



Building mobile web app

that don't suck

Frédéric Harper

Sr. Technical Evangelist @ Mozilla

@fharper | outofcomfortzone.net

Web Unleashed

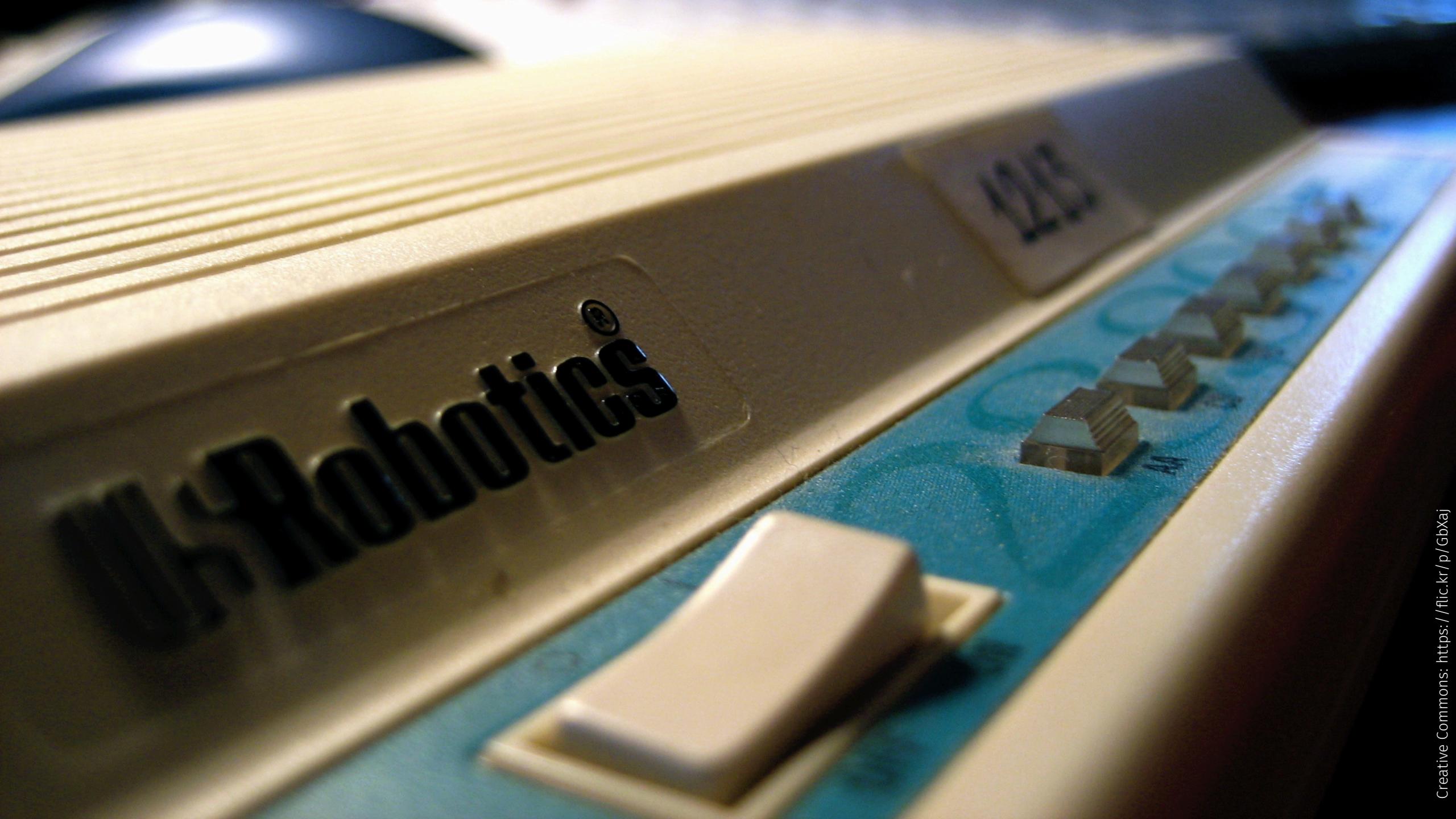
2014-09-18











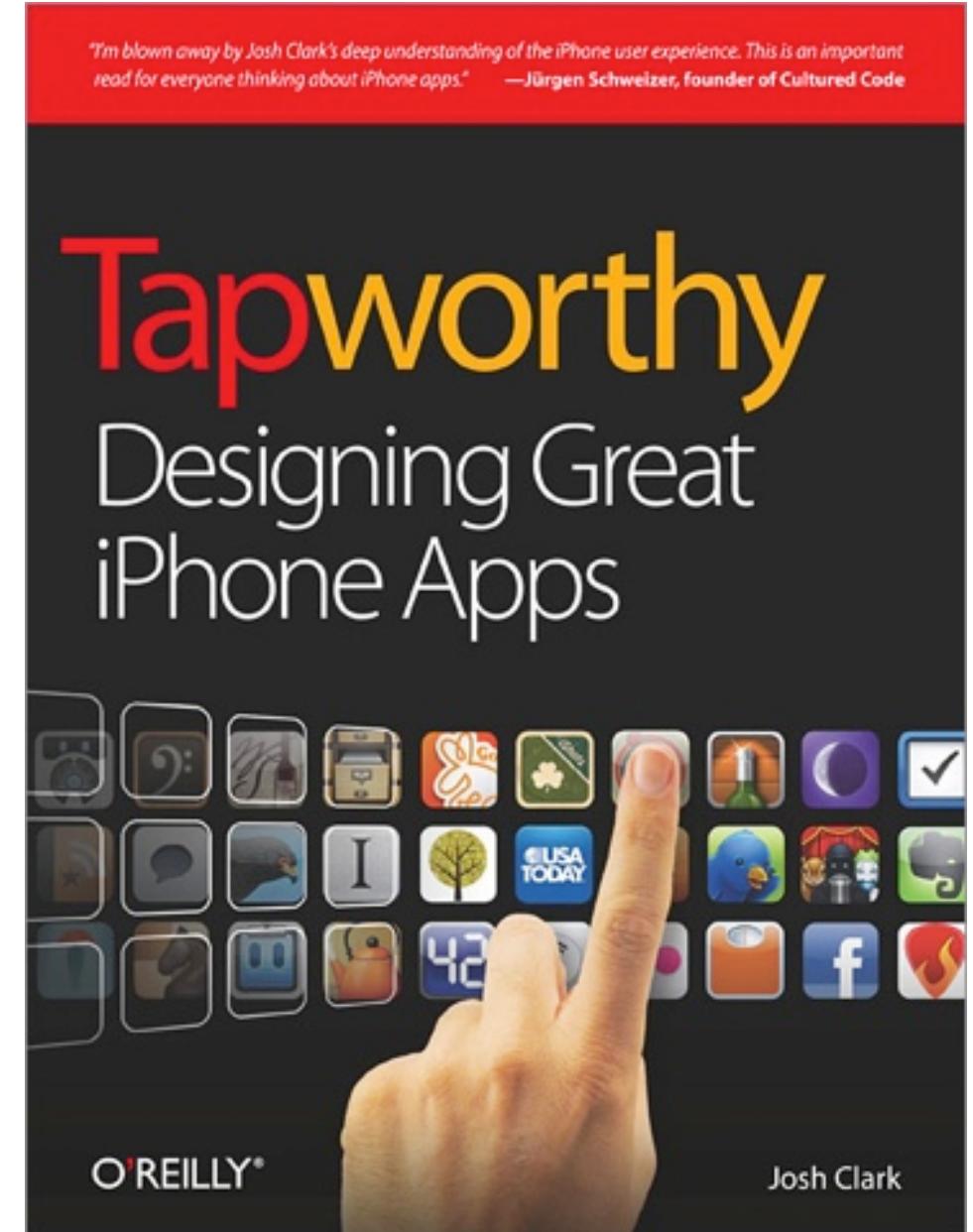
**38 billion devices
connected in 2020**

