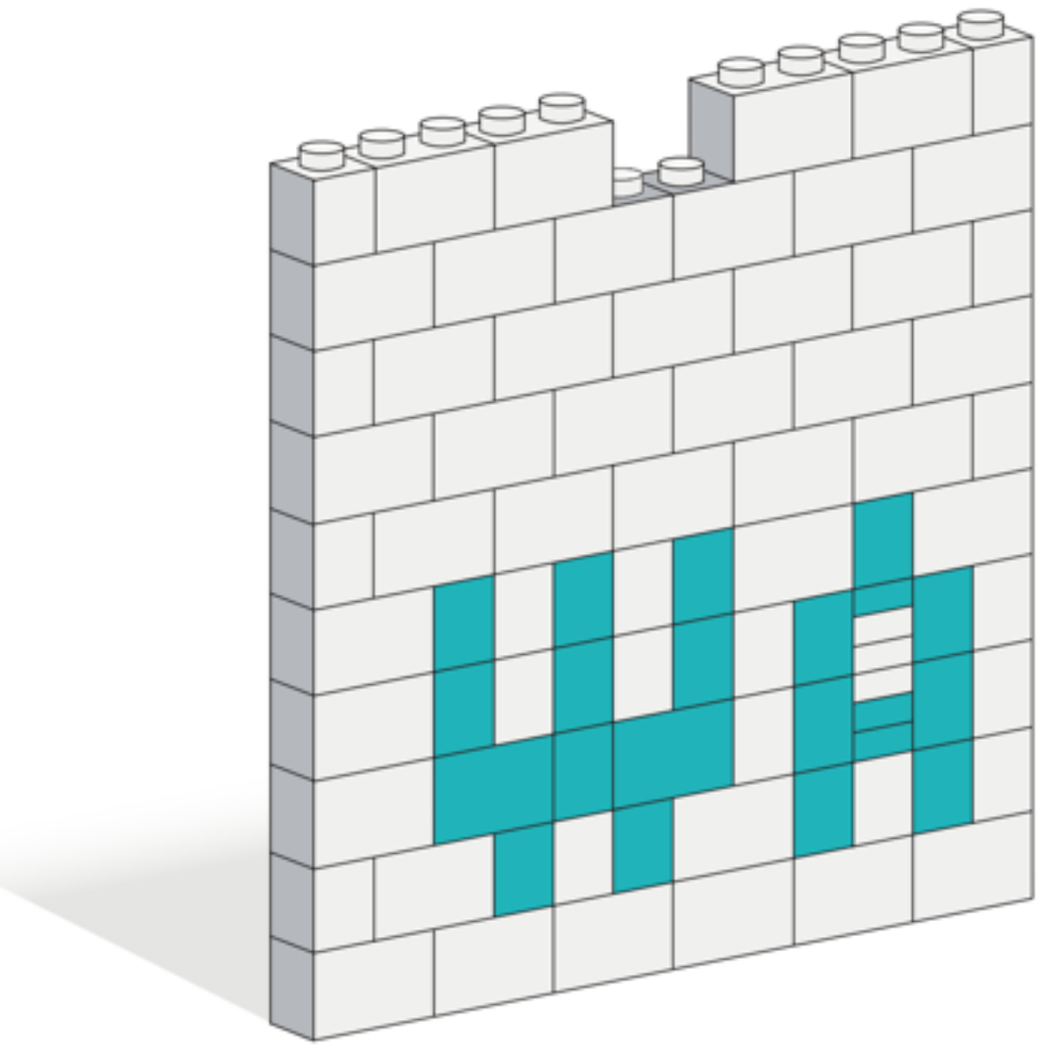
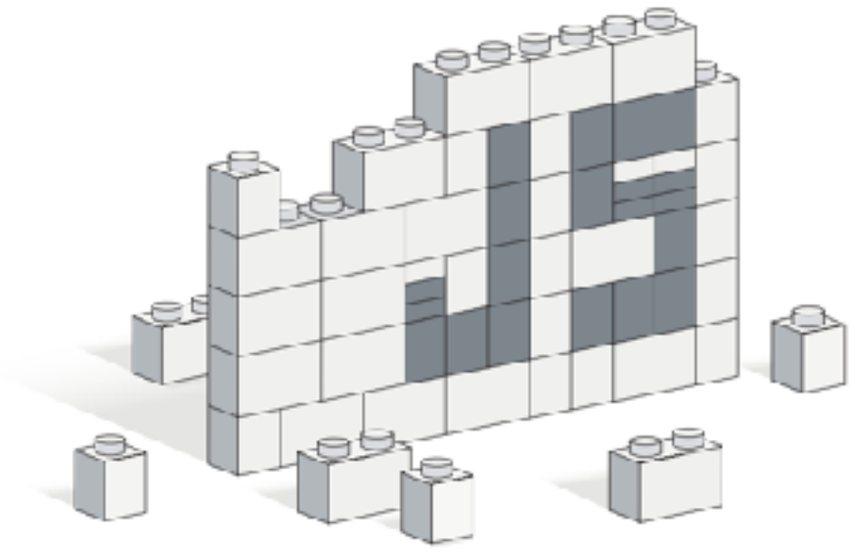


