

JavaScript in 2016: Beyond Harmony

Brendan Eich
Big Techday Munich 2016

Life has been Rough on the Web

You can get your hand cut off...



We all deserve a better Web

And a chainsaw for a hand
(optional)

Stuff Happened Last Year

- Firefox OS (\$200M?) failed to get traction
- Tizen (\$10B?) failed to get traction
- Smartphones and the Web grew together
- Flipkart took down its website, went native-app-only
- Flipkart came back as a Progressive Web App



There have been a few turning points in the history of the web platform that radically changed how web apps were built, deployed and experienced. Ajax was one such pivot that led to a profound shift in web engineering. It allowed web applications to be responsive enough to challenge the conventional desktop apps. However on mobile, the experience was defined by native apps and web apps hardly came close to them, at least until now. Mobile Engineering team at Flipkart discovered that with right set of capabilities in a browser, a mobile web app can be as performant as a native app.

Thanks to [Extensible Web Manifesto](#)'s efforts to tighten the feedback loop between the editors of web standards and web developers, browser vendors started introducing new low-level APIs based on the feedback from developers. The advent of these APIs brings unprecedented capabilities to the web. We, at Flipkart, decided to live on this bleeding edge and build a truly powerful and technically advanced web app while working to further evolve these APIs.

EWM FTW!



**IN THE FUTURE, SERVICE
WORKERS WILL BE ABLE**

TO TALK TO EACH OTHER.



What's the Problem?

- Nothing, really (walled gardens, crap security aside)
- The Web is not going away
- Web Developers are better off than ever
- The Web is still messy and NON-MINIMAL
- It always will be, because it grows compatibly
- Compatibility breaks happen only over decades

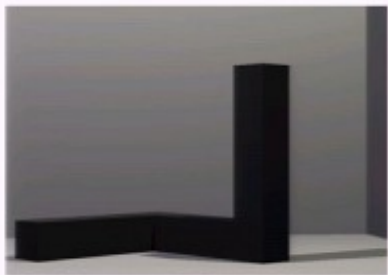
Hybrid Apps Live

- Amazon
- iOS App Store and iTunes
- Twitter
- Instagram (and even Facebook)
- and many more

Problems / Solutions

- Lack of Offline operation / Service Workers
- Startup Time / Server-Side Render + "rehydration"
- 60fps Touch / React Canvas
- C++-level Performance / asm.js + WebAssembly
- Data Plan & Battery / Ad + Tracker Blocking

Minimalism



The Web isn't Minimalist

**Systems Software Research is
Irrelevant**

Rob Pike

Bell Labs

Lucent Technologies

`rob@plan9.bell-labs.com`

Feb 21, 2000

"After 20 years, this is still the best exposition of the workings of a 'real' operating system."

Ken Thompson

Lions' Commentary on UNIX[®]

6th Edition

with Source Code

John Lions

Foreword by Dennis Ritchie



```
2904: /*
2905:  * This table is the switch used to transfer
2906:  * to the appropriate routine for processing a system call.
2907:  * Each row contains the number of arguments expected
2908:  * and a pointer to the routine.
2909:  */
2910: int      sysent[]
2911: {
2912:     0, &nullsys,          /* 0 = indir */
2913:     0, &rexit,            /* 1 = exit */
2914:     0, &fork,             /* 2 = fork */
2915:     2, &read,             /* 3 = read */
2916:     2, &write,           /* 4 = write */
2917:     2, &open,            /* 5 = open */
2918:     0, &close,           /* 6 = close */
2919:     0, &wait,            /* 7 = wait */
2920:     2, &creat,           /* 8 = creat */
2921:     2, &link,            /* 9 = link */
2922:     1, &unlink,          /* 10 = unlink */
2923:     2, &exec,            /* 11 = exec */
2924:     1, &chdir,           /* 12 = chdir */
2925:     0, &ptime,           /* 13 = time */
2926:     3, &mknod,           /* 14 = mknod */
2927:     2, &chmod,           /* 15 = chmod */
2928:     2, &chown,           /* 16 = chown */
2929:     1, &sbreak,          /* 17 = break */
2930:     2, &stat,            /* 18 = stat */
2931:     2, &seek,            /* 19 = seek */
2932:     0, &getpid,         /* 20 = getpid */
```

The Web is not Unix

Hot from TC39

- Google is Microsoft
- Microsoft is Mozilla
- Apple is Apple
- `async await` — coming soon to V8
- @wycats' decorators proposal