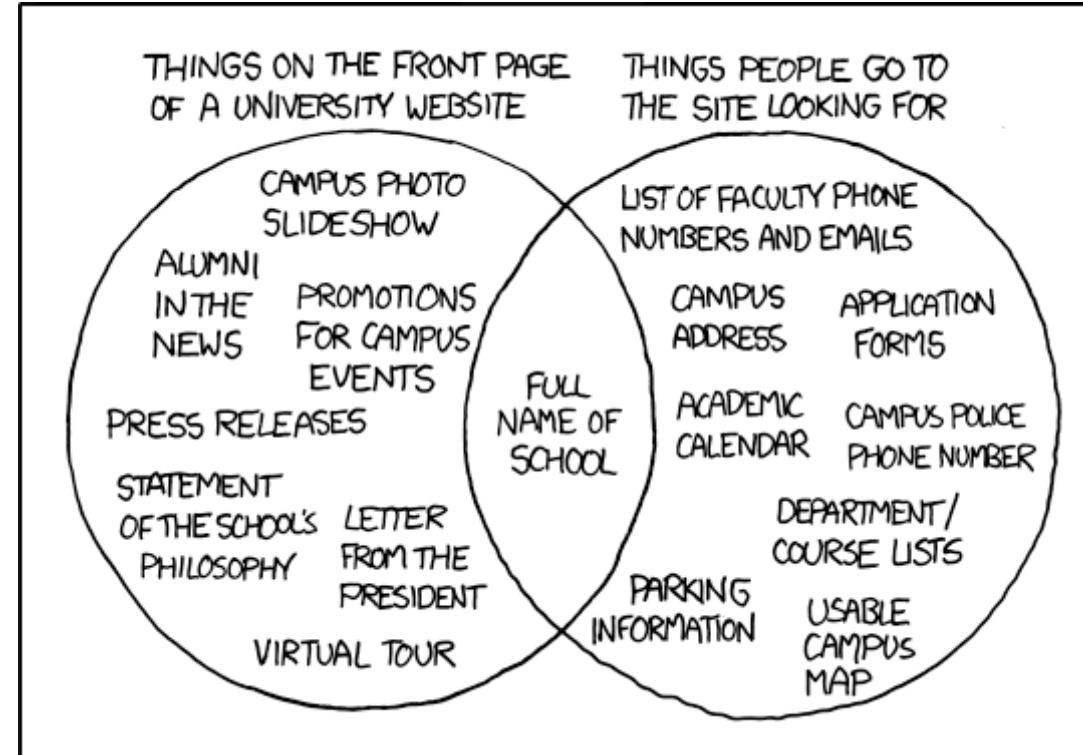
ABI Research - 2013-05-09 - <http://j.mp/38billion>

Think differently

The 3 motivations

1. I'm microtasking
2. I'm local
3. I'm bored





Mobile First



Responsive Web Design



Responsive Web Design

1. A flexible, grid-based layout,
2. Flexible images and media,
3. Media queries.

DEMO

Offline mode

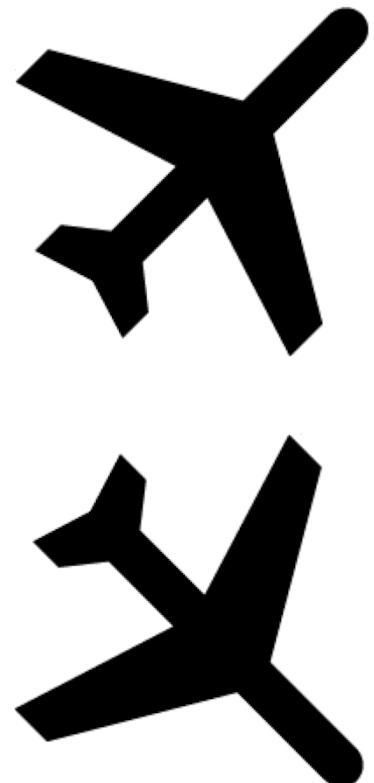
1. Application cache
2. localStorage
3. IndexedDB
4. Native Offline Support (ex.: package app on Firefox OX)