WebAssembly

and the death of JavaScript?

@ColinEberhardt

JavaScript
Brendan Eich
1995
10 days



ActiveX

Flash

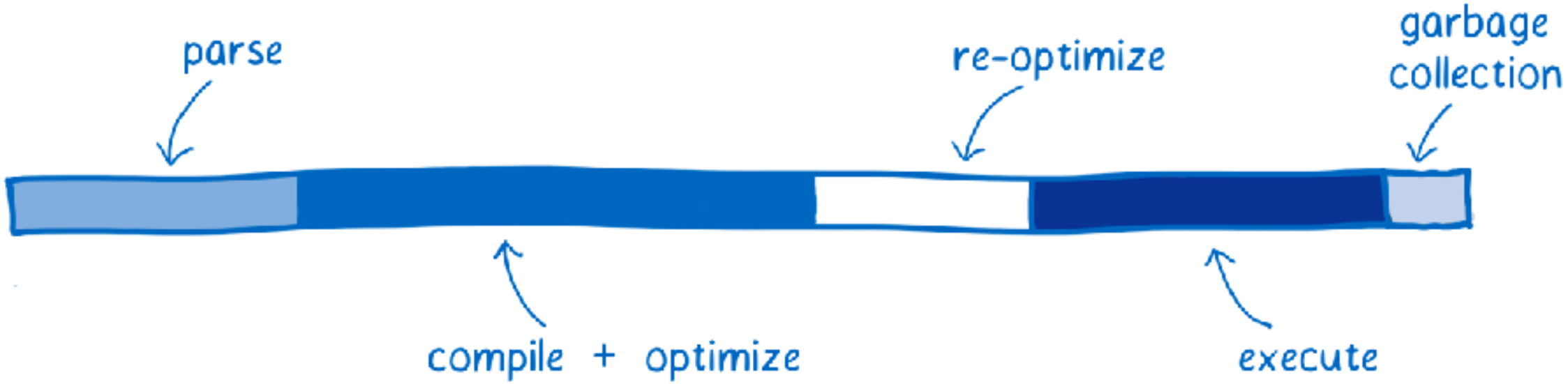
Silverlight

Dart

2018, JavaScript … still

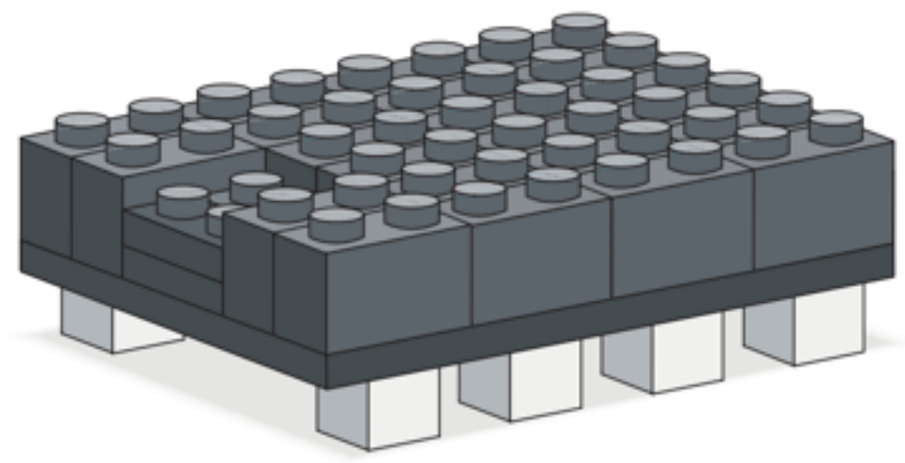
```
pe&&(b[c]=d.value)},r="undefined"!=typeof window&&window===this?this:"undefined"!=
(b){for(var c=r,d=["Promise"],e=0;e<d.length-1;e++){var f=d[e];f in c||(c[f]={});c
{configurable:!0,writable:!0,value:b})}},ca=function(){ca=function(){};
ol=da)},da=function(){var b=0;return function(c){return"jscomp_symbol_"+(c||"")+b+
or;b||(b=r.Symbol.iterator=r.Symbol("iterator"));typeof Array.prototype[b]!=m&&aa(
:function(){return ea(this)}});fa=function(){},ea=function(b){var c=0;return ha(f
)},ha=function(b){fa();b={next:b};b[r.Symbol.iterator]=function(){return this};
tion(b){fa();var c=b[Symbol.iterator];return c?c.call(b):ea(b)};ba(function(b){fun
new f(function(c){c(b)})}if(b)return b;c.prototype.c=function(b){null==this.b&&(th
ction(){var b=this;this.f(function(){b.i()})};var e=r.setTimeout;c.prototype.f=fun
s.b.length;){var b=this.b;this.b=[];for(var c=0;c<b.length;++c){var d=
try{d()}catch(l){this.h(l)}}}this.b=null};c.prototype.h=function(b){this.f(functio
s.b=[];var c=this.f();try{b(c.resolve,c.reject)}catch(n){c.reject(n)}};f.prototype
```

