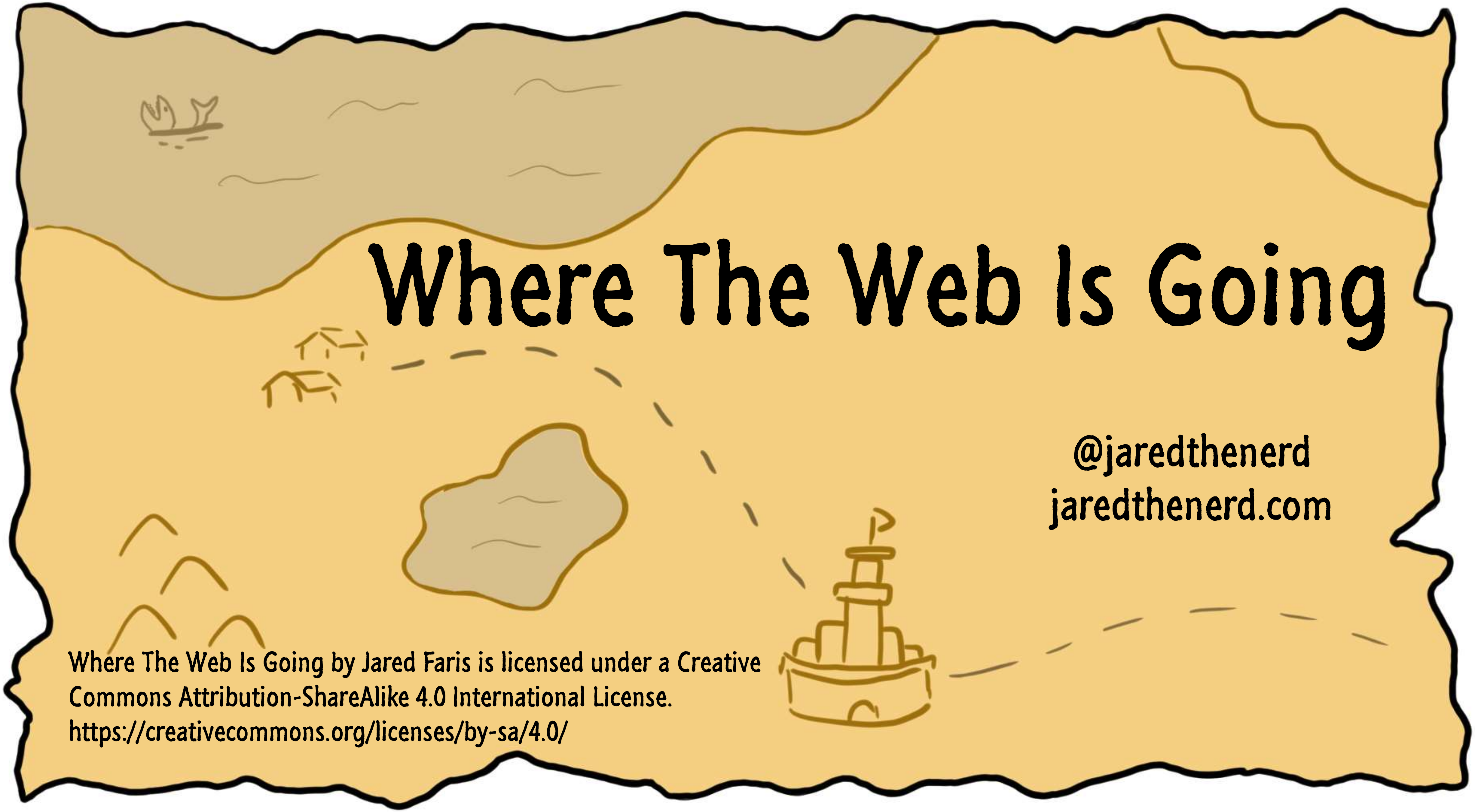
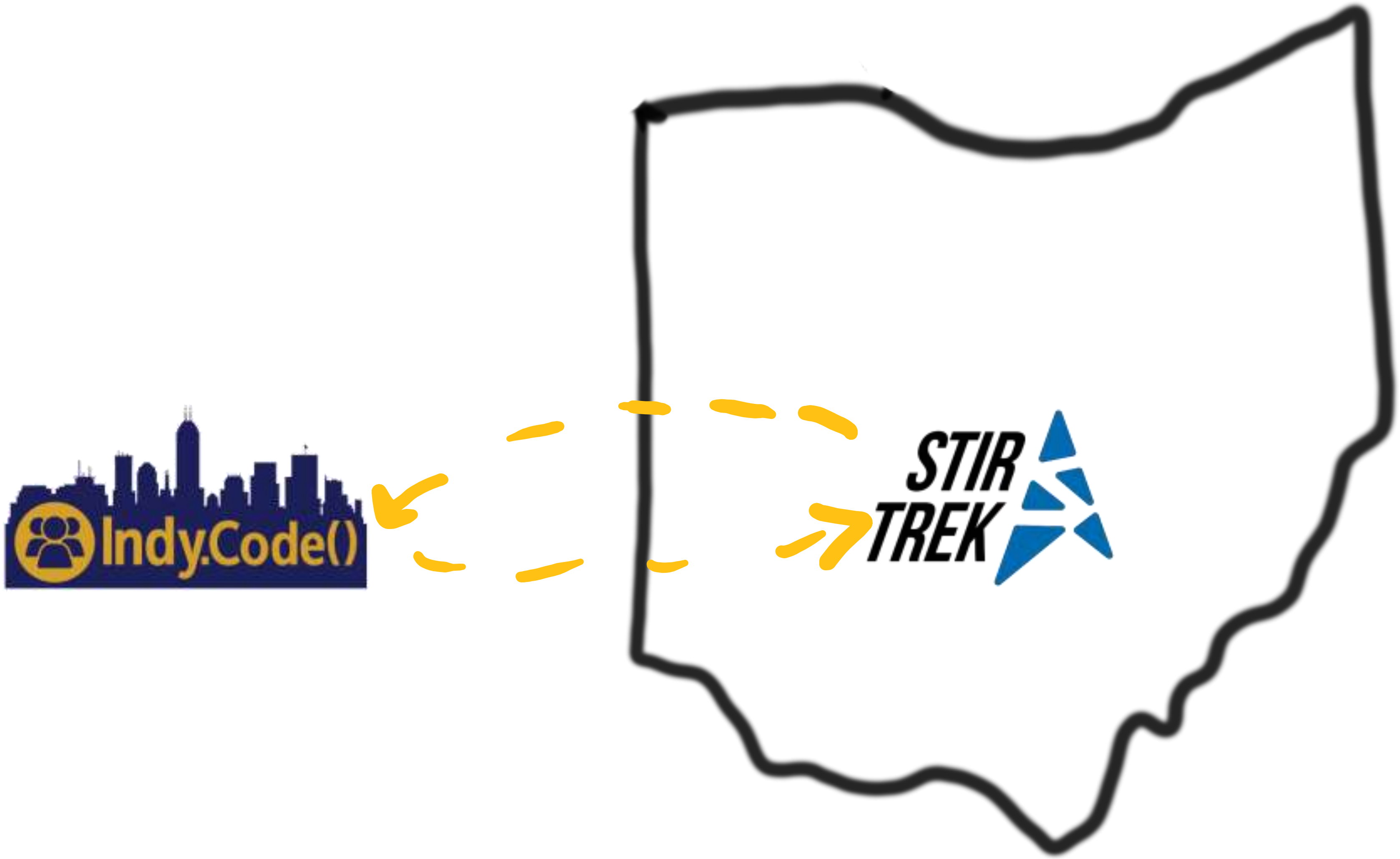
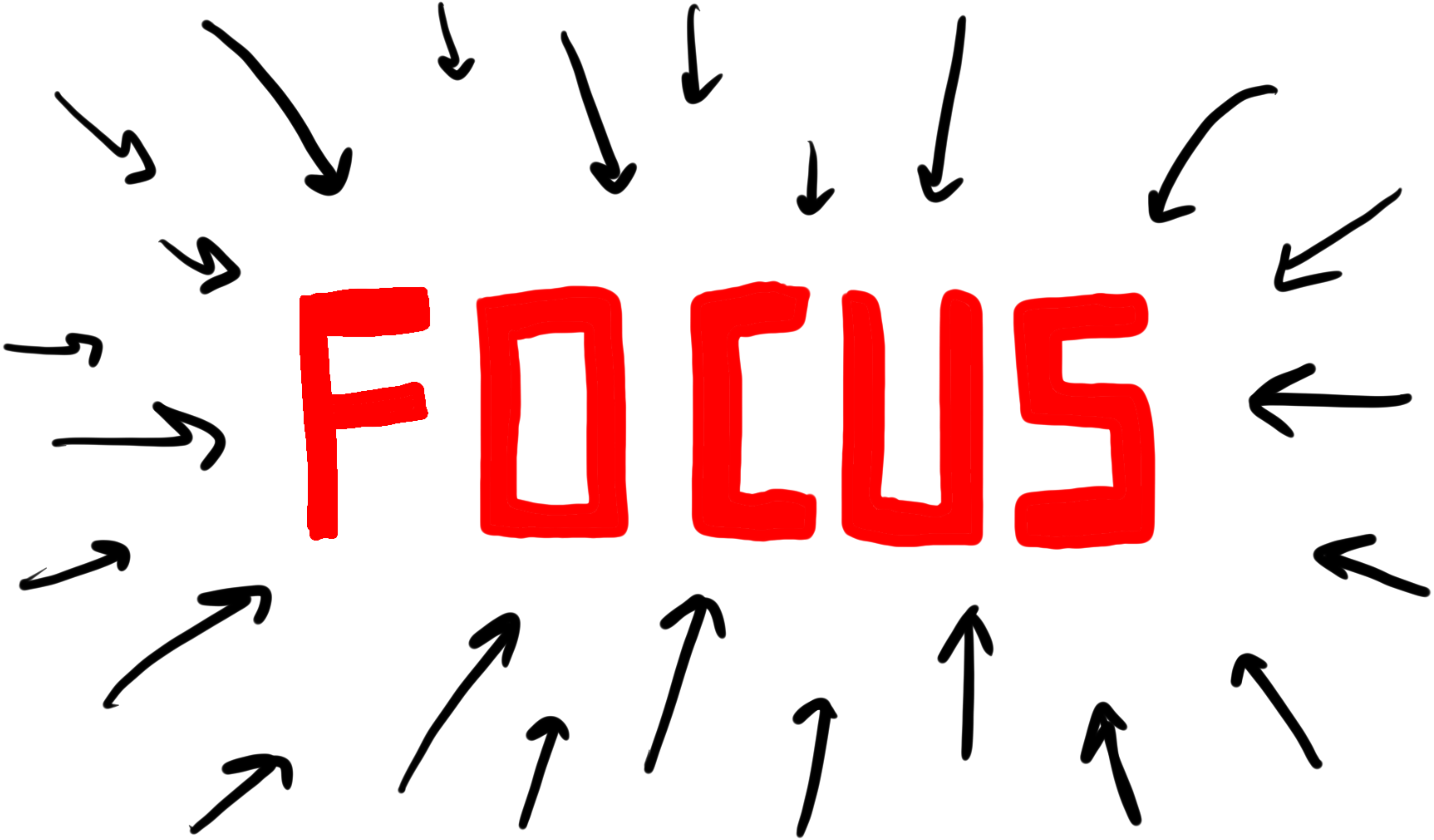