UI & UX are important

Firefox OS



Firefox OS



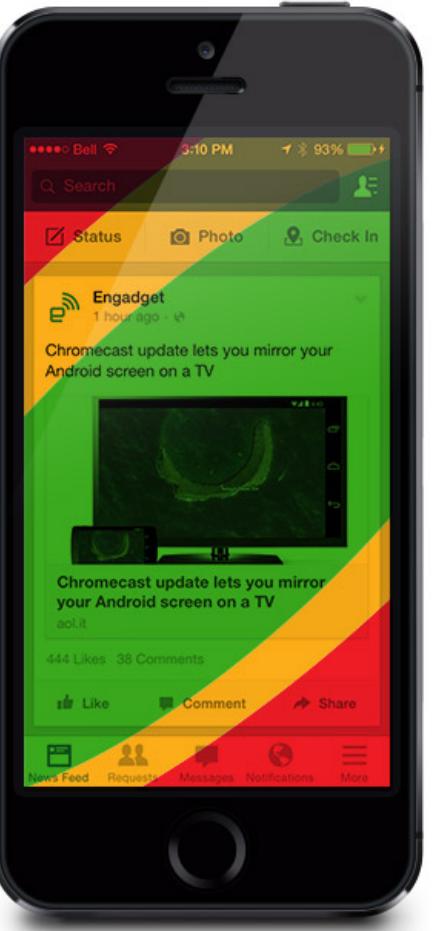
Take off

Land

Firefox OS



The comfort zone



Fitt's law

$$T = a + b \times \log_2 (1 + D / W)$$

T: Average time taken to complete the movement

a: Start/stop time of the device

b: Inherent speed of the device

D: Distance from the starting point to target center

W: Width of target, measured along axis of motion

Fitt's law, translated

The bigger and closer a target is,
the easier it is to hit.

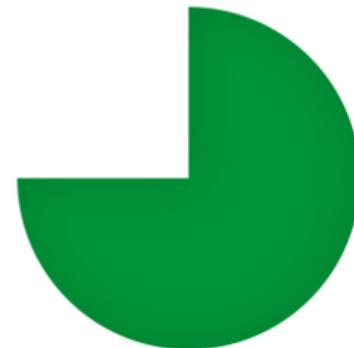
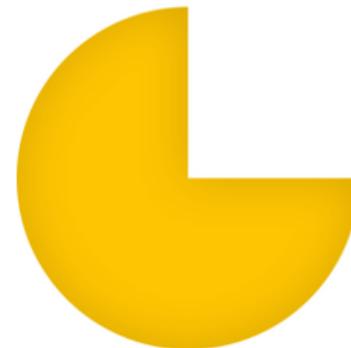
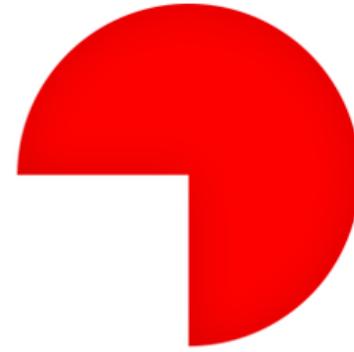
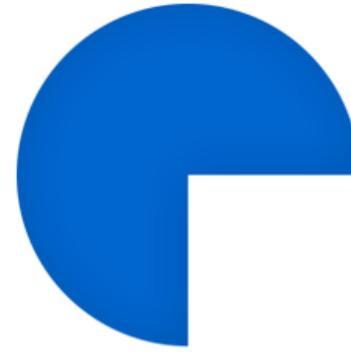
Optimize

(when it make sense)

JavaScript

Google Closure is my preferred one

- Hoisting (joining variables declarations)
- Removing unreachable code
- Obfuscate your code
- Renaming variables (shortening)
- Remove unnecessary spaces/code
- Reduce files size