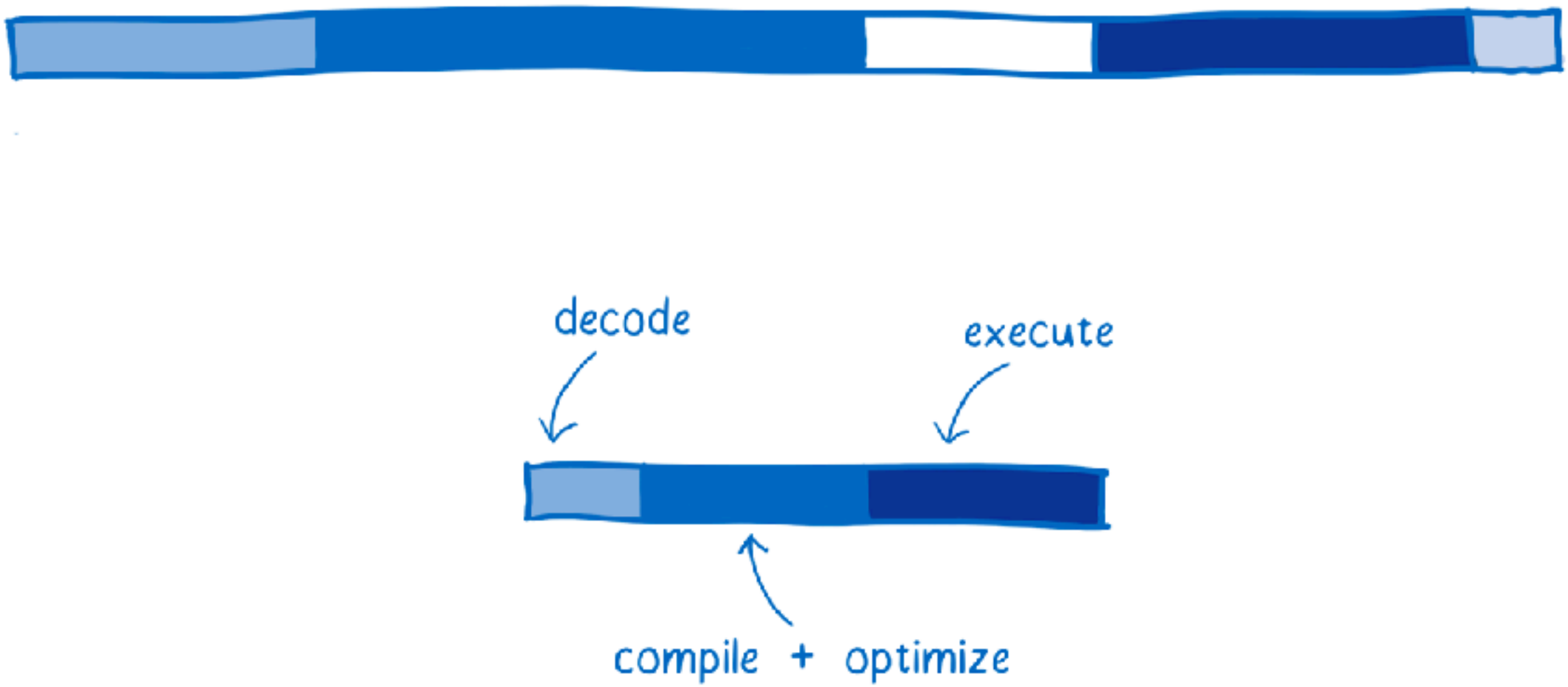
```
ll(c,e))}}var c=this,d=!1;return{resolve:b(this.B),reject:b(this.g)}};f.prototype.
nise cannot resolve to itself"));else if(b instanceof f)this.C(b);
eof b){case "object":var c=null!=b;break a;case m:c=!0;break a;default:c=!1}c?this
{c=b.then}catch(n){this.g(n);return}typeof c==m?this.D(c,b):this.i(b)};f.prototype
ction(b){this.j(1,b)};f.prototype.j=function(b,c){if(0!=this.c)throw Error("Cannot
"+this.c);this.c=b;this.h=c;this.l()};f.prototype.l=function(){if(null!=this.b){for
ngth;++c)b[c].call(),b[c]=null;this.b=null}};var g=new c;f.prototype.C=function(b)
ction(b,c){var d=this.f();try{b.call(c,d.resolve,d.reject)}catch(l){d.reject(l)}};
b==m?function(c){try{e(b(c))}catch(Za){g(Za)}}:c}var e,g,h=new f(function(b,c){e=b
h"]=function(b){return this.then(void 0,b)};f.prototype.o=function(b,
switch(e.c){case 1:b(e.h);break;case 2:c(e.h);break;default:throw Error("a`"+e.c);}
)})};f.resolve=d;f.reject=function(b){return new f(function(c,d){d(b)})};f.race=fu
```


