"The best way to predict the future is to invent it."

--Alan Kay

credit: David Hasan

brief^history lesson

and incomplete!

(of American spaceflight)

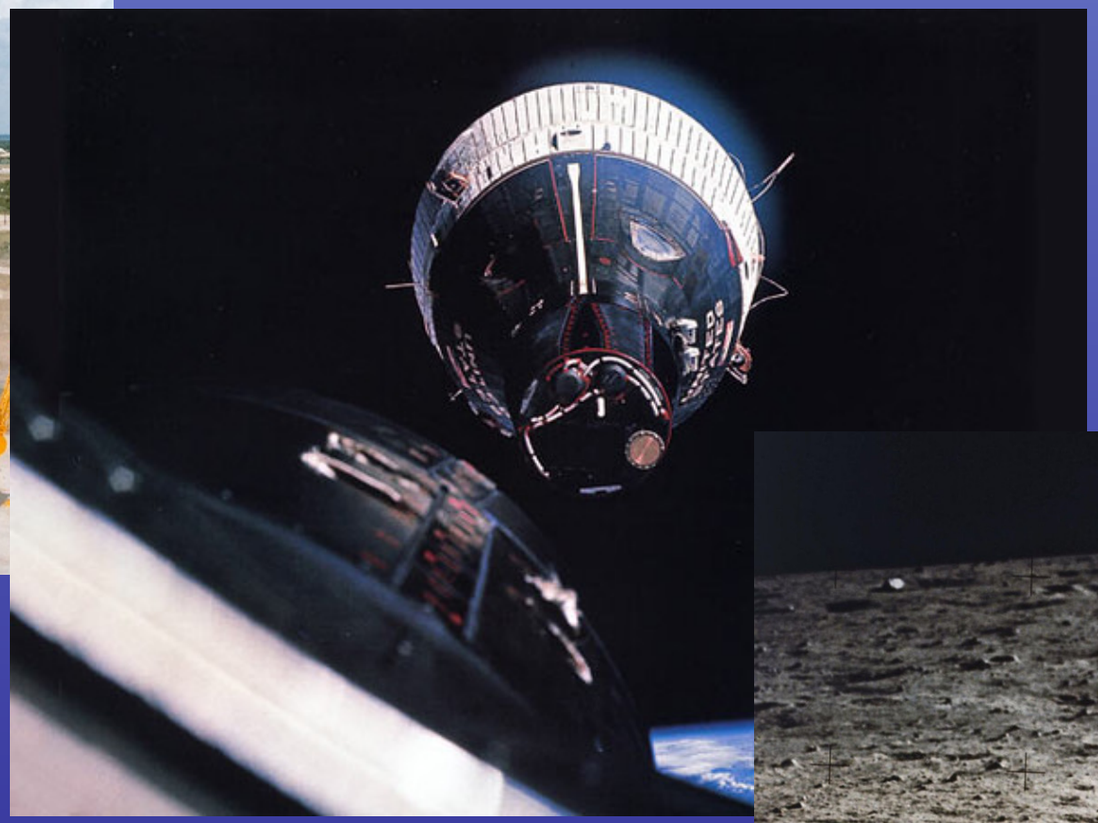
credit: David Hasan

Mercury (Atlas)


GT-3
GT-4
GT-5
GT-7
GT-6
GT-8
GT-9
GT-10
credit: David Hasan
Gemini (Titan)


Apollo 8
Apollo 9
Apollo 10
Apollo 11
Apollo 12
Apollo 13
credit: David Hasan
USA
Apollo (Saturn)

credit: David Hasan



lots of baby steps

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

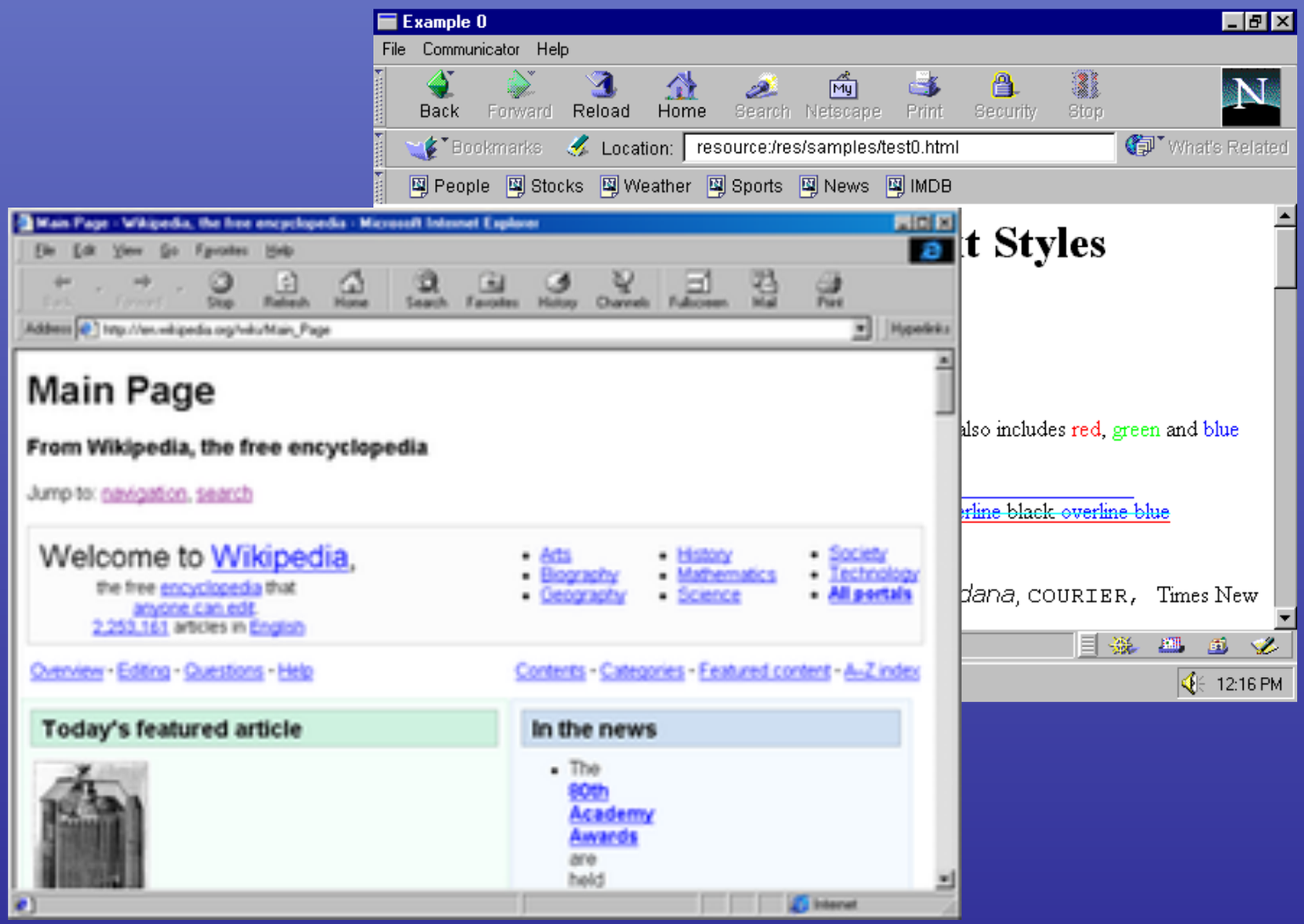
Brendan Eich

Keep Betting On JavaScript

Kyle Simpson
getify

another brief history

(of my journey into JavaScript)


