



Electron.js and the Browser Wars

By James Reynolds

First Browser War (1995-2001) - IE vs Netscape

- 1983 - IPv4 deployed on the ARPANET
- 1993 - Mosaic is released
- 1993 - Steve Balmer learns about TCP/IP



<https://www.theguardian.com/global/2015/mar/22/web-browser-came-back-haunt-microsoft>

- 1995 - "Netscape would reduce Windows to a 'poorly debugged set of device drivers'"
Ian Murdock quoting Marc Andreessen (a Netscape founder)
<https://web.archive.org/web/20060812205515/http://www.ianmurdock.com/>
- 1995 - Gates pens the internal memo "The Internet Tidal Wave"
- 1998 - United States v. Microsoft Corp.

First Browser War (1995-2001) - IE vs Netscape

- 2000 - "Microsoft is a company with an institutional disdain for both the truth and for rules of law" - Judge Thomas Jackson
 - The 2001 settlement gave Microsoft "a special antitrust immunity to... destroy freedom of competition." - Andrew Chin
 - "Now the only way Microsoft can die is by suicide" - Cringely
- https://en.wikipedia.org/wiki/United_States_v._Microsoft_Corp.
- But, was the war really over web browsers?

A photograph of a dense forest with many tall, thin trees. The trees are mostly bare, suggesting a late autumn or winter setting. The ground is covered in fallen leaves and branches. In the lower center of the image, a person is standing, looking up at the trees. The text "DON'T MISS THE FOREST FOR THE TREES" is overlaid in large, white, bold, sans-serif capital letters.