parse
re-optimize
garbage collection
compile + optimize
execute

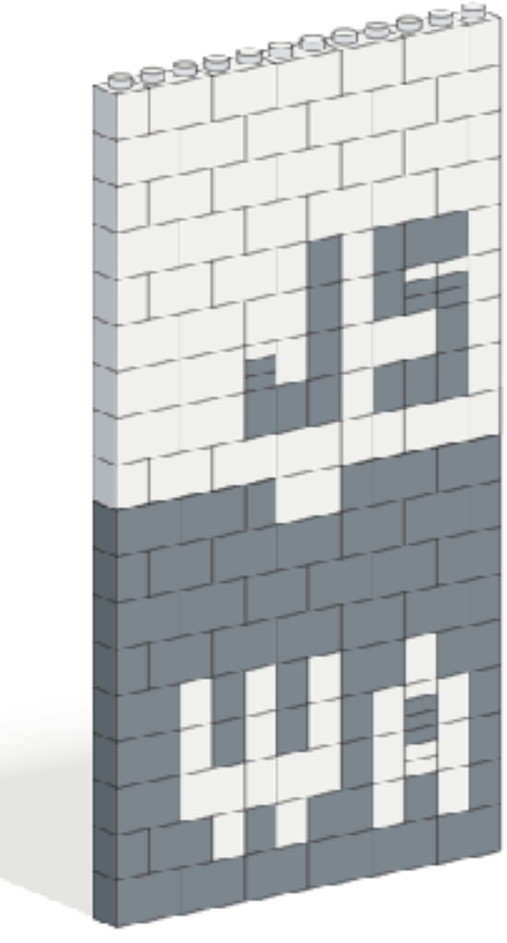
"the Web has become the most ubiquitous application platform ever, and yet by historical accident the only natively supported programming language for that platform is JavaScript!"

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




decode
execute
compile + optimize

```
00000000: 0061 736d 0100 0000 0107 0160 027d 7f01  .asm.......`.}..
00000010: 7d03 0201 0004 0401 7000 0005 0301 0001  }.......p.......
00000020: 0712 0206 6d65 6d6f 7279 0200 0570 6f77  ....memory...pow
00000030: 6572 0000 0a33 0131 0101 7d02 4020 0141  er...3.1..}.@ .A
00000040: 0248 0d00 2001 417f 6a21 0120 0021 0203  .H.. .A.j!. .!..
00000050: 4020 0220 0094 2102 2001 417f 6a22 010d  @ . ..!. .A.j"..
00000060: 000b 2002 0f0b 2000 0b                   .. ... ..
```



```
// read the binary into a buffer
const fs = require("fs");
const buf = fs.readFileSync("./add.wasm");

// create a wasm module
const wasmModule = new WebAssembly.Module(new Uint8Array(buf));

// construct an instance of the module
const wasmInstance = new WebAssembly.Instance(wasmModule);

// invoke the exported function
const result = wasmInstance.exports.power(2, 3)
console.log(result);
```