Example 0
File Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: resource:/res/samples/test0.html What's Related
People Stocks Weather Sports News IMDB
t Styles
lso includes red, green and blue
erline black overline blue
dana, COURIER, Times New
12:16 PM
Main Page - Wikipedia, the free encyclopedia - Microsoft Internet Explorer
Main Page
From Wikipedia, the free encyclopedia
Jump to: navigation, search
Welcome to Wikipedia,
the free encyclopedia that anyone can edit
2,263,161 articles in English
Arts History Society
Biography Mathematics Technology
Geography Science All portals
Overview - Editing - Questions - Help
Contents - Categories - Featured content - A–Z index
Today's featured article
In the news
The 80th Academy Awards are held

00*

ext

Back

Forward

Reload

Save As...

Print...

Cast...

Translate to English

View Page Source

Inspect

```
34  {
35      documentHTMLText = "";
36  }
37
38  function Start_HTML_Layer(layerID,left_var,top
39  {
40      if ((ie4) || (ns6))
41      {
42          documentHTMLText += "<div id='" + layerI
43          documentHTMLText += "left:" + left_var +
44          documentHTMLText += "width:" + width_var
45          if (visibility_var != "") { documentHTML
46          if (zindex_var != -1) { documentHTMLText
47          if (backgroundcolor_var != "")
48          {
49              documentHTMLText += "background-color
50          }
51          documentHTMLText += "' ";
52      }
```

```
var image9Info = "BW009;SUMMER;Scratchboard/Col

var image1Array = image1Info.split(";");
var image2Array = image2Info.split(";");
var image3Array = image3Info.split(";");
var image4Array = image4Info.split(";");
var image5Array = image5Info.split(";");
var image6Array = image6Info.split(";");
var image7Array = image7Info.split(";");
var image8Array = image8Info.split(";");
var image9Array = image9Info.split(";");

function parseimgstr(strnum)
{
   return eval("image" + viewimagenum + "Array"
}

function imagex()
{
```

Book A Flight

Roundtrip | One Way

From: Where?

To: Where?

Accepting reservations through January 11,

Depart: NOV 5

December 2006

Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Close

Seniors

s 65+

Check Your Flight Status

View or Change Your Flight

hwest

e Your Travel
ySouthwest

LOG IN

View travel itineraries
Set online travel preferences
View Rapid Rewards credit details

Fly For Less

More

$99 or Less ONE-WAY

Philadelphia, PA
Fly for $99 or less one-way.
14-day advance purchase.
Book by Jun 30!

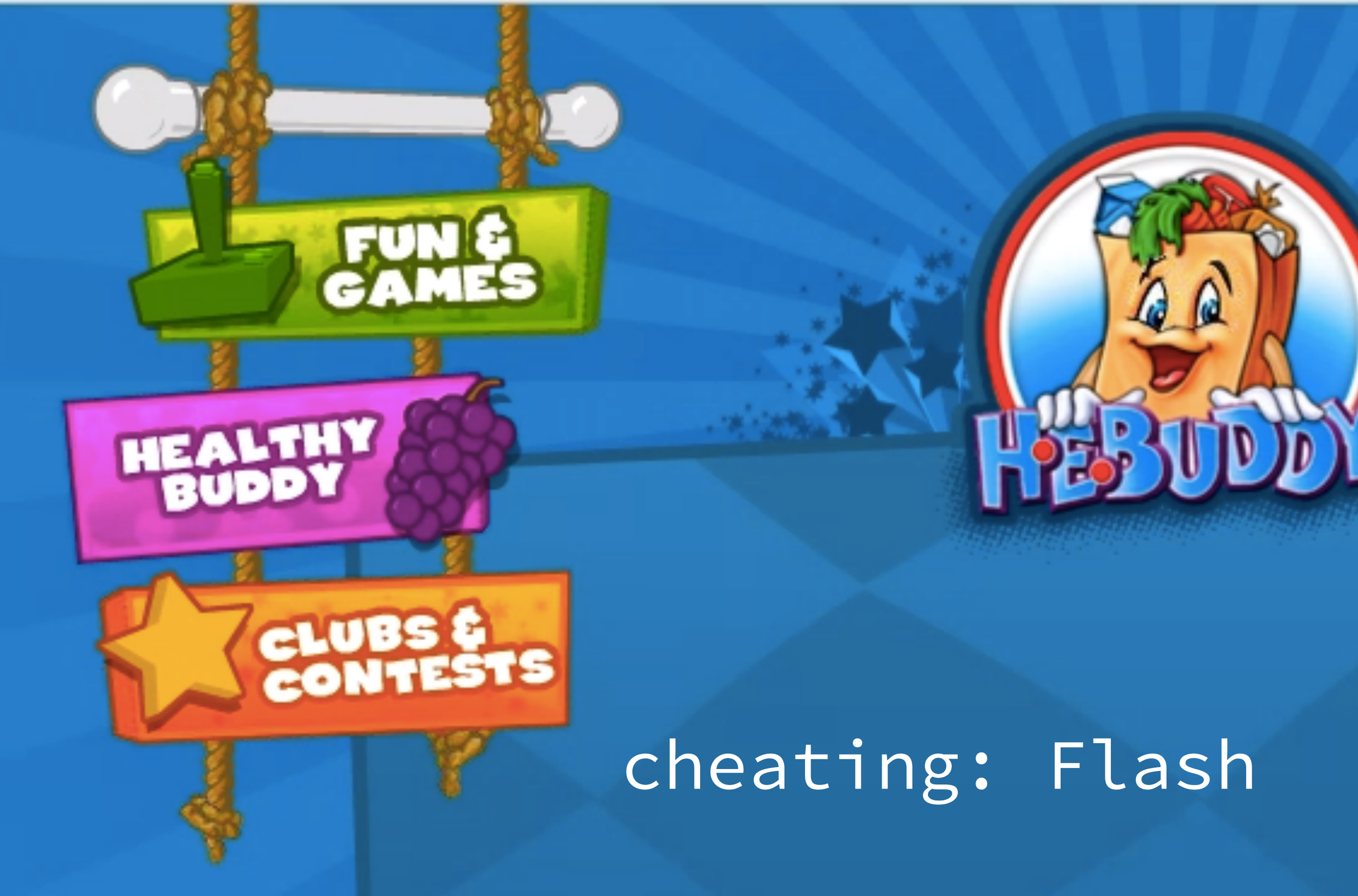
Starting at

Featured Destination: New

What's New

Support our Troop
Donate Now!
Help our military ph
with a pre-paid calli

://hebuddy.com



cheating: Flash

$("#menu")

```
function load_script(src,type,language) {
  for (var k=0; k<scrlen; k++) {
    if (typeof scriptArry[k].src !== UNDEF) {
      if (scriptArry[k].src.indexOf(src) >= 0) { break; }
    }
  }
  var scriptElem = doc.createElement("script");
  scriptElem.setAttribute("src",_flensed.base_path+src);
  if (typeof type !== UNDEF) { scriptElem.setAttribute("ty
  if (typeof language !== UNDEF) { scriptElem.setAttribute
  doc.getElementsByTagName("head")[0].appendChild(scriptEl
}

if ((typeof scriptArry !== UNDEF) && (scriptArry !== null)
  if (!base_path_known) {
    var idx=0;
```