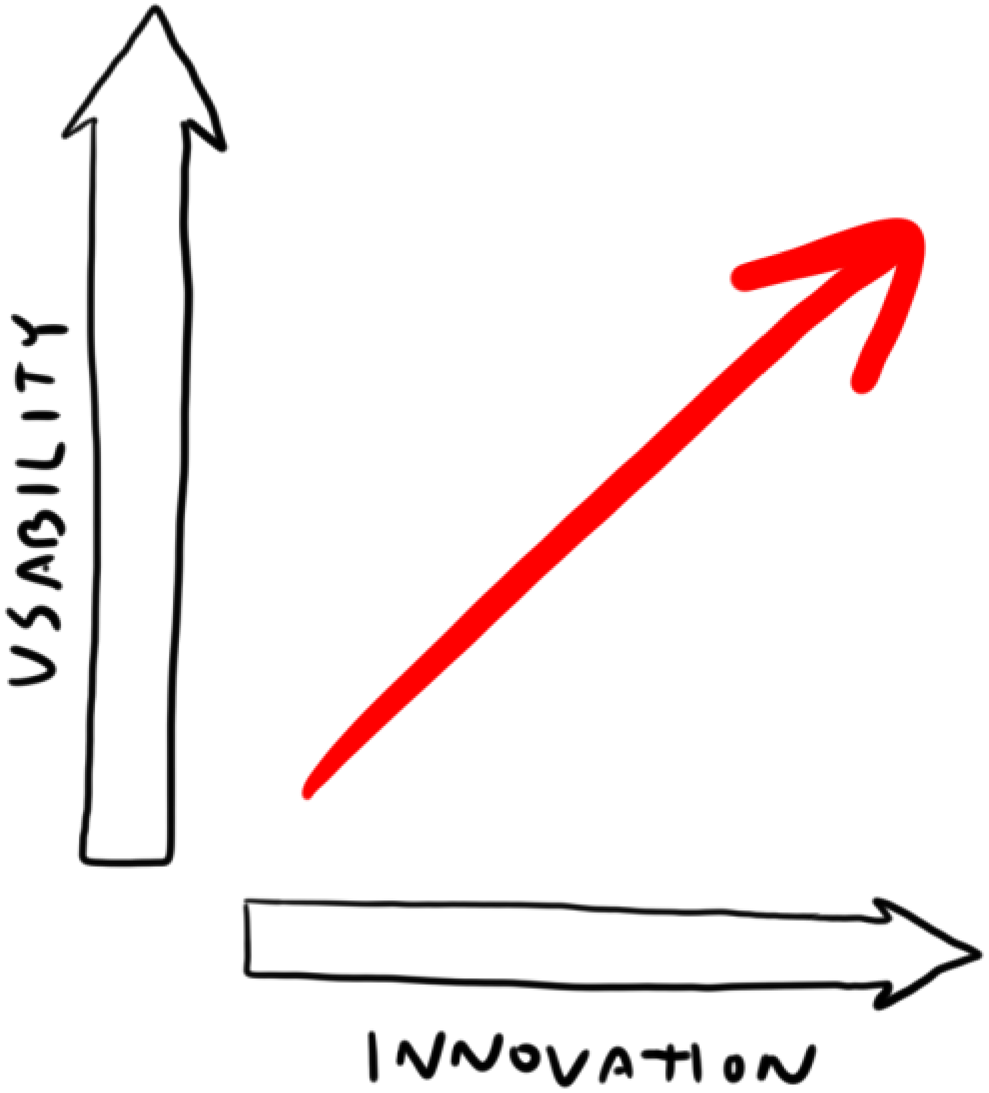
Where The Web Is Going

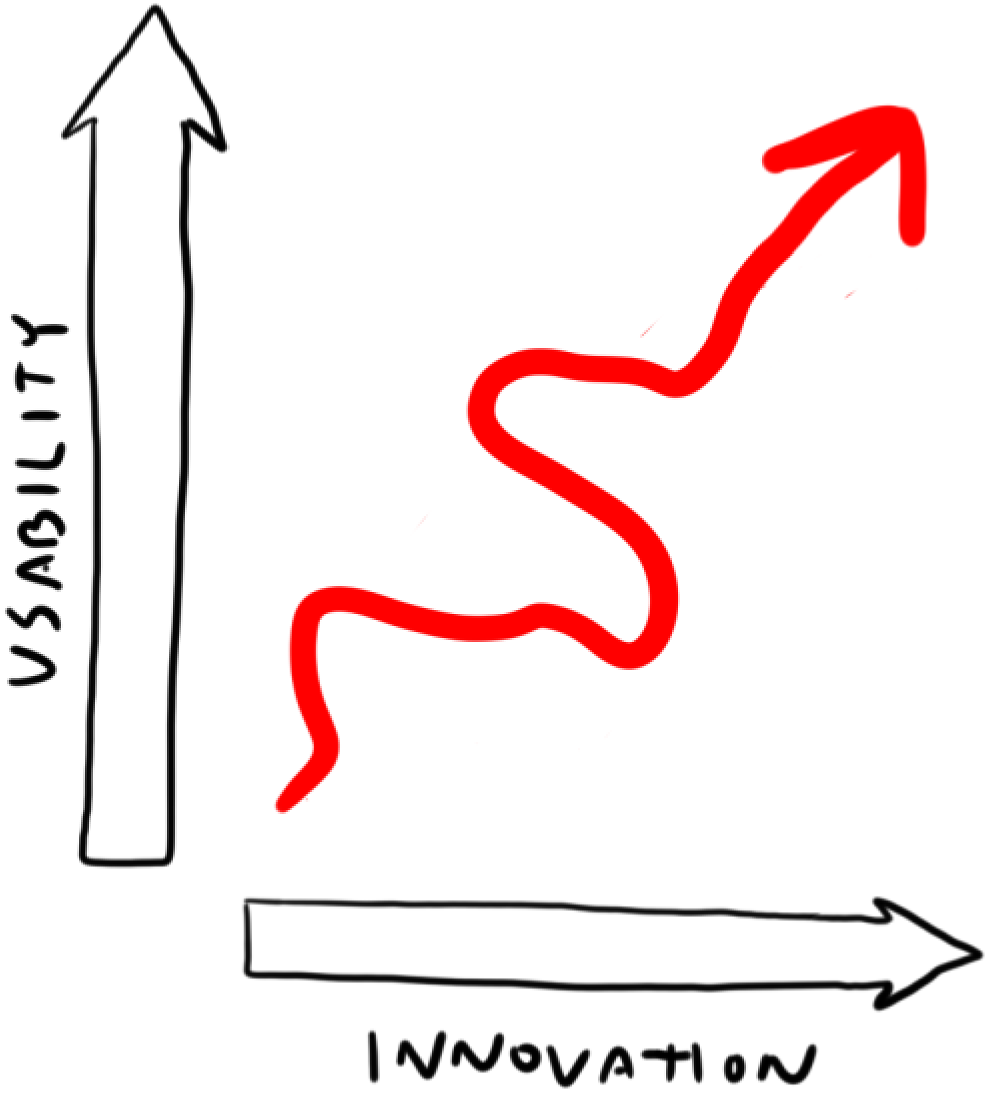
@jaredthenerd
jaredthenerd.com

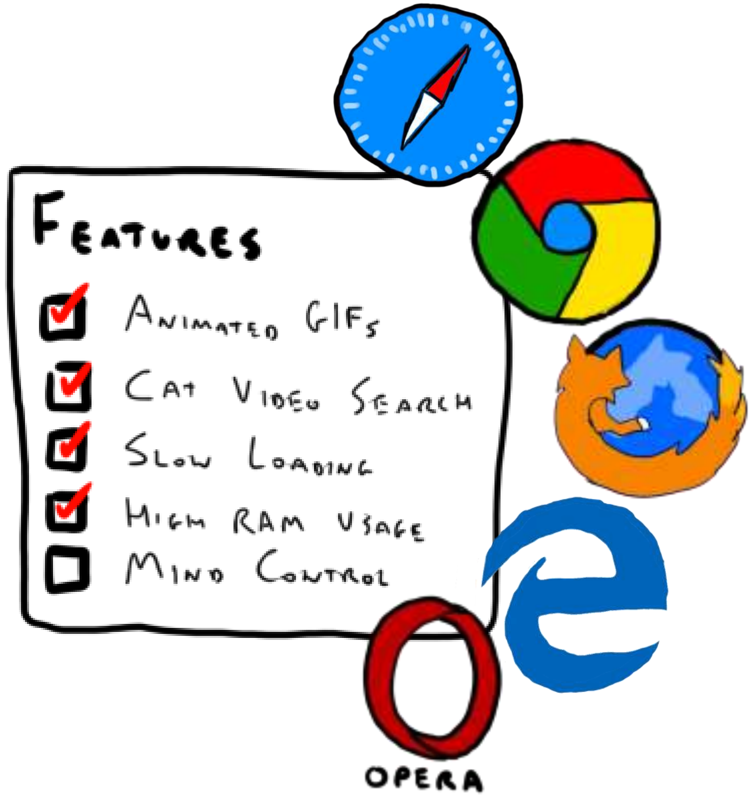
HMB


Indy.Code()
STIR TREK


FOCUS


USABILITY
INNOVATION


USABILITY
INNOVATION

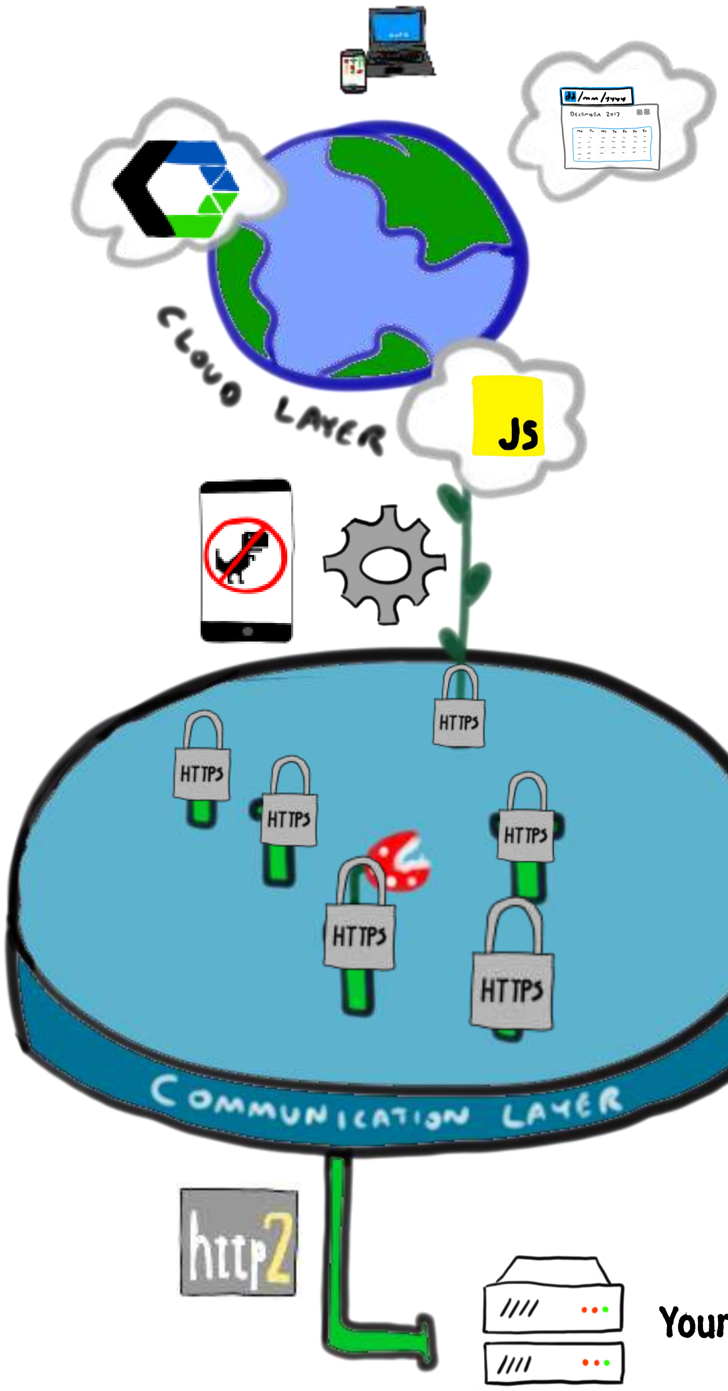


Features

- ☑ Animated GIFs
- ☑ Cat Video Search
- ☑ Slow Loading
- ☑ High RAM Usage
- □ Mind Control

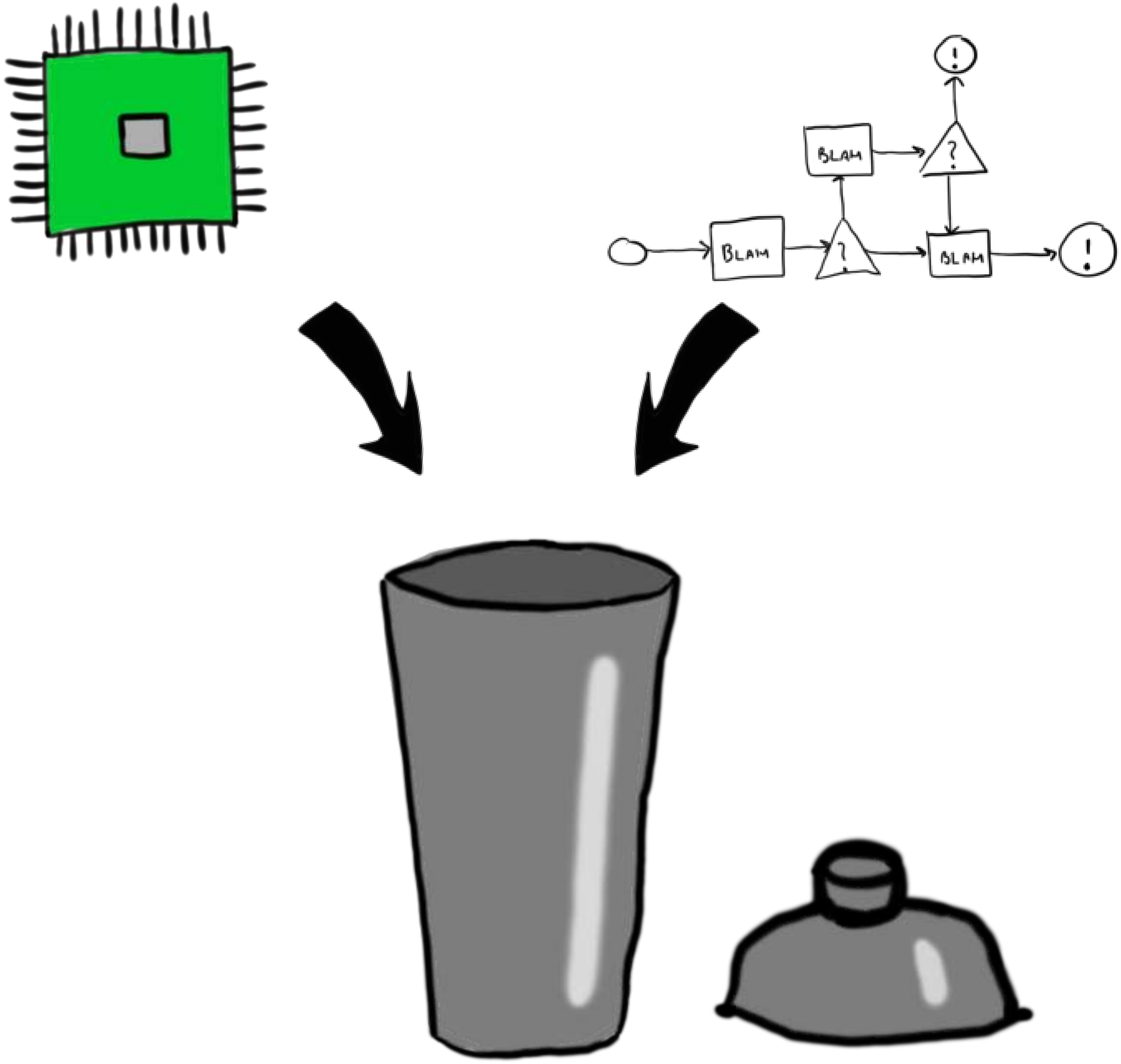
How the Web Literally* Works

and what we'll be talking about



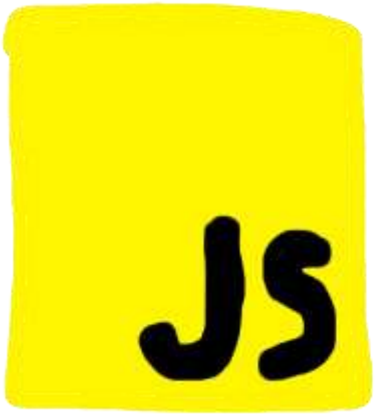
Your servers sit here

*figuratively


BLAM
?
!
BLAM
?
BLAM
!

HTML
5

CSS
3


JS

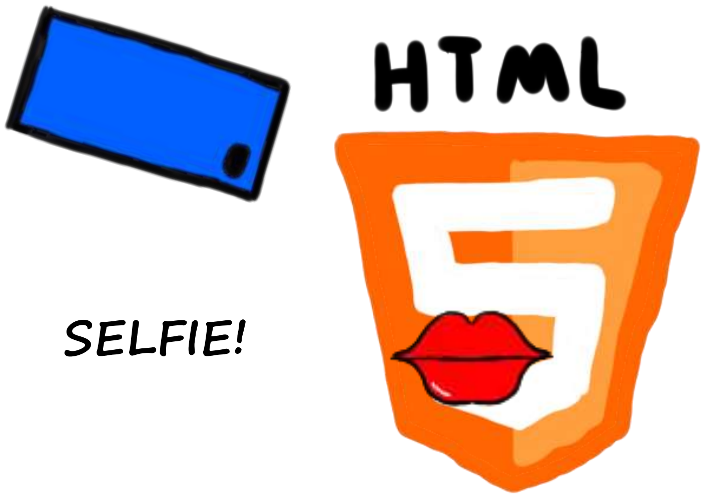
http2

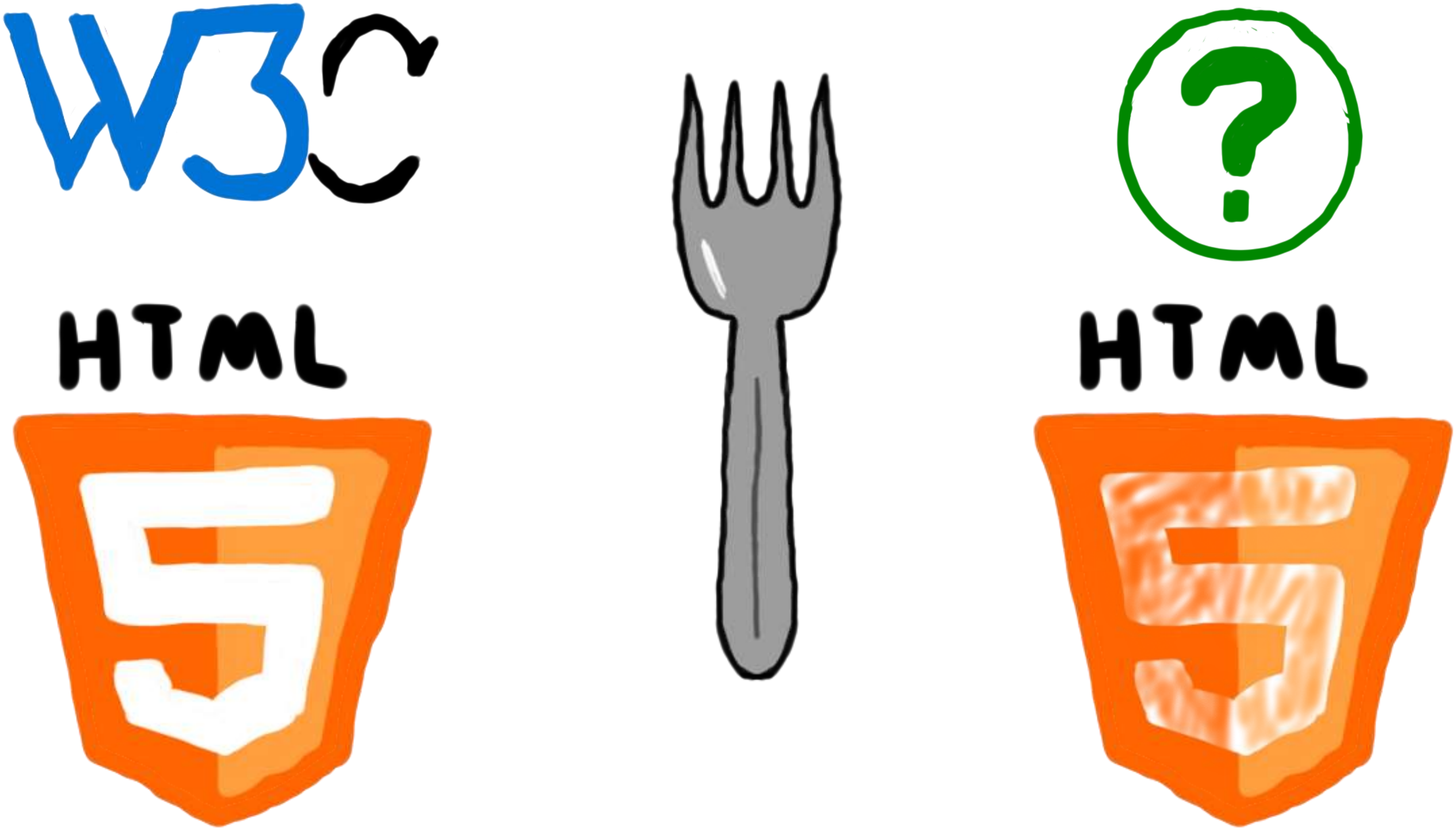
BBQLOL

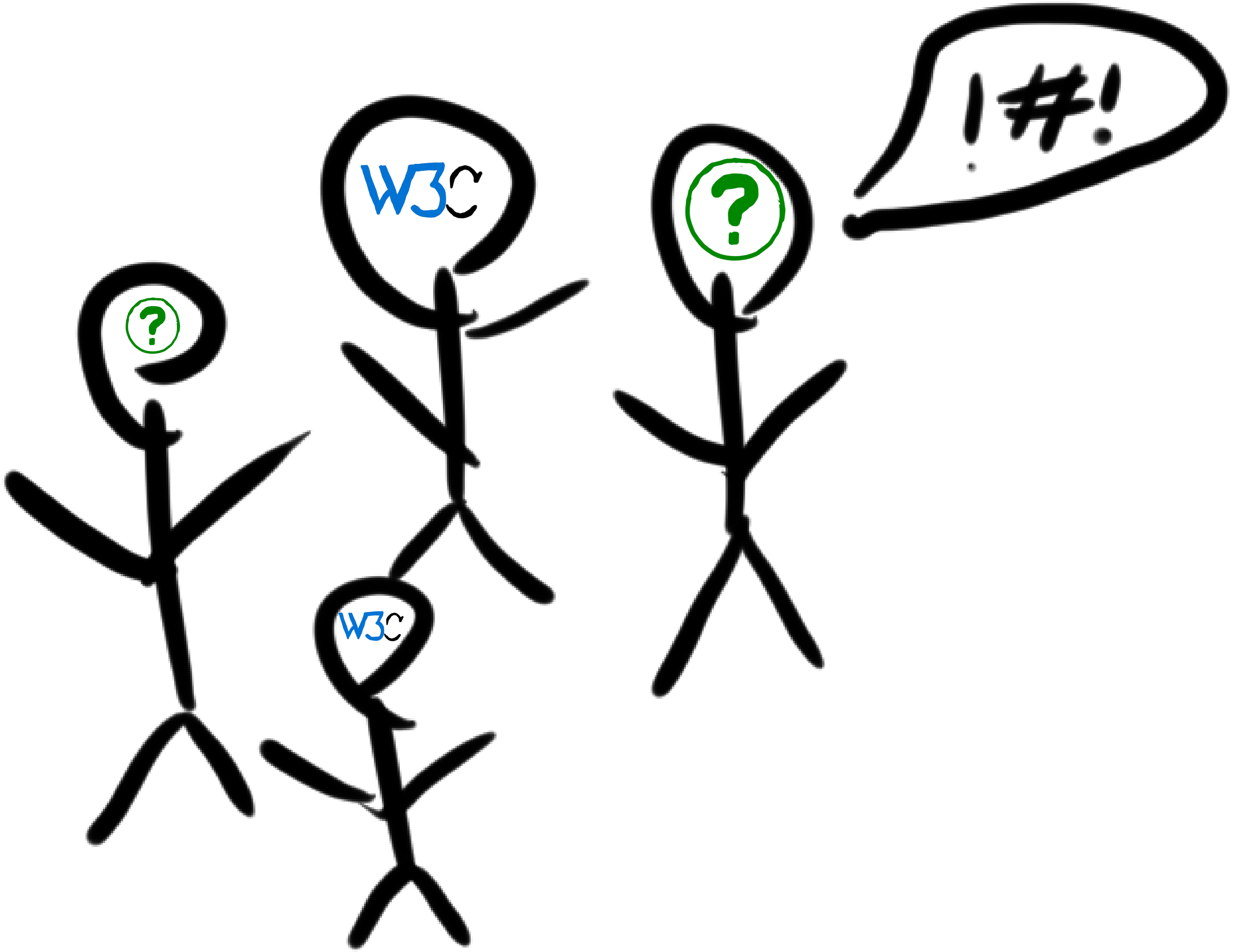
W3C
HTML
5
?

