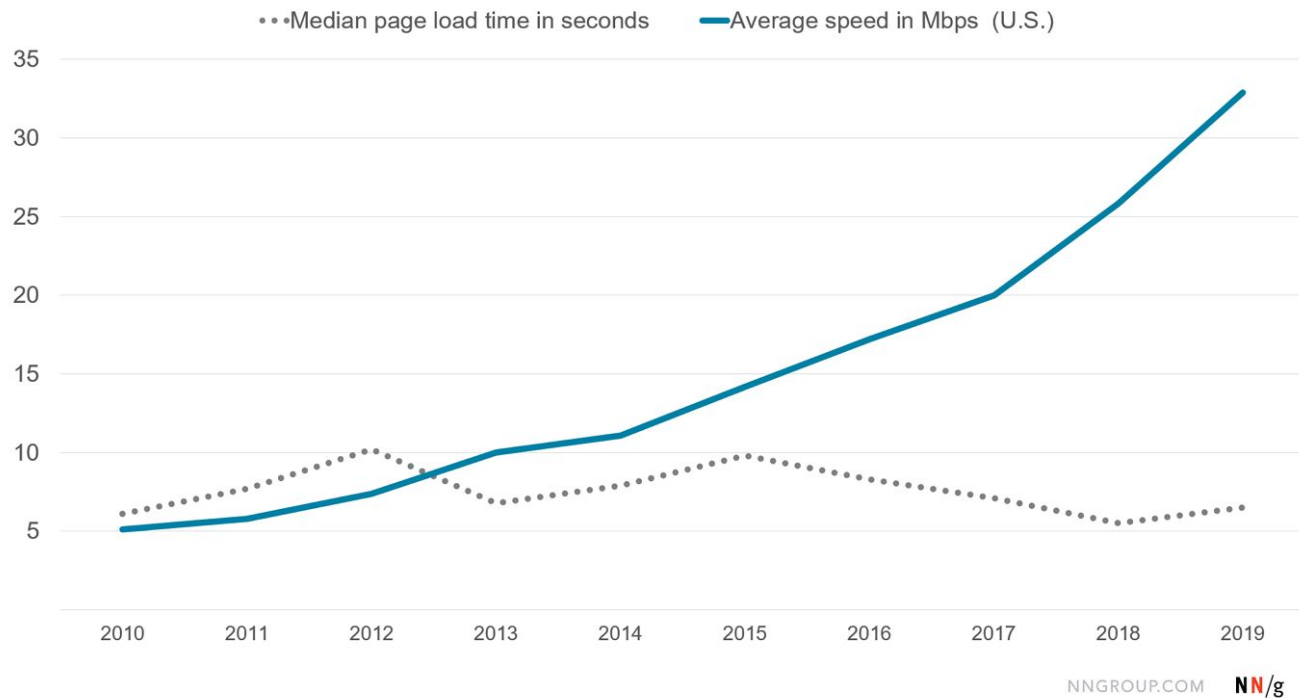comes great responsibility

WEBSITES TESTED **5.8M**

DATA PROCESSED **20.9 TB**

Networks

The need for speed

20м
10м
30м
5м
Mbps
50м
1м
75м
0м
30,15
100м

Change in desktop page load time vs. average connection speed

•••Median page load time in seconds ——Average speed in Mbps (U.S.)

https://www.nngroup.com/articles/the-need-for-speed/

Change in **mobile** page load time vs. average mobile connection speed

•••• Median page load time on mobile devices in seconds — Average mobile speed in Mbps (U.S.)

NNGROUP.COM   N N/g
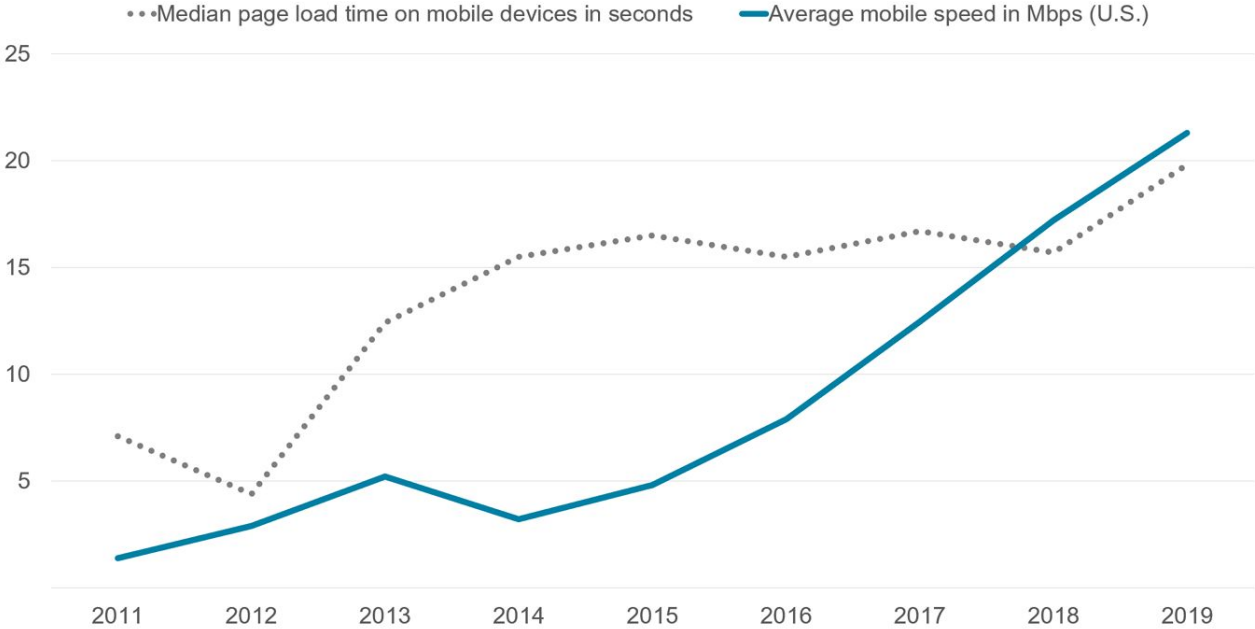
This site sucks ☹️
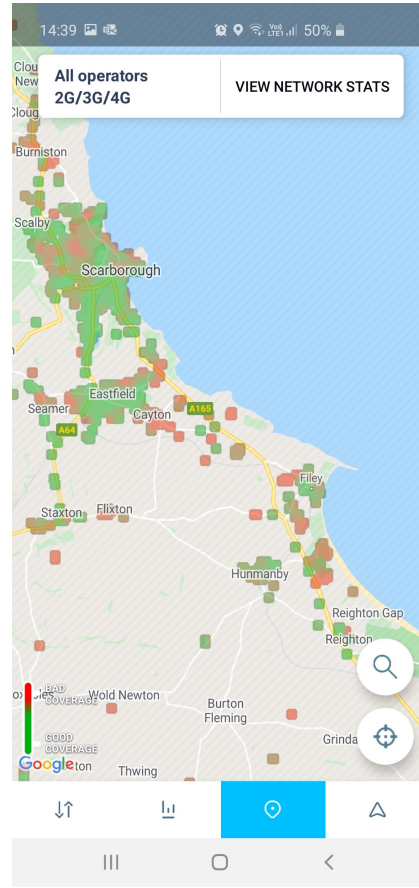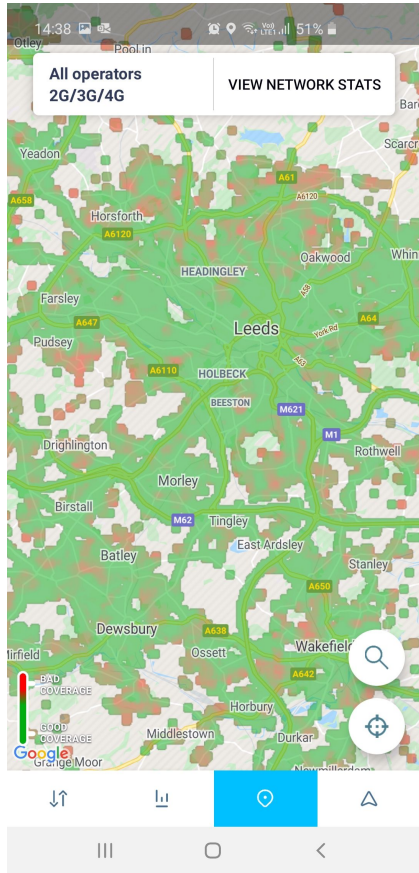
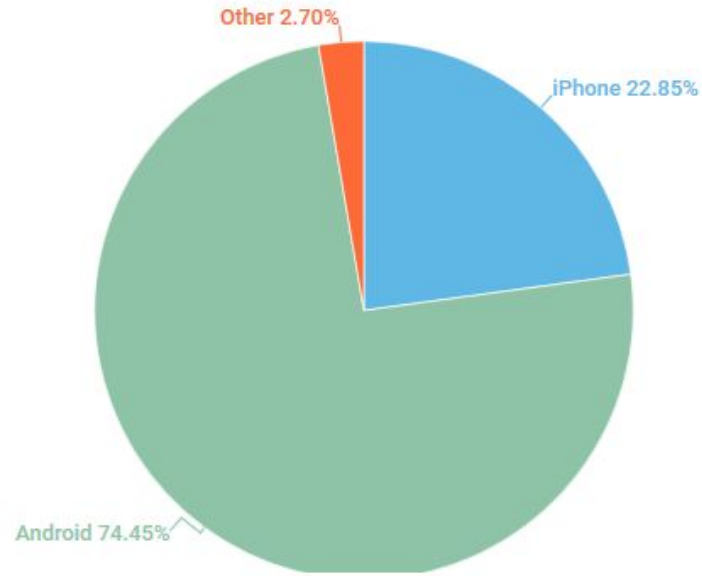Assumptions will naturally be made at some point. But I want to make as few of those assumptions as possible. Because every assumption I make introduces fragility.

Every assumption introduces another way that my site can break.