flXHR

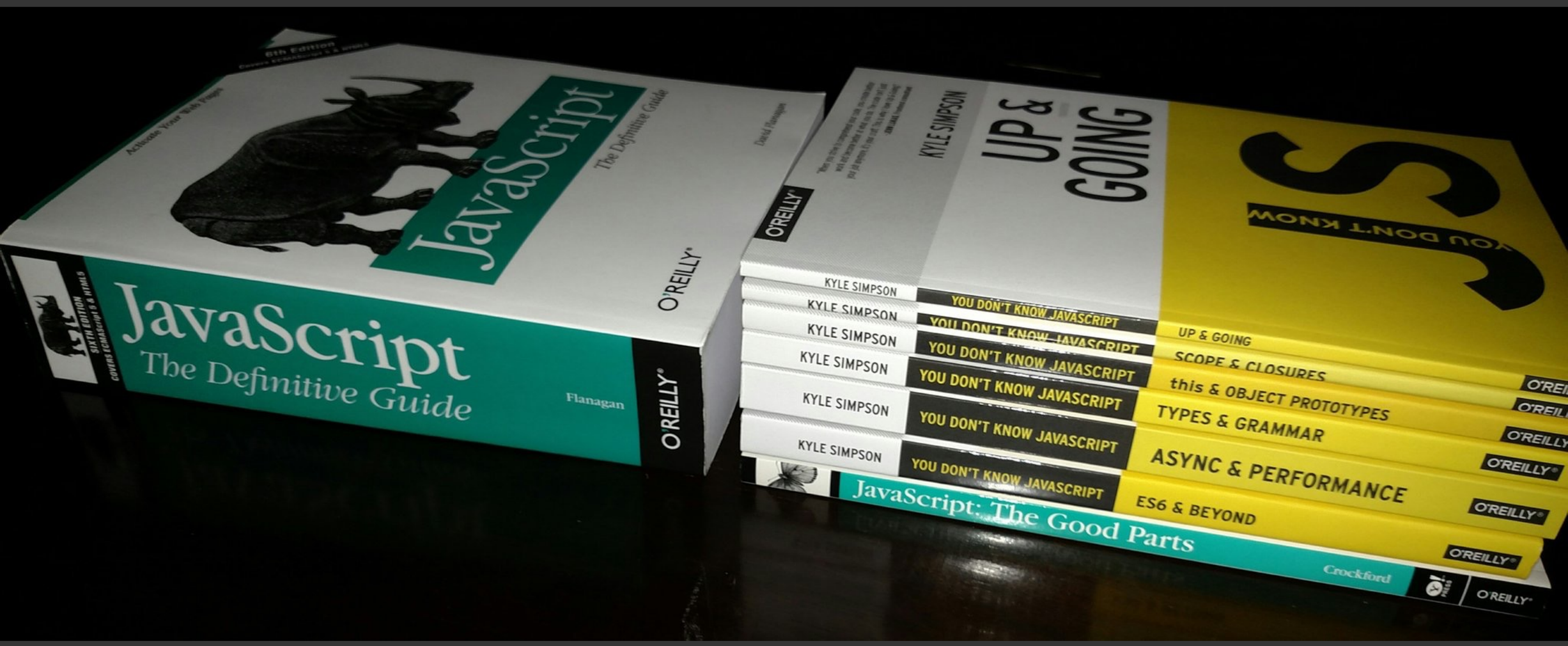
```
script:function(){
  for (var i=0; i<arguments.length; i++) {
    (function(script_obj,script_list){
      var splice_args;

      if (!is_array(script_obj)) {
        script_list = [script_obj];
      }
      for (var j=0; j<script_list.length; j++) {
        init_script_chain_group();
        script_obj = script_list[j];

        if (is_func(script_obj)) script_obj = script_obj();
        if (!script_obj) continue;
```

LABjs


JavaScript
The Definitive Guide
David Flanagan
O'REILLY
JavaScript
The Good Par
O'REILLY
YAHOO! PRESS


JavaScript
The Definitive Guide
JavaScript
The Good Par
Britannica
CLIFFS NOTES on
DICKENS' GREAT EXPECTATIONS
Cliffs

"You Don't Know JS" books



There, I fixed it for you.

lots of
baby steps

totally not a history lesson

(of JavaScript the language)

```
1  function unquote (str) {
2      if (str.length == 0) return str;
3      var i=0, j=0;
4      var ostr = "";
5      while ((i = str.indexOf('"',j)) != -1) {
6          ostr += str.substring(j,i) + '*';
7          j = i+1;
8      }
9      return ostr + str.substring(j,str.length);
10 }
```

ES-early

```
1  function Foo(x) {
2      this.val = x;
3  }
4
5  function identify() {
6      alert(this.val);
7  }
8
9  Foo.prototype.identify = identify;
10
11 var p = new Foo(42);
12
13 p.identify();
```