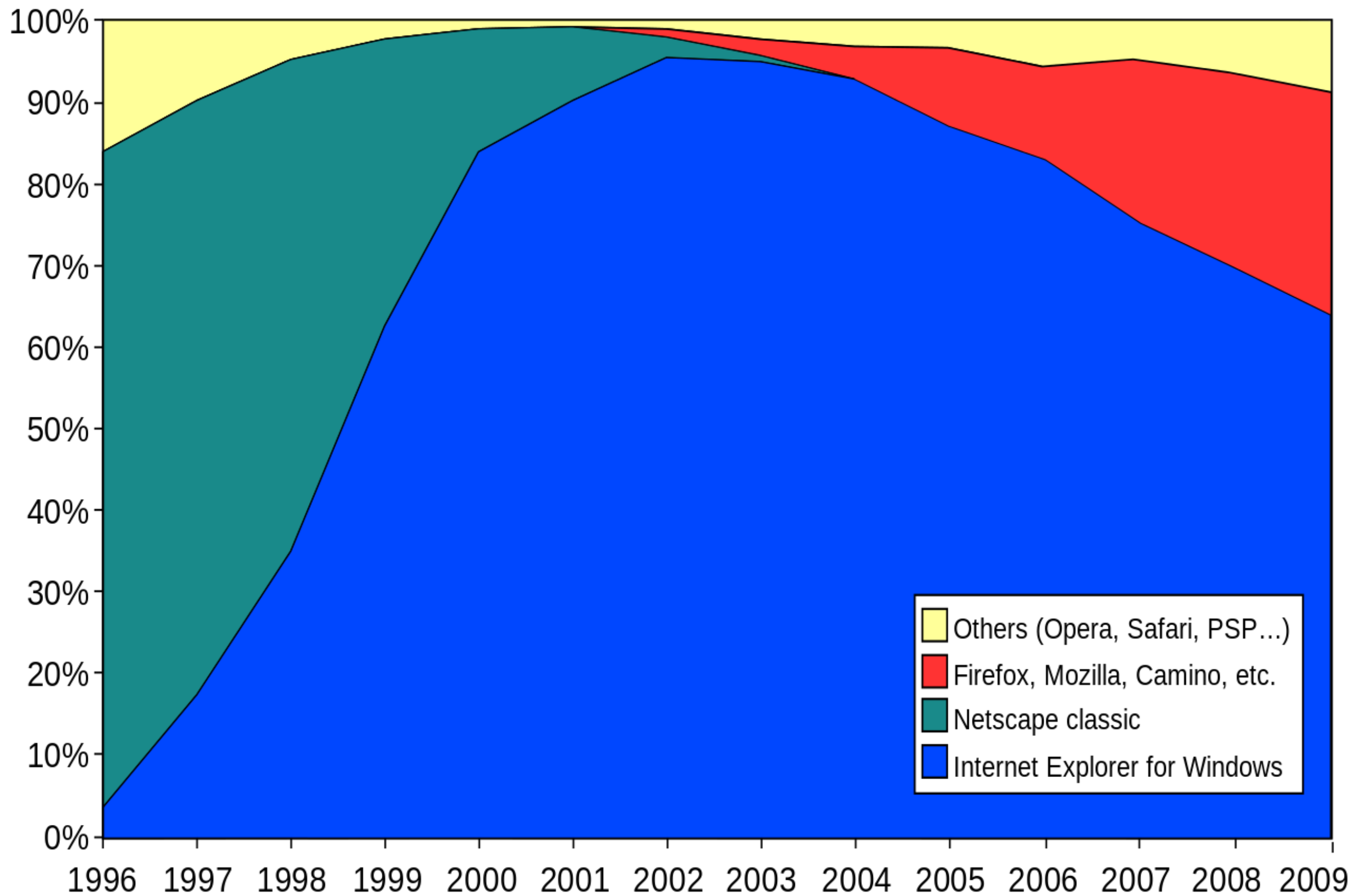
**DON'T MISS
THE FOREST
FOR THE TREES**

Second Browser War (2004-2017) - Enter Google

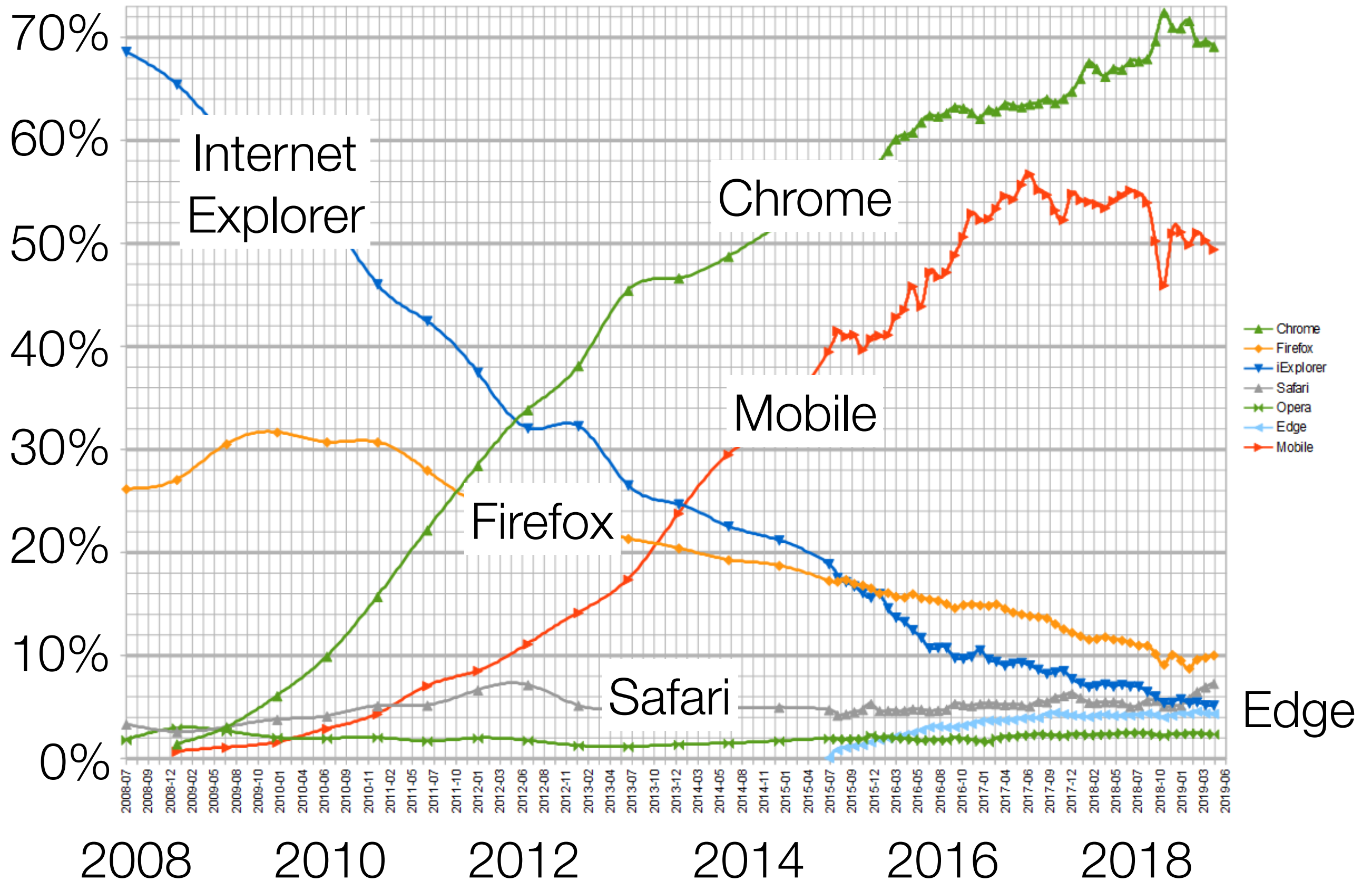
- 2001-2006 saw only one new version of IE
- 2003, MS announced IE for Mac discontinued
- 2003, Safari; 2005, WebKit released
- Opera and Firefox grow
- 2007, iPhone, "Thoughts on Flash"
- 2008, Chrome released (uses WebKit); and Android
- 2011, Chromebook released