Decorators

- ```
class Person {
 @nonenumerable
 get kidCount() {
 return this.children.length;
 }
}
```
- ```
function nonenumerable(target, name,  
                           descriptor) {  
    descriptor.enumerable = false;  
    return descriptor;  
}
```

Decorators (2)

```
• class C {  
    @enumerable(false)  
    method() { }  
}  
  
function enumerable(value) {  
    return function (target, key, desc) {  
        desc.enumerable = value;  
        return desc;  
    }  
}
```

“WebAssembly may go live in browsers
this year” - InfoWorld, 3/2/2016



WebAssembly

- Started from asm.js, which (V8 helped) killed PNaCl
- Self-verifying, goto-free AST bitcode, not bytecode.
- Mozilla, Google, Microsoft and Apple fully on board
- ML specification (definitional interpreter, with tests)
- Emscripten support via JS wasm=>asm.js shim
- Engine getting native WebAssembly support NOW

Coming to Top JS Engines

- ChakraCore: <https://github.com/Microsoft/ChakraCore/wiki/Roadmap>
- Chrome: <https://www.chromestatus.com/features/5453022515691520>
- SpiderMonkey: <https://platform-status.mozilla.org/#web-assembly>
- WebKit: <https://webkit.org/status/#specification-webassembly>
- Compatible experimental implementations in the Nightly/Canary/TechPreview channels THIS YEAR



Threads for JS?!

"... over your dead body!"

Brendan Eich, 12 Feb 2007

```
$ ./bin/asm2wasm test/hello_world.asm.js
(module
  (export $add "add" $add)
  (func $add (param $x i32) (param $y i32) (result i32)
    (i32.add
      (get_local $x)
      (get_local $y)
    )
  )
)
```

WebAssembly isn't LISP

but it does have an s-expression syntax for view-source
and tooling convenience

WebAssembly is Unix

Gone Non-Native

- Lots of Games based on Unity 5
 - Dead Trigger 2 (demo)
 - Alphabear
 - Ski Safari
 - Many others in the pipe
- Unreal Engine 4 under way (needs threads)

Screenshot of an email client window. The window title is "Outlook - Mail". The interface shows a list of messages with columns for "From", "Subject", and "Received".

From	Subject	Received
John	Compassionate and professional support for your business	11/14/11
Microsoft	Microsoft Office 2010	11/14/11
Microsoft	Microsoft Office 2010	11/14/11
Microsoft	Microsoft Office 2010	11/14/11

Screenshot of a presentation slide titled "Microarray Data Analysis". The slide content includes a table with columns for "Gene", "Expression", and "Significance".

Gene	Expression	Significance
Gene 1	1.2	0.05
Gene 2	1.5	0.01
Gene 3	1.8	0.001
Gene 4	2.1	0.0001
Gene 5	2.5	0.00001

Screenshot of a presentation slide titled "Demos". The slide content includes a list of items:

- 1. Introduction
- 2. Data Analysis
- 3. Results
- 4. Conclusion

Screenshot of a Windows desktop with a purple background. The desktop contains several icons, including "My Computer", "Recycle Bin", and "Network". A large window titled "Outlook - Mail" is open, displaying a list of messages. A smaller window titled "Microarray Data Analysis" is also open, showing a table of data. A third window titled "Demos" is open, showing a list of items. The taskbar at the bottom shows the Start button and several open applications.

Screenshot of a Windows taskbar showing several open applications. The taskbar includes the Start button and icons for "Outlook", "Microarray Data Analysis", and "Demos".

WebAssembly

[Overview](#)[Demo](#)[Design](#)[Specification](#)[Community Group](#)

WebAssembly or *wasm* is a new portable, size- and load-time-efficient format suitable for compilation to the web.