ES1/ES2

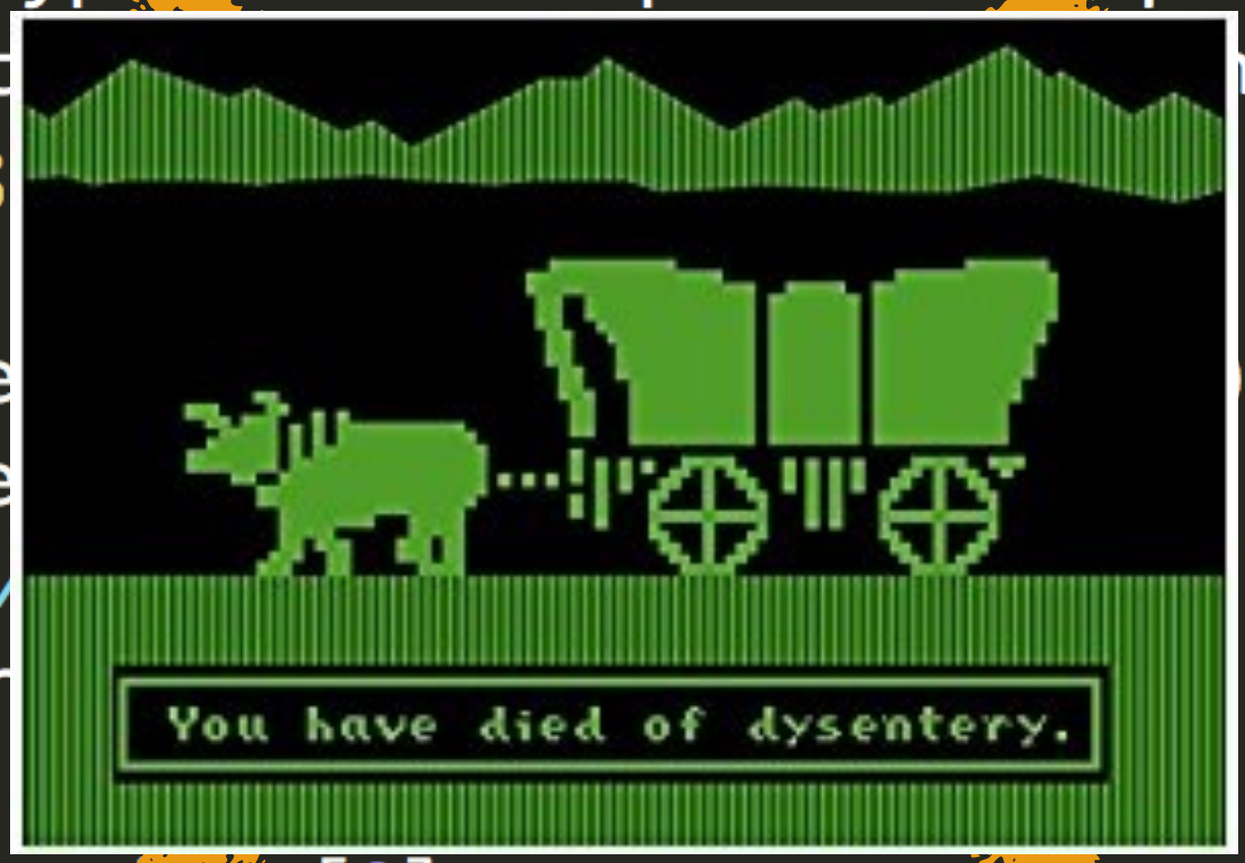
```
function foo(str) {
    return function(){
        try {
            var r = /^he/i;
            if (r.test(str) === false) {
                throw new Error("Oops!");
            }
            alert(str);
        }
        catch (err) {
            alert(err);
        }
    };
}

var f = foo("Hello World");
f();
```

ES3

```
1  package {
2
3      public class Foo {
4          var id:int
5
6          public function Foo(x:int):void {
7              secret(x);
8          }
9
10         public function identify() {
11             trace(id);
12         }
13
14         private function secret(x:int) {
15             id = x * 2;
16         }
17
18     }
19 }
```

~~ES4~~ (aka ActionScript3)

```
1  var sales = <sales vendor="John">
2        <item type="peas" price="4" quantity="6"/>
3        <item type="carrot" price="3" quantity="10"/>
4        <item t                                ntity="3"/>
5      </sales>;
6
7  alert( sale                                ).@quantity );
8  alert( sale
9  for each( v                                ) {
10     alert( pr
11 }
12 delete sales.item[0];
13 sales.item = <item type="oranges" price="4"/>;
14 sales.item.(@type == "oranges").@quantity = 4;
```



~~E4X~~ (totally not ES4)

```
1  "use strict";
2
3  var Foo = {
4      get x() { return this.__x; },
5      set x(v) { this.__x = v; }
6  };
7
8  function Bar(v) {
9      this.x = v;
10 }
11
12 Bar.prototype = Object.create(Foo);
13
14 Bar.prototype.identify = function(){
15     console.log(this.x);
16 };
17
18 var p = new Bar(42);
19
20 p.identify();
```

ES5

```
var Foo = {
    *[Symbol.iterator]() {
        for (let [idx,val] of this.nums.entries()) {
            yield label =>
                `${label}: ${val}
                    (${idx+1} / ${this.nums.length})`;
        }
    },
    bar([x = 1,y = 2,...nums] = []) {
        this.nums = [x,y,...nums];
    }
};

Foo.bar([10,20,30,40,50]);

for (let f of Foo) {
    console.log( f("Foo") );
}
```

ES6 (aka ~~ES2015~~)

```
var nums = [];

for (let i = 1; i < 5; i++) {
    nums.push( i ** (i+1) );
}

if (nums.includes( 81 )) {
    console.log("Yay!");
}
```

ES2016 (aka ~~ES7~~)

```
async function foo(obj,) {
    for (let [prop,val,] of Object.entries(obj)) {
        let x = await val;
        console.log(
            `| ${prop.padStart(5)} | ${x.padEnd(10)} |`
        );
    }
}

var prs = {
    a: Promise.resolve("hello"),
    bc: Promise.resolve("friends"),
};

foo(prs,);
```

ES2017 (aka ~~ES8~~)

```
async function* gimmeStuff(arr) {
    for (let v of arr) {
        let { ok, ...data } = await fetchData(v);
        if (ok) {
            yield data;
        }
    }
}

async function main() {
    for await (let v of gimmeStuff([1,2,3])) {
        let data = { id: uniqID(), ...v, };
        let re = /(?<!Other )(Name: (?<name>\w+))/;
        let { groups: { name } = {} } =
            data.message.match(re);
        console.log(`User: ${name} (${id})`);
    }
}
```

ES2018 (aka ~~ES9~~)


...blah, blah...

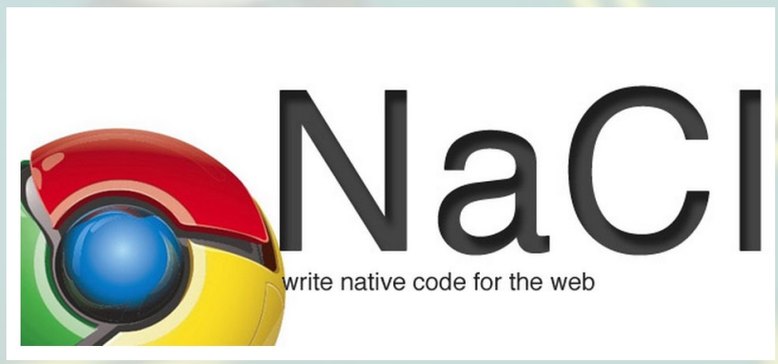
best progress

is incremental

place your bets?