Browser Wars



Usage share of browsers (source StatCounter)



The winner is...



Meanwhile, Down a Different Alley...

- GitHub founded (2008)
- Google Chromium/V8 (2008)
- Node.js (2009)
- MS TypeScript (2012)
- GitHub makes Electron (2013)
- MS buys GitHub (2018)
- MS goes to Chromium (2019)

```
it repository in /tmp/t
EOF '
ted revision control sy

t) e4dcc69] You can ed

insertion(+)
README
igin git@github.com:cdc
n master
```

Let's Talk About Chromium

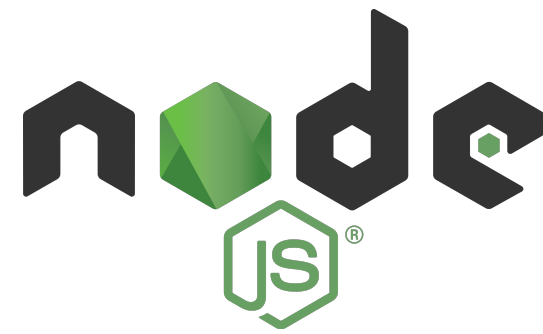
- Free, open-source, minimal web browser by Google
- Written in C, C++, JavaScript
- Includes the V8 JavaScript engine
- 293 MB download
- Many web browsers based on it including
 - Chrome, Edge, Opera, Vivaldi, Yandex, Silk, Brave





Let's Talk About Node.js

- Uses Google's V8 JavaScript runtime minus the web browser
- Joyent is the corporate steward (builds cloud apps, owned by Samsung)
- Part of OpenJS Foundation, which also includes jQuery
- `/usr/local/bin/node`
- Server-side scripting (competes with PHP and Java)
- Builds custom web server (competes with Apache)



Let's Talk About Node.js

- JavaScript now does front and back end web dev
- Event driven and asynchronous (compared to Apache and PHP scripts that blocked)
- npm (2010) is its package manager (cf. brew, fink, ports, yum, apt-get, cpan, pip, gem, etc etc etc)
- Thousands of open-source libraries for Node.js
- Atom, Brackets, VS Code all target Node.js development
- About 44 MB without any modules installed

Now Let's Talk About Electron

- "Build cross-platform desktop apps with JavaScript, HTML, and CSS"
- Created by Github (now owned by Microsoft)
- Chromium + Node.js (V8)
 - This is a web browser folks...
- Mac, Linux, Windows
- "JavaScript everywhere" (WORA)



Déjà Vu Anyone?