New Yorker – The Group That Rules The Web

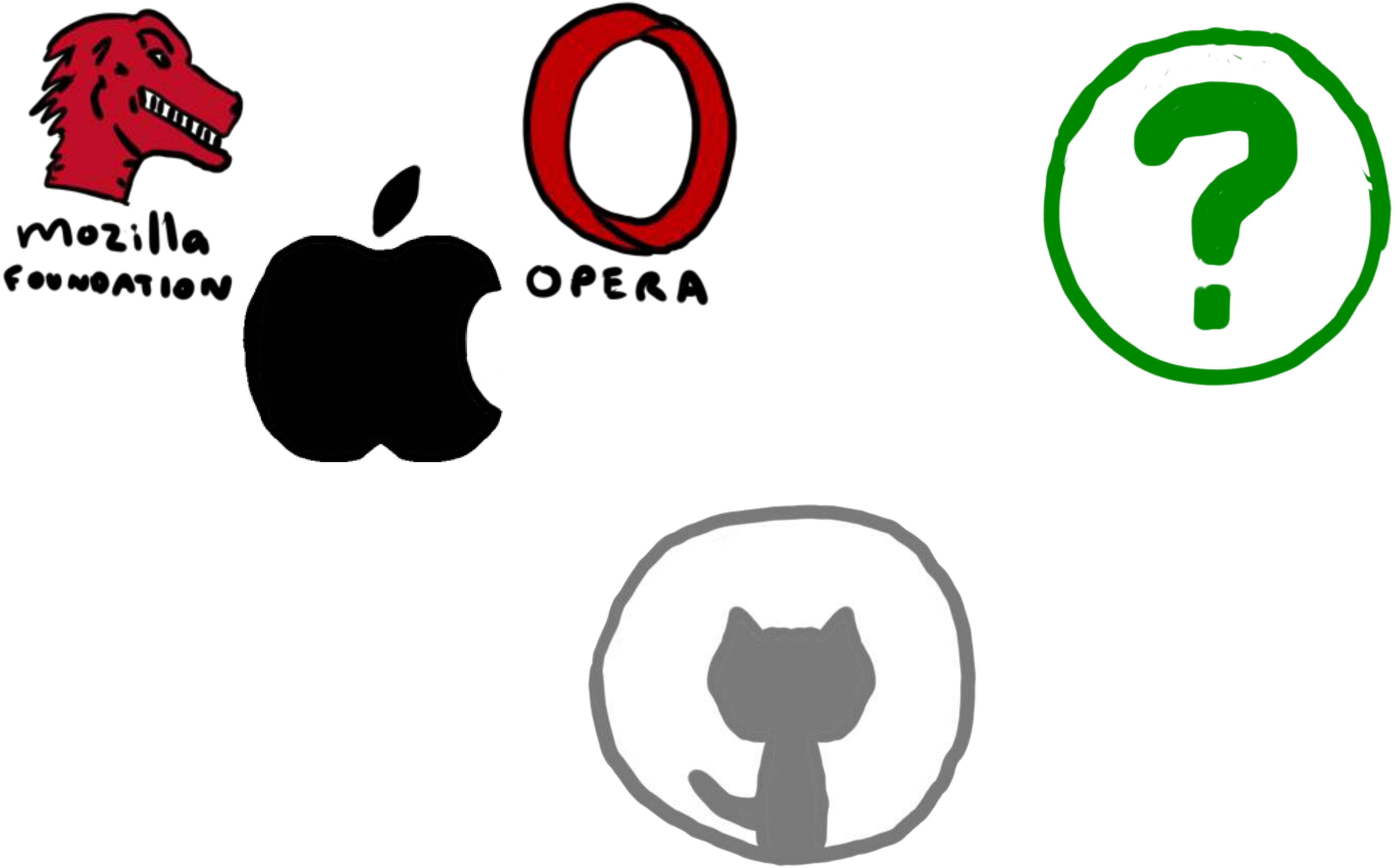
http://bit.ly/RulesTheWeb




W3C
HTML
?
HTML



Web Standards People Swear A Lot


mozilla
FOUNDATION
OPERA
?

“Consensus is not a useful value in Web spec development, because it doesn't accurately reflect the power dynamics at work.

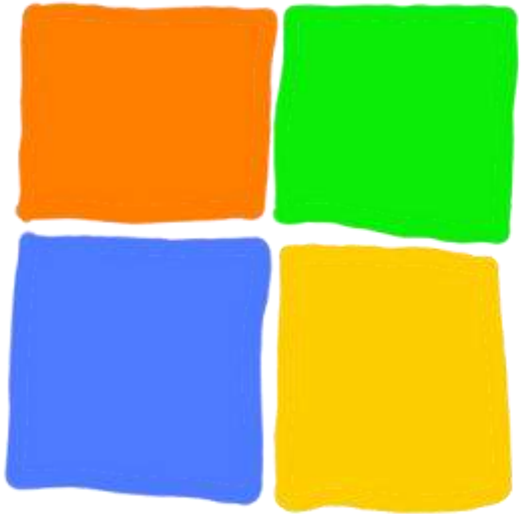
In practice, implementers have the ultimate say, not spec editors, not authors, not users, and not interested bystander standardistas.

The W3C and the WHATWG thus have no power. (The WHATWG is built around that realization; the W3C, however, is built on the assumption that it does have power.

That's how it ends up making mistakes like RDF, XForms, or XHTML2.)”

Ian Hickson – April 23, 2014

<3 OPEN STAND

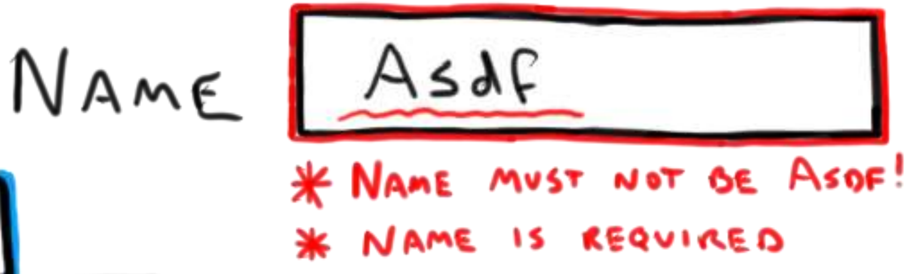


Me Too!

http://bit.ly/WHATWGIPR

HTML 5.1

Form Validation & Forced Spellcheck



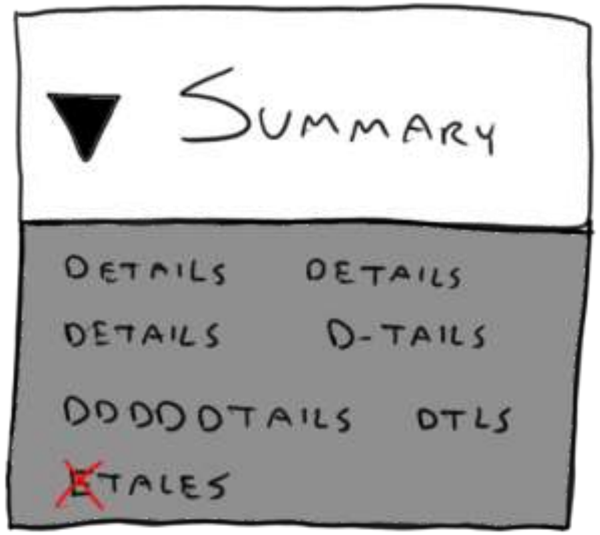
dd /mm /yyyy
DECEMBER 2017
Mo Tu We Th Fr Sa Su

Date Pickers

LAUNCH ROCKET
ABORT LAUNCH
ABORT LUNCH
BACK
FORWARD
VIEW SOURCE
RUNNING OUT
OF THINGS TO
PUT HERE

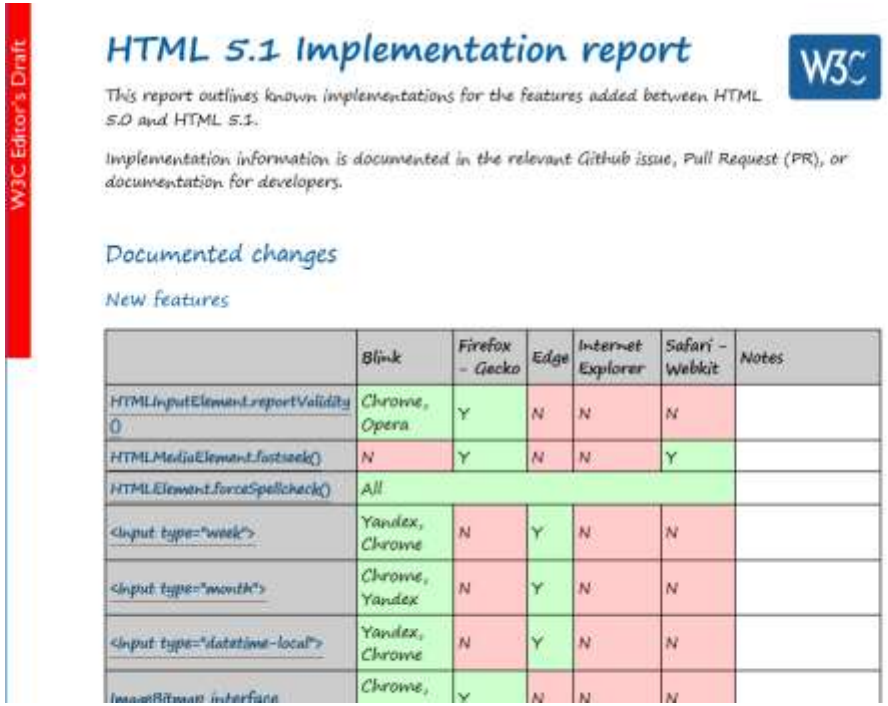
Context Menus

Expanding Detail Panels



HTML
5.1

¯_(ツ)_/¯



W3C Editor's Draft

HTML 5.1 Implementation report

W3C

This report outlines known implementations for the features added between HTML 5.0 and HTML 5.1.

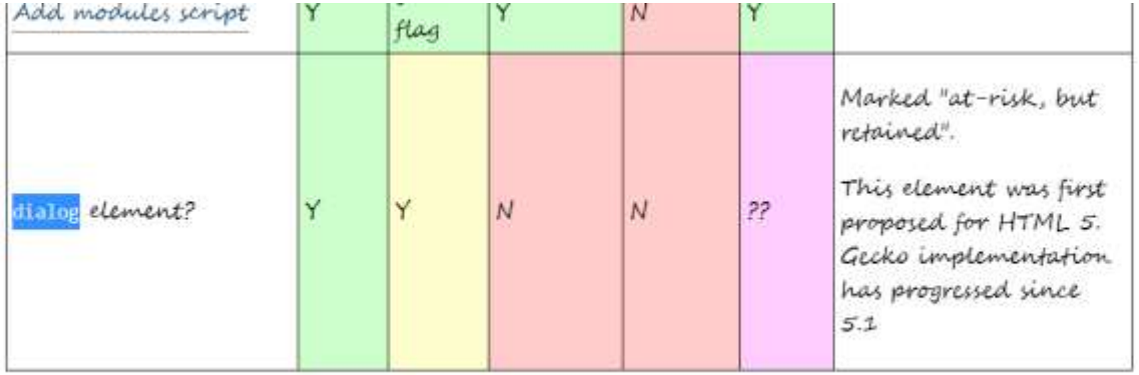
Implementation information is documented in the relevant Github issue, Pull Request (PR), or documentation for developers.

Documented changes

New features

	Blink	Firefox - Gecko	Edge	Internet Explorer	Safari - Webkit	Notes
HTMLInputElement.reportValidity()	Chrome, Opera	Y	N	N	N	
HTMLMediaElement.fastseek()	N	Y	N	N	Y	
HTMLElement.forceSpellcheck()	All					
<input type="week">	Yandex, Chrome	N	Y	N	N	
<input type="month">	Chrome, Yandex	N	Y	N	N	
<input type="datetime-local">	Yandex, Chrome	N	Y	N	N	
ImageBitmap interface	Chrome,	Y	N	N	N	

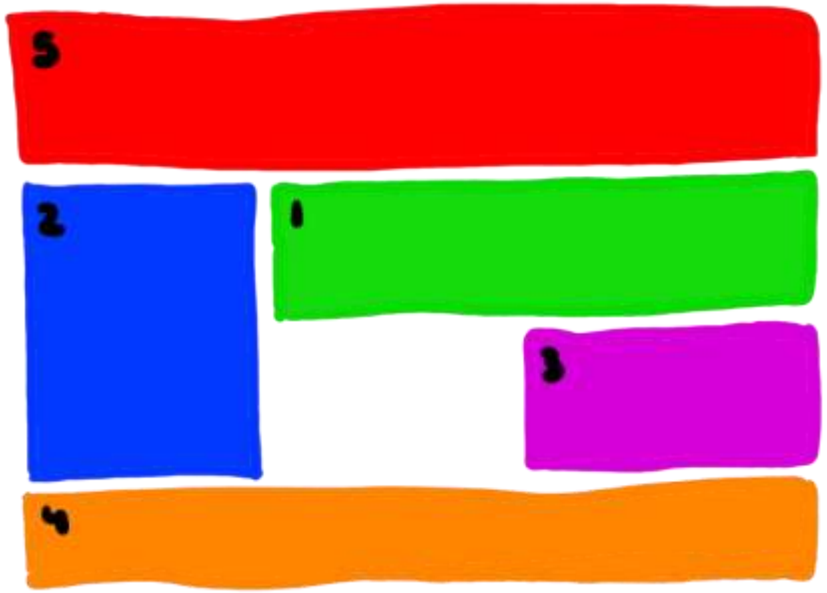
http://bit.ly/HTML5Implementation

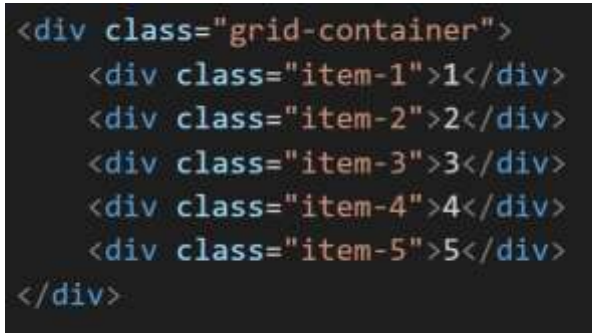


Add modules script	Y	flag	Y	N	Y	
dialog element?	Y	Y	N	N	??	Marked "at-risk, but retained". This element was first proposed for HTML 5. Gecko implementation has progressed since 5.1

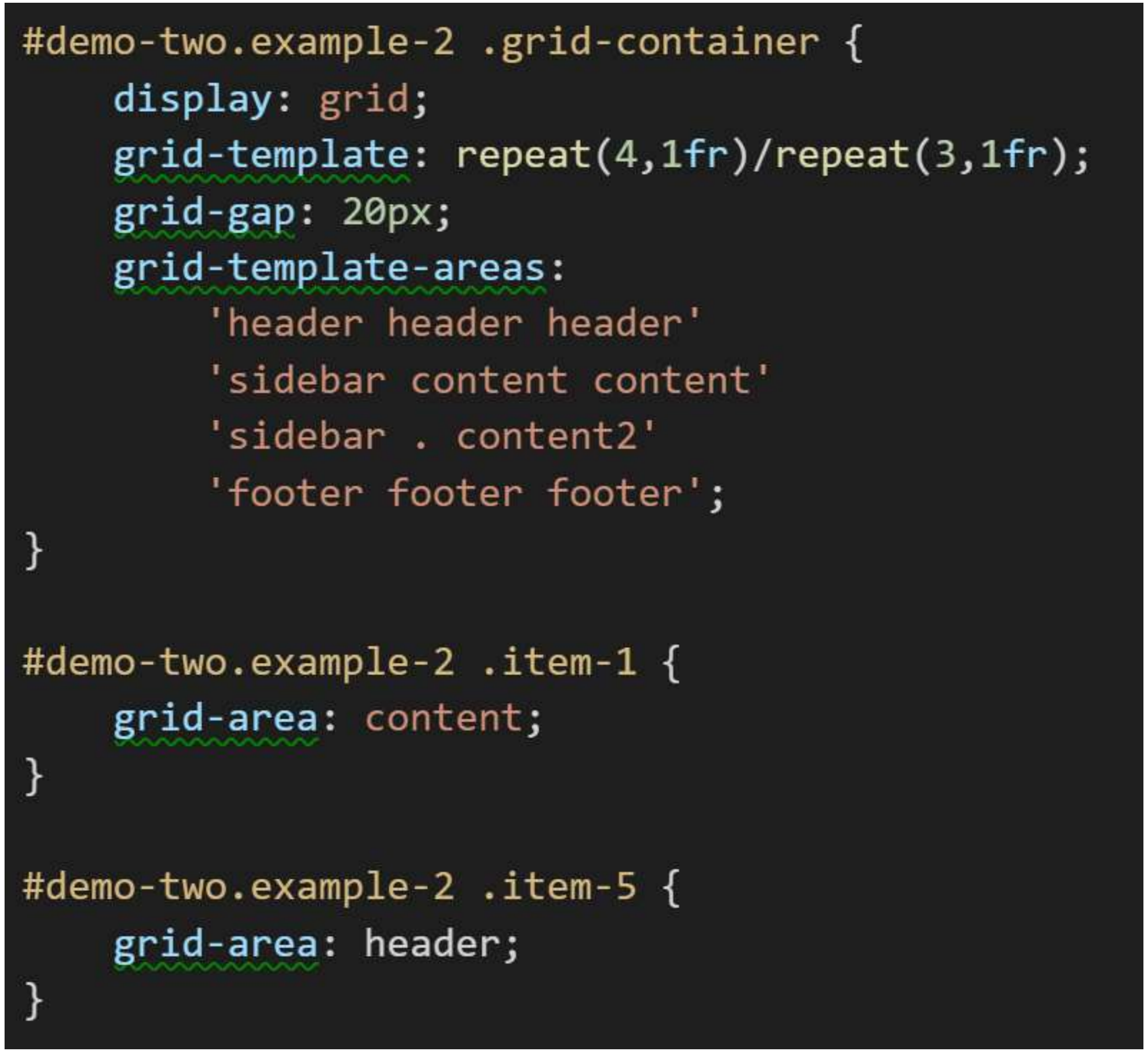
More W3C? Stuff

CSS Grid



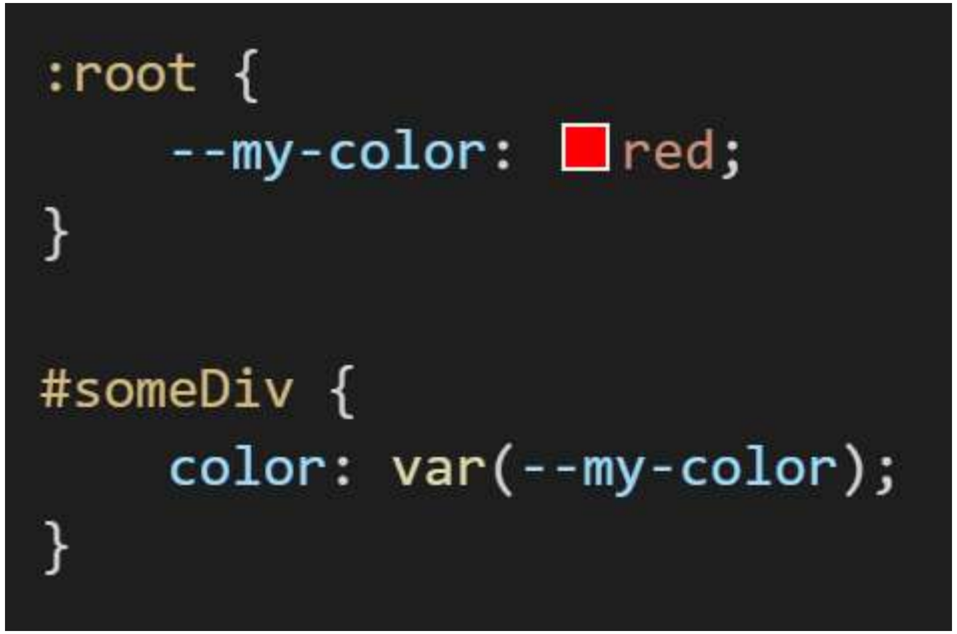


```
<div class="grid-container">
    <div class="item-1">1</div>
    <div class="item-2">2</div>
    <div class="item-3">3</div>
    <div class="item-4">4</div>
    <div class="item-5">5</div>
</div>
```



