Progressive Enhancement is Dead

LONG LIVE PROGRESSIVE ENHANCEMENT
progressive enhancement
(n.) rendering HTML on the server, then adding behavior using JavaScript
Only you can prevent JAVASCRIPT SHAMING.
Sigh, JavaScript

The best sites, only accessible to some.

Basecamp.

AUG. 27 2013   2 NOTES
Pages that are empty without JS: dead to history (archive.org), unreliable for search results (despite any search engine claims of JS support, check it yourself), and thus ignorable. **No need to waste time reading or responding.**

Because in 10 years nothing you built today that depends on JS for the content will be available, visible, or archived anywhere on the web.

**All your fancy front-end-JS-required frameworks are dead to history, a mere evolutionary blip in web app development practices.** Perhaps they provided interesting ephemeral prototypes, nothing more.
Classic PC Games

Take a step back in time and revisit your favorite DOS and Windows games. The files available in this collection consist primarily of PC demos, freeware, and shareware. These files are the original releases which will require intermediate to
don't take THE WEB for granted
Mike Matas

Great news for the future of honeybees. Plus, the photography in this article is amazing!
Quest for a Superbee

Can the world’s most important pollinators be saved? How scientists and breeders are trying to create a hardier honeybee.

BY CHARLES C. MANN
PHOTOGRAPHS BY ANAND VARMA

News
ubiquity = inconsistency
With #emberjs you'll be developing apps in 5 minutes! Oh sorry did I say 5 minutes? I meant 5 to 10 months.
Using JavaScript has REAL BENEFITS
• Works offline
• No page reloads *(great for e.g. music players)*
• Fast
• Rich interaction
• Access to device features *(camera, storage, GPS)*
Over the next few years almost all of the people who don't yet have a phone will get one, and almost all of the phones on earth will become smartphones.

A decade ago some of that was subject to debate - today it isn't. **What all those people pay for data, and how they charge their phones, may be a challenge**, but the smartphone itself is close to a universal product for humanity - the first the tech industry has ever had.

With billions of people buying a device every two years, on average, the phone business dwarfs the PC business, which has an install base of 1.5-1.6bn devices replaced every 4-5 years. PC sales are a bit over 300m units a year where phone sales are now close to 2bn, of which well over half, and growing, are now smartphones.

That in turn means that the smartphone supply chain is replacing the PC supply chain as a key driver of the tech industry.
A Tale of Two Supercomputers
A Tale of Two Supercomputers
SMARTPHONES will be more pervasive than CONNECTIVITY.
When you have a supercomputer in your pocket, but intermittent connectivity, the ability to work offline is the key to ubiquity
HTML
HTTP!
User Agent
progressive enhancement

JavaScript

technology

HTML + HTTP
+ maybe JS

best for

slow devices, good network

fast devices, bad network
Because internet connections can be flakey or non-existent, you need to consider **offline first**: write your app as if it has no internet connection. Once your app works offline, add whatever network functionality you need for your app to do more when it's online. Read on for tips on implementing your offline-enabled app.
Use local data whenever possible.

When using resources from the internet, use `XMLHttpRequest` to get it, and then save the data locally. You can use the Chrome Storage API, IndexedDB, or Filesystem API to save data locally.

Separate your app’s UI from its data.

Separating the UI and data not only improves your app's design and eases the task of enabling offline usage, but also lets you provide other views of the user's data. An MVC framework can help you keep the UI and data separate.

Assume your app can be closed at any time.

Save application state (both locally and remotely, when possible) so that users can pick up wherever they left off.
In other words
IF
progressive enhancement is about preserving the ubiquity of the web

AND
the age of the slow, incapable device is coming to an end

BUT
connectivity remains imperfect, even in developed countries
We should all be building offline apps with... JavaScript!
BUT!
Progressive Enhancement

- Works if JavaScript fails to load
- Easier to archive and index
- Faster initial load times
MOBILE-FIRST
RESPONSIVE-FIRST
ACCESSIBILITY-FIRST
CONTENT-FIRST
SECURITY-FIRST
OFFLINE-FIRST
DOCUMENTATION-FIRST
API-FIRST
PERFORMANCE-FIRST
EGO DEPLETION
The best way to make someone do something is to make it free
Do Defaults Save Lives?

Eric J. Johnson* and Daniel Goldstein
Do Defaults Save Lives?

Eric J. Johnson* and Daniel Goldstein

<table>
<thead>
<tr>
<th>Country</th>
<th>Effective Consent Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denmark</td>
<td>4.25</td>
</tr>
<tr>
<td>Netherlands</td>
<td>27.5</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>17.17</td>
</tr>
<tr>
<td>Germany</td>
<td>12</td>
</tr>
<tr>
<td>Austria</td>
<td>99.98</td>
</tr>
<tr>
<td>Belgium</td>
<td>98</td>
</tr>
<tr>
<td>France</td>
<td>99.91</td>
</tr>
<tr>
<td>Hungary</td>
<td>99.97</td>
</tr>
<tr>
<td>Poland</td>
<td>99.5</td>
</tr>
<tr>
<td>Portugal</td>
<td>99.64</td>
</tr>
<tr>
<td>Sweden</td>
<td>85.9</td>
</tr>
</tbody>
</table>
FastBoot
Progressive Enhancement for Ember.js
Progressive Enhancement vs. JavaScript Frameworks
I agree:
The web is about **UBIQUITY**
Sometimes we need to **RETHINK** our assumptions.
• Smartphones are eating the world
• Connectivity is the weak link, even in developed countries
• JavaScript frameworks are taking over, for good reasons
Let’s *take advantage* of these trends while preserving the ubiquity of the web
Not try to shame people back to an outdated model
THANK YOU

tomdale