- Java
- Qt
- Delphi
- GTK+
- Tcl/Tk
- wxWidgets

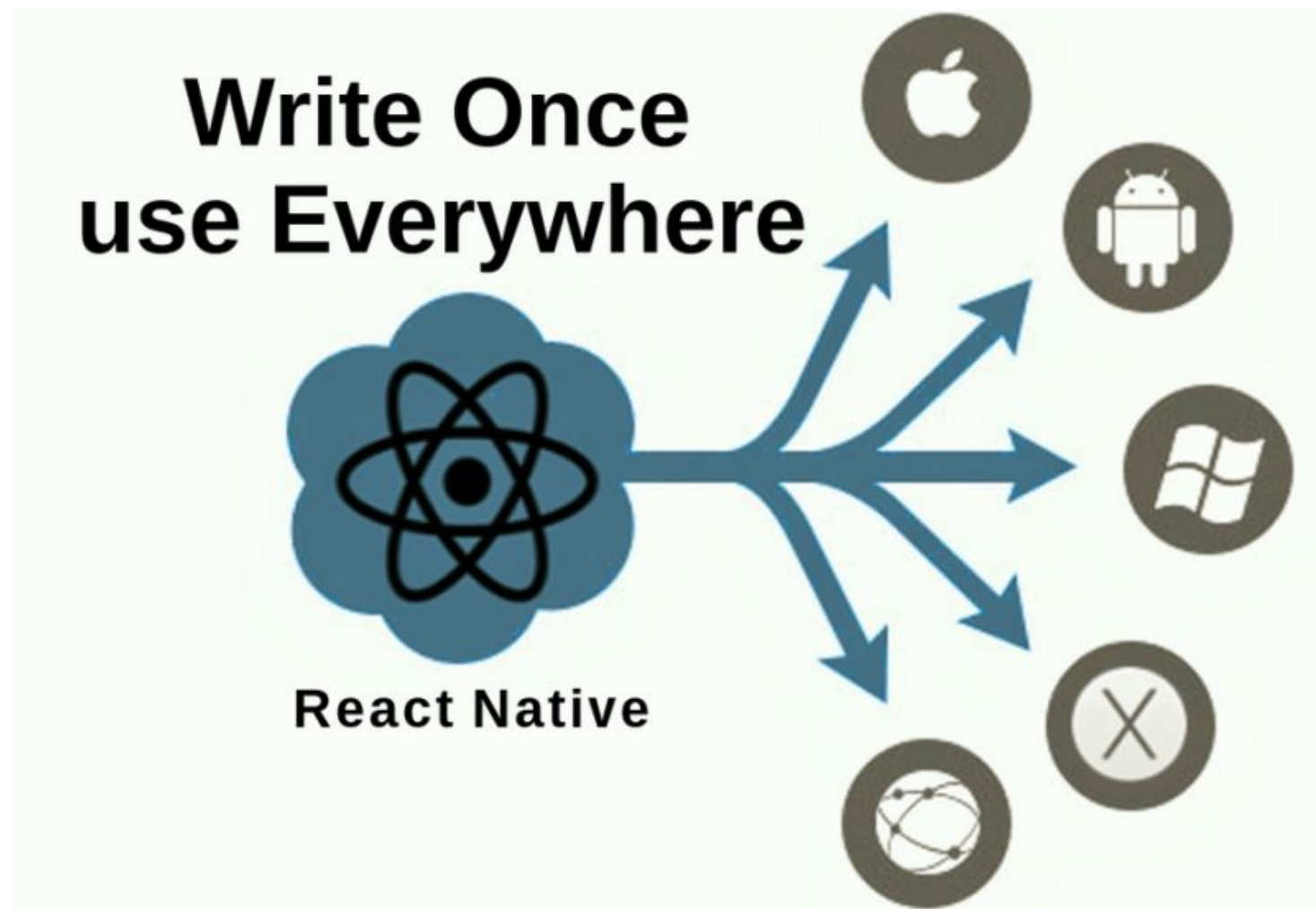
Anton Epple Java everywhere



Write Once Run Anywhere with DukeScript

React Native Is Facebook's Attempt at WORA

- JavaScript "rendered" to native code (no HTML or CSS)



Other Attempts At WORA

- Flutter from Google (2017), uses Dart
- "Progressive Web Apps" (Apple: "HTML Apps" or Web Apps")
 - Service Worker (JavaScript network proxy, allows offline use)
 - Web apps added to home screen and don't open as a Safari tab
 - Try 2048game.com
 - Does not require a special bundling or distribution (i.e. store)
 - Apple pushing this instead of "native apps that wrap a webpage"



Other Attempts At WORA



- WebAssembly (Wasm)
 - W3C announced in Dec 2019 Wasm is the "fourth" native web language after HTML, CSS, and JavaScript
 - Right now 91% of all installed browsers support Wasm
 - About 40 LLVM-supported languages can compile to a Wasm binary, e.g. C, C++, D, Rust, etc; eventually C#, Python
 - "For voice recognition, computer vision, etc"
 - A June 2019 study found that over 50% of Wasm sites were malware

<https://en.wikipedia.org/wiki/WebAssembly>

Advantages of Electron

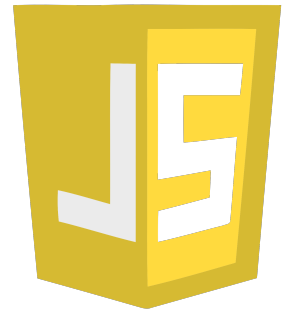
- No need to support multiple browsers
- Access to file system
- Code is already downloaded
- Huge community and momentum (a lot of work is going into making this as easy as possible)
- HTML/CSS/JavaScript

JavaScript

Why Does Electron use JavaScript/HTML/CSS?

