ES6, ES7, WHERE DO I START???

OH CRAP

JAVASCRIPT CHANGED AGAIN!

HI, I'M RAYMOND!

- Developer Advocate for IBM
- Focused on Node, API, Serverless, and Enterprise Cat Demos
- Blogging at www.raymondcamden.com
- Tweeting at @raymondcamden
- I have opinions.

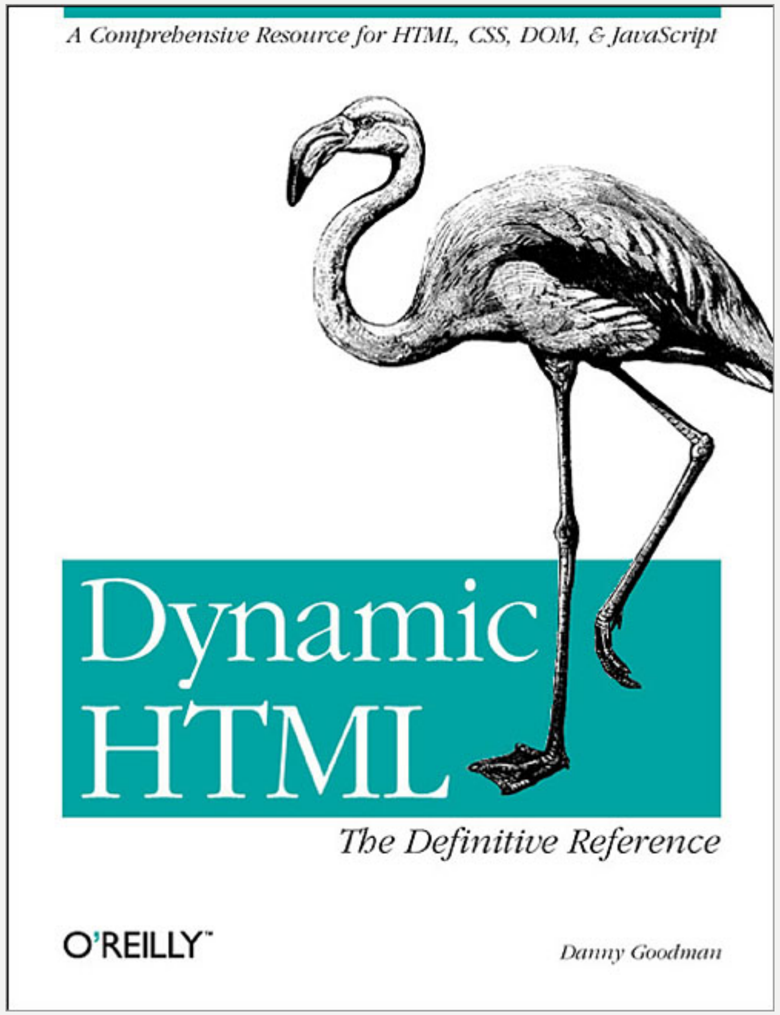
HOW AWESOME I AM AT JAVASCRIPT

- My first demos were epic…

N
Netscape
Navigator 2.02
Copyright © 1995, 1996 Netscape
Communications Corp. All rights reserved