Devices

You never know what you're gonna get

iPhone vs Android market share
Infogram

| Global | Europe | North America | China |

Other 1.14%

iPhone 27.95%

Android 70.91%
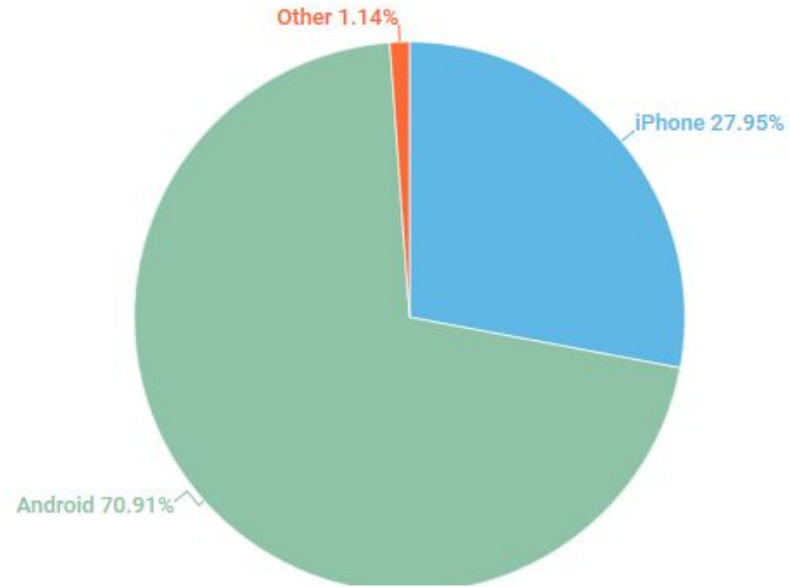
iPhone vs Android market share
Infogram

Geekbench Single-Core Scores

- iOS Single-Core
- Android Single-Core
- Budget Android Single-Core
- Low-End Single-Core

iPhone 11
iPhone XS
iPhone 8
iPhone 7
iPhone 6s
iPhone 6
iPhone 5s
iPhone 5

Galaxy S10+
Galaxy S9+
Galaxy S8+
Galaxy S7
Galaxy S6
Galaxy S4
Galaxy S5
Galaxy S III
Galaxy S II

HTC Evo 4G
Sony Xperia U
Sony Xperia M
Moto E
Moto E5
Moto G4
Moto G4+
Honor X7
Moto G7
Nokia 2

4000
2000

2011 2012 2013 2014 2015 2016 2017 2018 2019
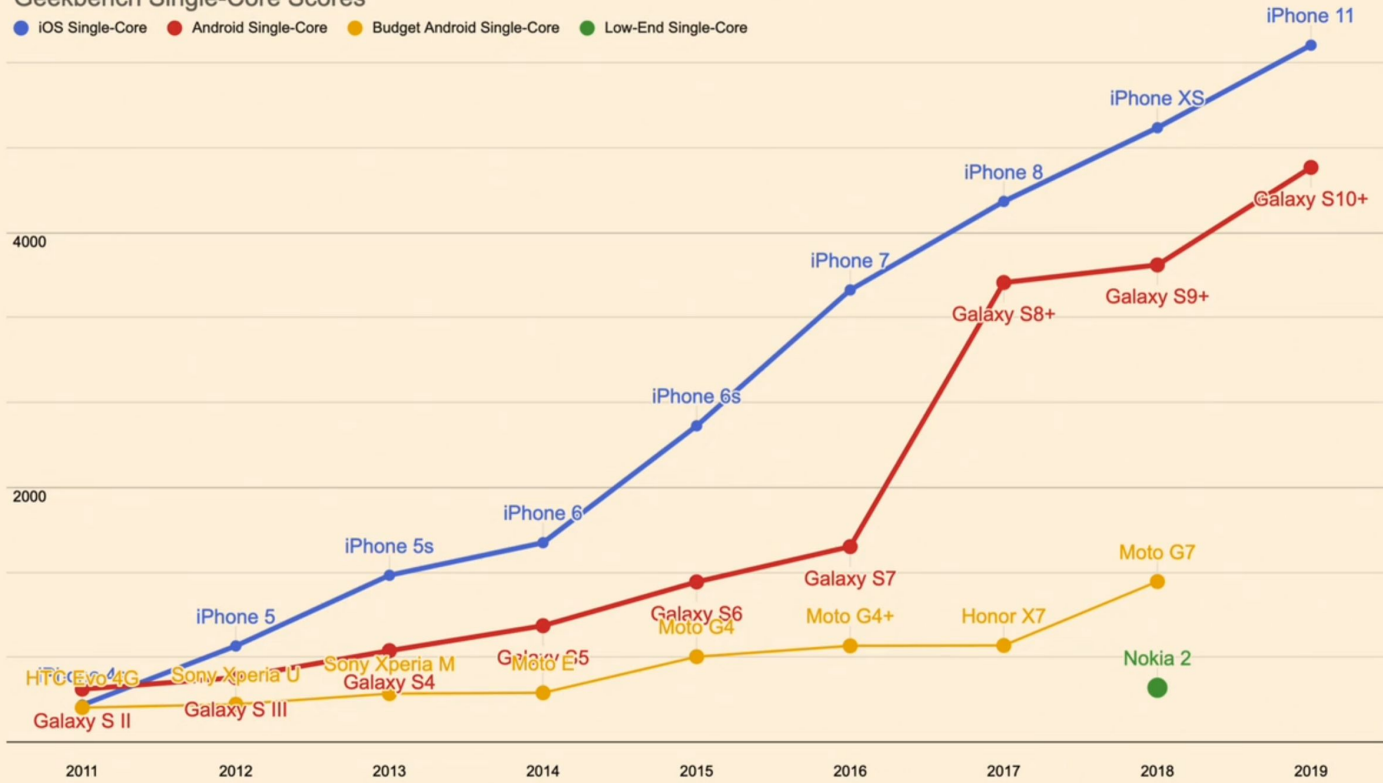
1. Samsung Galaxy A50 (£309)
2. Samsung Galaxy A40 (£219)
3. Samsung Galaxy A20e (£169)
4. Redmi Note 7 (£168)
5. Apple iPhone XR (£629)
6. Samsung Galaxy A10 (£139)
7. Samsung Galaxy S10 (£799)
8. Apple iPhone 8 (£479)
9. Samsung Galaxy A70 (£369)
10. Samsung Galaxy S10+ (£899)

	Samsung Galaxy A50	**Apple iPhone XR**

HARDWARE & PERFORMANCE

	Samsung Galaxy A50	Apple iPhone XR
System chip	Samsung Exynos 7 Octa 9610	Apple A12 Bionic APL1W81
Processor	Octa-core, 2300 MHz, ARM Cortex-A73 and ARM Cortex-A53, 64-bit, 10 nm	Hexa-core, 2490 MHz, Vortex and Tempest, 64-bit, 7 nm
GPU	Mali-G72 MP3	Apple 4-core GPU
RAM	4GB	3GB
Internal storage	128GB	64GB, not expandable
Storage expansion	microSDXC up to 512 GB	
OS	Android (10, 9.0 Pie), Samsung One UI	iOS (13.x, 12.x)

Samsung Galaxy A50 Benchmarks

Benchmark results for the Samsung Galaxy A50 can be found below. The data on this chart is gathered from user-submitted Geekbench 5 results from the Geekbench Browser.

Geekbench 5 scores are calibrated against a baseline score of 1000 (which is the score of an Intel Core i3-8100). Higher scores are better, with double the score indicating double the performance.

CPU Benchmark Scores	
333	**1177**
Single-Core Score	Multi-Core Score

iPhone XR Benchmarks

Benchmark results for the iPhone XR can be found below. The data on this chart is gathered from user-submitted Geekbench 5 results from the Geekbench Browser.

Geekbench 5 scores are calibrated against a baseline score of 1000 (which is the score of an Intel Core i3-8100). Higher scores are better, with double the score indicating double the performance.

CPU Benchmark Scores	
1107	**2571**
Single-Core Score	Multi-Core Score

Geekbench Single-Core Scores

● iOS Single-Core   ● Android Single-Core   ● Budget Android Single-Core   ● Low-End Single-Core
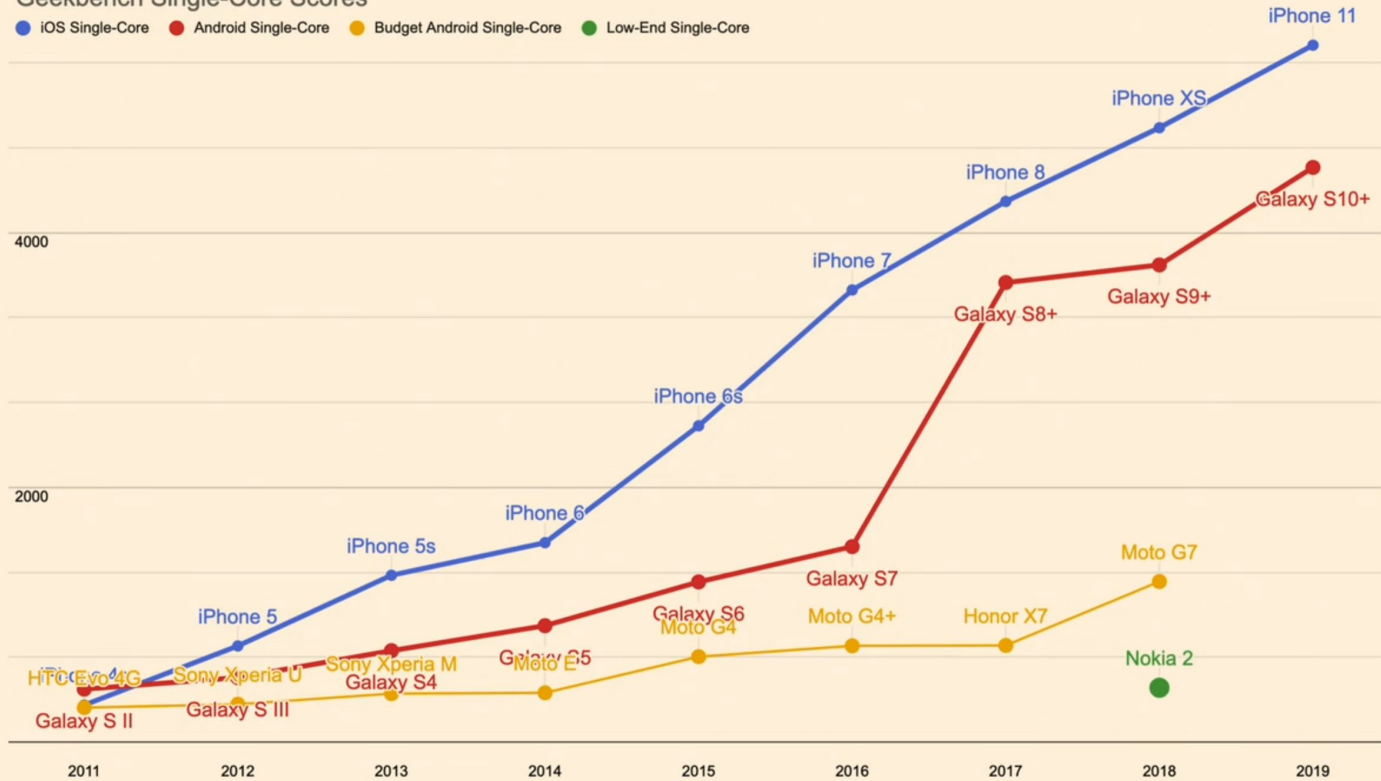
This site sucks 😩