PSPDFKit for Web using WebAssembly ✅

PSPDFKit for Web
Demo

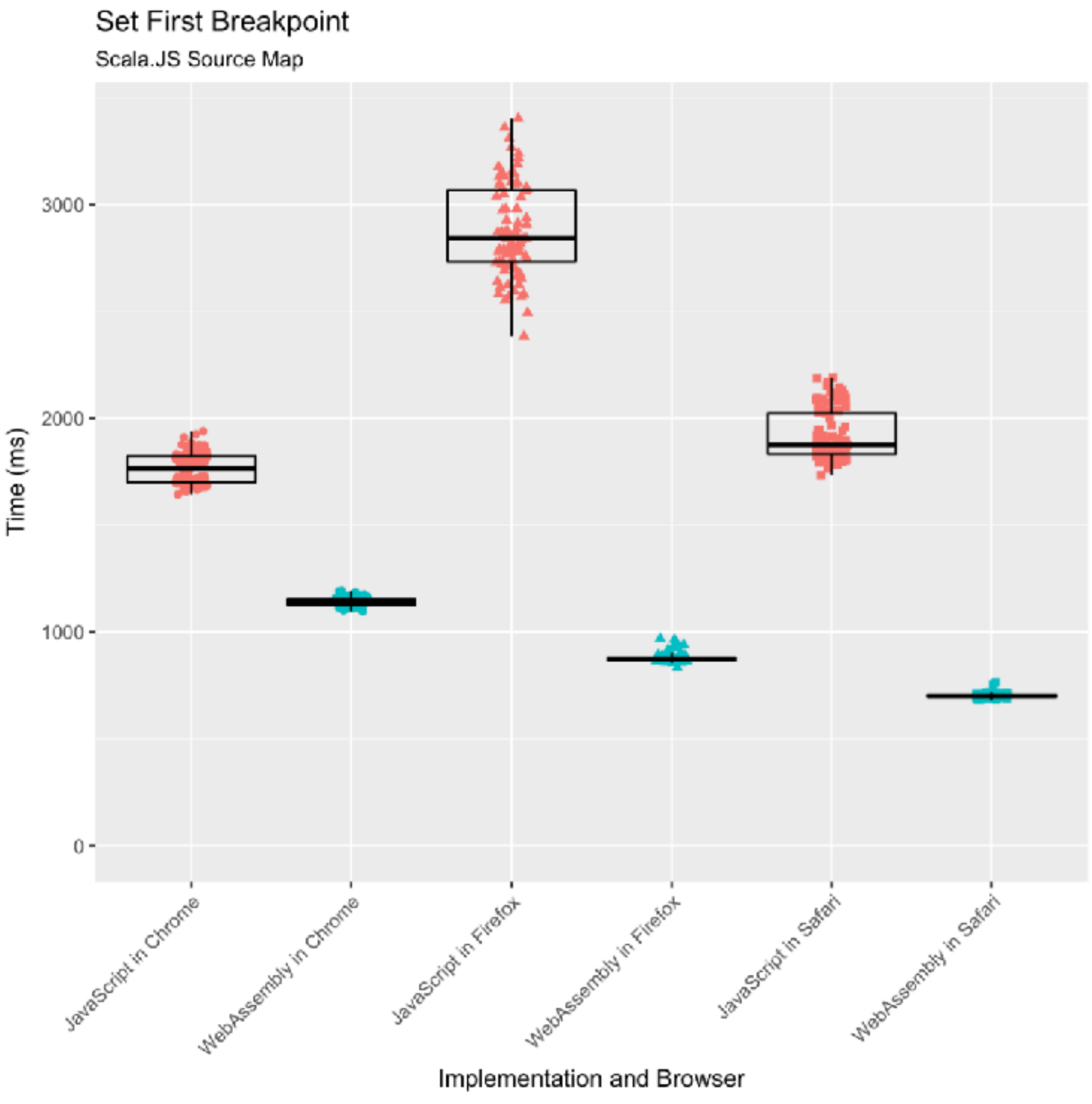
Hello!

This is a sample PDF document that showcases the functionality and rendering of PSPDFKit for Web, our JavaScript-based PDF SDK. If you have any additional questions regarding PSPDFKit for Web or our other SDK products, please get in touch!