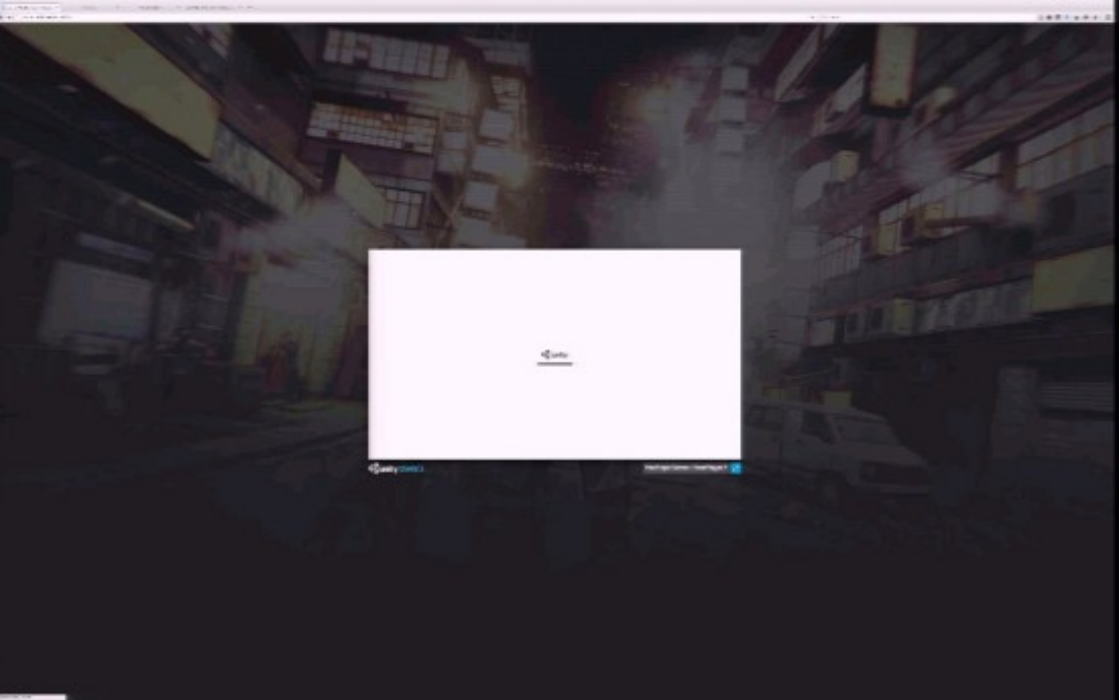
WebAssembly is currently being designed as an open standard by a [W3C Community Group](#) that includes representatives from all major browsers. Expect the contents of this website and its associated design repositories to be in flux: everything is still under discussion and subject to change.

Overview

- **Efficient and fast:** The *wasm AST* is designed to be encoded in a size- and load-time-efficient [binary format](#). WebAssembly aims to execute at native speed by taking advantage of [common hardware capabilities](#) available on a wide range of platforms.
- **Safe:** WebAssembly describes a memory-safe, sandboxed [execution environment](#) that may even be implemented inside existing JavaScript virtual machines. When [embedded in the web](#), WebAssembly will enforce the same-origin and permissions security policies of the browser.
- **Open and debuggable:** WebAssembly is designed to be pretty-printed in a [textual format](#) for debugging, testing, experimenting, optimizing, learning, teaching, and writing programs by hand. The textual format will be used when [viewing the source](#) of *wasm* modules on the web.
- **Part of the open web platform:** WebAssembly is designed to maintain the versionless, feature-tested, and backwards-compatible [nature of the web](#). WebAssembly modules will be able to call into and out of the JavaScript context and access browser functionality through the same Web APIs accessible from JavaScript. WebAssembly also supports [non-web](#) embeddings.

Read the project's [high-level goals](#) and consult the [FAQ](#) section for more information.





 [Link 1](#)

[Link 2](#) 



If you want to drift, you need to be here to watch.

CURRENT OBJECTIVE

Escort Dr. Adamos to the chopper

LAST INTERCOM



Escort Dr. Adamos to the chopper

GAME PAUSED

END GAME

BACK

CRITICAL MESSAGE

Evac Dr. Adams to the chopper

CRITICAL MESSAGE

Evac Dr. Adams to the chopper

GAME PAUSED

END GAME

BACK

← [Return to Game](#)

[Return to Game](#) →

SKI SAFARI

CREATED BY
BRENDAN WATTS AND SHAWN EUSTACE

WebAssembly

[Overview](#)[Demo](#)[Design](#)[Specification](#)[Community Group](#)

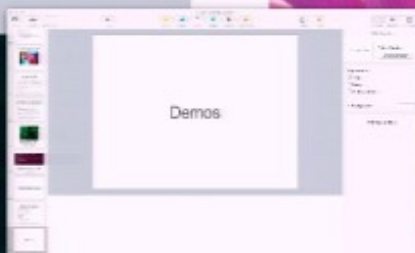
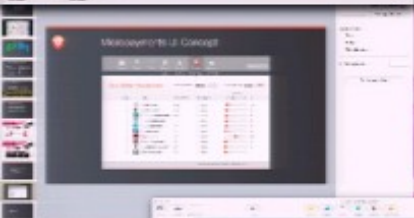
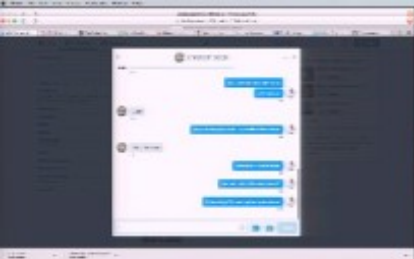
WebAssembly or *wasm* is a new portable, size- and load-time-efficient format suitable for compilation to the web.