Scripts

JAVASCRIPT ALL THE THINGS!
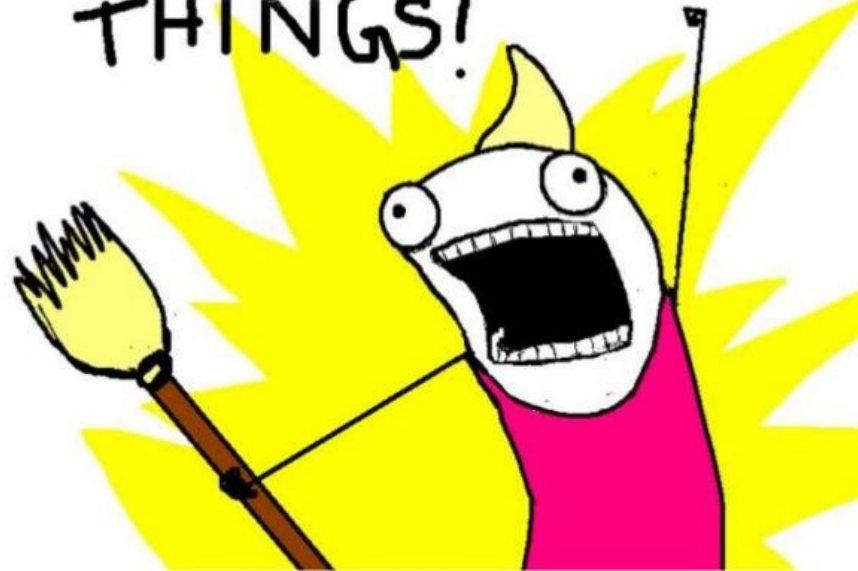
Figure 3. V8 Main thread processing times by device.

Distribution of JavaScript bytes (Mobile)

Web Almanac 2019: JavaScript
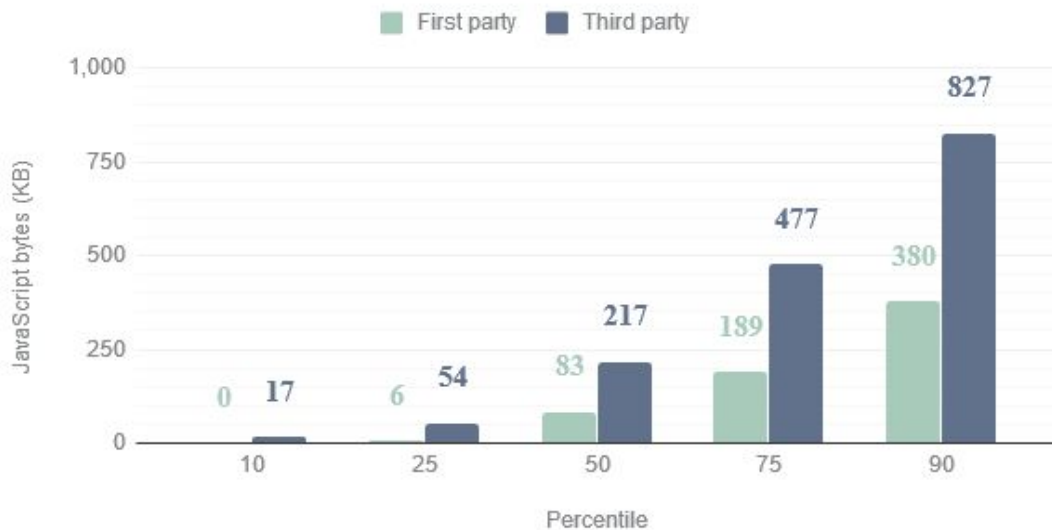
Figure 9. Distribution of total JavaScript downloaded on mobile.

Pinned Tweet

Jerry Jones
@juryjowns

Love like you've never been hurt.
Dance like no one is watching.
Code like JavaScript is going to fail.
#progressiveEnhancement

4:07 PM · Feb 17, 2016 · Twitter for Mac

JavaScript Bytes Served to Mobile Devices, by Percentile

	10TH	25TH	50TH	75TH	90TH
All Sites	93.4kb	196.6kb	413.5kb	746.8kb	1,201.6kb
Sites with jQuery	110.3kb	219.8kb	430.4kb	748.6kb	1,162.3kb
Sites with Vue.js	244.7kb	409.3kb	692.1kb	1,065.5kb	1,570.7kb
Sites with Angular	445.1kb	675.6kb	1,066.4kb	1,761.5kb	2,893.2kb
Sites with React	345.8kb	441.6kb	690.3kb	1,238.5kb	1,893.6kb
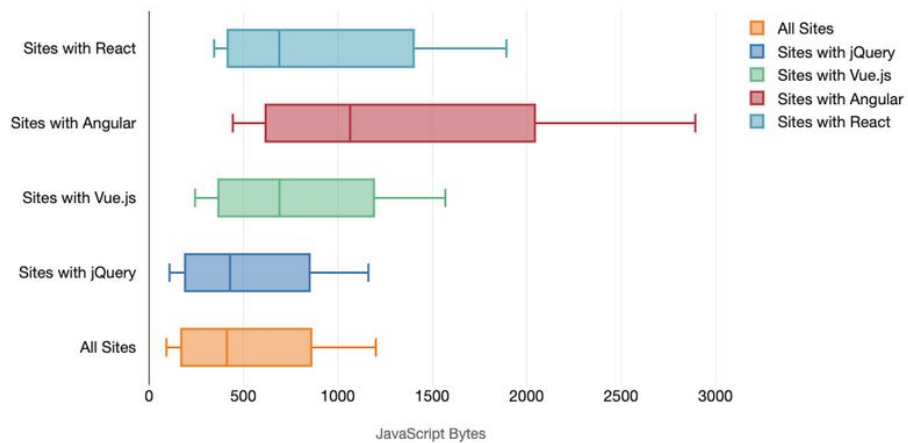
Scripting related CPU time (in milliseconds) for mobile devices, in percentiles

	10TH	25TH	50TH	75TH	90TH
All Sites	356.4ms	959.7ms	2,372.1ms	5,367.3ms	10,485.8ms
Sites with jQuery	575.3ms	1,147.4ms	2,555.9ms	5,511.0ms	10,349.4ms
Sites with Vue.js	1,130.0ms	2,087.9ms	4,100.4ms	7,676.1ms	12,849.4ms
Sites with Angular	1,471.3ms	2,380.1ms	4,118.6ms	7,450.8ms	13,296.4ms
Sites with React	2,700.1ms	5,090.3ms	9,287.6ms	14,509.6ms	20,813.3ms

JavaScript CPU Time

https://timkadlec.com/remembers/2020-04-21-the-cost-of-javascript-frameworks/
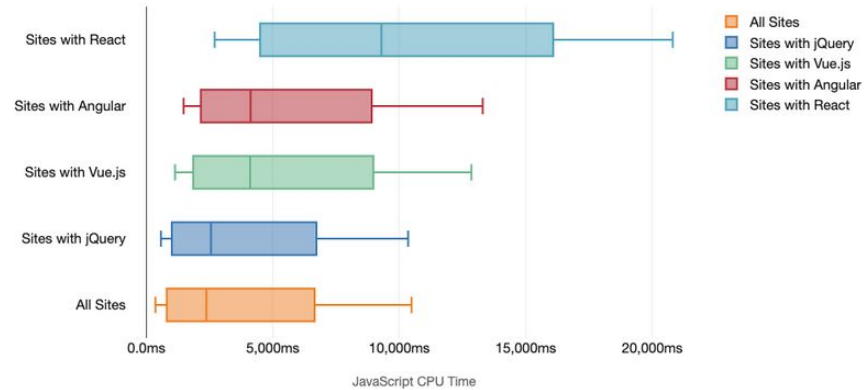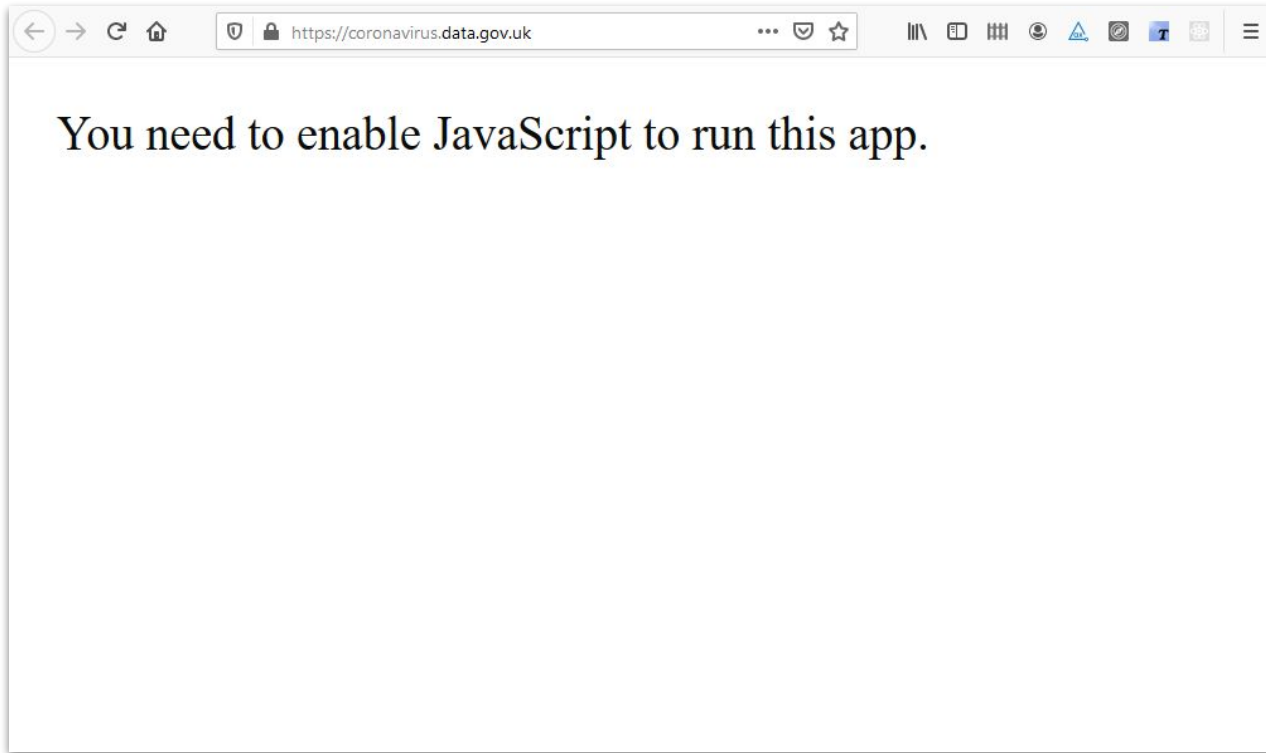
We need JavaScript to show this mostly static content which is updated about once per day.