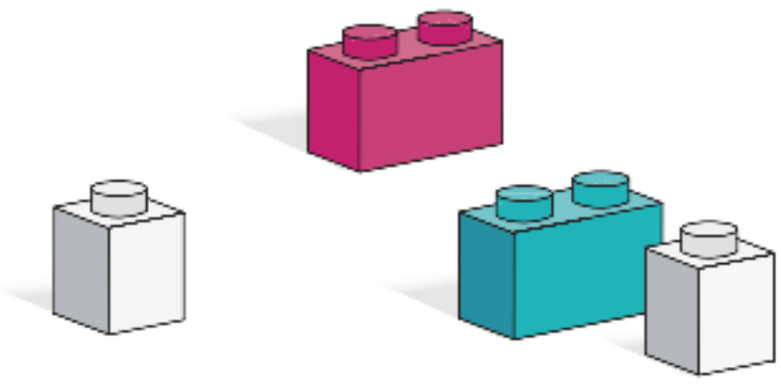
— the PSPDFKit Team

Scroll down for more!

```
JSC.js Shell
Downloading contents, please wait...
Preparing...
All downloads complete.
Running...
JSC >>>
JSC >>>
JSC >>> Date();
Fri Sep 15 2017 20:23:37 GMT-0700
JSC >>> Date.now();
1505532221655
JSC >>>
JSC >>>
JSC >>> 0.1 + 0.2
0.30000000000000004
JSC >>>
```