WebAssembly is currently being designed as an open standard by a [W3C Community Group](#) that includes representatives from all major browsers. Expect the contents of this website and its associated design repositories to be in flux: everything is still under discussion and subject to change.

Overview

- **Efficient and fast:** The *wasm AST* is designed to be encoded in a size- and load-time-efficient [binary format](#). WebAssembly aims to execute at native speed by taking advantage of [common hardware capabilities](#) available on a wide range of platforms.
- **Safe:** WebAssembly describes a memory-safe, sandboxed [execution environment](#) that may even be implemented inside existing JavaScript virtual machines. When [embedded in the web](#), WebAssembly will enforce the same-origin and permissions security policies of the browser.
- **Open and debuggable:** WebAssembly is designed to be pretty-printed in a [textual format](#) for debugging, testing, experimenting, optimizing, learning, teaching, and writing programs by hand. The textual format will be used when [viewing the source](#) of *wasm* modules on the web.
- **Part of the open web platform:** WebAssembly is designed to maintain the versionless, feature-tested, and backwards-compatible [nature of the web](#). WebAssembly modules will be able to call into and out of the JavaScript context and access browser functionality through the same Web APIs accessible from JavaScript. WebAssembly also supports [non-web](#) embeddings.

Read the project's [high-level goals](#) and consult the [FAQ](#) section for more information.





Micropayments UI Concept

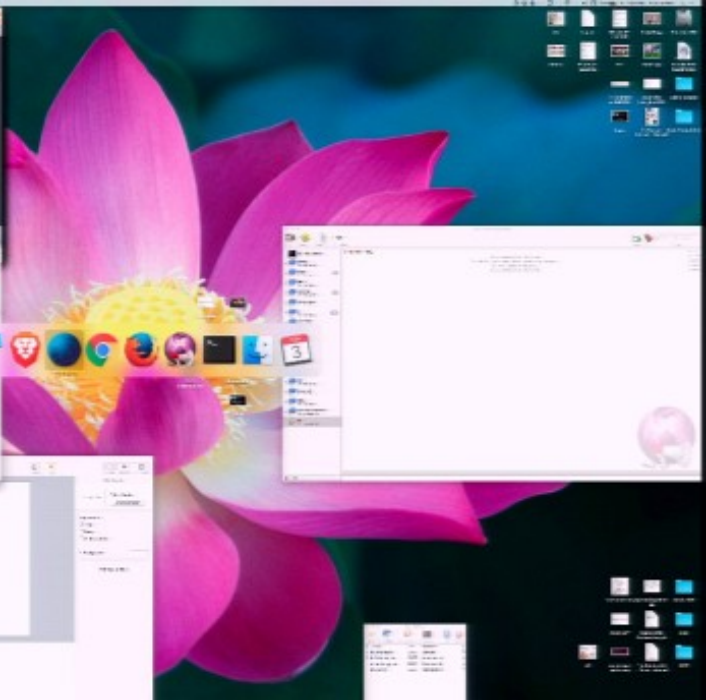
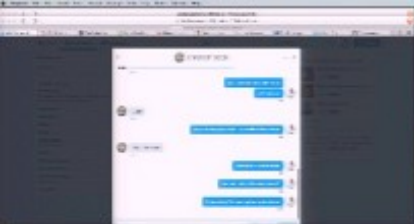
The screenshot shows a browser interface with a top navigation bar containing icons for General, Search, Tabs, Privacy, Security, **Micropayments**, and Advanced. Below these are links for Sync, Options, Personalize, and Insert Map. A search bar is on the right.

The main content area features a section titled "Top Sites Recipients". To the right of this title are two controls: "account balance \$85.00" with a "+" button, and "monthly budget \$5.00" with an "edit" button.