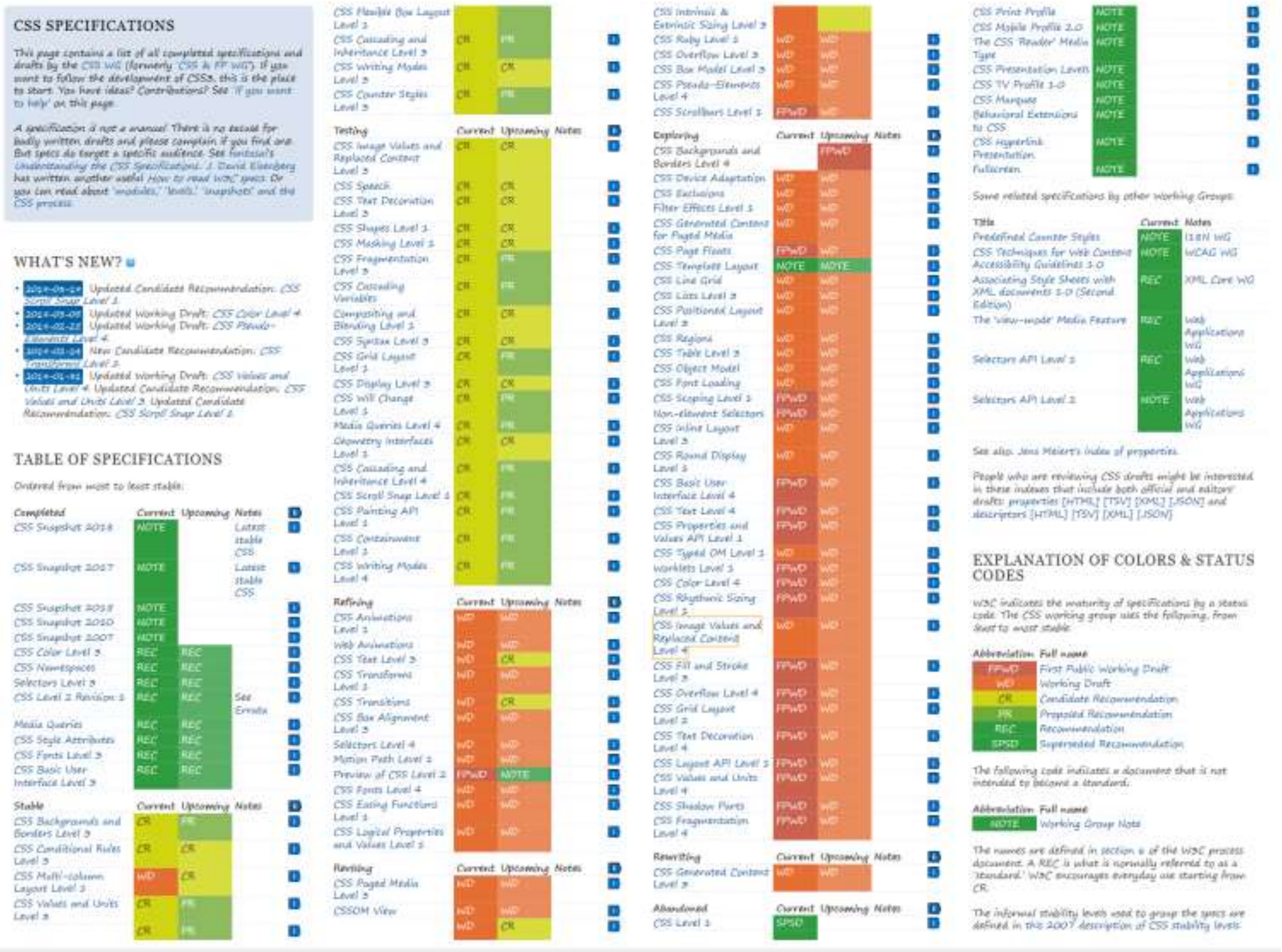
```
#demo-two.example-2 .grid-container {
    display: grid;
    grid-template: repeat(4,1fr)/repeat(3,1fr);
    grid-gap: 20px;
    grid-template-areas:
        'header header header'
        'sidebar content content'
        'sidebar . content2'
        'footer footer footer';
}

#demo-two.example-2 .item-1 {
    grid-area: content;
}

#demo-two.example-2 .item-5 {
    grid-area: header;
}
```

CSS Variables



```
:root {
    --my-color: 🟥red;
}

#someDiv {
    color: var(--my-color);
}
```

UMMMMMMMM… WHAT?



CSS SPECIFICATIONS

This page contains a list of all completed specifications and drafts by the CSS WG (formerly 'CSS & FP WG'). If you want to follow the development of CSS3, this is the place to start. You have ideas? Contributions? See 'If you want to help' on this page.

A specification is not a manual. There is no excuse for badly written drafts and please complain if you find one. But specs do target a specific audience. See fantasai's Understanding the CSS Specifications. J. David Eisenberg has written another useful How to read W3C specs. Or you can read about 'modules,' 'levels,' 'snapshots' and the CSS process.

WHAT'S NEW?

- 2019-03-19 Updated Candidate Recommendation: CSS Scroll Snap Level 1
- 2019-03-05 Updated Working Draft: CSS Color Level 4
- 2019-02-25 Updated Working Draft: CSS Pseudo-Elements Level 4
- 2019-02-14 New Candidate Recommendation: CSS Transforms Level 1
- 2019-01-31 Updated Working Draft: CSS Values and Units Level 4. Updated Candidate Recommendation: CSS Values and Units Level 3. Updated Candidate Recommendation: CSS Scroll Snap Level 1

TABLE OF SPECIFICATIONS

Ordered from most to least stable:

Completed	Current	Upcoming	Notes
CSS Snapshot 2018	NOTE		Latest stable CSS
CSS Snapshot 2017	NOTE		Latest stable CSS
CSS Snapshot 2015	NOTE		
CSS Snapshot 2010	NOTE		
CSS Snapshot 2007	NOTE		
CSS Color Level 3	REC	REC	
CSS Namespaces	REC	REC	
Selectors Level 3	REC	REC	
CSS Level 2 Revision 1	REC	REC	See Errata
Media Queries	REC	REC	
CSS Style Attributes	REC	REC	
CSS Fonts Level 3	REC	REC	
CSS Basic User Interface Level 3	REC	REC	

Stable	Current	Upcoming	Notes
CSS Backgrounds and Borders Level 3	CR	PR	
CSS Conditional Rules Level 3	CR	CR	
CSS Multi-column Layout Level 1	WD	CR	
CSS Values and Units Level 3	CR	PR	
	CR	PR	
CSS Flexible Box Layout Level 1			
CSS Cascading and Inheritance Level 3	CR	PR	
CSS Writing Modes Level 3	CR	CR	
CSS Counter Styles Level 3	CR	PR	

Testing	Current	Upcoming	Notes
CSS Image Values and Replaced Content Level 3	CR	CR	
CSS Speech	CR	CR	
CSS Text Decoration Level 3	CR	CR	
CSS Shapes Level 1	CR	CR	
CSS Masking Level 1	CR	CR	
CSS Fragmentation Level 3	CR	PR	
CSS Cascading Variables	CR	PR	
Compositing and Blending Level 1	CR	CR	
CSS Syntax Level 3	CR	CR	
CSS Grid Layout Level 1	CR	PR	
CSS Display Level 3	CR	CR	
CSS Will Change Level 1	CR	PR	
Media Queries Level 4	CR	PR	
Geometry Interfaces Level 1	CR	CR	
CSS Cascading and Inheritance Level 4	CR	PR	
CSS Scroll Snap Level 1	CR	PR	
CSS Painting API Level 1	CR	PR	
CSS Containment Level 1	CR	PR	
CSS Writing Modes Level 4	CR	PR	

Refining	Current	Upcoming	Notes
CSS Animations Level 1	WD	WD	
Web Animations	WD	WD	
CSS Text Level 3	WD	CR	
CSS Transforms Level 1	WD	WD	
CSS Transitions	WD	CR	
CSS Box Alignment Level 3	WD	WD	
Selectors Level 4	WD	WD	
Motion Path Level 1	WD	WD	
Preview of CSS Level 2	FPWD	NOTE	
CSS Fonts Level 4	WD	WD	
CSS Easing Functions Level 1	WD	WD	
CSS Logical Properties and Values Level 1	WD	WD	

Revising	Current	Upcoming	Notes
CSS Paged Media Level 3	WD	WD	
CSSOM View	WD	WD	
	WD	CR	
CSS Intrinsic & Extrinsic Sizing Level 3			
CSS Ruby Level 1	WD	WD	
CSS Overflow Level 3	WD	WD	
CSS Box Model Level 3	WD	WD	
CSS Pseudo-Elements Level 4	WD	WD	
CSS Scrollbars Level 1	FPWD	WD	

Exploring	Current	Upcoming	Notes
CSS Backgrounds and Borders Level 4		FPWD	
CSS Device Adaptation	WD	WD	
CSS Exclusions	WD	WD	
Filter Effects Level 1	WD	WD	
CSS Generated Content for Paged Media	WD	WD	
CSS Page Floats	FPWD	WD	
CSS Template Layout	NOTE	NOTE	
CSS Line Grid	WD	WD	
CSS Lists Level 3	WD	WD	
CSS Positioned Layout Level 3	WD	WD	
CSS Regions	WD	WD	
CSS Table Level 3	WD	WD	
CSS Object Model	WD	WD	
CSS Font Loading	WD	WD	
CSS Scoping Level 1	FPWD	WD	
Non-element Selectors	FPWD	WD	
CSS Inline Layout Level 3	WD	WD	
CSS Round Display Level 1	WD	WD	
CSS Basic User Interface Level 4	FPWD	WD	
CSS Text Level 4	FPWD	WD	
CSS Properties and Values API Level 1	FPWD	WD	
CSS Typed OM Level 1	WD	WD	
Worklets Level 1	FPWD	WD	
CSS Color Level 4	FPWD	WD	
CSS Rhythmic Sizing Level 1	FPWD	WD	
CSS Image Values and Replaced Content Level 4	WD	WD	
CSS Fill and Stroke Level 3	FPWD	WD	
CSS Overflow Level 4	FPWD	WD	
CSS Grid Layout Level 2	FPWD	WD	
CSS Text Decoration Level 4	FPWD	WD	
CSS Layout API Level 1	FPWD	WD	
CSS Values and Units Level 4	FPWD	WD	
CSS Shadow Parts	FPWD	WD	
CSS Fragmentation Level 4	FPWD	WD	

Rewriting	Current	Upcoming	Notes
CSS Generated Content Level 3	WD	WD	

Abandoned	Current	Upcoming	Notes
CSS Level 1	SPSD		
CSS Print Profile	NOTE		
CSS Mobile Profile 2.0	NOTE		
The CSS 'Reader' Media Type	NOTE		
CSS Presentation Levels	NOTE		
CSS TV Profile 1.0	NOTE		
CSS Marquee	NOTE		
Behavioral Extensions to CSS	NOTE		
CSS Hyperlink Presentation	NOTE		
Fullscreen	NOTE		

Some related specifications by other Working Groups:

Title	Current	Notes
Predefined Counter Styles	NOTE	I18N WG
CSS Techniques for Web Content Accessibility Guidelines 1.0	NOTE	WCAG WG
Associating Style Sheets with XML documents 1.0 (Second Edition)	REC	XML Core WG
The 'view-mode' Media Feature	REC	Web Applications WG
Selectors API Level 1	REC	Web Applications WG
Selectors API Level 2	NOTE	Web Applications WG

See also: Jens Meiert's index of properties.

People who are reviewing CSS drafts might be interested in these indexes that include both official and editors' drafts: properties [HTML] [TSV] [XML] [JSON] and descriptors [HTML] [TSV] [XML] [JSON]

EXPLANATION OF COLORS & STATUS CODES

W3C indicates the maturity of specifications by a status code. The CSS working group uses the following, from least to most stable:

Abbreviation	Full name
FPWD	First Public Working Draft
WD	Working Draft
CR	Candidate Recommendation
PR	Proposed Recommendation
REC	Recommendation
SPSD	Superseded Recommendation

The following code indicates a document that is not intended to become a standard:

Abbreviation	Full name
NOTE	Working Group Note

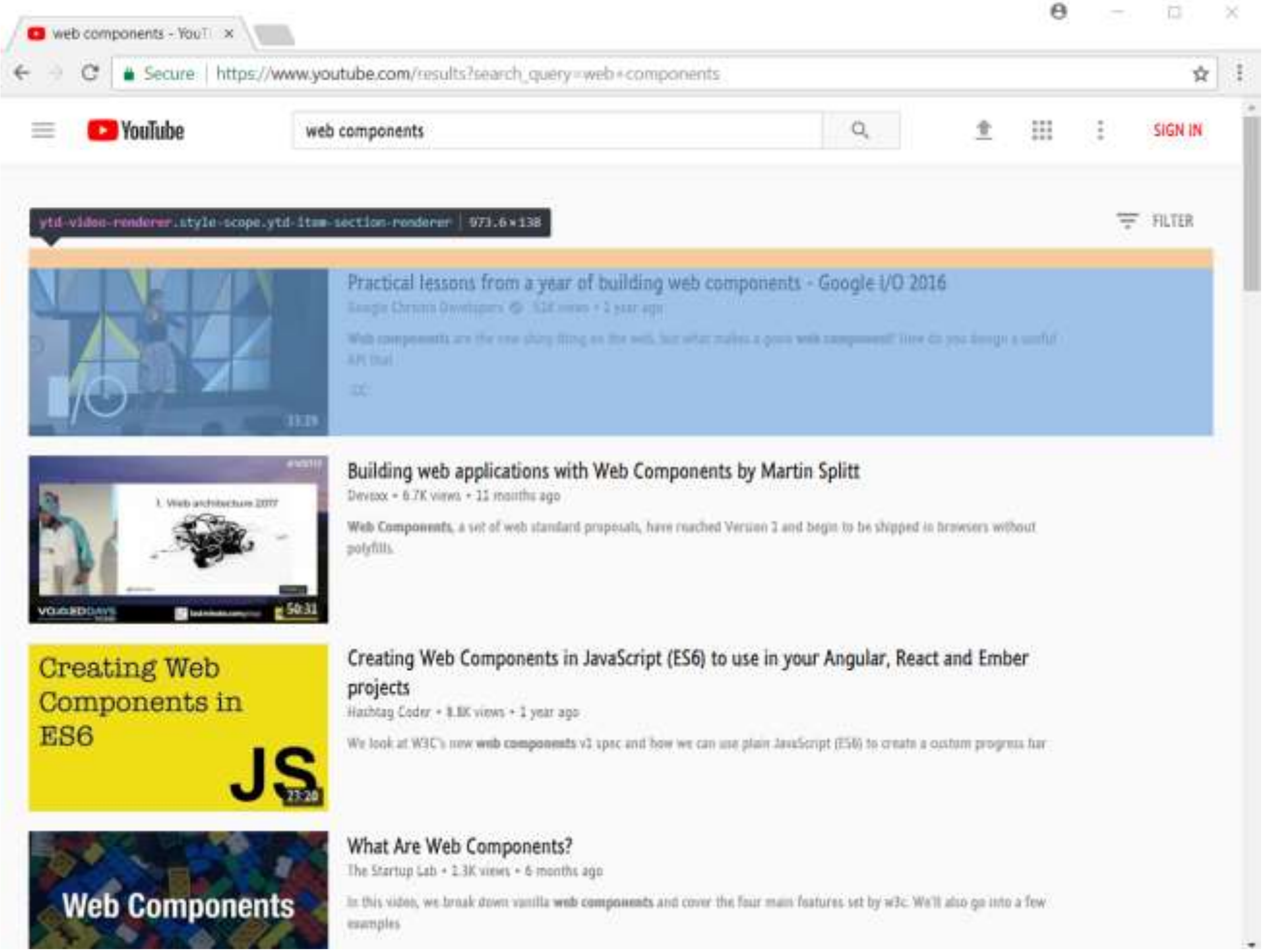
The names are defined in section 6 of the W3C process document. A REC is what is normally referred to as a 'standard.' W3C encourages everyday use starting from CR.

The informal stability levels used to group the specs are defined in this 2007 description of CSS stability levels.

Web Components

- HTML Templates
- Shadow DOM
- Custom Elements
- ~~HTML Imports~~/ES6 Modules

Web Components In Action

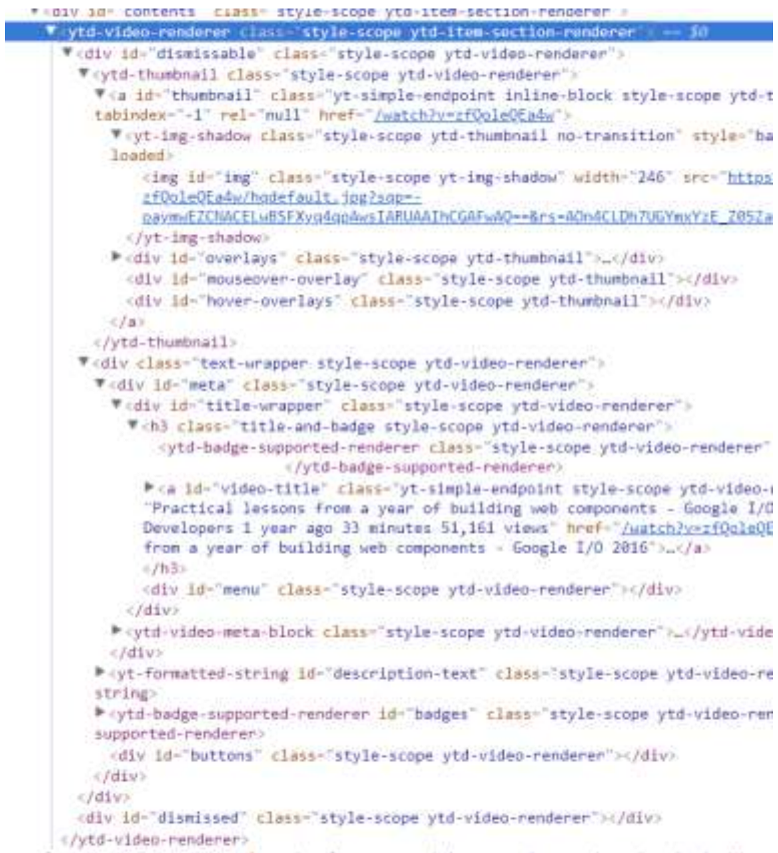

web components - YouT
Secure
https://www.youtube.com/results?search_query=web+components
YouTube
web components
SIGN IN
ytd-video-renderer.style-scope.ytd-item-section-renderer | 973.6 × 138
FILTER
13:29
50:31
Creating Web Components in ES6
JS
23:20
Web Components

Practical lessons from a year of building web components - Google I/O 2016
Google Chrome Developers · 51K views · 1 year ago
Web components are the new shiny thing on the web, but what makes a good web component? How do you design a useful API that
CC

Building web applications with Web Components by Martin Splitt
Devoxx • 6.7K views • 11 months ago
Web Components, a set of web standard proposals, have reached Version 1 and begin to be shipped in browsers without polyfills.