Set First Breakpoint
Scala.JS Source Map
Time (ms)
3000
2000
1000
0
JavaScript in Chrome
WebAssembly in Chrome
JavaScript in Firefox
WebAssembly in Firefox
JavaScript in Safari
WebAssembly in Safari
Implementation and Browser

WebAssembly Predictions

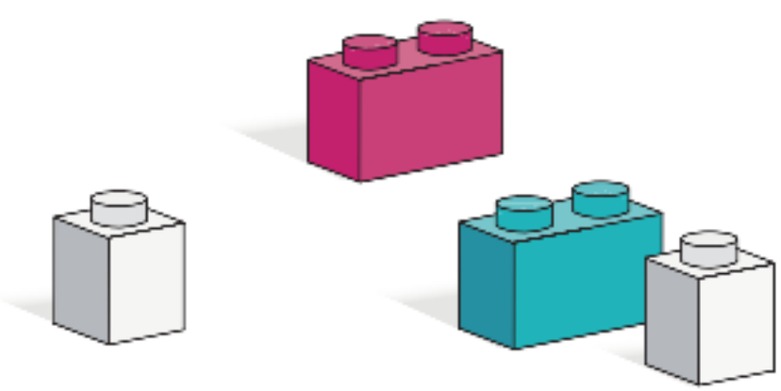


2018

- Rust, C, C++ used in production for performance critical, algorithmic tasks

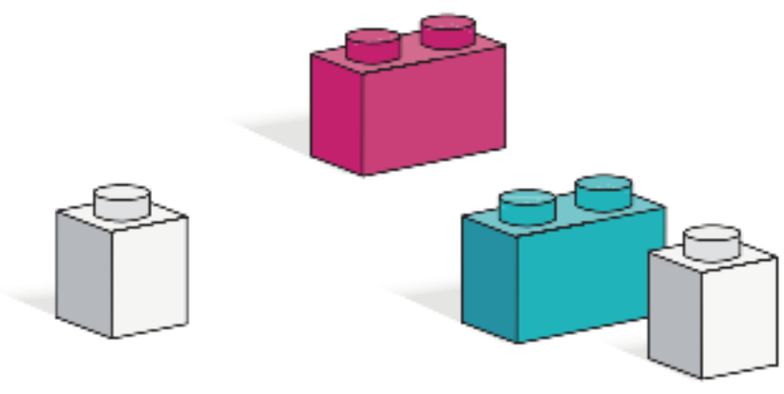
- Java, C#, Typescript lots of creative experiments / POCs

- Garbage Collection lands in WebAssembly



2019 - 2020

- Java & C# - was considered production ready
- Heavyweight productivity apps move to the web
- Another wave of mobile, desktop and server-side UI frameworks will re-target the web - *write one, run everywhere*
- React for Rust
- Native Android apps die out in favou

