CONTINUE? Y/N
PROGRESSIVE WEB APPS
THE NEW NORMAL

PRESS START

@addyosmani
1 billion husband points
BET ON THE MOBILE WEB! IT CAN BE FAST, RELIABLE AND ENGAGING.
5B devices connected to web

source: we are social
Progressive Web Apps:
Radically better web experiences

- Reliable
- Fast
- Engaging
PWAs are added to homescreen in one tap

https://www.trivago.co.uk/
Displayed in **app launcher**, and integrated into **Android settings**

https://www.trivago.co.uk/
Lyft Download Size:

- Android App: 17 MB
- iOS App: 75 MB
- PWA: <1 MB
Fast loading on slow networks
Use less data
Works well on smartphones

https://mobile.twitter.com
Fast, smooth scrolling

https://mobile.twitter.com
Access media to tweet

https://mobile.twitter.com
Receive notifications

https://mobile.twitter.com
65% increase pages/session
75% increase in Tweets
1M Daily visits from homescreen icon

https://mobile.twitter.com
PWAs are a cross-browser initiative
53% increase in session length on iOS

https://lancome-usa.com
“By 2020, Progressive Web Apps will have replaced 50% of general-purpose consumer facing apps.”

// Gartner Research

Source: [Gartner post](http://gartner.com/post)
USE A JAVASCRIPT FRAMEWORK THAT GIVES YOU HEADROOM ON MOBILE!
Test on real phones & real networks
There's no substitute.
1. Verizon is killing Tumblr's fight for net neutrality
   78 points by alithebest 1 hour ago | 15 comments

2. Trump administration has plan to scrap 'startup visa' rule
   105 points by storej 4 hours ago | 77 comments

3. Teller – API for your bank account
   353 points by idn_tech_exect 6 hours ago | 163 comments

4. 14 People Make 500K Tons of Steel a Year in Austria
   136 points by ania1ik 17 hours ago | 109 comments

5. Show HN: ShakyDraw – Hand-drawn block diagrams from plain text
   49 points by sankha93 4 hours ago | 17 comments

6. Help EFF Track the Progress of AI and Machine Learning
   34 points by sinak 1 day ago | 10 comments

These audits validate the aspects of a Progressive Web App, as specified by the baseline PWA Checklist.

These metrics encapsulate your app’s performance across a number of dimensions.
The bloat of your **baseline** defines how much headroom you have for **app code**.

How much is taken up by your framework?
Plenty of lightweight options for mobile

Lower total cost on size + parse times from the get-go
Ask HN: For those programming 10+ years, what do you wish you knew 4 years in?

223 points by shubhaker

I've been programming for ~3 years now, first just on the side and now full-time. My learning has been in stops and starts. Now I can feel the cumulative learning paying off. I'm also realizing there's SO much more to learn. What advice do you have for programmers just hitting their stride?

377 comments

Everyone's code basically sucks, including yours. The boundary between a beginner and a coding rockstar really isn't very great unless you are getting into the realm of computer sciences, like machine learning. Learn from your seemingly smarter peers, but don't idolize them. Chances are they will leave the company long before you do, and you will figure out that their coding skills aren't all that they are cracked up to be, especially if their code is poorly documented. You will soon understand that they were focusing on getting the job done, while you were expending all that effort to be clever. In that sense, they were totally right.
A zero configuration tool makes this all easier!
create-react-app now creates a PWA by default
create-react-app

Progressive Web App support

Code-splitting with dynamic import()

Webpack 2

Service Worker for offline caching

Helpful overlay for uncaught errors

Jest 20
Headroom for application code: 1.5s

https://www.webpagetest.org/result/170503_25_5d1c63ea614dc8696e69f1e6d1de2bfb/
Preact
- React Router + React
- Used preact-compat for 2 months
- Mostly a drop-in

- 15% average improvement in TTI
- Median TTI of 5.5 seconds
- 24KB shaved off largest bundle
- 500ms script bootup on 3G
MAKE YOUR CODE, NOT YOUR FRAMEWORK
THE LARGEST PART OF YOUR APP!
100/100 Lighthouse

Automatic code-splitting across routes

Service Worker for offline caching

PRPL

Built in tracking for bundle sizes

Zero-config pre-rendering
Headroom for application code: **3s**

https://www.webpagetest.org/result/170503_01_c3c6734aea4e13427fc2a8dacfc95a3c/
- Built with Vue.js 2.x & Webpack 2 for code-splitting
- Global event bus (but exploring Veux)
- Experimenting with Nuxt.js for component SSR

- Ads and images use AMP-style placeholders & lazy loading
- For larger images use WEBP with source-set for HiDPI displays
- Shipped `<link rel="preload">` support

<table>
<thead>
<tr>
<th>Mobile</th>
<th>Time To Interactive (alpha):</th>
<th>Target: 5,000ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>📲</td>
<td>2199.2ms</td>
<td>2199.2ms</td>
</tr>
<tr>
<td>📲</td>
<td>1696.5ms</td>
<td>1696.5ms</td>
</tr>
</tbody>
</table>

- Service Worker caching Shell, JS, CSS and images with `sw-precache`
USE A PROGRESSIVE FRAMEWORK FOR PROGRESSIVE WEB APPS!
vue init pwa
Progressive Web App support

Code-splitting with dynamic import()

Service Worker for offline caching

vue init pwa

JS chunks preloaded or prefetched

Version hashes for long-term caching

Bundle size analytics

npm run build --report
Headroom for application code: 2s

https://www.webpagetest.org/result/170503_71_046b4347f08805859cb638d48e8831c2/
PROGRESSIVE WEB APPS ARE THE NEW NORMAL