Below these controls is a table with the following columns: Rank, URL, Page Views, Time Spent, and a micropayment control. The micropayment control consists of a slider with a red dot and a percentage value to its right. The table lists the top 10 sites:

Rank	URL	Page Views	Time Spent	micropayment control
1	reddit.com	149	5h 12m	10%
2	nytimes.com	120	4h 34m	9%
3	huffingtonpost.com	111	4h 11m	8%
4	theguardian.com	82	3h 42m	8%
5	engadget.com	61	2h 42m	8%
6	forbes.com	53	1h 42m	0%
7	stgate.com	46	1h 42m	50%
8	bloomberg.com	32	1h 42m	4%
9	dudgereport.com	18	1h 42m	2%
10	cbenews.com	14	1h 42m	1%

At the bottom of the page, there is a small text line: "Allocation will process at 12:00am on March 20, 2010".



WebAssembly

Assembly

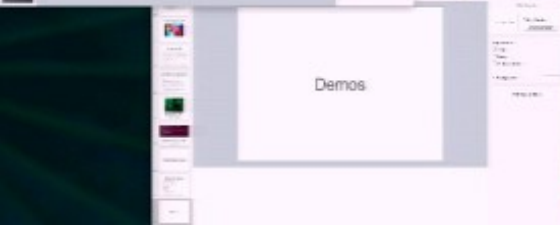
Assembly

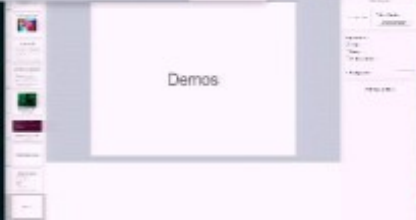


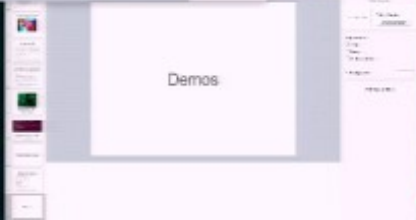
Microarray

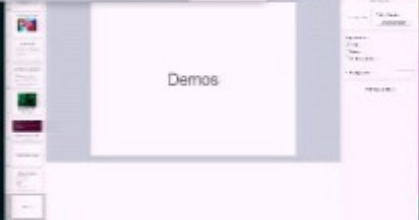


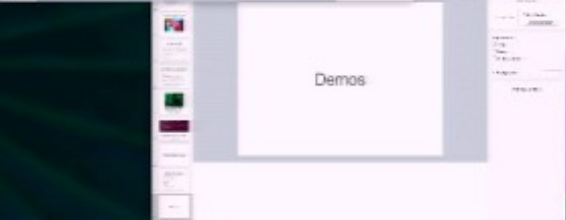
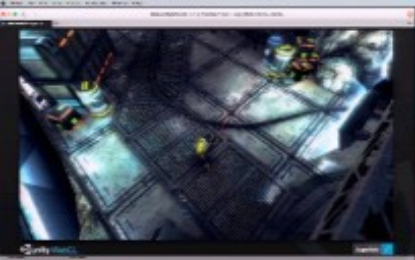
Demos

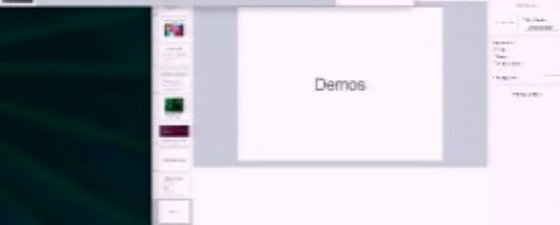
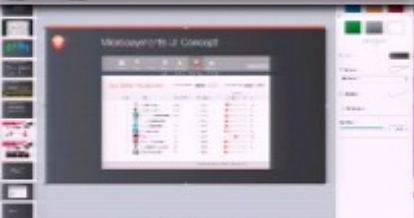


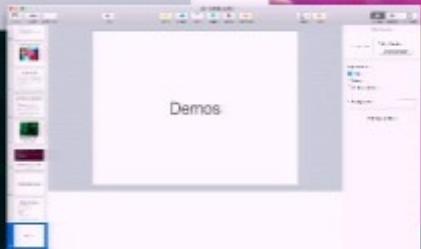
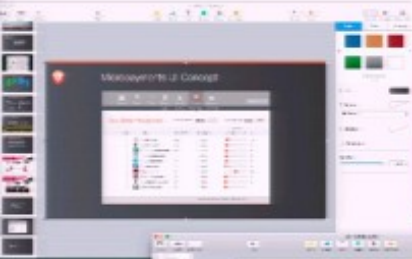












Demos

Always bet on JS

- First they said JS couldn't be useful for building "rich Internet apps"
- Then they said it couldn't be fast
- Then they said it couldn't be fixed
- Then it couldn't do multicore/GPU
- Wrong every time!
- My advice: **always bet on JS & WASM!**