This site sucks 😡

Assets

They should be called "liabilities"

Becoming a jedi

Be patient you must...

[Use the force] [Join the dark side]

Percentile	Total (KB)	HTML (KB)	JS (KB)	CSS (KB)	Image (KB)	Document (KB)
90	6226	107	1060	234	4746	49
75	3431	56	668	122	2270	25
50	1745	26	360	56	893	13
25	800	11	164	22	266	7
10	318	6	65	5	59	4

Figure 3. Page weight on mobile broken down by resource type.

25% of web pages are larger than the complete works of Shakespeare

25% of web pages are larger than the complete works of Shakespeare

Percentile	Total (KB)	HTML (KB)	JS (KB)	CSS (KB)	Image (KB)	Document (KB)
90	+376	-50	+46	+36	+648	+2
75	+304	-7	+34	+21	+281	0
50	+179	-1	+27	+10	+106	0
25	+110	-1	+16	+5	+36	0
10	+72	0	+13	+2	+20	+1

Figure 5. Change in mobile page weight since 2018.

Cognitive load associated with stressful situations

Bar chart values (cognitive load, scale 0 to 1):
- Waiting in line at retail store: ~0.57
- Watching a melodramatic TV show: ~0.61
- Standing at the edge of a virtual cliff: ~0.75
- Watching a horror movie: ~0.8
- Experiencing mobile delays: ~0.82
- Solving a math problem: ~0.82

The level of stress caused by mobile delays was comparable to watching a horror movie
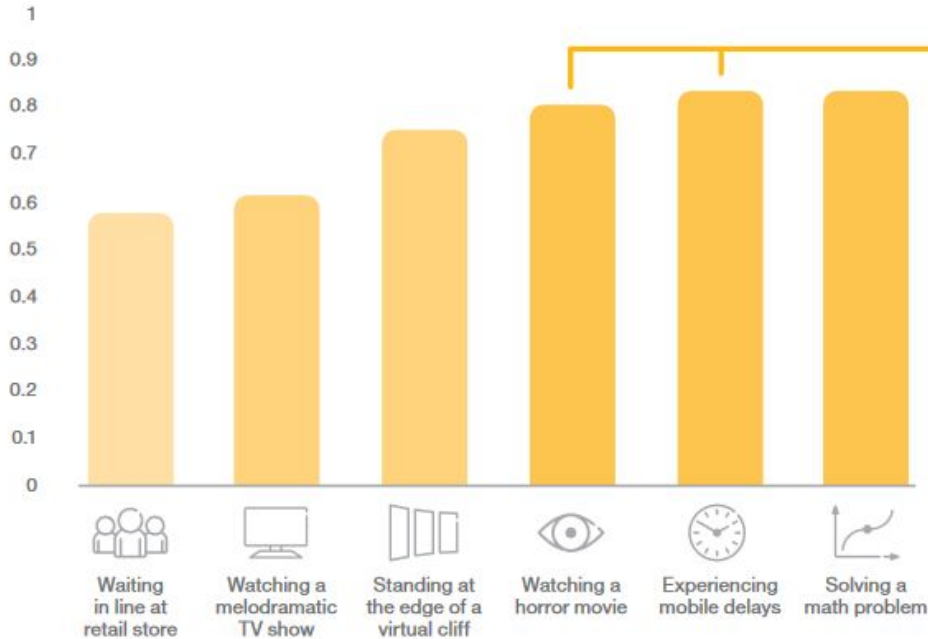
Source: Ericsson ConsumerLab, Neurons Inc., 2015

What might go wrong?

- The server hosting the file may be offline
- The file may be temporarily unreadable
- The URL for the file may be wrong
- DNS settings may be incorrect, meaning the server can't be found
- The CDN may be down
- The file may be empty, or return an incorrect HTTP status code
- The file contains syntax error(s)
- The code may clash with other 3rd party code
- The file may have been modified in transit
- Concatenation/minification may have altered the file badly
- The browser may not fully understand the file (lacks JavaScript APIs, for example)

This site sucks 😒

Percentage of pages using each heading level

Web Almanac 2019: Accessibility
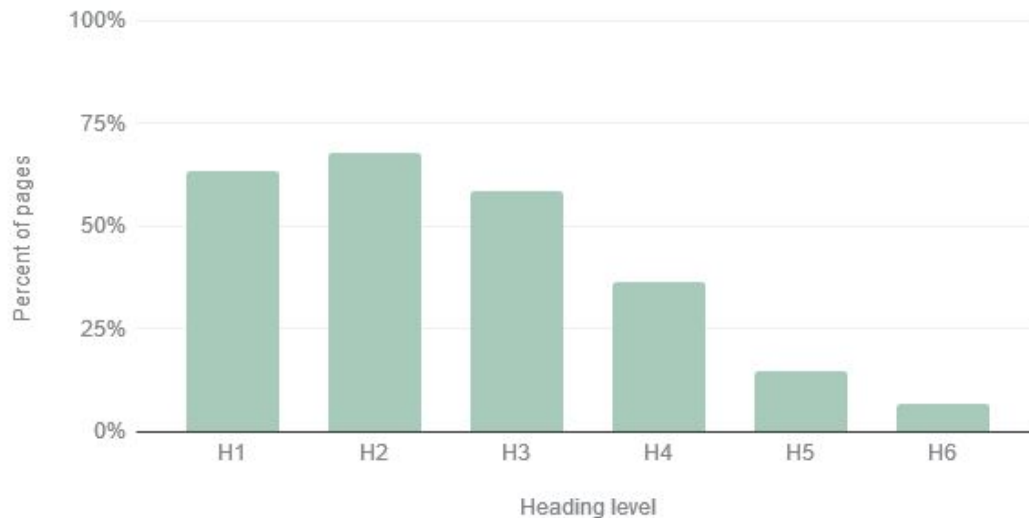
Show description of Figure 3

Figure 3. Popularity of heading levels.

There are 140
HTML elements

The average 2
year-old knows
200-300 words

Even though alt attributes have been around for 25 years, 49.91% of pages still fail to provide alt attributes for some of their images,

and 8.68% of pages never use them at all.

Only 22.04% of sites gave all of their text sufficient color contrast.

Or in other words: 4 out of every 5 sites have text which easily blends into the background, making it unreadable.

https://almanac.httparchive.org/en/2019/accessibility

Equality Act 2010 (EQA)

The EQA imposes a duty on service providers to make "reasonable adjustments" to enable disabled persons to access their services.

This site sucks 😢

Does it matter?

Works On My Machine

Certification of Attainment

Dennis Nedry

For outstanding achievement

11 June 2020

Ancestry.com saw a 7% increase in conversions after improving render time by 68%, page weight by 46% and load time by 64%.

https://www.dropbox.com/s/8rffw1pxoyno5gm/Velocity-NY-Jed-Wood-Ancestry-User-Centered-Metrics.pdf?dl=0

Fashion retailer Missguided removed
BazaarVoice for Android visitors.

Median page load time improved by 4 seconds,
and revenue increased by 26%.

The Trainline reduced latency by 0.3 seconds across their funnel and customers spent an extra £8 million a year.

DoubleClick by Google found 53% of mobile site visits were abandoned if a page took longer than 3 seconds to load.

Amazon sees a 1% decrease in revenue for every 100ms increase in load time.

http://radar.oreilly.com/2008/08/radar-theme-web-ops.html

There are over 11 million people with a limiting long term illness, impairment or disability [in the UK]

More than two million people are estimated to be living with sight loss in the UK today.

This sight loss is severe enough to have a significant impact on their daily lives.

In each of the three years to 2018/19, mobility was the most prevalent impairment reported,

however it has decreased from 7.1 million to 6.8 million people (from 51 per cent to 48 per cent).

Loss of money for business per month

Type of business	Loss of income
High street shop	£267 million
Restaurant / pub / club	£163 million
Supermarket	£501 million
Energy company	£44 million
Phone / internet provider	£49 million
Transport provider	£42 million
Bank or building society	£935 million
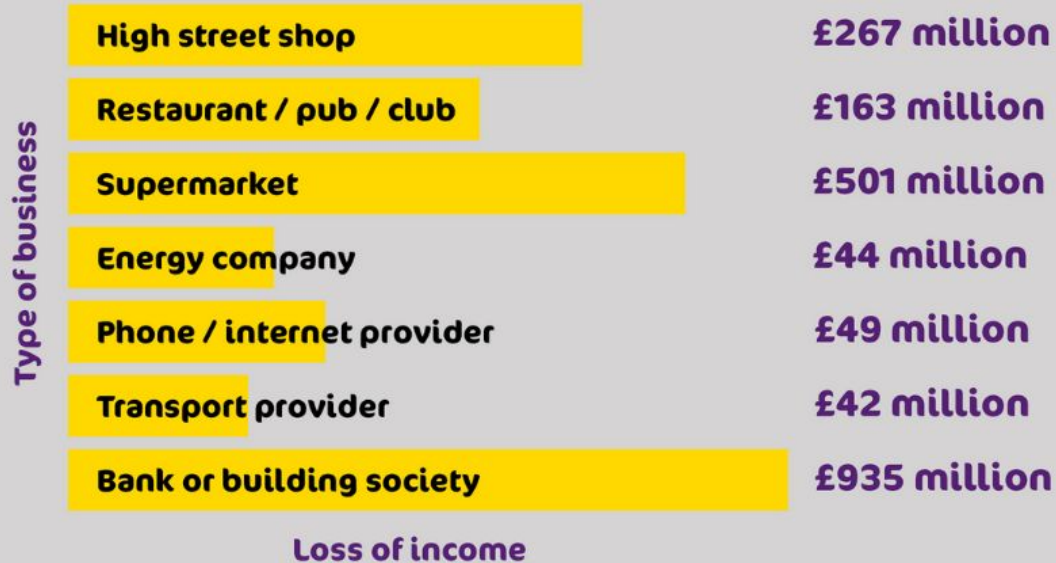
That's a lot of suck.