Creating Web Components in JavaScript (ES6) to use in your Angular, React and Ember projects
Hashtag Coder • 8.8K views • 1 year ago
We look at W3C's new web components v1 spec and how we can use plain JavaScript (ES6) to create a custom progress bar

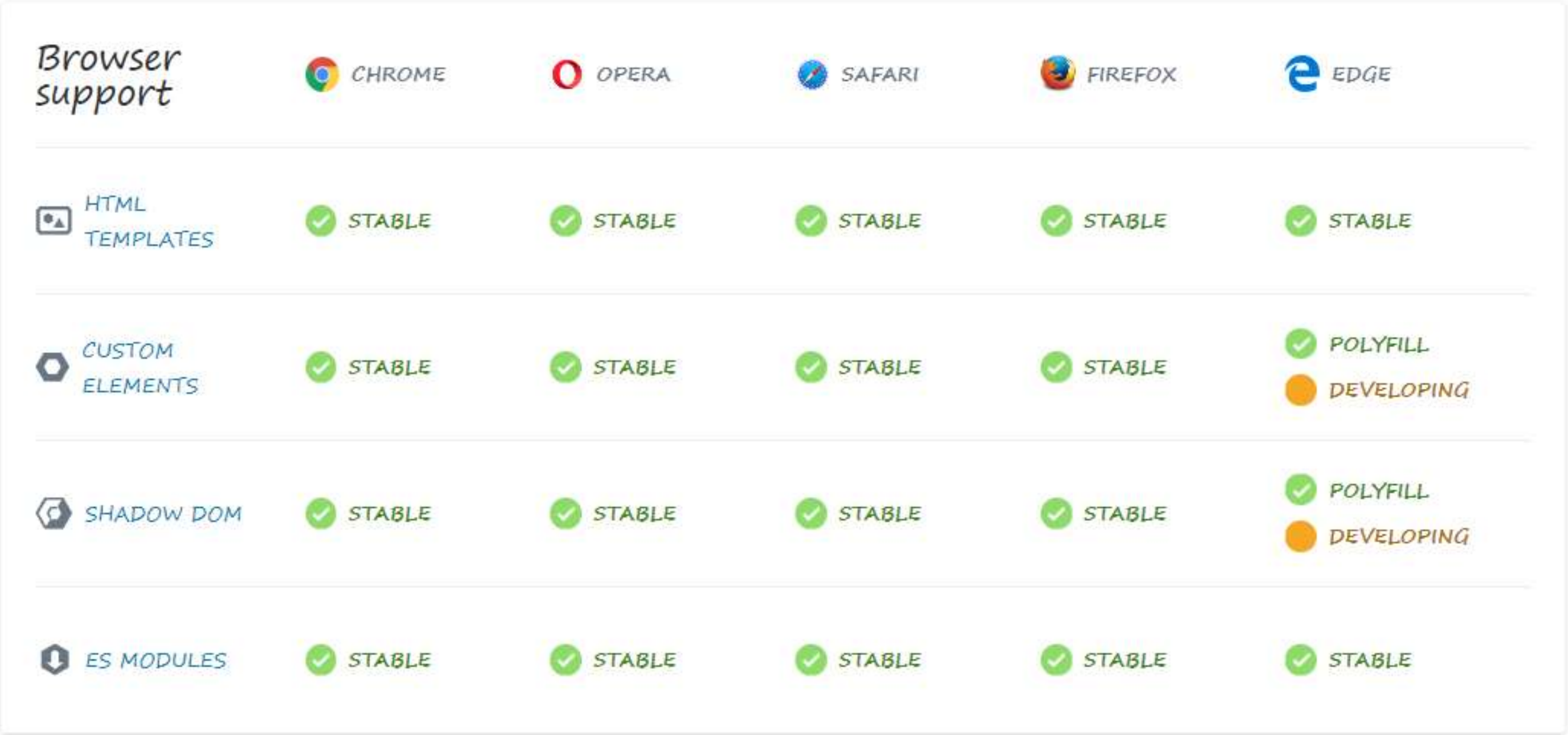
What Are Web Components?
The Startup Lab • 2.3K views • 6 months ago
In this video, we break down vanilla web components and cover the four main features set by w3c. We'll also go into a few examples



```
▼<div id="contents" class="style-scope ytd-item-section-renderer">
  ▼<ytd-video-renderer class="style-scope ytd-item-section-renderer"> == $0
    ▼<div id="dismissable" class="style-scope ytd-video-renderer">
      ▼<ytd-thumbnail class="style-scope ytd-video-renderer">
        ▼<a id="thumbnail" class="yt-simple-endpoint inline-block style-scope ytd-t
        tabindex="-1" rel="null" href="/watch?v=zfQoleQEa4w">
          ▼<yt-img-shadow class="style-scope ytd-thumbnail no-transition" style="ba
          loaded>
              <img id="img" class="style-scope yt-img-shadow" width="246" src="https
              zfQoleQEa4w/hqdefault.jpg?sqp=-
              oaymwEZCNACELwBSFXyq4qpAwsIARUAAIhCGAFwAQ==&rs=AOn4CLDh7UGYmxYzE_Z05Za
            </yt-img-shadow>
          ▶<div id="overlays" class="style-scope ytd-thumbnail">…</div>
            <div id="mouseover-overlay" class="style-scope ytd-thumbnail"></div>
            <div id="hover-overlays" class="style-scope ytd-thumbnail"></div>
          </a>
        </ytd-thumbnail>
      ▼<div class="text-wrapper style-scope ytd-video-renderer">
        ▼<div id="meta" class="style-scope ytd-video-renderer">
          ▼<div id="title-wrapper" class="style-scope ytd-video-renderer">
            ▼<h3 class="title-and-badge style-scope ytd-video-renderer">
                <ytd-badge-supported-renderer class="style-scope ytd-video-renderer"
                                </ytd-badge-supported-renderer>
              ▶<a id="video-title" class="yt-simple-endpoint style-scope ytd-video-
              "Practical lessons from a year of building web components - Google I/O
              Developers 1 year ago 33 minutes 51,161 views" href="/watch?v=zfQoleQE
              from a year of building web components - Google I/O 2016">…</a>
              </h3>
              <div id="menu" class="style-scope ytd-video-renderer"></div>
            </div>
          ▶<ytd-video-meta-block class="style-scope ytd-video-renderer">…</ytd-vide
          </div>
        ▶<yt-formatted-string id="description-text" class="style-scope ytd-video-re
        string>
        ▶<ytd-badge-supported-renderer id="badges" class="style-scope ytd-video-ren
        supported-renderer>
          <div id="buttons" class="style-scope ytd-video-renderer"></div>
        </div>
      </div>
      <div id="dismissed" class="style-scope ytd-video-renderer"></div>
    </ytd-video-renderer>
```

These get includes with Polymer and Shady DOM. That is its own talk!

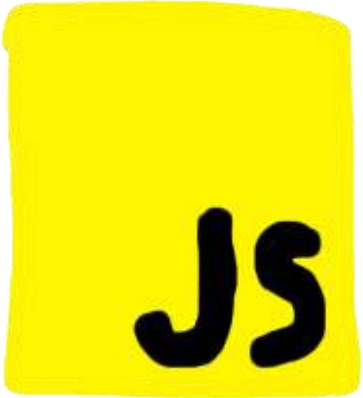
HTML

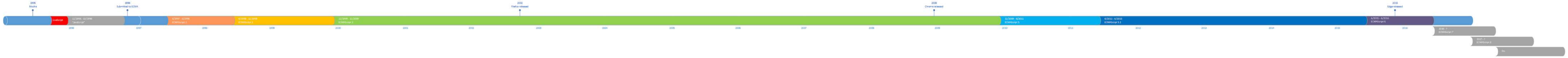


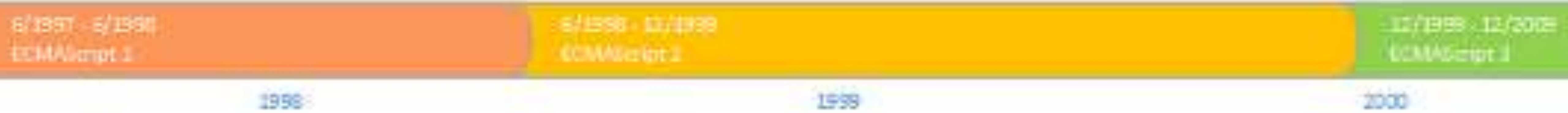
Browser support	CHROME	OPERA	SAFARI	FIREFOX	EDGE
HTML TEMPLATES	✓ STABLE	✓ STABLE	✓ STABLE	✓ STABLE	✓ STABLE
CUSTOM ELEMENTS	✓ STABLE	✓ STABLE	✓ STABLE	✓ STABLE	✓ POLYFILL ● DEVELOPING
SHADOW DOM	✓ STABLE	✓ STABLE	✓ STABLE	✓ STABLE	✓ POLYFILL ● DEVELOPING
ES MODULES	✓ STABLE	✓ STABLE	✓ STABLE	✓ STABLE	✓ STABLE

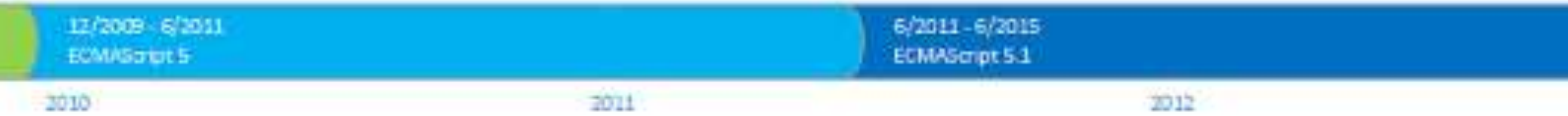
https://www.webcomponents.org/

"ECMAScript was always an unwanted trade name that sounds like a skin disease." – Brendan Eich



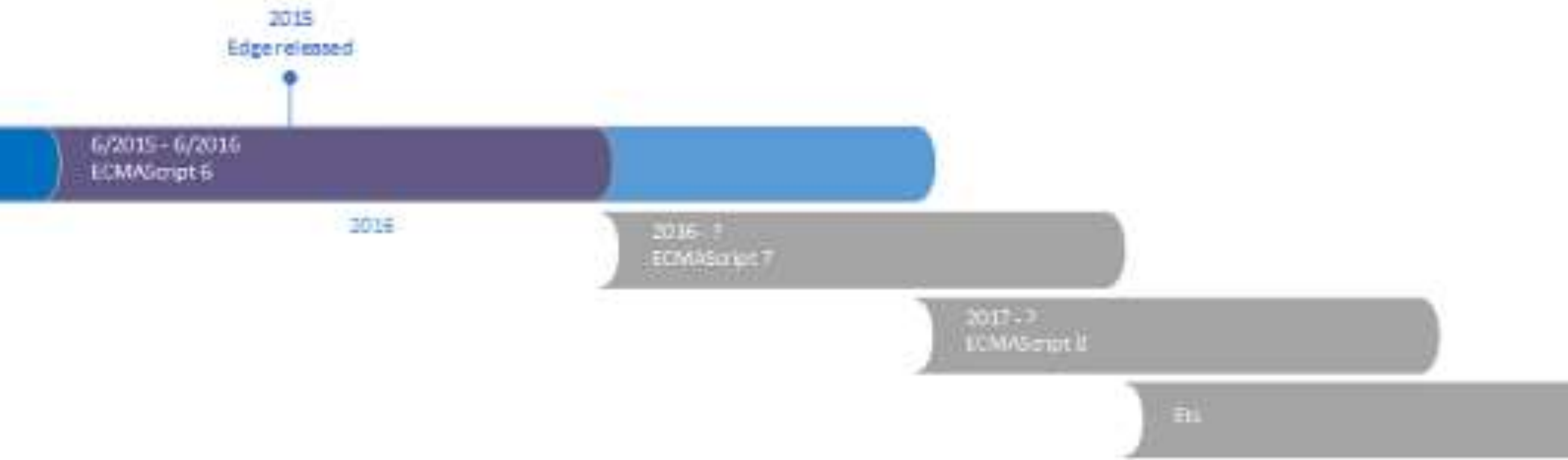

1995
Mocha
1996
Submitted to ECMA
2002
Firefox released
2008
Chrome released
2015
Edge released
LiveScript
12/1995 - 10/1996
"JavaScript"
6/1997 - 6/1998
ECMAScript 1
6/1998 - 12/1999
ECMAScript 2
12/1999 - 12/2009
ECMAScript 3
12/2009 - 6/2011
ECMAScript 5
6/2011 - 6/2015
ECMAScript 5.1
6/2015 - 6/2016
ECMAScript 6
2016 - ?
ECMAScript 7
2017 - ?
ECMAScript 8
Etc.
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016


6/1997 - 6/1998
ECMAScript 1
6/1998 - 12/1999
ECMAScript 2
12/1999 - 12/2009
ECMAScript 3
1998
1999
2000



12/2009 - 6/2011
ECMAScript 5

6/2011 - 6/2015
ECMAScript 5.1



2015
Edge released

6/2015 - 6/2016
ECMAScript 6

2016

2016 - ?
ECMAScript 7

2017 - ?
ECMAScript 8

Etc

ES 2016

Array.includes

** operator

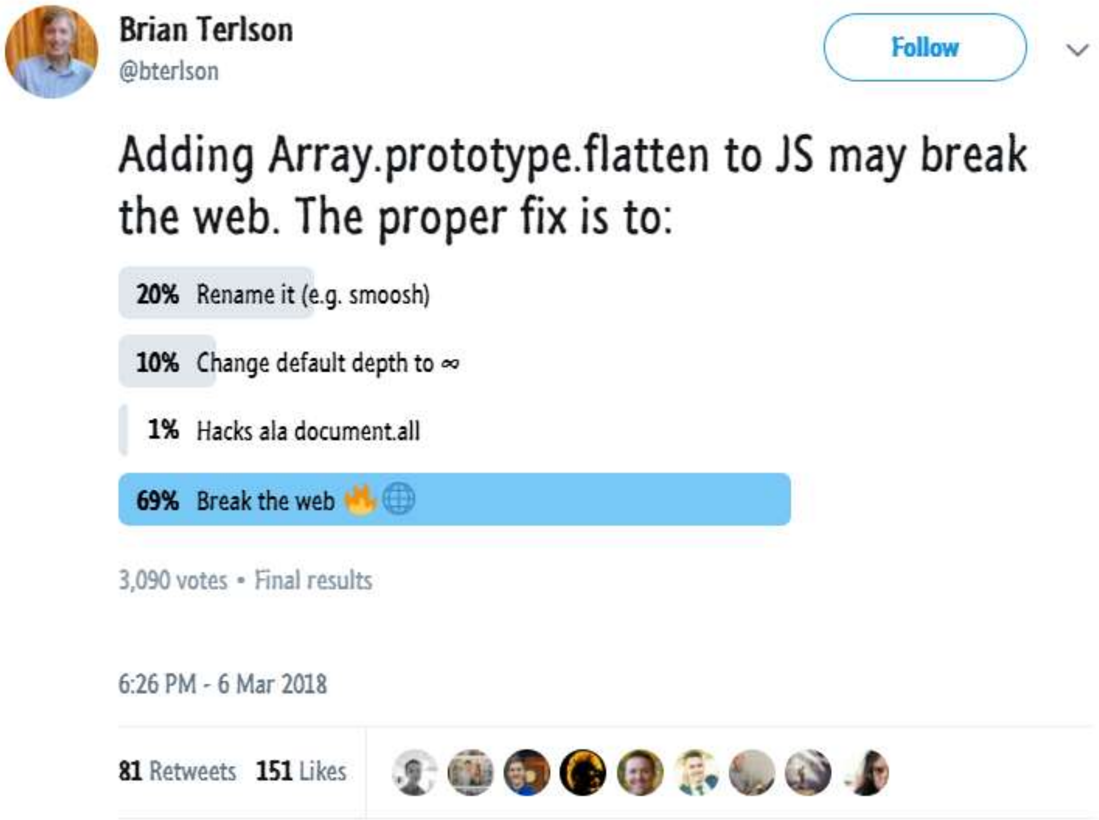
Array.contains ???

“You're telling me I should ship a browser that chokes on thousands of web sites that work fine today. That would be bad for our users, so I'm not planning on doing that."

"Patching 3.5 million websites is not a 'small fix' in any relevant sense.“

– Jason Orendorff (Mozilla)

SMOOSH? Seriously?