activex


Java


NaCl
write native code for the web

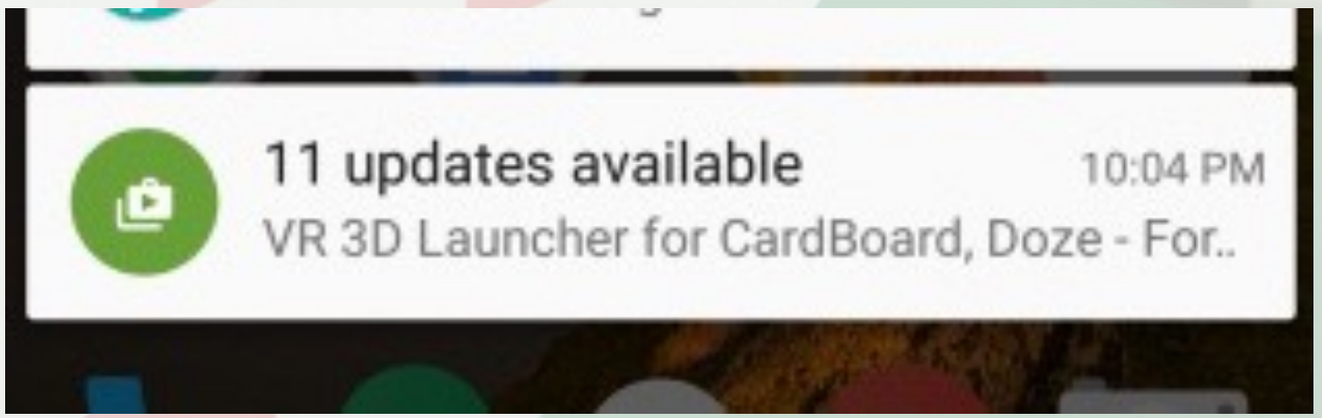
JS

HTML

CSS

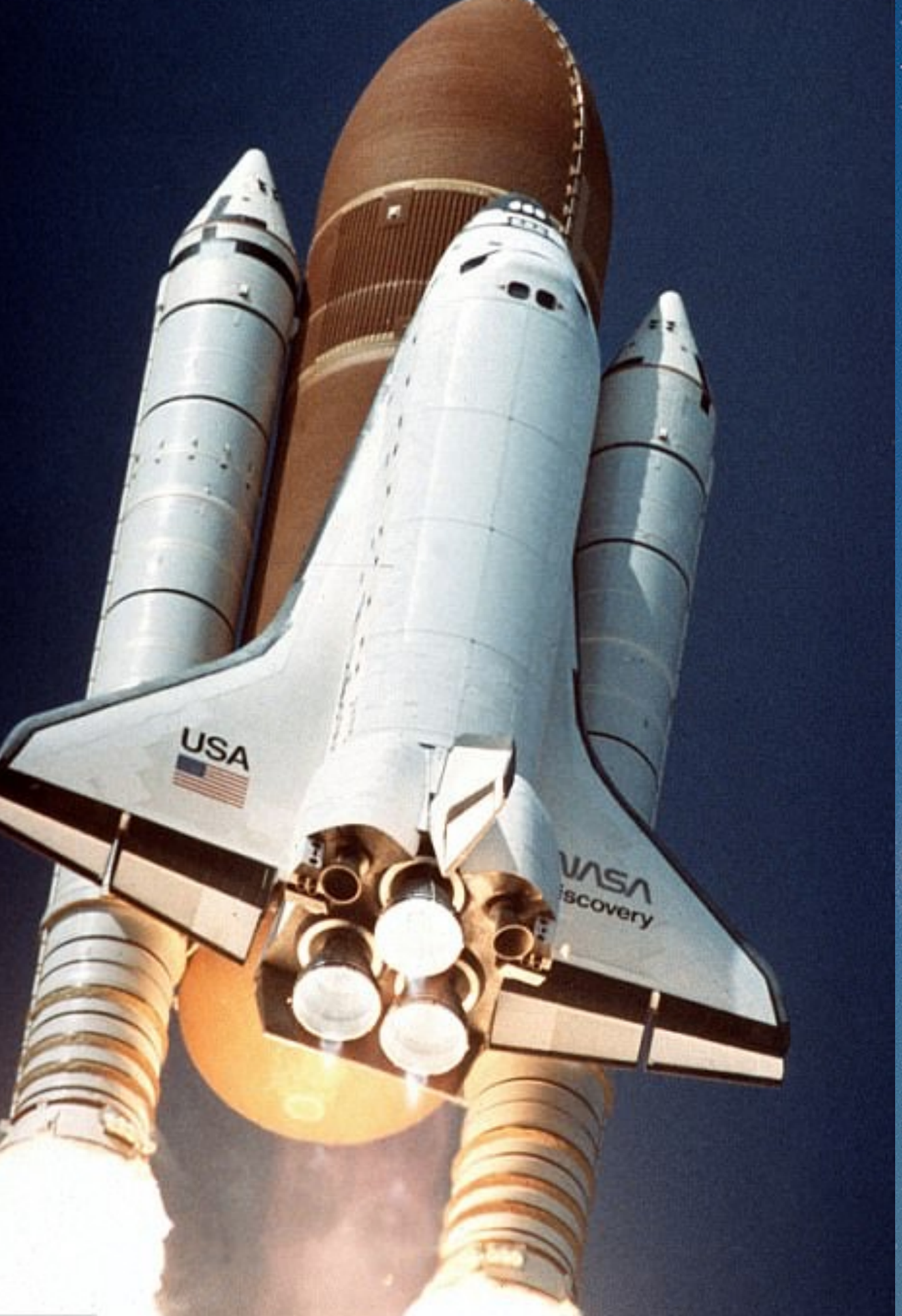
PWA

PWA


11 updates available
10:04 PM
VR 3D Launcher for CardBoard, Doze - For..