Can we fix it?

Yes, we can!

Tip 1:

Learn how browsers work

Chickens go in, pies come out

Microsoft Office Word

Microsoft Office Word cannot open this file because it is an unsupported file type.

OK

GET HTML → Parse HTML → GET CSS → Parse CSS
Parse HTML → GET JavaScript → Parse JavaScript → Run JavaScript → GET content → Parse content → Full page

Error
Error (empty page)
Error (empty page)
Error (empty page)
Error (empty page)
Error (empty page)
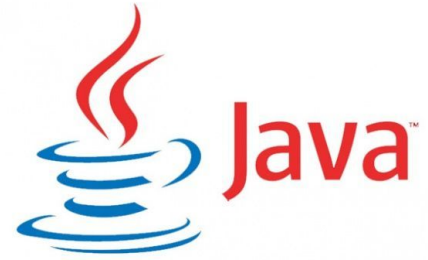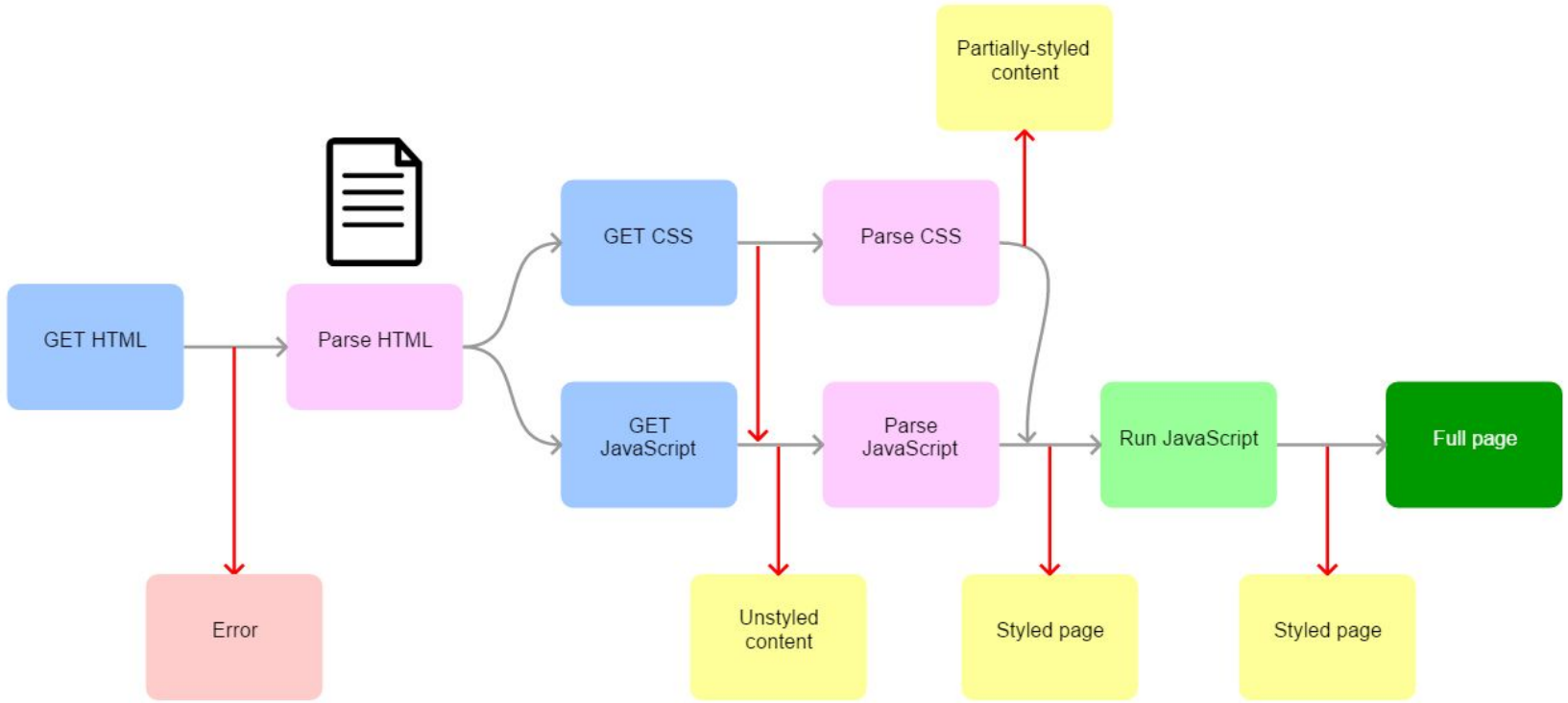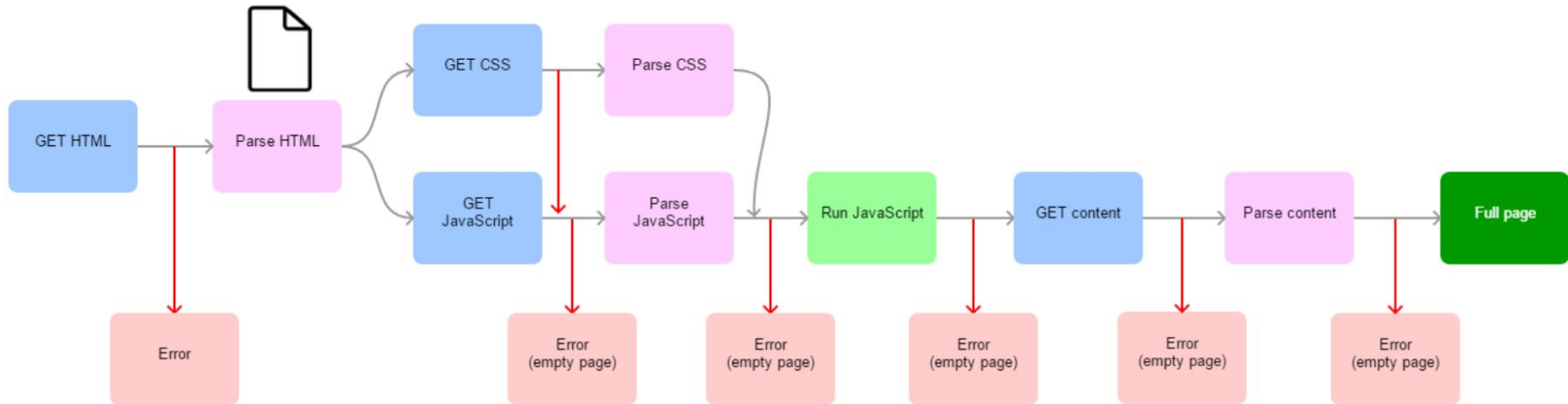
In traditional software development,
we have some say in the execution environment.

On the Web, we don't.
On the Web … all bets are off.

Tip 2:

Learn and use semantic HTML

Time flies like an arrow; fruit flies like a banana.

Groucho Marx

❌

```html
<span class="button"
onclick="magic()">I'm a button,
honest!</span>


<div goto="page.html">Yo bro, I'm a
link innit</div>


<div style="font-size:mega;
font-weight:boldiest">I AM THE MAIN
TITLE, BOW BEFORE ME</div>
```

✅

```html
<button>I'm actually a
button</button>


<a href="page.html">I'm a link, the
superpower of the web</a>


<h1>I think you'll find I am the
main title of this page</h1>
```

Developer who wrote vanilla HTML+CSS in Notepad declared a witch

SAM SYDEFFEKT
21 MAY 2020 · 1 MIN READ

https://www.theolognion.com/developer-who-wrote-vanilla-html-css-in-notepad-declared-a-witch/