- JavaScript is currently the most popular language

JavaScript



<https://insights.stackoverflow.com/survey/2019>

<https://github.blog/2018-11-15-state-of-the-octoverse-top-programming-languages/>

<https://www.techrepublic.com/article/the-worlds-most-popular-programming-language-is-javascript-but-why/>

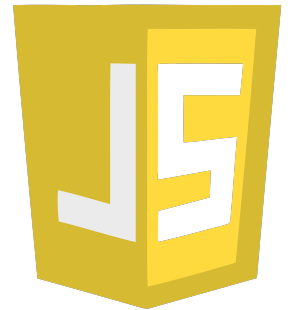
- When it comes to web dev, there is no other choice, it's a funnel forcing devs to learn them

JavaScript

Why Does Electron use JavaScript/HTML/CSS?

- Companies need only one set of developers
- Plugins are very easy to create
- GUI's are very easy to create
- Native coding is hard
- No need to follow the platform's HIG

JavaScript



Example

How can I create a UILabel with strikethrough text?

Asked 7 years, 6 months ago Active 6 days ago Viewed 65k times



I want to create a `UILabel` in which the text is like this

117

~~\$198~~



How can I do this? When the text is small, the line should also be small.



33

ios

swift

cocoa-touch

UILabel

strikethrough

Swift 4:

```
let attributeString: NSMutableAttributedString = NSMutableAttributedString(string: "$198")
attributeString.addAttribute(
    NSAttributedString.Key.strikethroughStyle, value: 2, range: NSRange(0, attributeString.length))
```

HTML:

`<s>$198</s>`

<https://stackoverflow.com/questions/13133014/how-can-i-create-a-UILabel-with-strikethrough-text>

Example - Apple Tries To Stop Lazy Devs

- This class supports multiple text styles through use of the **attributedText** property. Setting a value for this property causes the text view to use the style information provided in the attributed string. You can still use the **font**, **textColor**, and **textAlignment** properties to set style attributes, but those properties apply to all of the text in the text view. It's recommended that you use a text view—and not a **UIWebView** object—to display both plain and rich text in your app.

<https://developer.apple.com/documentation/uikit/uitextview>

What's The Problem?



- JavaScript might have problems
 - `alert((![]+[])[+[]]+(![]+[])[+!+[]]+(![]+[])[+!+[]]+(![]+[])[+!+[]]+(![]+[])[+!+[]]+(![]+[])[+!+[]]);`
<https://whydoesitsuck.com/why-does-javascript-suck/>
- I have nothing against Node.js
- I have nothing against Chromium (I use Brave)
- But every app a cross-platform web browser?
 - This also means every dev is now a web browser dev



Every-App-Is-A-Web-Browser Security



- <https://www.electronjs.org/docs/tutorial/security>
 - Keep application up-to-date with latest Electron
 - Choose trusted Node.js libraries and keep them up-to-date
 - XSS is worse for Electron apps, so code securely
 - Be careful executing code from remote servers (e.g. MITM)
 - Lists 17 security recommendations
 - e.g. "Do not use allowpopups"

Every-App-Is-A-Web-Browser Security



- There's 3 CVE's for Electron (2018, versions 1-3)
 - e.g. IFRAME remote code execution
- <https://snyk.io/vuln/npm:electron> lists vulns
- Chromium release every 6 weeks
 - Chrome fixes about 3 security vulns a weeks
- Electron release every 12 weeks
- Electron supports 4 release versions, Currently: 9, 8, 7, 6

Every-App-Is-A-Web-Browser Overhead

- Memory use
- CPU usage (battery drain)
- Hard disk use (some people still have 256 GB drives...)
 - App, modules, & caches! (these are web browsers)
- We generally use only one browser because of this



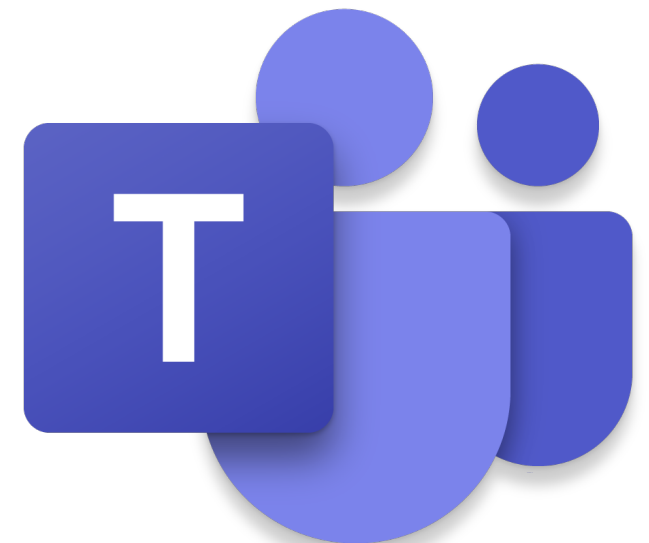
Case Study: Mailspring 1.7.6 (2020-04)

- Includes Electron 4.2.2, 2019-05 (latest 9)
 - snyk.io:13 High, 2 Medium vulns; 150+ vulns?
- Many emails are HTML from untrusted sources
- Do I dare use this?