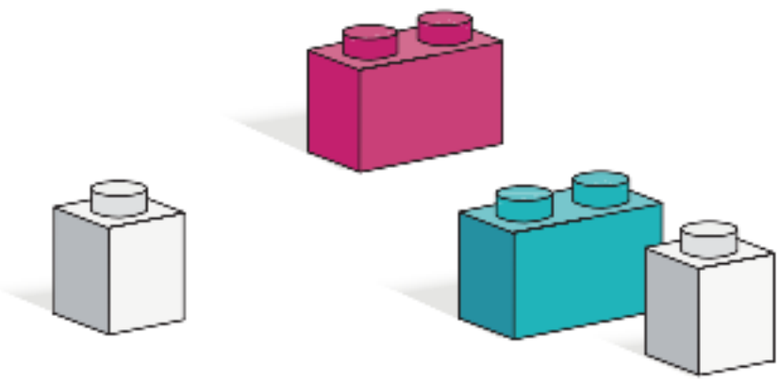


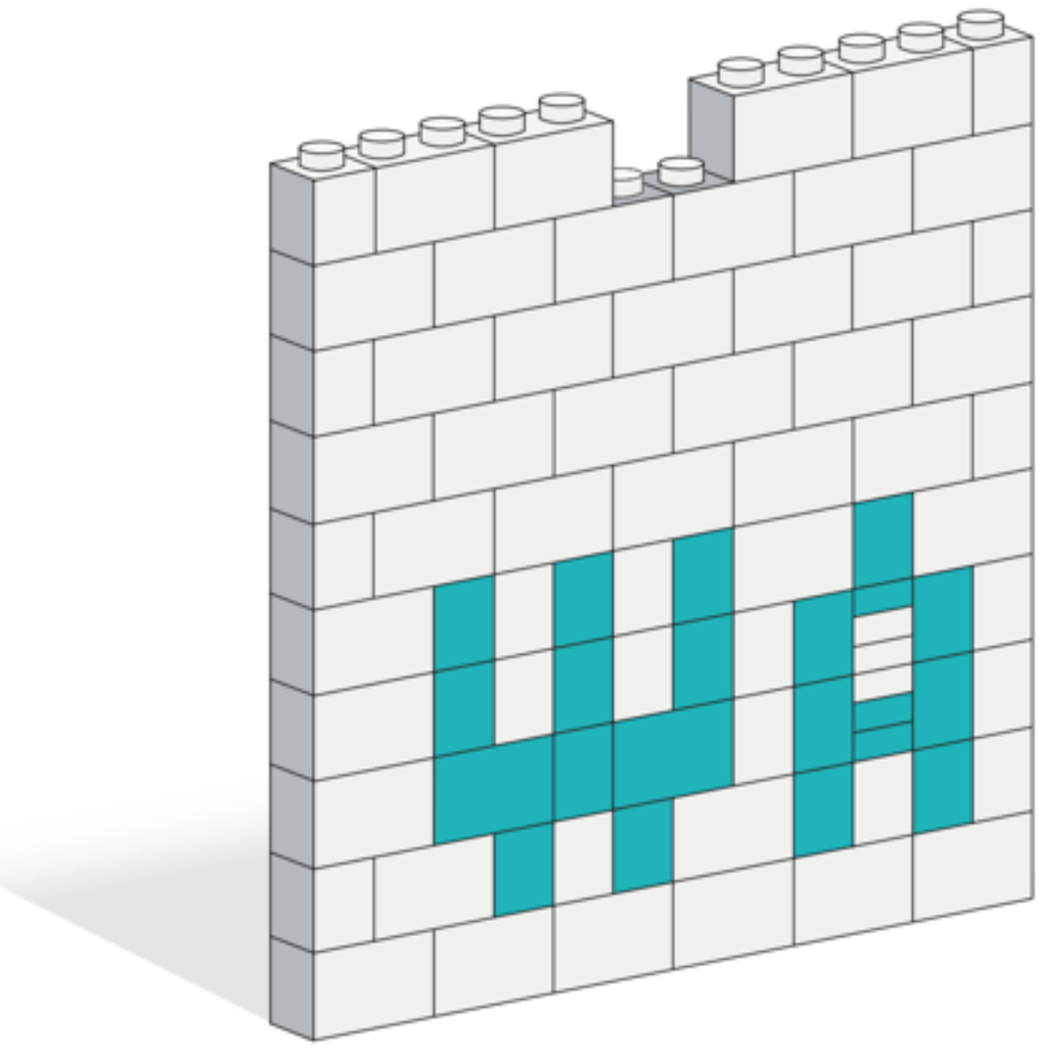
2021 - and beyond

- Windows Store drops support for non-web technologies

- MacOS drops support for non-web technology apps, resulting in a single unified runtime across desktop, web and mobile

- As WebAssembly has replaced JavaScript, a replacement for the DOM emerges





WebAssembly

and the death of JavaScript?

@ColinEberhardt