Headings

- Groups heading 1
  - The ▨▨, ▨▨ ▨▨▨ ▨▨▨ discussion group heading 2
  - Unread announcements 4 See All heading 2
    - ▨▨▨▨▨ heading 3
      - Comments heading 4
  - New activity heading 2
    - ▨▨▨▨▨▨▨ heading 3
      - Comments heading 4
    - ▨▨▨▨▨ heading 3
      - Comments heading 4
    - ▨▨▨▨▨▨ heading 3
      - Comments heading 4
    - ▨▨▨ shared a link. heading 3
      - Comments heading 4
    - ▨▨▨ shared a link. heading 3
      - Comments heading 4
    - ▨▨▨ heading 3
      - Comments heading 4
    - ▨▨ shared a link. heading 3
      - Comments heading 4
  - About heading 2
  - Recent Media heading 2

```html
<div class="rqescxv l9j0dhe7 du4w35lb j83agx80 cbu4d94t d2edcug0 rj1gh0hx buofh1pr g5gj957u hpfvmrgz dp1hu0rb" aria-label="Preview of a group" role="main"> flex
  <div class="j83agx80 cbu4d94t buofh1pr dp1hu0rb hpfvmrgz l9j0dhe7 du4w35lb"> flex
    <div class="l9j0dhe7 dp1hu0rb cbu4d94t j83agx80" role="main"> flex
      <div class=""> </div>
      <div class="rqescxv l9j0dhe7 du4w35lb"></div>
      <div class="rqescxv lpgh02oy tkr6xdv7 rek2kq2y" style="top: 60px;"> </div>
      <div class="d2edcug0 cbu4d94t j83agx80 bp9cbjyn"> flex
        <div class="bp9cbjyn j83agx80 cbu4d94t k4urcfbm"> flex
          <div class="rqescxv d2edcug0 tn7ubyq0 discj3wi">
            <div class="rqescxv l9j0dhe7 du4w35lb j83agx80 taijpn5t gs1a9yip kbz25j0m btwxx1t3 sv5sfqaa o22cckgh obtkqiv7 fop5sh7t"> flex
              <div class="rqescxv l9j0dhe7 du4w35lb qmfd67dx gile2uim buofh1pr g5gj957u hpfvmrgz aov4n071 oi9244e8 bi6gxh9e h676nmdw aghb5jc5">
                <div class="sjgh6510"> </div>
              </div>
              <div class="j83agx80 l9j0dhe7 k4urcfbm"> flex
                <div class="hybvsw6c ue3kfks5 pw54ja7n uo3d90p7 l82x9zwi rqescxv ni8dbmo4 stjgntxs l9j0dhe7 k4urcfbm">
                  <div class="ihqw7lf3"> </div>
                  <div class="oijh8qal f9o22wc5 myj7ivm5 ad2k81qe km676qk1"> </div>
                  <div class="b20td4e0">
                    <div class="sjgh6510 l9j0dhe7 k4urcfbm du4w35lb">
                      <div class="du4w35lb l9j0dhe7">
                        <div class="" data-testid="Keycommand_wrapper_feed_story">
                          <div class="" data-testid="Keycommand_wrapper">
                            <div class="lzcic4wl" aria-posinset="1" aria-labelledby="jsc_c_an" role="article">
                              <div class="cbu4d94t j83agx80"> flex
                                <div class="">
                                  <div class="j83agx80 l9j0dhe7 k4urcfbm"> flex
                                    <div class="hybvsw6c ue3kfks5 pw54ja7n uo3d90p7 l82x9zwi rqescxv ni8dbmo4 stjgntxs l9j0dhe7 k4urcfbm sbcfpzgs">
                                      <div>
                                        <div></div>
                                        <div>
                                          <div></div>
                                          <div> </div>
                                        </div>
                                        <div>
                                          <div class="" dir="auto">
                                            <div id="jsc_c_an" class="ecm0bbzt hv4rvrfc e5nlhep0 datiiw0a" data-ad-comet-preview="message" data-ad-preview="message">
                                              <div class="j83agx80 cbu4d94t ew0dbk1b irj2b8pg"> flex
                                                <div class="qzhwtbm6 knvmm38d">
                                                  <span class="o1732d6d 1k7dh3pa d2edcug0 qv66sw1b c1et5uql a8c37x1j muag1w35 ew0dbk1b jq4qc12q a3bd9o3v knj5qynh oo9gr5id hzawkc8m" dir="auto">
                                                    <div class="kvgmc6g5 cxmmr5t8 oygrvhab hcukyx3x c1et5uql ii04i59q"> </div>
                                                    <div class="o9v6fnle cxmmr5t8 oygrvhab hcukyx3x c1et5uql ii04i59q">
                                                      <div dir="auto" style="text-align: start;"> </div>
                                                    </div>
                                                    <div class="o9v6fnle cxmmr5t8 oygrvhab hcukyx3x c1et5uql ii04i59q"> </div>
                                                  </span>
                                                  ::after
                                                </div>
                                              </div>
                                            </div>
                                          </div>
                                        </div>
                                      </div>
                                      <div class="stjgntxs ni8dbmo4 l82x9zwi uo3d90p7 h90sisnu monazrh9" data-vc-ignore-dynamic="1"> </div>
                                    </div>
                                  </div>
                                </div>
                              </div>
                            </div>
                          </div>
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
          <div role="feed"> </div>
          <div class="rqescxv l9j0dhe7 du4w35lb qmfd67dx o387gat7 buofh1pr g5gj957u hpfvmrgz aov4n071 oi9244e8 bi6gxh9e h676nmdw aghb5jc5"> </div>
        </div>
      </div>
    </div>
  </div>
</div>
<div> </div>
```
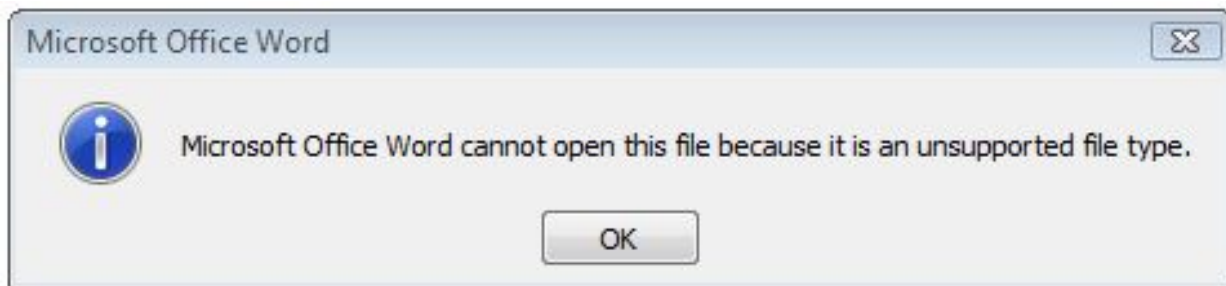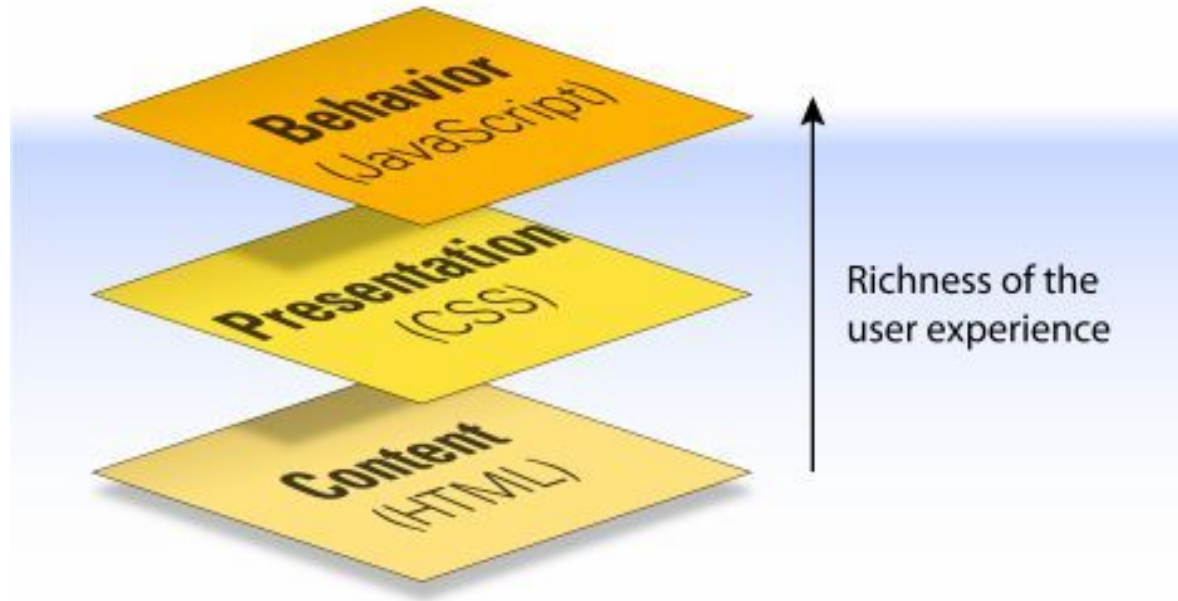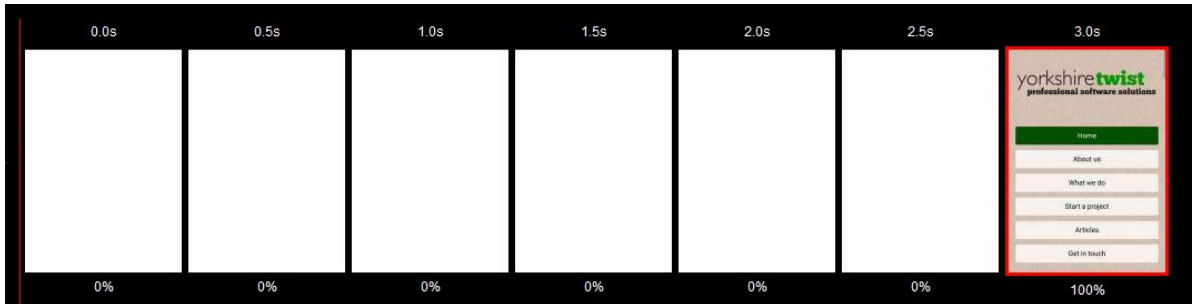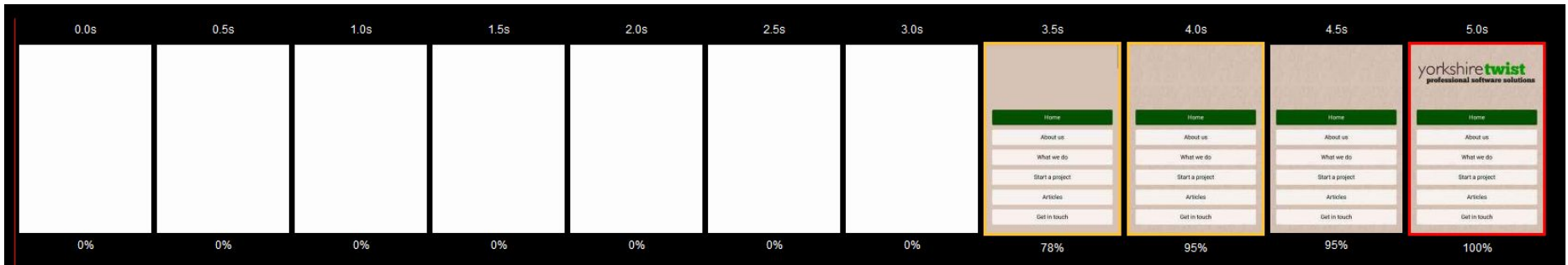
Avoid an excessive DOM size

May 2, 2019 · Updated Oct 4, 2019

Appears in: Performance audits

A large DOM tree can slow down your page performance in multiple ways:

- **Network efficiency and load performance**

- **Runtime performance**

- **Memory performance**

Tip 3:

Make testing tools your best friends

Name six animals which live specifically in the Arctic.

Two polar bears
~~Three~~ Four Seals

scripts 6MiB 55544 count

moz-extension://6...
js::BaseScript

https://www.yout...
js::BaseScript

ht...

https://www.youtub...
js::BaseScript

m...

ht...

ht...

noFilename 1MiB 16634 count
js::jit::JitCode

se...

j...

js::Scope 1MiB 16408 count

Reset camera   Retake snapshot

z-index: 25
z-index: 59
z-index: auto
z-index: 0
z-index: auto
z-index: 35
z-index: 24
z-index: auto
z-index: auto
z-index: auto
z-index: 36
z-index: auto
z-index: 34
z-index: 60
z-index: 43
z-index: 33
z-index: auto
z-index: auto
z-index: 0
z-index: 1000
z-index: 0
z-index: 0
z-index: 0
index: 0
z-index: 0

JS Flame Chart panel:

▽   ≡ Waterfall   ≡ Call Tree   ≣ JS Flame Chart

0 ms   800 ms   1600 ms   2400 ms   3200 ms   4000 ms

max 60 fps
min 1.36 fps

20 ms   2340 ms   2360 ms   2380 ms   2400

https://www.youtube.com/watch?v=UkDg8nUVxLQ
playerBootstrap (watch:43)
HS.create (base.js:6778)
HS (base.js:3372)
Gla (base.js:3407)
MS (...   cT (base.js:3411)
YJ (base.js:2672)
UJ (base.js:2666)   XJ (base...
g.h.tr (base.js:6072)   n...   g...
Bka (base.js:2670)   S...   g...
VJ (base.js:2671)   dr.prototype.then (ba...   Z...   g...
Hja (base....   dr.prototype.then (ba...   Bka/< (base.js:2670)   G...   g...
g.h.yn (ba...   g.h.iM (base.js:6074)   OJ (base.js:2739)   g.h...   g...
Zha (b...   vK (base.js:2696)   g.N.prototype.T (base...   L...   g...
Yga...   g.h.initialize (b...   g.h.T (base.js:5178)   g...   c...
D...   g.N.prot...   UF ...   g.h.kt (base.js:6817)   g...   c...
C...   g.h.T (b...   g.x...   XS (base.js:3401)   Q...   y...
m...   g.h.SG (...   UF/...   g...   g.h.Ow (ba...   g...   x...
z...   cK (ba...   H...   S...   g.h.ud (ba...
K...   g...   JIT   u...   g.h.or...
g...   p...   g.h.lo...
K...   g...
W...
U...
g...
g...