Case Study: Microsoft Teams 1.00.309271

- 81 MB pkg download, 205 MB app
- Includes Electron 4.2.12 (2020-10)
 - snyk.io: 13 High, 2 Medium vulns; 84+ vulns?
- Application Support/Microsoft/Teams
 - After a few minutes of use 357 MB
 - In the wild 900 MB
- 406 MB RAM, just chat, nothing else, 365 MB in background



Case Study: Atom 1.47 (2020-05)

- Includes Electron 5.0.13 (2019-12)
 - snyk.io: 12 High, 2 Medium vulns; 65+ vulns?
- 190 MB download, 586 MB app
- Reported to use 1 GB of RAM to edit a single file (I didn't see this, maybe it's fixed)



Case Study: Microsoft VS Code 1.45.1 (2020-04)

- 86 MB download, 246 MB app
- Includes Electron 7.2.4
 - snyk.io: 9 High vulns
- Caches com.microsoft.VSCode.Shipit 246 MB
- ~/.vscode - 327 MB










Case Study: Slack 4.5.1 (2020-05)







- 68 MB dmg download, 186 MB app
- Includes Electron 8.2.5
 - snyk.io: 2 High vulns
- Application Support/Slack
 - After a few minutes: 86 MB
 - In the wild: 503 MB
- 300 MB RAM scrolling, 220 MB idle (Slack+3 Slack helpers)



Popular Electron Apps

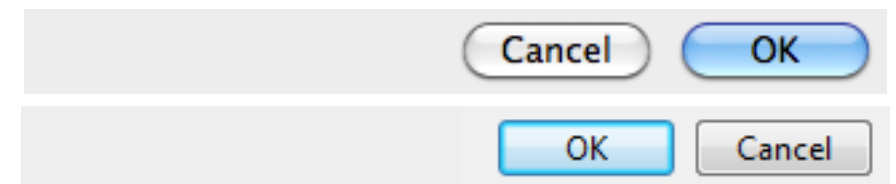


- Atom (MS) 
- GitHub Desktop (MS) 
- Teams (MS) 
- Skype (MS) 
- VS Code (MS) 
- Slack 
- WhatsApp (FB) 

- WordPress Desktop 
- Basecamp 3 
- CrashPlan 
- Discord 
- Signal 
- Arq 6 

What's Wrong With Cross-Platform Apps?

- "Write once, debug everywhere"
- Lowest common denominator features
- Different user interface conventions
- Performance penalty: interpreters, virtual machines
- Installers
- Fertile environment for malware



Atom and VS Code DO NOT PRINT



- And the devs don't care
- Plugin will render to HTML and open it in a browser
- Deal breaker for me (I bought a color laser printer)

Other Comments Online

- "Electron is without question a scourge" - John Gruber
https://daringfireball.net/2018/12/electron_and_the_decline_of_native_apps
- "Electron is a cancer murdering both macOS and Windows as it proliferates." - SwiftOnSecurity
<https://twitter.com/SwiftOnSecurity/status/1069776335336292352>
- "Electron enables lazy developers to write garbage" - Drew DeVault
<https://drewdevault.com/2016/11/24/Electron-considered-harmful.html>
- "Developers don't let friends write electron apps" - Joseph Gentle
<https://josephg.com/blog/electron-is-flash-for-the-desktop/>

Other Comments Online

- "Atom uses around around one gigabyte of memory to edit a single medium sized file... 'Well, it works fine on my machine, and I only have 32 gigabytes of ram.' - Silicon Valley Developer, 2017" - Casper Beyer
<https://medium.com/commitlog/electron-is-cancer-b066108e6c32>
- "Electron's bulk is primarily a symptom... where there's such an apparent abundance of a commonly-shared resource that people use far more of it than they should, leading the resource to be wiped out." - Meredith Espinosa
<https://medium.com/@boundarybreaker/electron-is-a-hulking-monstrosity-of-a-wora-framework-and-it-needs-to-be-replaced-25e9d849b0e>

But... There's Something Else Going On Here



This is a "web thin client"
\$400



This is not.
Starts at \$5000



- No secret they believe the future is web thin clients
- By March 2018, Chromebooks made up 60% of computers purchased by schools in the United States.
<https://en.wikipedia.org/wiki/Chromebook>
- I can't really judge how well Google has switched the computing world to web thin apps but I'm not too worried (I am worried about the rising generation though)