Brian Terlson
@bterlson

Adding Array.prototype.flatten to JS may break the web. The proper fix is to:

20% Rename it (e.g. smoosh)

10% Change default depth to ∞

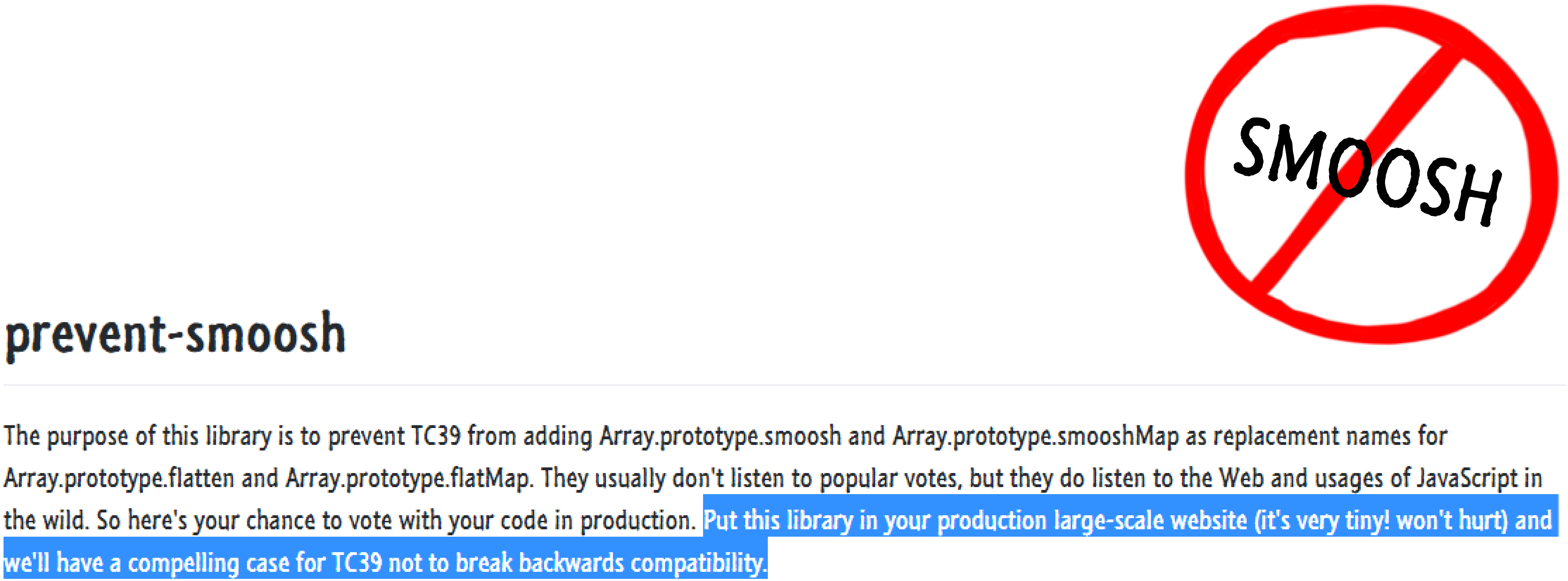
1% Hacks ala document.all

69% Break the web 🔥🌐

3,090 votes • Final results

6:26 PM - 6 Mar 2018

81 Retweets 151 Likes



prevent-smoosh

The purpose of this library is to prevent TC39 from adding Array.prototype.smoosh and Array.prototype.smooshMap as replacement names for Array.prototype.flatten and Array.prototype.flatMap. They usually don't listen to popular votes, but they do listen to the Web and usages of JavaScript in the wild. So here's your chance to vote with your code in production. Put this library in your production large-scale website (it's very tiny! won't hurt) and we'll have a compelling case for TC39 not to break backwards compatibility.

https://github.com/staltz/prevent-smoosh

ES 2016

Array.includes

** operator

ES 2017

Async

Lots of little stuff

(Not an actual feature name.
Object.values, String Padding, not breaking on trailing commas, etc.)

ES 2018

Shared Memory & Atomics

Async Iteration

Rest/Spread... operators

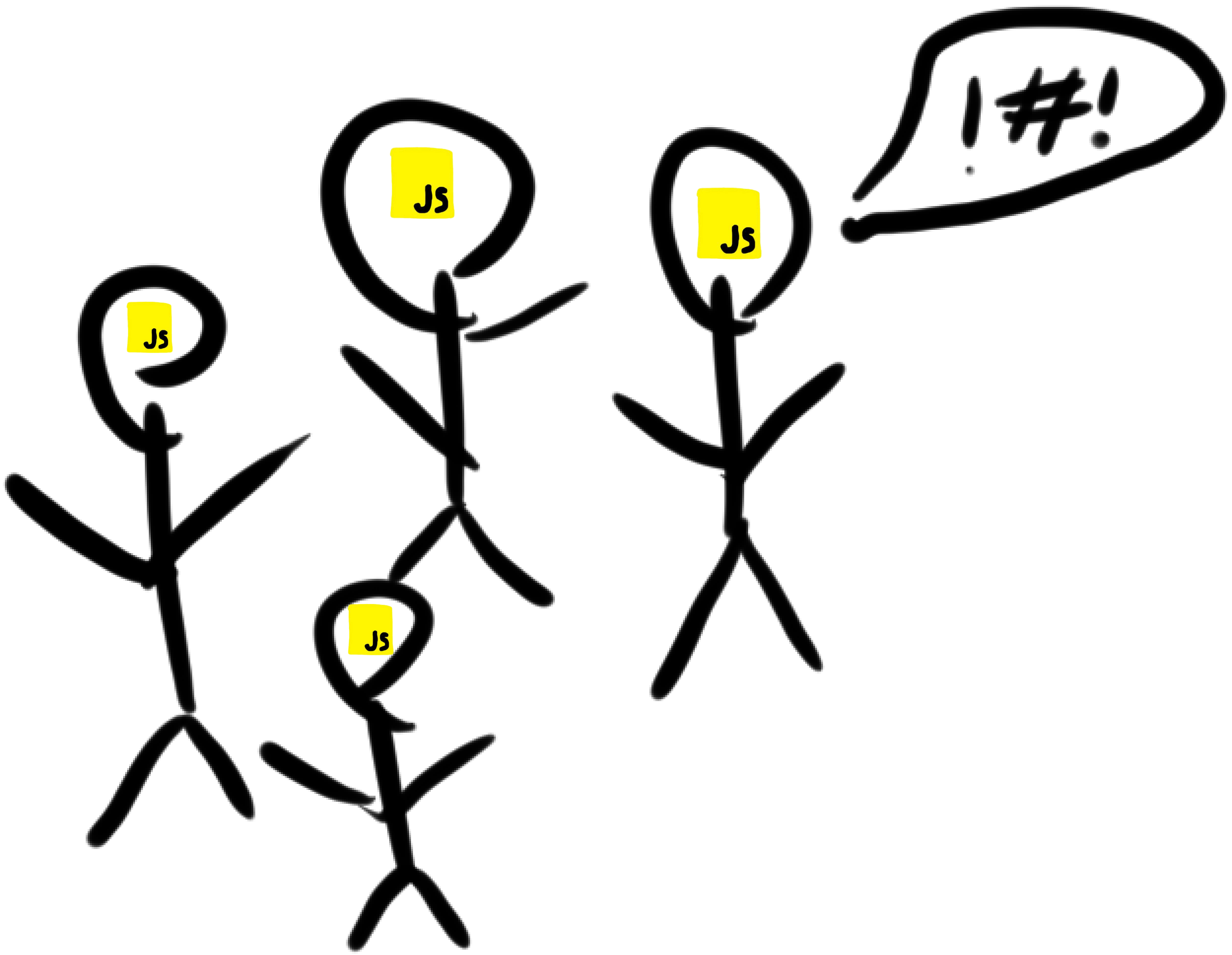
RegExp features

ES 2019

Array.prototype.{flat,flatMap}

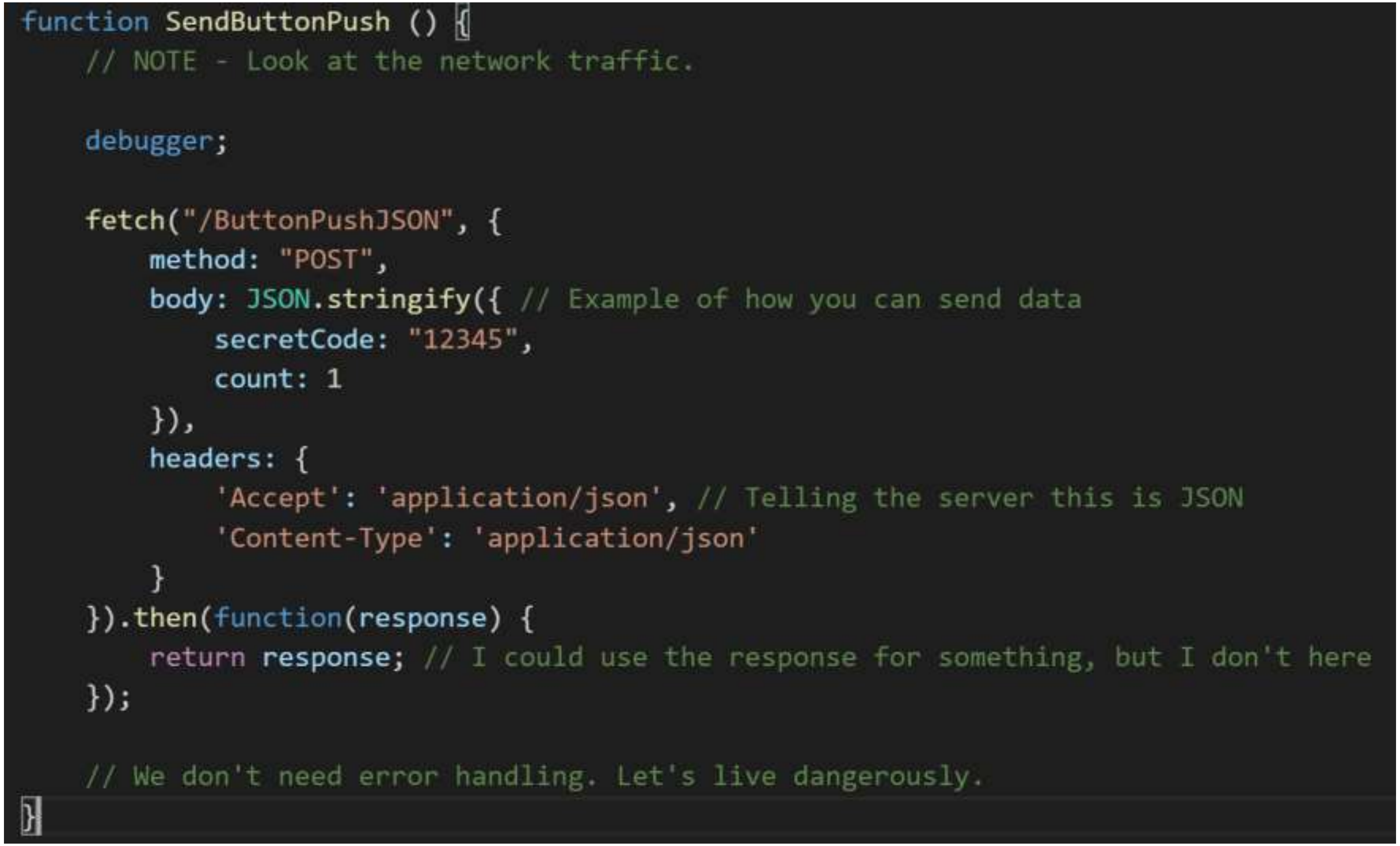
String.prototype.{trimStart, trimEnd}

Some string and JSON improvements

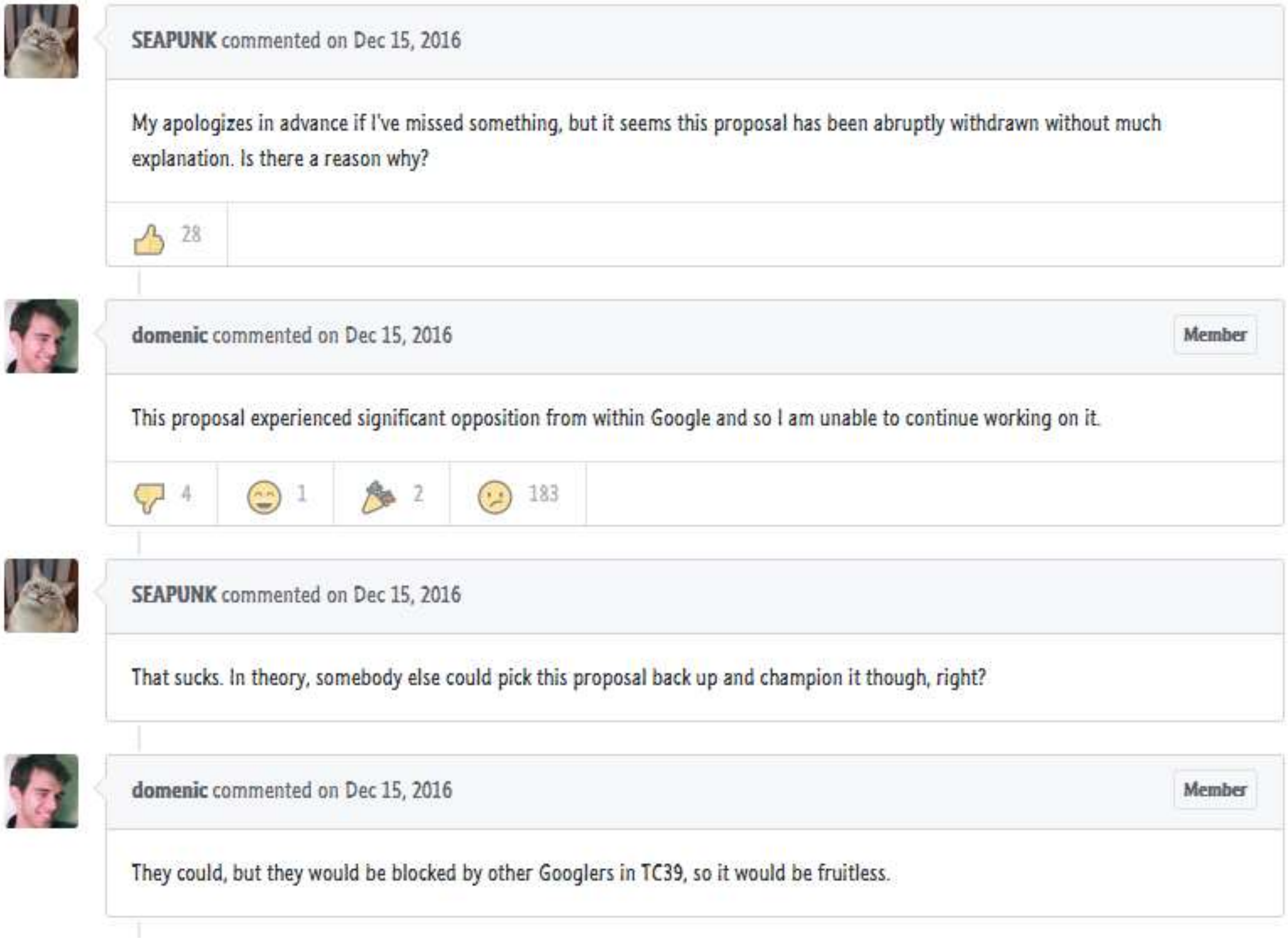


JavaScript Developers Swear A Lot

Fetch



```
function SendButtonPush () {
    // NOTE - Look at the network traffic.

    debugger;

    fetch("/ButtonPushJSON", {
        method: "POST",
        body: JSON.stringify({ // Example of how you can send data
            secretCode: "12345",
            count: 1
        }),
        headers: {
            'Accept': 'application/json', // Telling the server this is JSON
            'Content-Type': 'application/json'
        }
    }).then(function(response) {
        return response; // I could use the response for something, but I don't here
    });

    // We don't need error handling. Let's live dangerously.
}
```



SEAPUNK commented on Dec 15, 2016

My apologizes in advance if I've missed something, but it seems this proposal has been abruptly withdrawn without much explanation. Is there a reason why?

👍 28

domenic commented on Dec 15, 2016 Member

This proposal experienced significant opposition from within Google and so I am unable to continue working on it.

👎 4 😄 1 🎉 2 😕 183

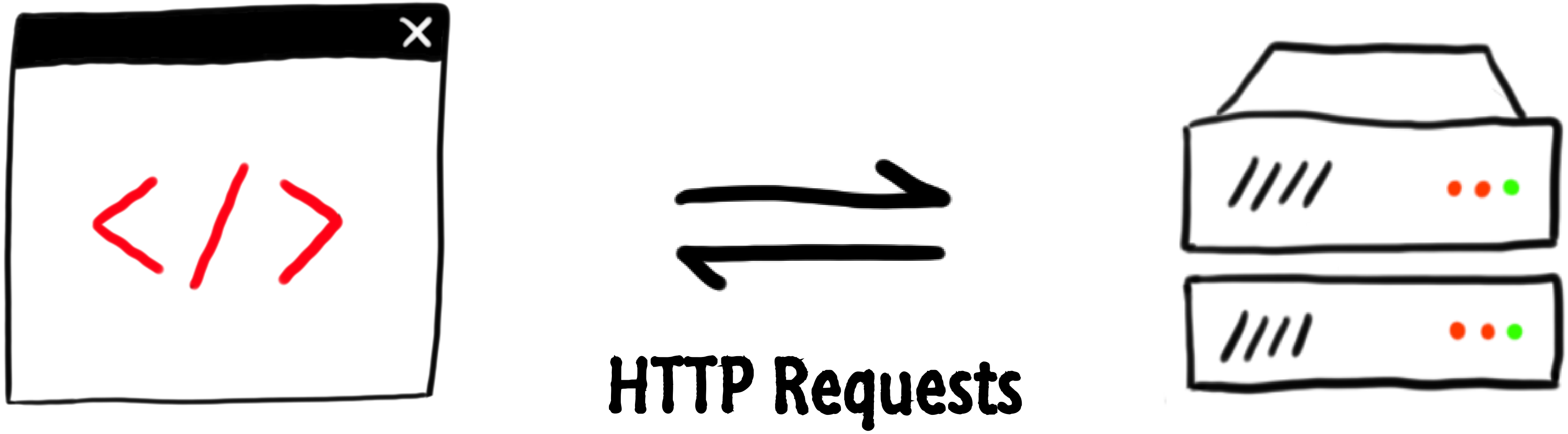
SEAPUNK commented on Dec 15, 2016

That sucks. In theory, somebody else could pick this proposal back up and champion it though, right?

domenic commented on Dec 15, 2016 Member

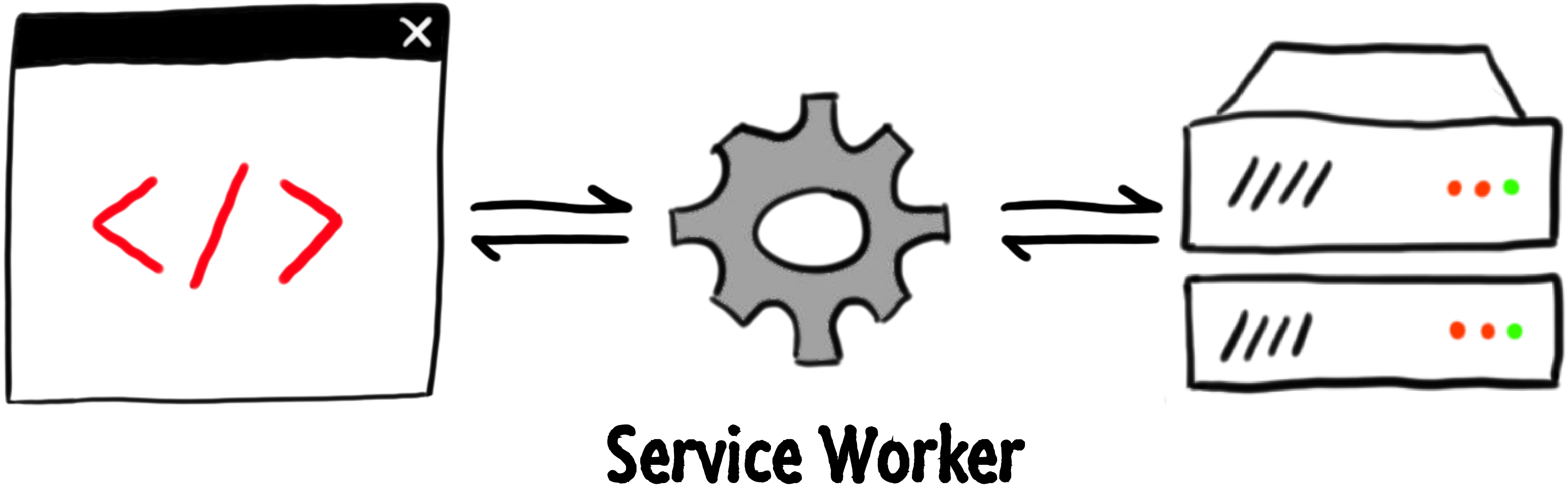
They could, but they would be blocked by other Googlers in TC39, so it would be fruitless.