open collaboration

...a good bet


USA
NASA
Discovery


SPACE
X

language

JS

machine

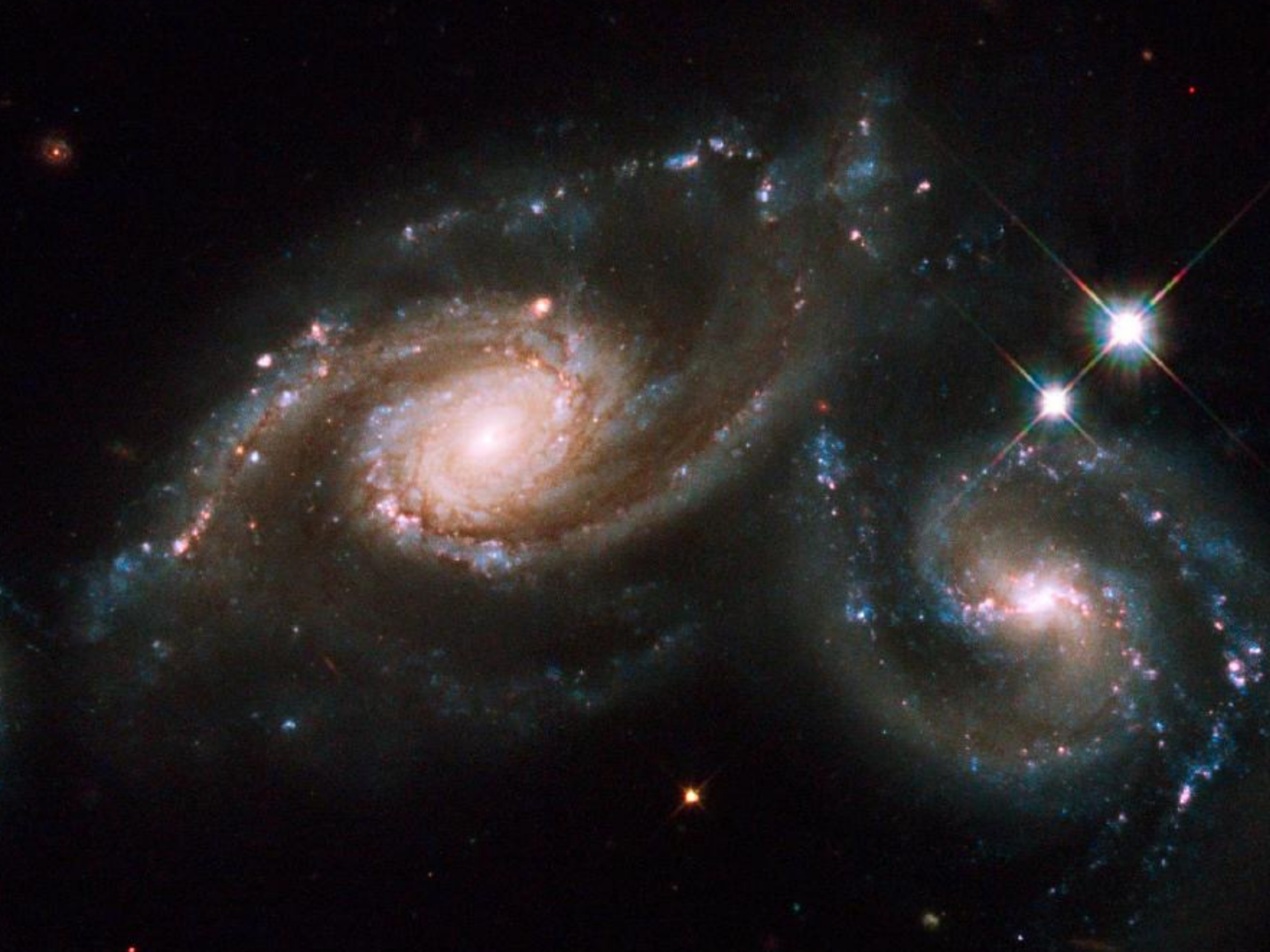
JS

language

machine

JS

JS



incremental

open

always good?