Web Page Performance Test for

https://yorkshiretwist.com/

From: Dulles, VA - **Moto G4** - Chrome - 3G Slow
08/06/2020, 20:45:32

Need help improving?

F	A	A	A	A	B	X
Security score	First Byte Time	Keep-alive Enabled	Compress Transfer	Compress Images	Cache static content	Effective use of CDN

Summary **Details** **Performance Review** **Content Breakdown** **Domains** **Processing Breakdown** **Screenshot** **Image Analysis** **Request Map**

Tester: MotoG4_09-192.168.1.109
Test runs: 3
Re-run the test

View JSON result
Raw page data - Raw object data
Export HTTP Archive (.har)
View Test Log

Performance Results (Median Run - SpeedIndex)

	First Byte	Start Render	First Contentful Paint	Speed Index	Last Painted Hero	Web Vitals			Document Complete			Fully Loaded			
						Largest Contentful Paint	Cumulative Layout Shift	Total Blocking Time	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 3)	2.261s	3.339s	3.331s	3.411s	3.910s	4.399s	0	0.070s	7.551s	9	89 KB	8.093s	10	90 KB	$---
Repeat View (Run 3)	2.286s	2.616s	2.598s	2.628s	2.764s	2.746s	0	0.086s	4.646s	2	3 KB	4.646s	2	3 KB	

Plot Full Results

https://webpagetest.org/result/200608_82_41290f4140d42390ef07f6f6ef898935/

Request Details

URL	/trustboxes/54ad5defc6454f065c28af8b/main.js
Host	widget.trustpilot.com
Response Code	200
Response Time	999ms
Load Time	2340ms
Type	Javascript
Size	47.665kB
Category:	Analytics
Vendor:	Trust Pilot
Product:	Trust Pilot
Timing:	8.31s to 9.651s in page load

(Loading) | (Interactivity) | (Visual Stability)

LCP
Largest Contentful Paint

GOOD | NEEDS IMPROVEMENT | POOR
2.5 sec | 4.0 sec

FID
First Input Delay

GOOD | NEEDS IMPROVEMENT | POOR
100 ms | 300 ms

CLS
Cumulative Layout Shift

GOOD | NEEDS IMPROVEMENT | POOR
0.1 | 0.25
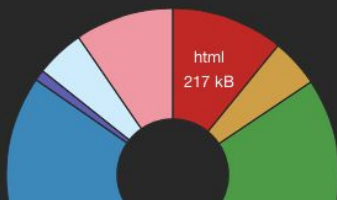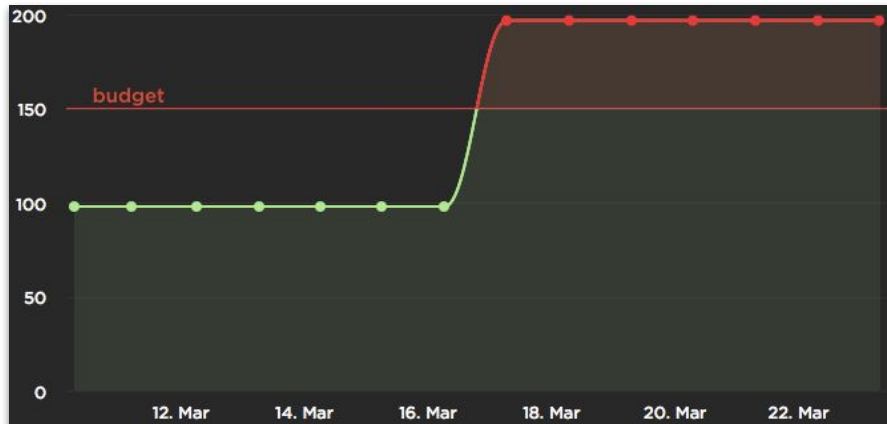
- **Largest Contentful Paint (LCP)**: measures *loading* performance. To provide a good user experience, LCP should occur within **2.5 seconds** of when the page first starts loading.

- **First Input Delay (FID)**: measures *interactivity*. To provide a good user experience, pages should have a FID of less than **100 milliseconds**.

- **Cumulative Layout Shift (CLS)**: measures *visual stability*. To provide a good user experience, pages should maintain a CLS of less than **0.1**.

For each of the above metrics, to ensure you're hitting the recommended target for most of your users, a good threshold to measure is the **75th percentile** of page loads, segmented across mobile and desktop devices.

https://web.dev/vitals/

Welcome to the wonderful world of Web Performance

Sitespeed.io is a set of Open Source tools that makes it easy to monitor and measure the performance of your web site.

Measuring performance shouldn't be hard: you should be able to have full control of your metrics, own your own data and you should be able to do it without paying top dollars.

That's why we created sitespeed.io.

|	Color screen (large breakpoint)	Non-color screen / colorblind palette (medium breakpoint)	Magnified screen / zoom (small breakpoint)	Audio / screen reader (Apple VoiceOver)	Braille terminal
Keyboard (desktop)	Works	Red text isn't visible on green background in protanopia check	Text is jamming together in header	Works	Works
Mouse (desktop)	Works	Same as above	Same as above	Not applicable for this page	Not applicable for this page
Voice / dictation (Apple VoiceOver)	Works	Same as above	Same as above	Works	Works
Touchscreen (Android / Google Chrome)	Can't see hover state on link	Can't see hover state on links	Same as above	Works	Not applicable for this page
Braille terminal	Can't get to hover state on link	Can't get to hover state on link	Can't get to hover state on link	Can't get to hover state on link	Can't get to hover state on link

Tip 4:

Constantly question your assumptions

All is not as it seems, Watson
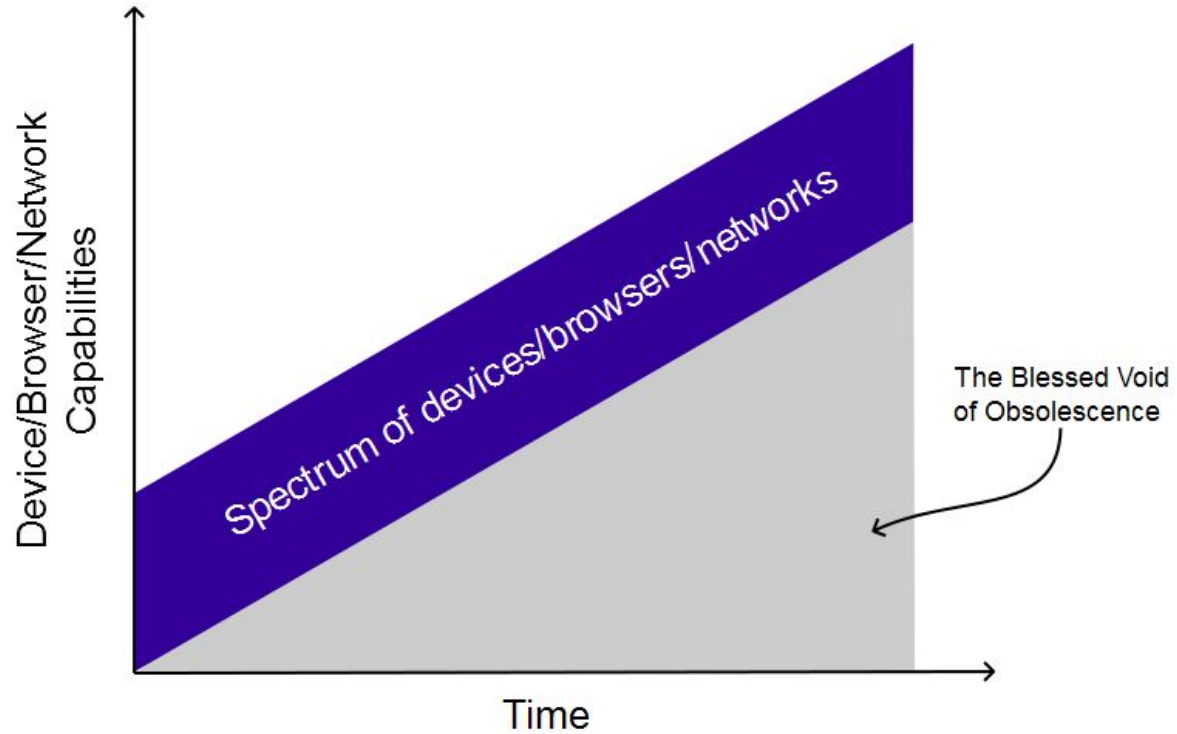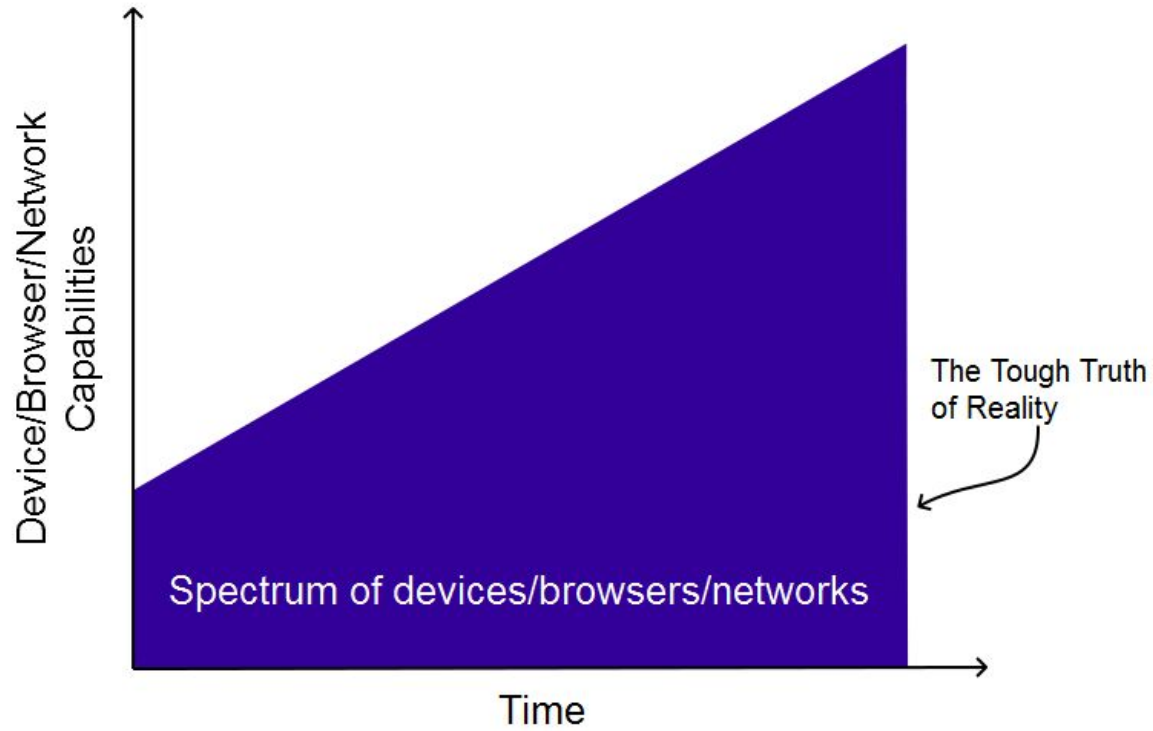
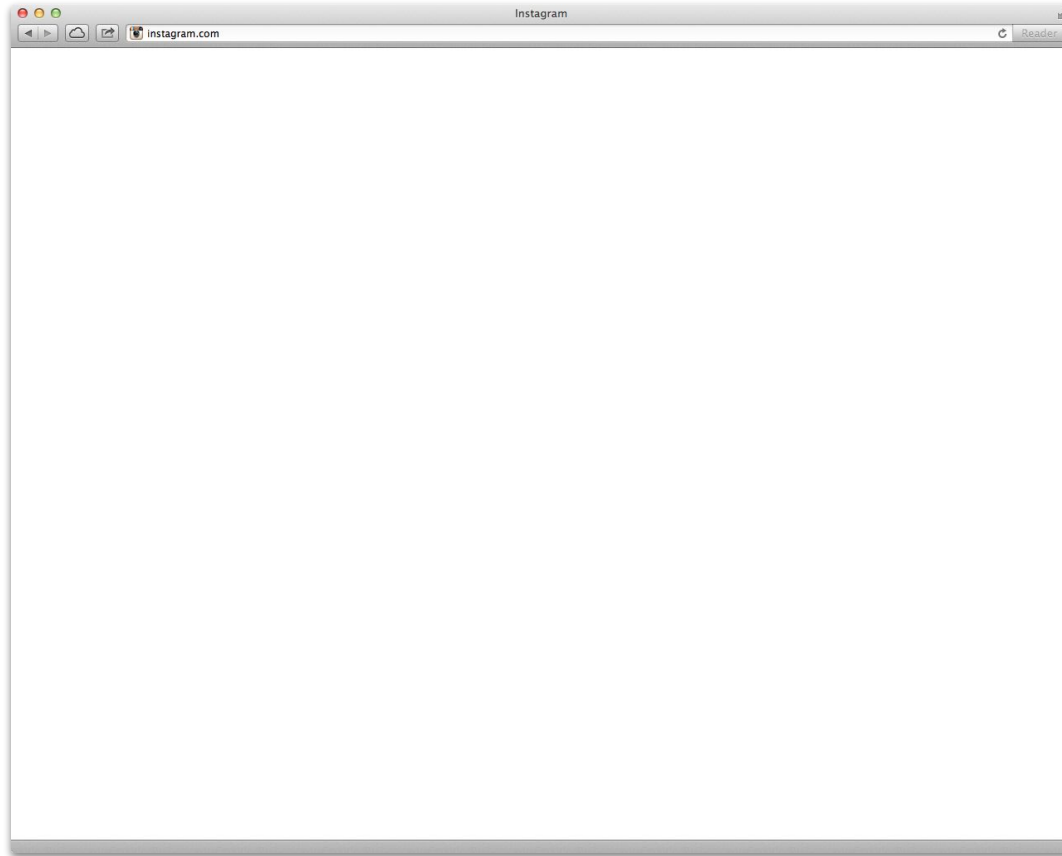# Tip 5:

Do you really need all that JavaScript?

Say when...
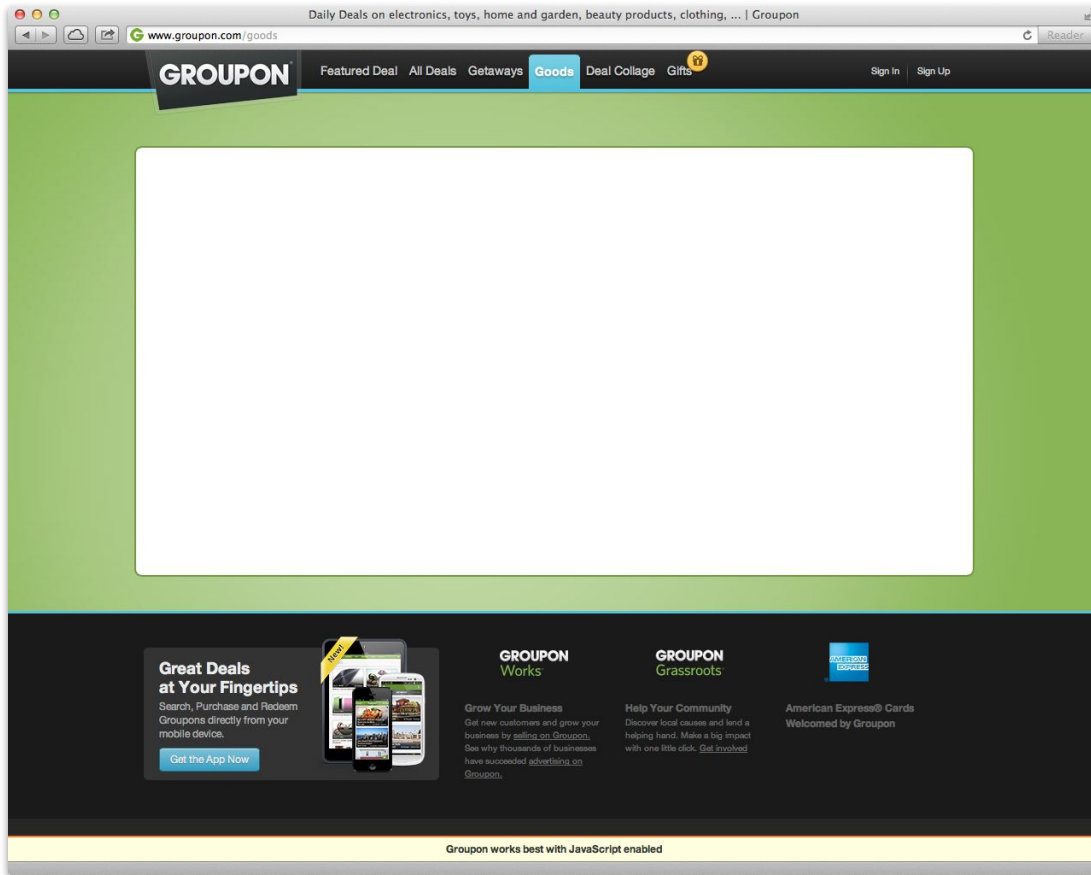
Lots of cool features on the Boston Globe
don't work when JS breaks;

"reading the news" is not one of them.

Dan Abramov
@dan_abramov

This hits the nail on the head. And is 100% matching our long term thinking. Client-side-only is not sustainable. We need to move more stuff to the server, but without sacrificing seamless composition of interactive pieces.

> **Tom MacWright** @tmcw · May 10
>
> okay i mulled over this idea in my head for a month and then just sat down and wrote this in one pass without switching tabs so here it is, raw and extra spicy:
>
> "Second guessing the modern web" macwright.org/2020/05/10/spa...
>
> Show this thread

11:39 PM · May 10, 2020 · Twitter Web App

Dan Abramov Follow ⌄

Working on @reactjs. Co-author of Redux and Create React App. Building tools for humans.

942 Following   68K Followers   ·

The majority of websites aren't, and don't need to be, single-page apps.

The majority of
websites aren't,
and don't need to be,
single-page apps.

https://reactjs.org/docs/add-react-to-a-website.html

Computers and browsers are stupid. They will fail, and they will fail in ways you didn't expect and you cannot reproduce.

Code as if everything will break.

Pre-rendered sites can be enhanced with JavaScript
and the growing capabilities of browsers
and services available via APIs.

Jamstack => JavaScript, APIs and Markup

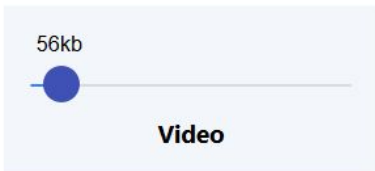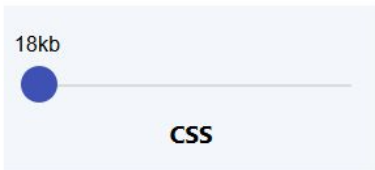# Tip 6:

Create a performance culture

I want my site to load in _____6_____ seconds on a Mobile 3G - Slow (780 Kbps) ▾ connection.

CALCULATE

Your budget:

573kb of 576kb used

Customise your budget

14kb	18kb	95kb
●	●	●
HTML	**CSS**	**JavaScript**
362kb	56kb	28kb
●	●	●
Images	**Video**	**Fonts**

Often, the secret of the most successful digital companies in the world is their obsession with their customers.

This stuff matters

The control which designers know in the print medium, and often desire in the web medium, is simply a function of the limitation of the printed page.

We should embrace the fact that the web doesn't have the same constraints, and design for this flexibility.

The control which *developers* know in the *desktop* medium, and often desire in the web medium, is simply a function of the *delivery mechanism* of *desktop apps*.

We should embrace the fact that the web doesn't have the same *delivery mechanism*, and *develop* for this *reality*.

Progressive enhancement is about
building robust products and
being paranoid about availability.

It is about asking "if" a lot.

Thank you

Let's go fix it!