W3C

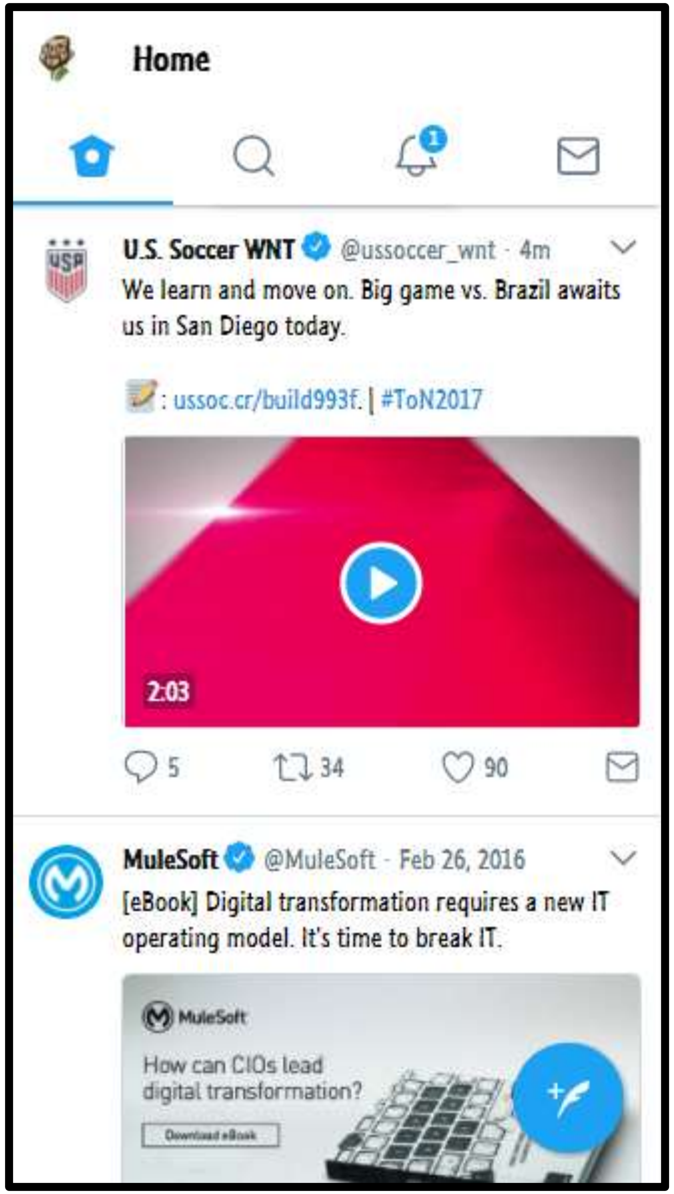


HTTP Requests

W3C



Progressive Web Apps



U.S. Soccer WNT @ussoccer_wnt · 4m

We learn and move on. Big game vs. Brazil awaits us in San Diego today.

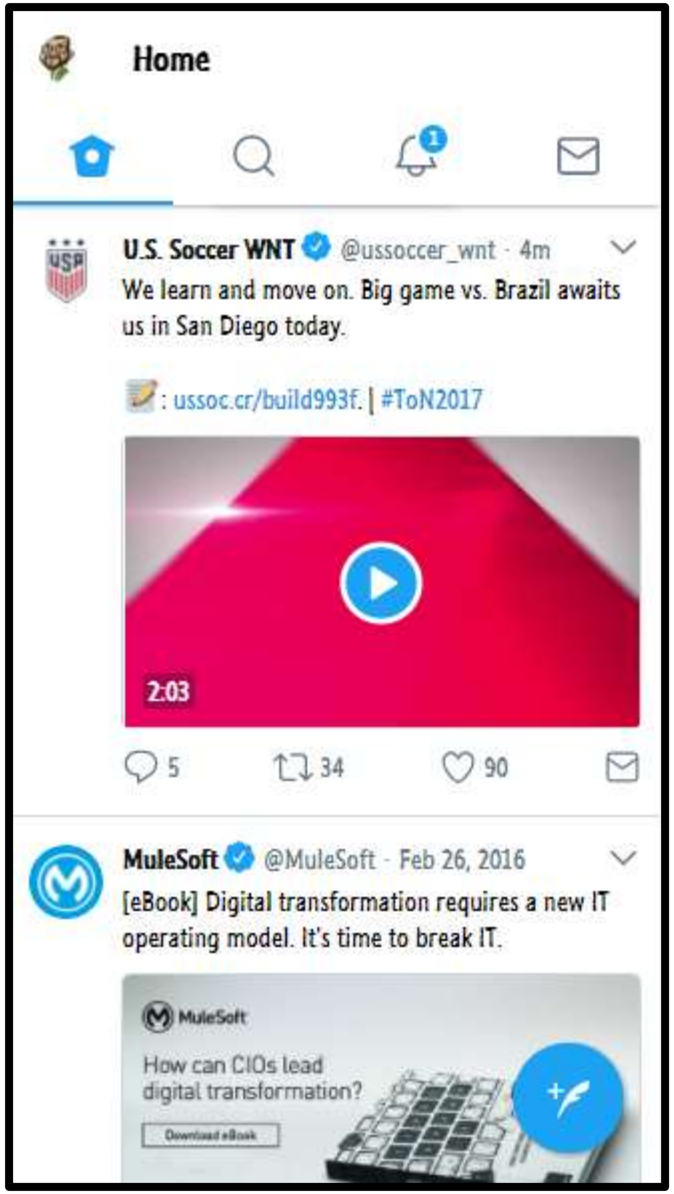
: ussoc.cr/build993f. | #ToN2017

MuleSoft @MuleSoft · Feb 26, 2016

[eBook] Digital transformation requires a new IT operating model. It's time to break IT.

Not Google

- 65% increase in pages per session
- 75% increase in Tweets sent
- 20% decrease in bounce rate



U.S. Soccer WNT @ussoccer_wnt · 4m
We learn and move on. Big game vs. Brazil awaits us in San Diego today.

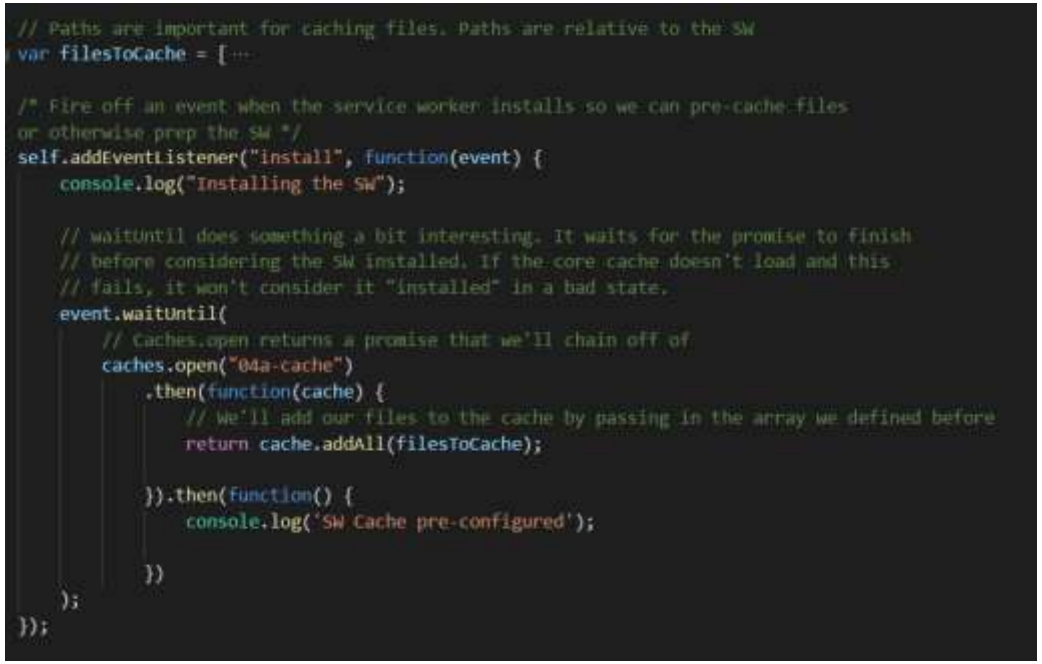
📝: ussoc.cr/build993f. | #ToN2017

MuleSoft @MuleSoft · Feb 26, 2016
[eBook] Digital transformation requires a new IT operating model. It's time to break IT.

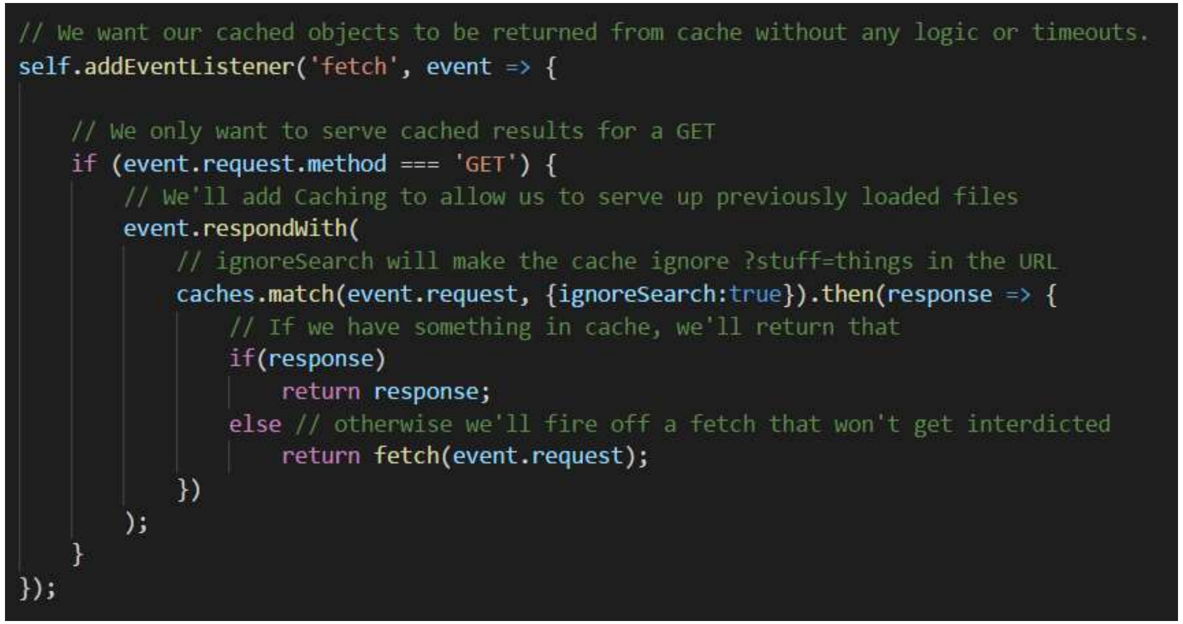
Not Google

Twitter Is Complicated…

Here's a dumb example

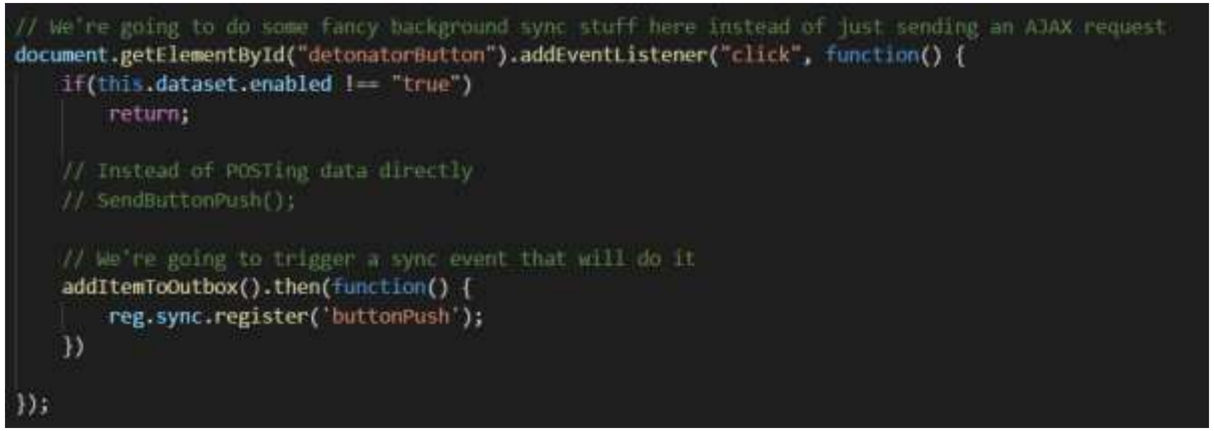


```
// Paths are important for caching files. Paths are relative to the SW
var filesToCache = [ ···

/* Fire off an event when the service worker installs so we can pre-cache files
or otherwise prep the SW */
self.addEventListener("install", function(event) {
    console.log("Installing the SW");

    // waitUntil does something a bit interesting. It waits for the promise to finish
    // before considering the SW installed. If the core cache doesn't load and this
    // fails, it won't consider it "installed" in a bad state.
    event.waitUntil(
        // Caches.open returns a promise that we'll chain off of
        caches.open("04a-cache")
            .then(function(cache) {
                // We'll add our files to the cache by passing in the array we defined before
                return cache.addAll(filesToCache);

            }).then(function() {
                console.log('SW Cache pre-configured');

            })
    );
});
```

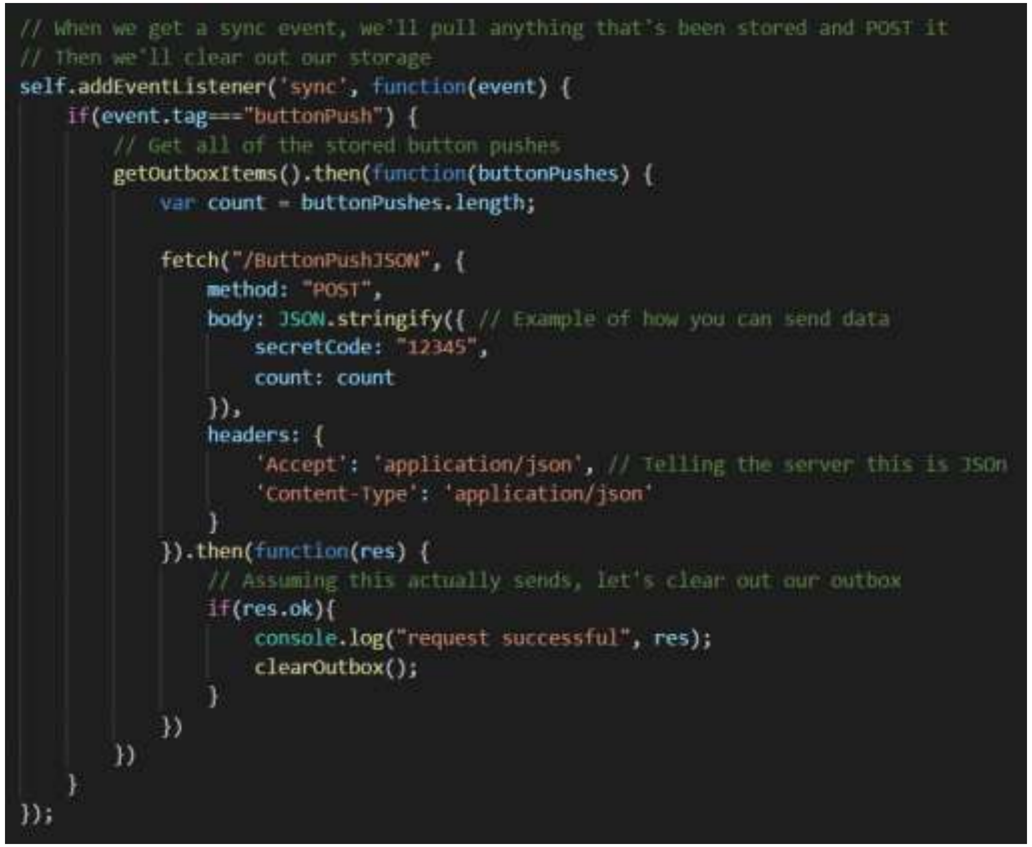


```
// We want our cached objects to be returned from cache without any logic or timeouts.
self.addEventListener('fetch', event => {

    // We only want to serve cached results for a GET
    if (event.request.method === 'GET') {
        // We'll add Caching to allow us to serve up previously loaded files
        event.respondWith(
            // ignoreSearch will make the cache ignore ?stuff=things in the URL
            caches.match(event.request, {ignoreSearch:true}).then(response => {
                // If we have something in cache, we'll return that
                if(response)
                    return response;
                else // otherwise we'll fire off a fetch that won't get interdicted
                    return fetch(event.request);
            })
        );
    }
});
```

Offline?



```
// We're going to do some fancy background sync stuff here instead of just sending an AJAX request
document.getElementById("detonatorButton").addEventListener("click", function() {
    if(this.dataset.enabled !== "true")
        return;

    // Instead of POSTing data directly
    // SendButtonPush();

    // We're going to trigger a sync event that will do it
    addItemToOutbox().then(function() {
        reg.sync.register('buttonPush');
    })

});
```



```
// When we get a sync event, we'll pull anything that's been stored and POST it
// Then we'll clear out our storage
self.addEventListener('sync', function(event) {
    if(event.tag==="buttonPush") {
        // Get all of the stored button pushes
        getOutboxItems().then(function(buttonPushes) {
            var count = buttonPushes.length;

            fetch("/ButtonPushJSON", {
                method: "POST",
                body: JSON.stringify({ // Example of how you can send data
                    secretCode: "12345",
                    count: count
                }),
                headers: {
                    'Accept': 'application/json', // Telling the server this is JSOn
                    'Content-Type': 'application/json'
                }
            }).then(function(res) {
                // Assuming this actually sends, let's clear out our outbox
                if(res.ok){
                    console.log("request successful", res);
                    clearOutbox();
                }
            })
        })
    }
});
```

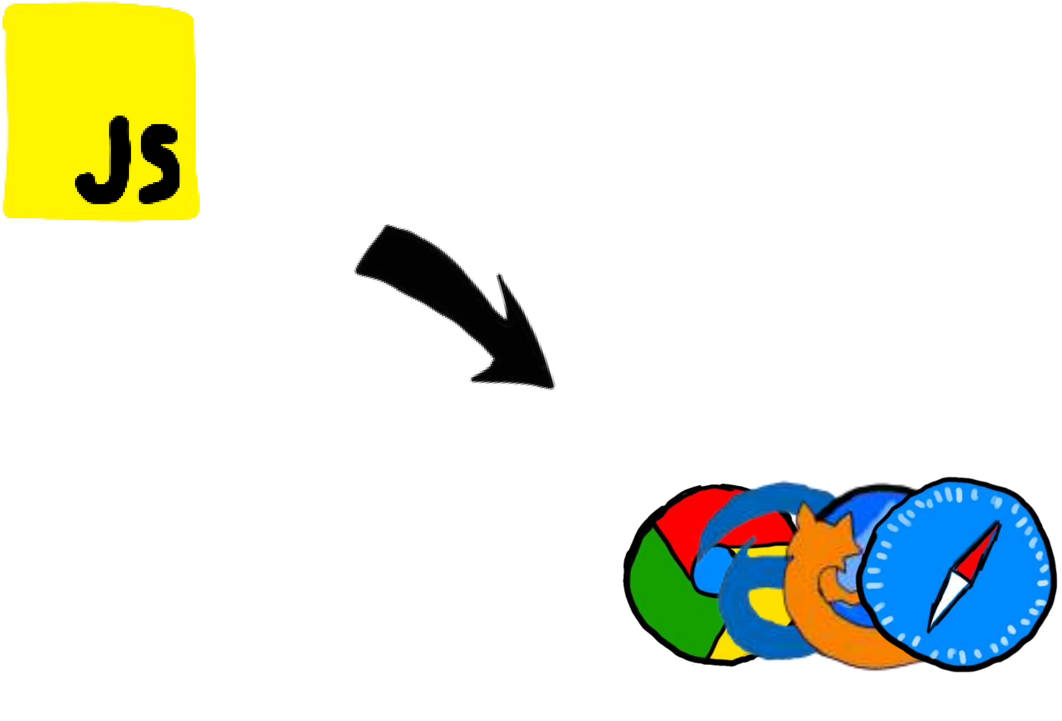
What about Web Assembly?