Welcome to Browser Wars 3.0

- June 4, 2018, Microsoft Buys GitHub for \$7.5 Billion
- Microsoft now owns Electron.js (and is using it heavily)



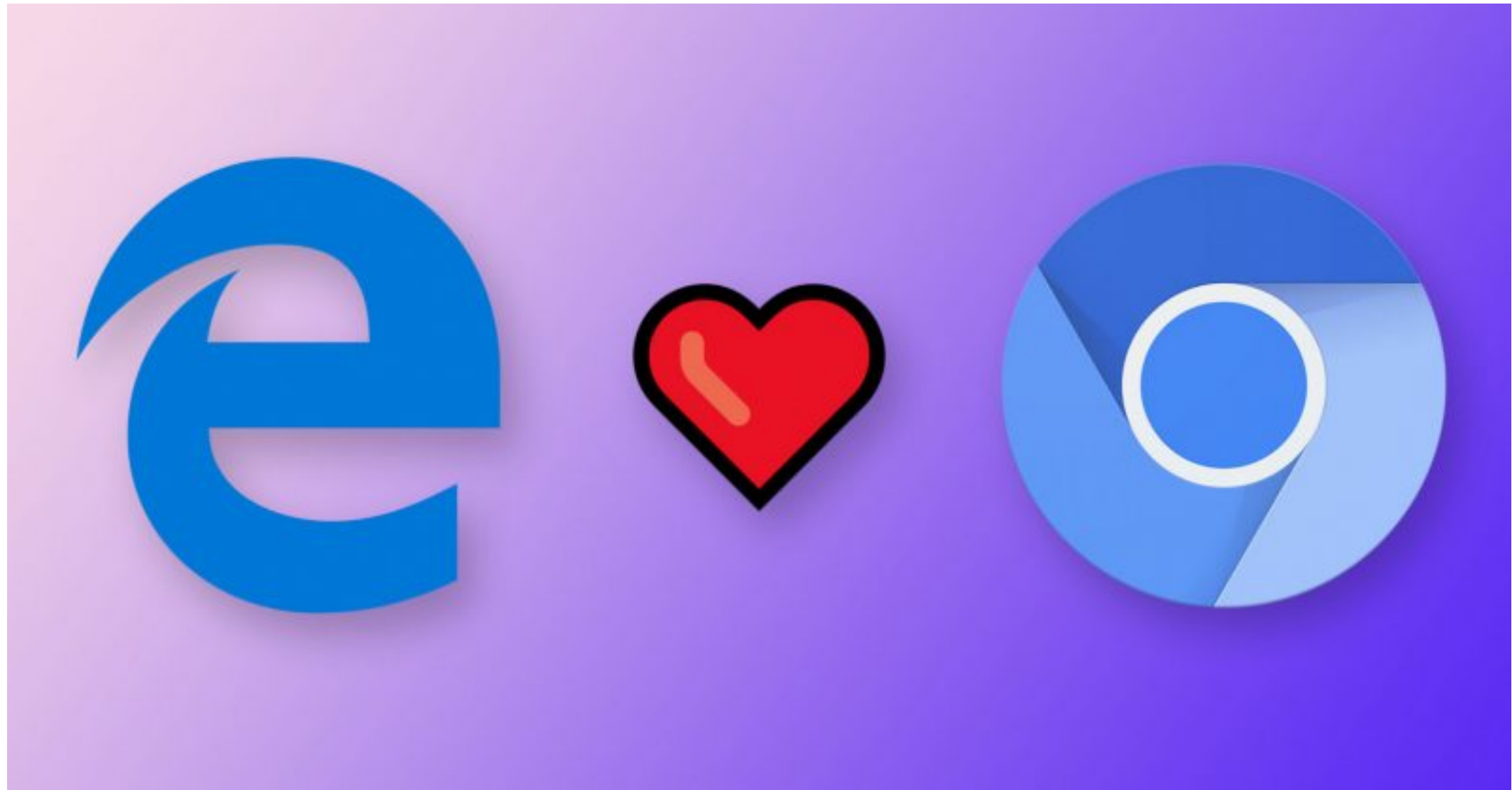
This is a web thin client
in disguise



This is now a web fat client

But... But... But...

- 2019 - Microsoft drops EdgeHTML in favor of...



How Can It Be A Browser War If...