language

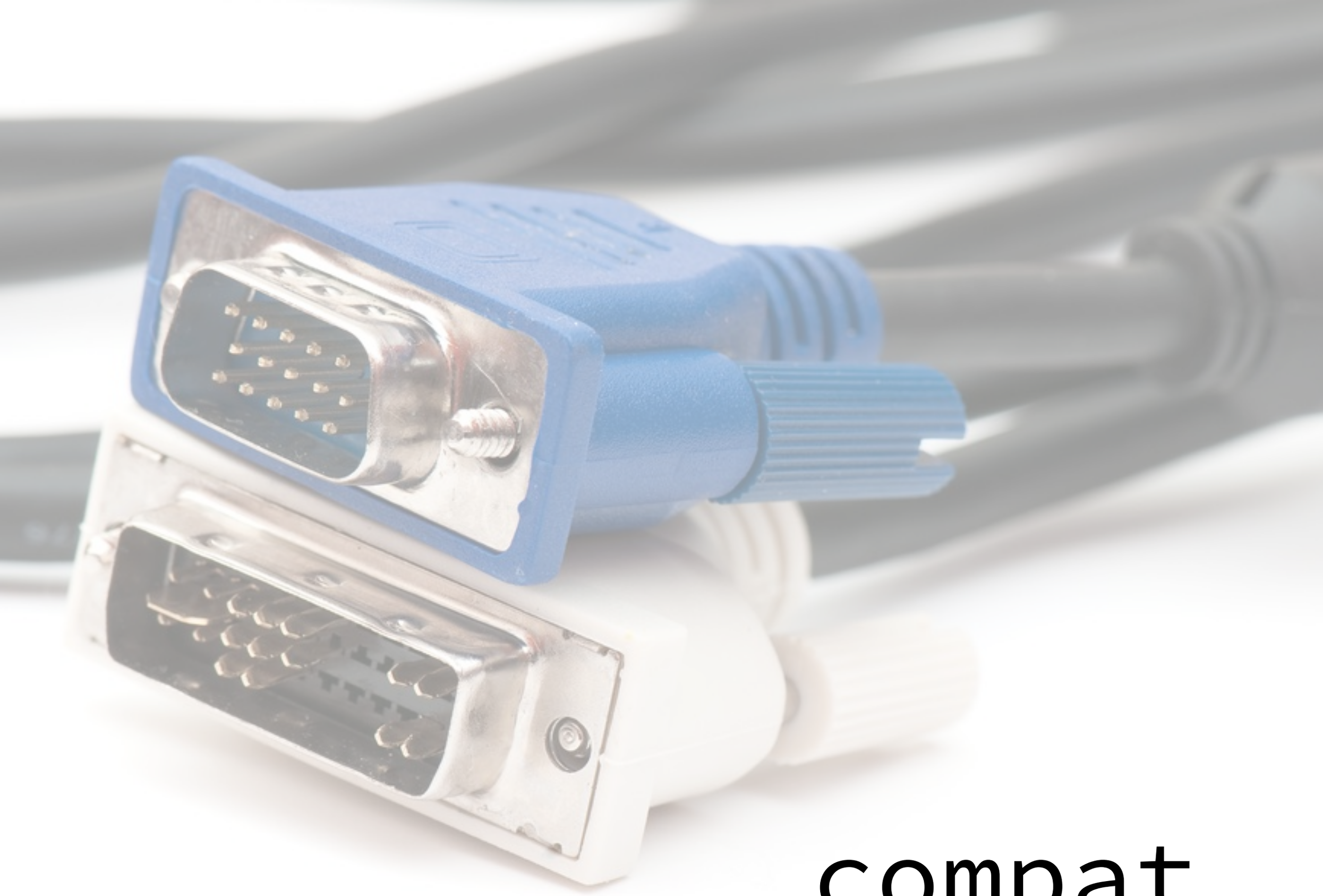
JS


code is
human communication

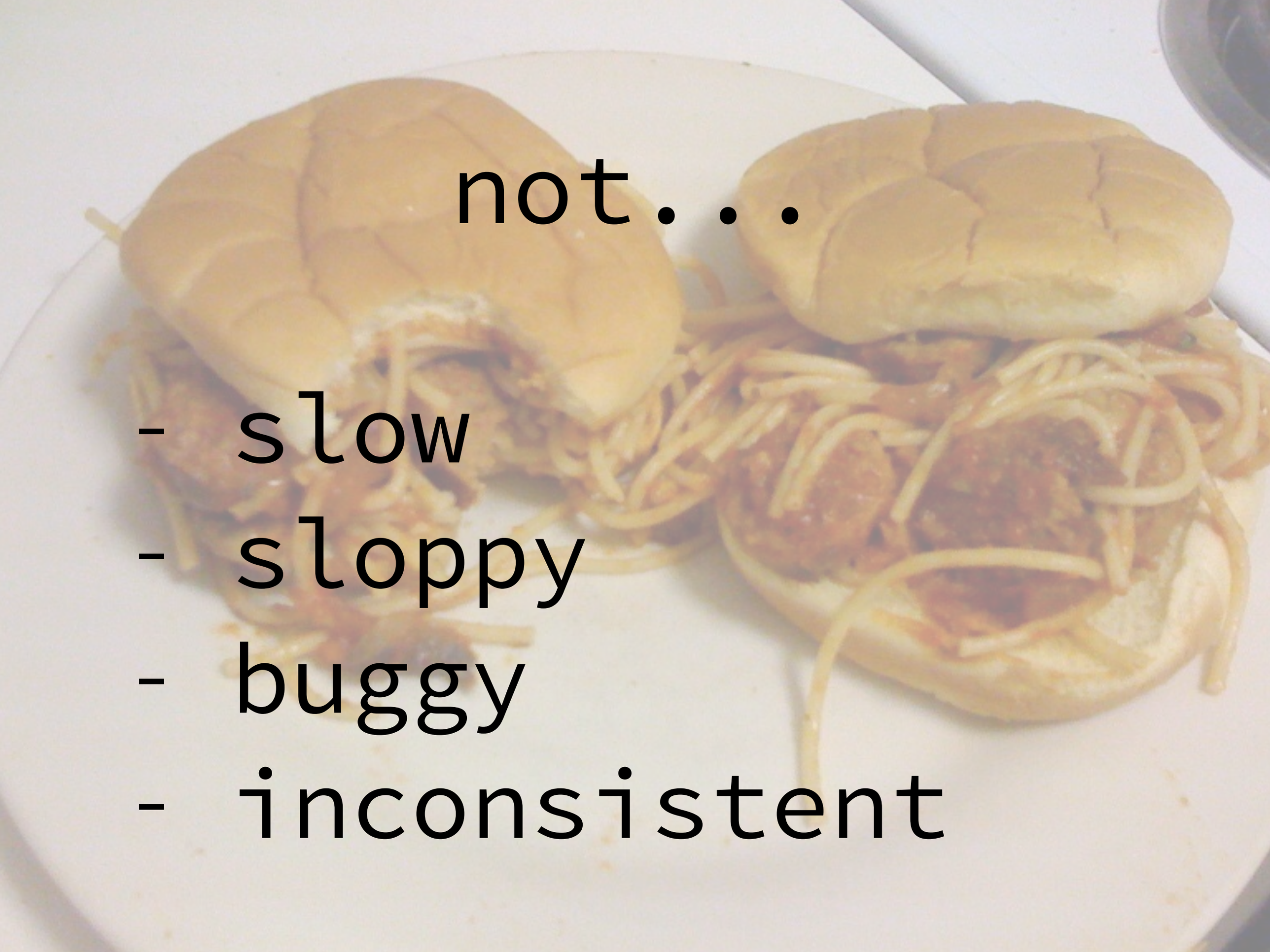
lingua franca

multi-paradigm

declarative



compat



not...

- slow
- sloppy
- buggy
- inconsistent

coming soon to a JavaScript near you

...probably...maybe

```
1  class Foo {
2      counter = 0;
3
4      gimmeTheCount() {
5          return this.counter++;
6      }
7  }
8
9  var x = new Foo();
10
11 x.gimmeTheCount();   // 0
12 x.gimmeTheCount();   // 1
13 x.counter;           // 2
```

class fields

```
1  class Foo {
2      #counter = 0;
3
4      gimmeTheCount() {
5          return this.#counter++;
6      }
7  }
8
9  var x = new Foo();
10
11 x.gimmeTheCount();   // 0
12 x.gimmeTheCount();   // 1
13 x.counter;           // undefined
```

private fields

```
class Foo {
    #counter = 0;

    gimmeTheCount() {
        this.#inc();
        return this.#counter;
    }

    #inc() { this.#counter++; }
}

var x = new Foo();

x.gimmeTheCount();  // 1
x.gimmeTheCount();  // 2
```

private methods

```
let parseResult = someFallbackValue;

try {
    parseResult = JSON.parse(val);
}
catch {}
```

optional catch binding

```
1 var x = (foo && foo.bar) ?
2                 foo.bar.x :
3                 undefined;
4
5 var y = foo?.bar?.y;
```

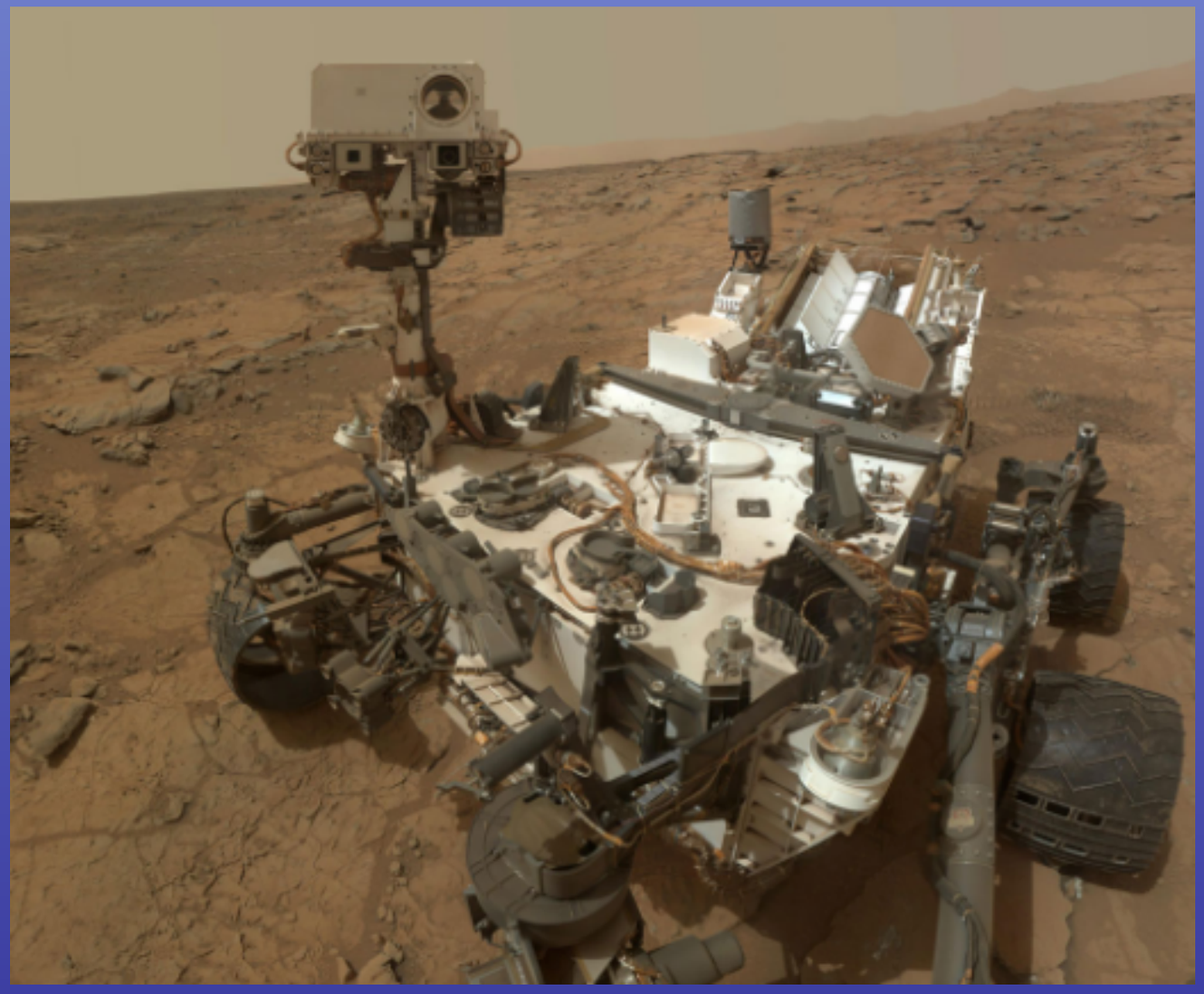
optional chaining

```
1 var x = 10000000000;
2
3 var y = 1_000_000_000;
4
5 x === y;      // true
```