<https://github.com/google/closure-compiler>

<http://closure-compiler.appspot.com/home>

Minimize or speed up HTTP requests

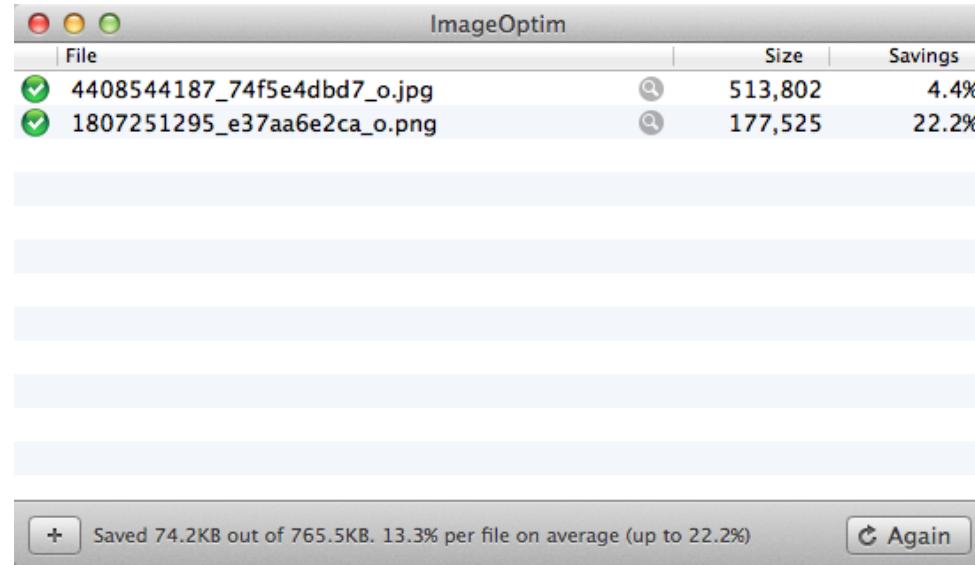
- Avoid or minimize 3xx redirections
- GZIP encoding (htaccess, web.config)
- Image sprites when it make sense
- CDN (Content Distribution Network)
- HTTP Cache Headers (Apache) or HTTP Expires Response Header (IIS)

Images

- Use Native Image Resolution (original width, height)
- Select the right image format (PNG, JPEG...)
- Use image preview for videos
- Compress your images

Be Flintheart Glomgold

Shrink your images
without losing quality



ImageOptim (OS X) - <https://imageoptim.com/>

Trimage (Linux) - <http://trimage.org/>

PNGGauntlet (Windows) - <http://pnggauntlet.com/>

Be aware of the rendering

Solid black texture sized 1920x1080



Once compressed = ~5kb PNG



Decompressed to memory = up to 8mb

($1920 \times 1080 \times 4$ bytes per pixels [rgba]) = 8294400 bytes

Small tips & tricks

- Avoid plugins usage
- Load JavaScript files at the end of the page
- Asynchronously load scripts (async) when it's possible
- JSON is faster than XML
- You don't always need a framework or library

Miscellaneous micro-optimization

- CSS for large bck images (DIV, CSS background under canvas)
- Multiple canvases for layers
- Don't scale images in drawImage (cache in offscreen canvas)
- Use WebGL* instead of Context 2D (free up CPU for GPU)
- Avoid creating new object when it's possible

In the end...

Google Developers PageSpeed Insights

Mobile Desktop

61 / 100 Speed

! Should Fix:

- Reduce server response time
 - › Show how to fix
- Eliminate render-blocking JavaScript and CSS in above-the-fold content
 - › Show how to fix

! Consider Fixing:

- Leverage browser caching
 - › Show how to fix
- Optimize images
 - › Show how to fix
- Minify CSS
 - › Show how to fix
- Minify JavaScript



<https://developers.google.com/speed/pagespeed/insights/>

JsPerf is your friend

	Test	Ops/sec
<u>Math.floor</u>	<code>Math.floor(89.938 / 293.3); Math.floor(83784 / 9289.2); Math.floor(7 / 60);</code>	1,768,597,852 ±0.84% fastest
Math.round	<code>Math.round(89.938 / 293.3); Math.round(83784 / 9289.2); Math.round(7 / 60);</code>	1,781,337,492 ±0.53% fastest
parseInt	<code>parseInt(89.938 / 293.3); parseInt(83784 / 9289.2); parseInt(7 / 60);</code>	31,229,167 ±0.53% 98% slower
Bitwise 	<code>89.938 / 293.3 0; 83784 / 9289.2 0; 7 / 60 0;</code>	1,768,534,912 ±0.91% fastest
Bitwise >>	<code>89.938 / 293.3 >> 0; 83784 / 9289.2 >> 0; 7 / 60 >> 0;</code>	1,747,552,494 ±1.93% 3% slower

<http://jsperf.com/browse>

Firefox Developer Tools



Try on low-end devices

Firefox OS Flame

<http://j.mp/1pd7p8g>

ZTE Open

<http://j.mp/1tPUIYN>



The philosophy behind great apps...

1. Don't make us think about the interface
2. Deal with complex tasks for us (insulate us from that complexity)
3. Make accomplishing our goals easier
4. Help users be awesome in the moment!



Frédéric Harper

fharper@mozilla.com

@fharper

<http://hacks.mozilla.org>

<http://outofcomfortzone.net>