¯_(ツ)_/¯

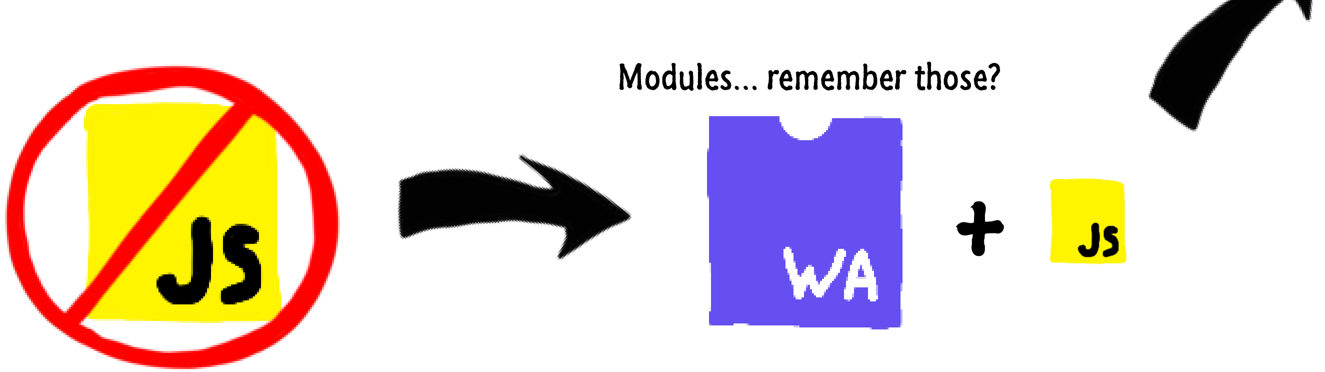
“WebAssembly (abbreviated Wasm) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable target for compilation of high-level languages like C/C++/Rust, enabling deployment on the web for client and server applications.”

How we code the web today*

* not shown: 400 npm packages and a complicated build system



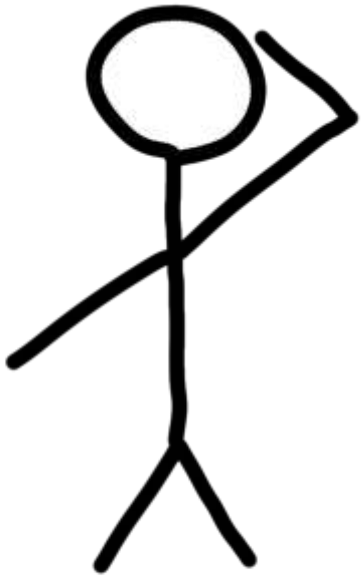
How Web Assembly works



Modules... remember those?

C#, Rust, C# (sorta), Java, Python, etc. etc.
Too lazy to draw a ton of language icons ;)

"Why would I WA...?"

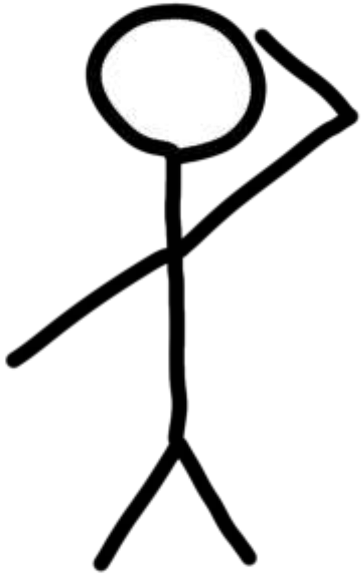


Good question

¯_(ツ)_/¯

Most use cases are dumb... today

“Oh, good...”



- Build high performing applications (video chat, gaming, CAD, etc.)
- Re-use complicated business logic
- Build things like ML or VR
- Avoid writing JavaScript

Tube

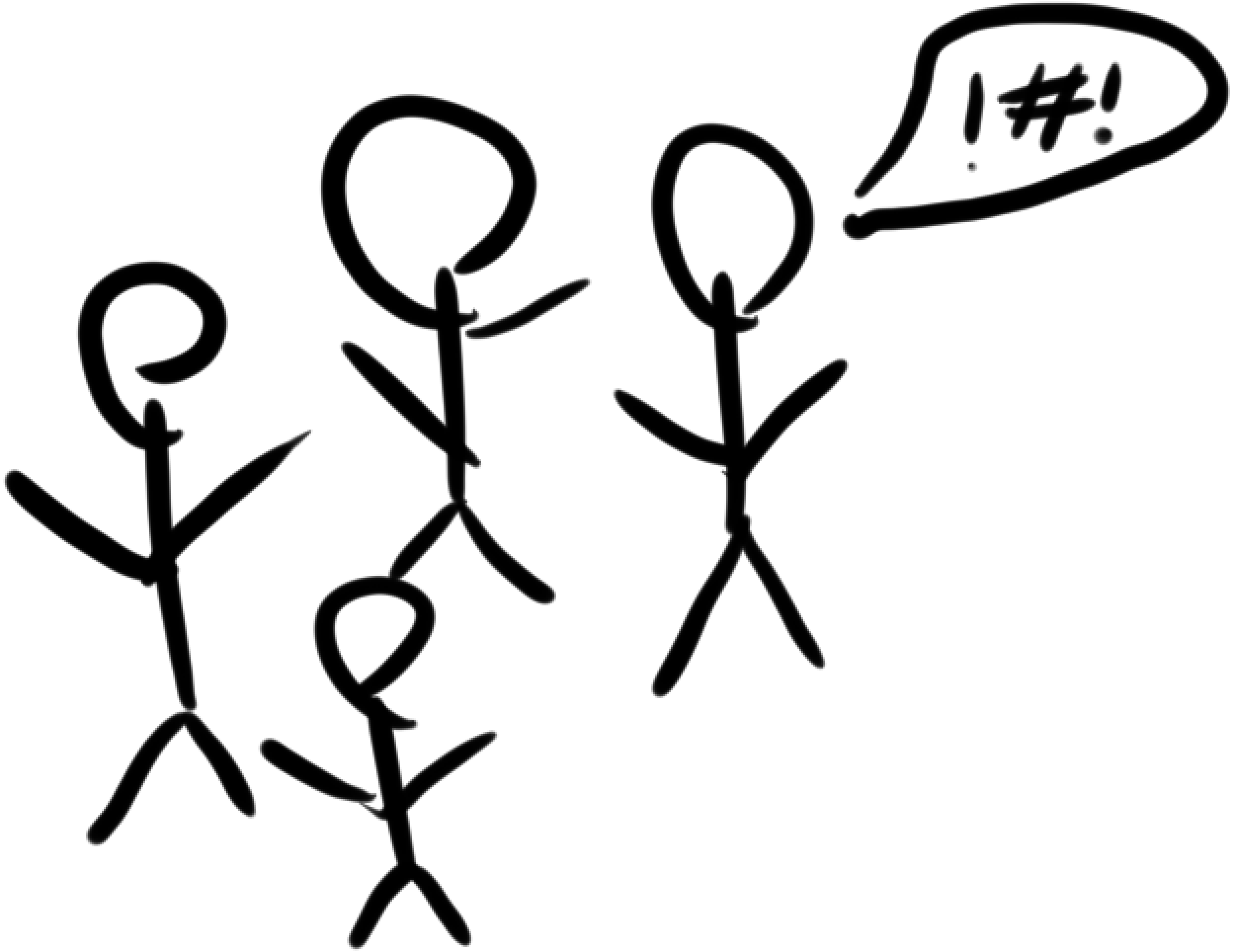
Internet Engineering ~~Task~~ Force

IETF

http2

Binary Framing Layer

¯_(ツ)_/¯



Security Researchers Swear A Lot

And what about...?

Internet Exchromedgium!*

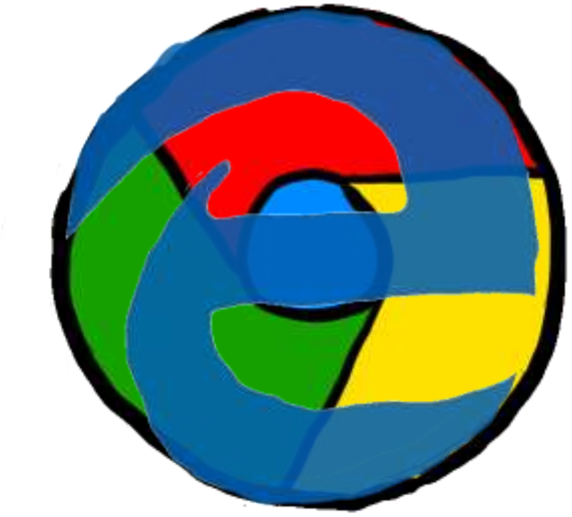
* Not official title

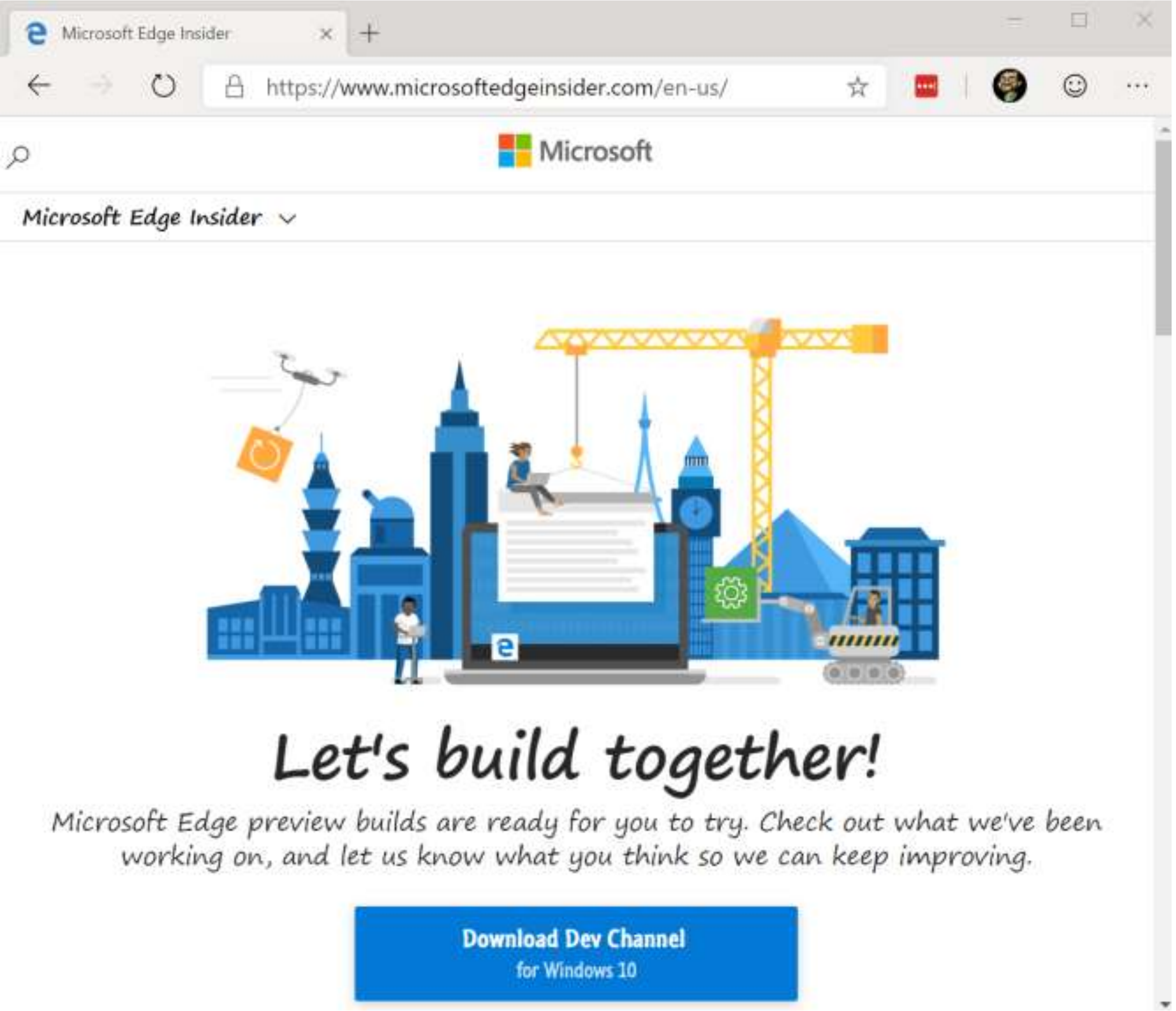
This impacts:

- Interoperability
- Standards compliance
- Developer experience

It doesn't directly change:

- Platform functionality
- Standards development

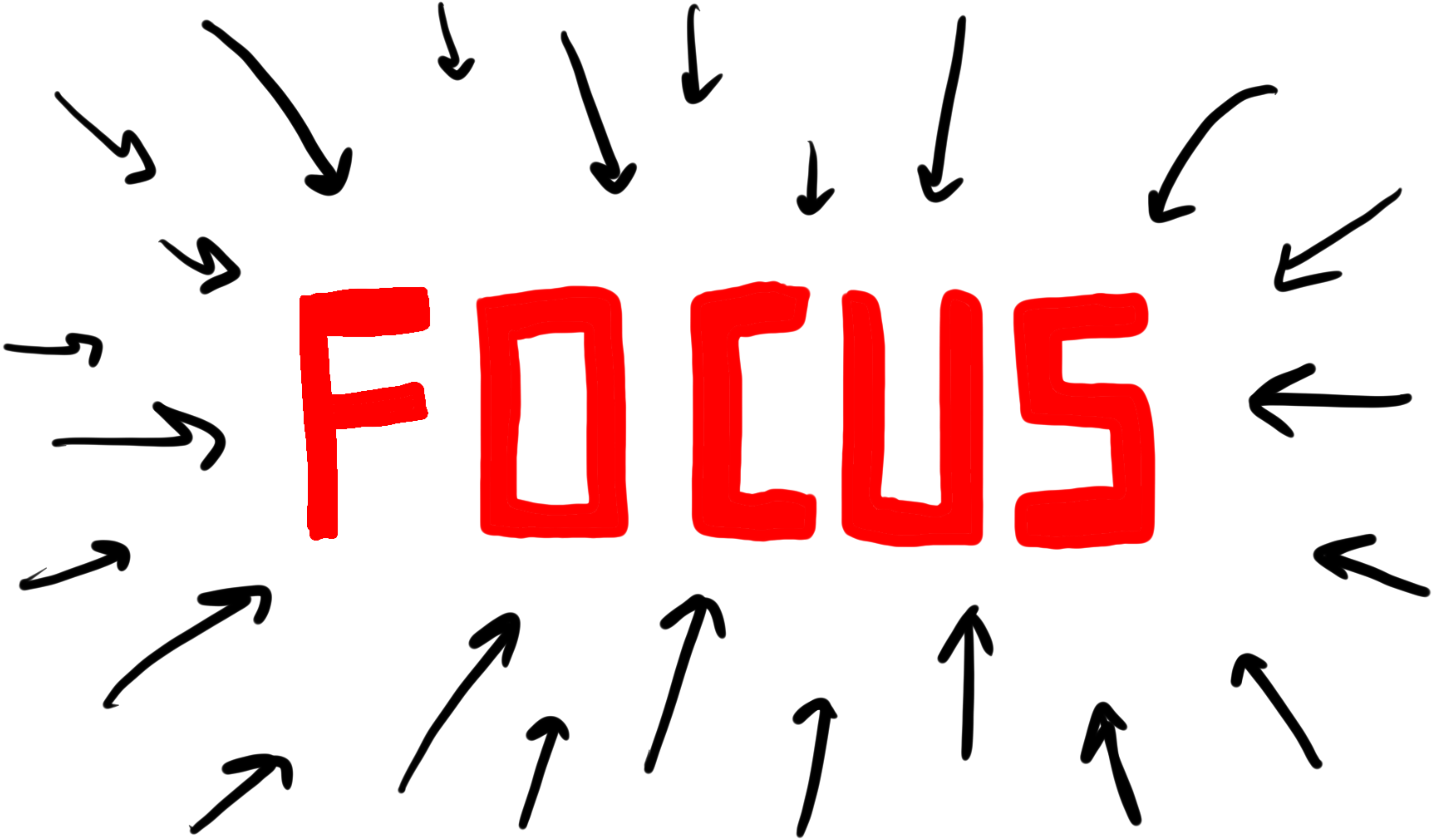




Let's build together!

Microsoft Edge preview builds are ready for you to try. Check out what we've been working on, and let us know what you think so we can keep improving.

https://www.microsoftedgeinsider.com



FOCUS

THANK YOU!

@jaredthenerd
jaredthenerd.com