readable numbers

```
1 var reallyBig = 2n ** 60n;
2
3 var x = 42;  // not a BigInt
4
5 reallyBig - BigInt(x);
6 // 1152921504606846934n
```

BigInt



from 10-days to now...

machine

JS

~~assembly~~

[redacted]

increased demands

Proxy

```
1  var obj = { a: 1 },
2
3      handlers = {
4          get(target,key,context) {
5              console.log(`accessing: ${key}`);
6              return Reflect.get(target,key,context);
7          }
8      },
9
10     pobj = new Proxy(obj,handlers);
11
12 obj.a;
13 // 1
14
15 pobj.a;
16 // accessing: a
17 // 1
```

TypedArray/DataView

```
1 var littleEndian = (function testEndianness(){
2     var buffer = new ArrayBuffer(2);
3     new DataView(buffer).setInt16(0,256,true);
4     return new Int16Array(buffer)[0] === 256;
5 })();
```

shared memory!?!

```
1  var worker = new Worker("worker.js");
2  var count = 10;
3
4  var sharedBuffer = new SharedArrayBuffer(
5      Int32Array.BYTES_PER_ELEMENT * count
6  );
7  var sharedArray = new Int32Array(sharedBuffer);
8
9  for (let i = 0; i < count; i++) {
10     sharedArray[i] = i * 2;
11 }
12
13 worker.postMessage(sharedBuffer);
```

shared memory!?!

```
1 self.addEventListener("message",function onmessage(m){
2     var sharedBuffer = m.data;
3     var sharedArray = new Int32Array(sharedBuffer);
4
5     console.log(sharedArray[3]);     // 6
6 });
```

shared memory!?!

```
 9 for (let i = 0; i < count; i++) {
10     sharedArray[i] = i * 2;
11 }
12
13 worker.postMessage(sharedBuffer);
14
15 while (sharedArray[0] === 0);
16
17 console.log(`Answer: ${sharedArray[0]}`);
```

Atomics

```
1 Atomics.store(sharedArray,0,42);
2 Atomics.wake(sharedArray,0,1);
3
4 // ..
5
6 Atomics.wait(sharedArray,0,0);
7 var answer = Atomics.load(sharedArray,0);
```

BABEL

```
async function foo(x) {
  var y = await x;
  console.log(y);
}
```

```
"use strict";

var foo = function () {
  var _ref = _asyncToGenerator( /*#__PURE__*/regeneratorR
    var y;
    return regeneratorRuntime.wrap(function _callee$(_con
      while (1) {
        switch (_context.prev = _context.next) {
          case 0:
            _context.next = 2;
            return x;

          case 2:
            y = _context.sent;

            console.log(y);

          case 4:
          case "end":
            return _context.stop();
        }
      }
    }, _callee, this);
  }));

  return function foo(_x) {
    return _ref.apply(this, arguments);
  };
}();

function _asyncToGenerator(fn) { return function () { var
```

sourcemaps

```
{
    version: 3,
    file: "script.js.map",
    sources: [
        "app.js",
        "content.js",
        "widget.js"
    ],
    sourceRoot: "/",
    names: ["slideUp", "slideDown", "save"],
    mappings: "AAA0B,kBAAhBA,QAAOC,SAAOC,OAAOC..."
}
```

sourcemaps

```
{
    version: 3,
    file: "script.js.map",
    sources: [
      "app.js",
      "content.js",
      "widget.js"
    ],
    sourceRoot: "/",
    names: ["slideUp", "slideDown", "save"],
    mappings: "AAA0B,kBAAhBA,QAAOC,SACjBD,OAAOC,OAAO..."
}
```

Problems:

- stack traces
- watch expressions

TypeScript

```typescript
interface Person {
    firstName: string;
    lastName: string;
}

function greeter(person: Person) {
    return `Hello, ${person.firstName} ${person.lastName}`;
}

let user = { firstName: "Jane", lastName: "User" };

greeter(user);
```

macros


```
syntax let = function (ctx) {
  let ident = ctx.next().value;
  ctx.next(); // eat `=`
  let init = ctx.expand('expr').value;
  return #`
    (function (${ident}) {
      ${ctx} // <2>
    }(${init}))
  `
};
```


```
1 for (var i = 0; i < 5; i++) {
2     let j = i;
3     console.log(j);
4 }
```

```
1 for (var i = 0; i < 5; i++) {
2     (function(j){
3         console.log(j);
4     })(i);
5 }
```

Tom Dale

tomdale.net/2017/09/compilers-are-the-new-frameworks/

experimentation in the wild?

```
1 function foo(x)(y) {
2     return x + y;
3 }
4
5 foo = (x)(y) => x + y;
6 // foo = x => y => x + y;
7
8 var f = foo(3);
9
10 f(4);          // 7
11 f(10);         // 13
```

yes! eh, not so much :(

asm.js

```
1 // calculate length of C string
2 function strlen(ptr) {
3     ptr = ptr|0;
4     var curr = 0;
5     curr = ptr;
6     while (MEM8[curr]|0 !== 0) {
7         curr = (curr + 1)|0;
8     }
9     return (curr - ptr)|0;
10 }
```

WA

WebAssembly

```
(module
  (memory (export "mem") 1)
  (func (export "accumulate") (param $ptr i32) (param $len i32) (result i32)
    (local $end i32)
    (local $sum i32)
    (set_local $end (i32.add (get_local $ptr) (i32.mul (get_local $len) (i32.const 4))))
    (block $break (loop $top
      (br_if $break (i32.eq (get_local $ptr) (get_local $end)))
      (set_local $sum (i32.add (get_local $sum)
                               (i32.load (get_local $ptr))))
      (set_local $ptr (i32.add (get_local $ptr) (i32.const 4)))
      (br $top)
    ))
    (get_local $sum)
  )
)
```

r Phone

$500*

n your next

Back
Forward
Reload

Save As...
Print...
Cast...
Translate to English

View Page Source
Inspect

language

JS

a tale of...

machine

JS

...two JavaScripts

language

JS

...decent bets

so far...

machine

JS

language

JS

...decent bets

so far...

machine

JS

where's your bet?

speakerdeck.com/getify/keep-betting-on-js

thanks

Kyle Simpson
getify