- Both sides are using (and improving) Google's Chromium
- "This isn't about Chrome. This is about ElectronJS. Microsoft thinks EdgeHTML cannot get to drop-in feature-parity with Chromium to replace it in Electron apps, whose duplication is becoming a significant performance drain. They want to single-instance Electron with their own fork" - SwiftOnSecurity
<https://twitter.com/SwiftOnSecurity/status/1069776335336292352>
- It's not a really a "browser" war...
- It's always been about Windows (aka Microsoft) dominance

Don't Miss the Forest For The Trees...

- "It's easy to imagine a single shared thread for Chromium on top of Windows, which can be accessed by any Electron-based instance."
- "If Electron was already overwhelmingly the platform of choice *despite its massive constraints*, this is going to open a new tidal wave of web-based apps on the desktop. Why would you build in any other language at this point, if you can write once and run everywhere?"

https://www.vice.com/en_us/article/59vke8/microsoft-putting-edge-on-chromium-will-fundamentally-change-the-web

Remember “The Internet Tidal Wave”? (1995)

- Looks like this
- Fast on Windows
- Bloated everywhere else
- And if it's not Electron it'll be React Native or something else
- Slowing down the competition has been Microsoft's game plan since forever
- Microsoft won't be taken to court because of "open source" and "but... Google won the browser war!"
- This is why Microsoft is trying to own the dev community and environment

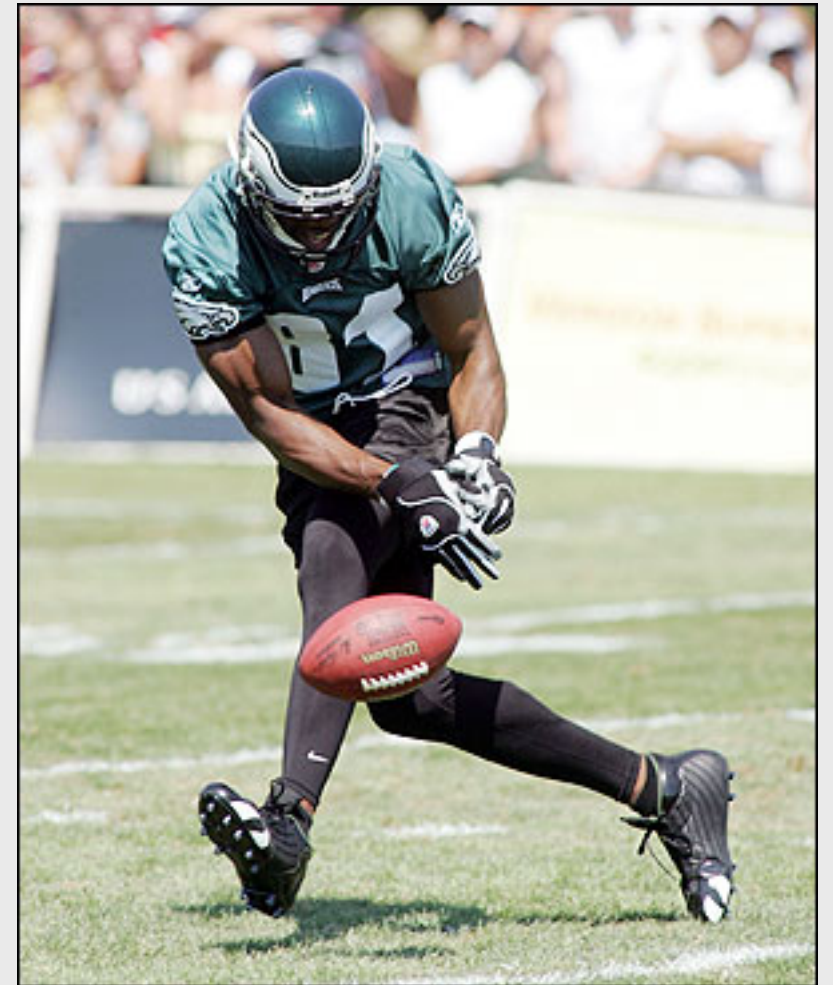


What About Apple?...

- "The users who really care about good native apps... were mostly already on the Mac....

Apple, of all companies, is shipping Mac apps with glaring un-Mac-like problems... 'Marzipan' apps...

The problem with Electron apps isn't really Electron — it's the decline in demand for well-made native Mac apps. And that is ominous." - John Gruber



https://daringfireball.net/2018/12/electron_and_the_decline_of_native_apps



“We have to let go of this notion that
for Apple to win,
Microsoft has to lose.
We have to embrace a notion that
for Apple to win,
Apple has to do a really good job.”

–Steve Jobs, 1997 MacWorld Expo

“Everything is based on a simple rule:
Quality is the best business plan, period.”

–Steve Jobs



About This Presentation

It started as an Electron rant...

Hope We Make It!

- Questions?