HOW AWESOME I AM AT JAVASCRIPT

- Professional DHTML Expert



HOW AWESOME I AM AT JAVASCRIPT

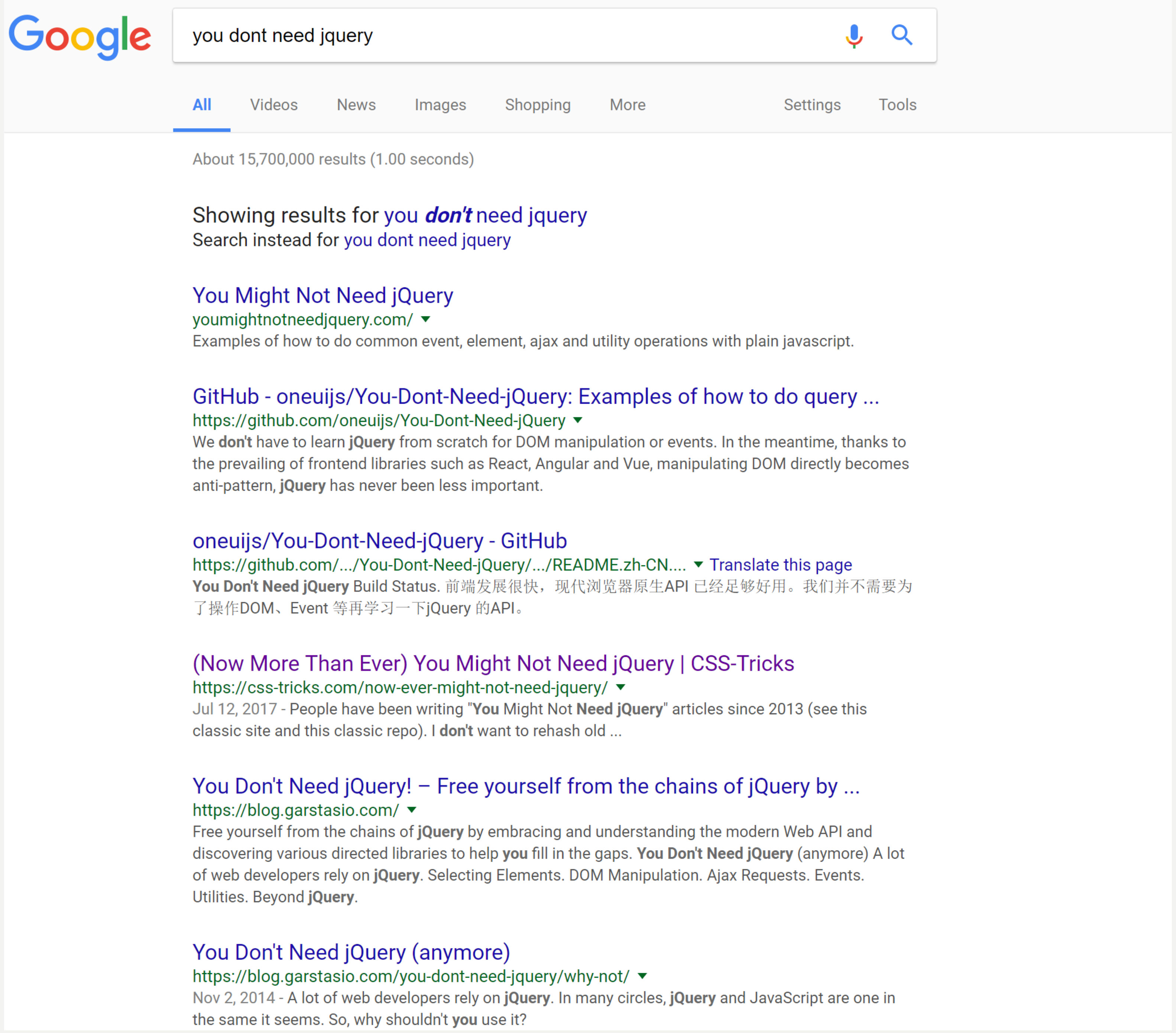
- Proud Failer (failee?) of the Google Code Interview!

OH THE THINGS I'VE SEEN
memecrunch.com

WHO ARE YOU?

TWO TYPES OF DEVELOPERS

- JavaScript Developers
- Developers who use JavaScript



About 15,700,000 results (1.00 seconds)

Showing results for you ***don't*** need jquery
Search instead for you dont need jquery

You Might Not Need jQuery
youmightnotneedjquery.com/
Examples of how to do common event, element, ajax and utility operations with plain javascript.

GitHub - oneuijs/You-Dont-Need-jQuery: Examples of how to do query ...
https://github.com/oneuijs/You-Dont-Need-jQuery
We **don't** have to learn **jQuery** from scratch for DOM manipulation or events. In the meantime, thanks to the prevailing of frontend libraries such as React, Angular and Vue, manipulating DOM directly becomes anti-pattern, **jQuery** has never been less important.

oneuijs/You-Dont-Need-jQuery - GitHub
https://github.com/.../You-Dont-Need-jQuery/.../README.zh-CN.... Translate this page
You Don't Need jQuery Build Status. 前端发展很快，现代浏览器原生API 已经足够好用。我们并不需要为了操作DOM、Event 等再学习一下jQuery 的API。

(Now More Than Ever) You Might Not Need jQuery | CSS-Tricks
https://css-tricks.com/now-ever-might-not-need-jquery/
Jul 12, 2017 - People have been writing "**You** Might Not **Need jQuery**" articles since 2013 (see this classic site and this classic repo). I **don't** want to rehash old ...

You Don't Need jQuery! – Free yourself from the chains of jQuery by ...
https://blog.garstasio.com/
Free yourself from the chains of **jQuery** by embracing and understanding the modern Web API and discovering various directed libraries to help **you** fill in the gaps. **You Don't Need jQuery** (anymore) A lot of web developers rely on **jQuery**. Selecting Elements. DOM Manipulation. Ajax Requests. Events. Utilities. Beyond **jQuery**.

You Don't Need jQuery (anymore)
https://blog.garstasio.com/you-dont-need-jquery/why-not/
Nov 2, 2014 - A lot of web developers rely on **jQuery**. In many circles, **jQuery** and JavaScript are one in the same it seems. So, why shouldn't **you** use it?

WHAT THE HECK IS GOING ON?

NO - SERIOUSLY

ES6

ES2016

ES7

ES2016

ES2017

ES6

ES2016

ES7

ES2016

ES2017

ECMASCRIPT

- The standard for JavaScript (and other scripting languages)
- Began way back in 1997
- Provides a "Bible" for JavaScript

ES6

ES2015

ES7

ES2016

ES2017

DATES

- ES6/ES2015 - finalized 2015
- ES7/ES2016 - finalized 2016
- ES8/ES2017 - finalized 2017

SYNTAX VERSUS "BROWSER FEATURES"

- x = [1,2,3]
  - x = new Array(3); x.push(1); x.push(2); x.push(3);
- navigator.serviceWorker

NOT EVERYTHING IS ABOUT YOU...

- Some language features are awesome!
- Some are awesome and you'll never use em!
- That's ok!

READ THE SPEC

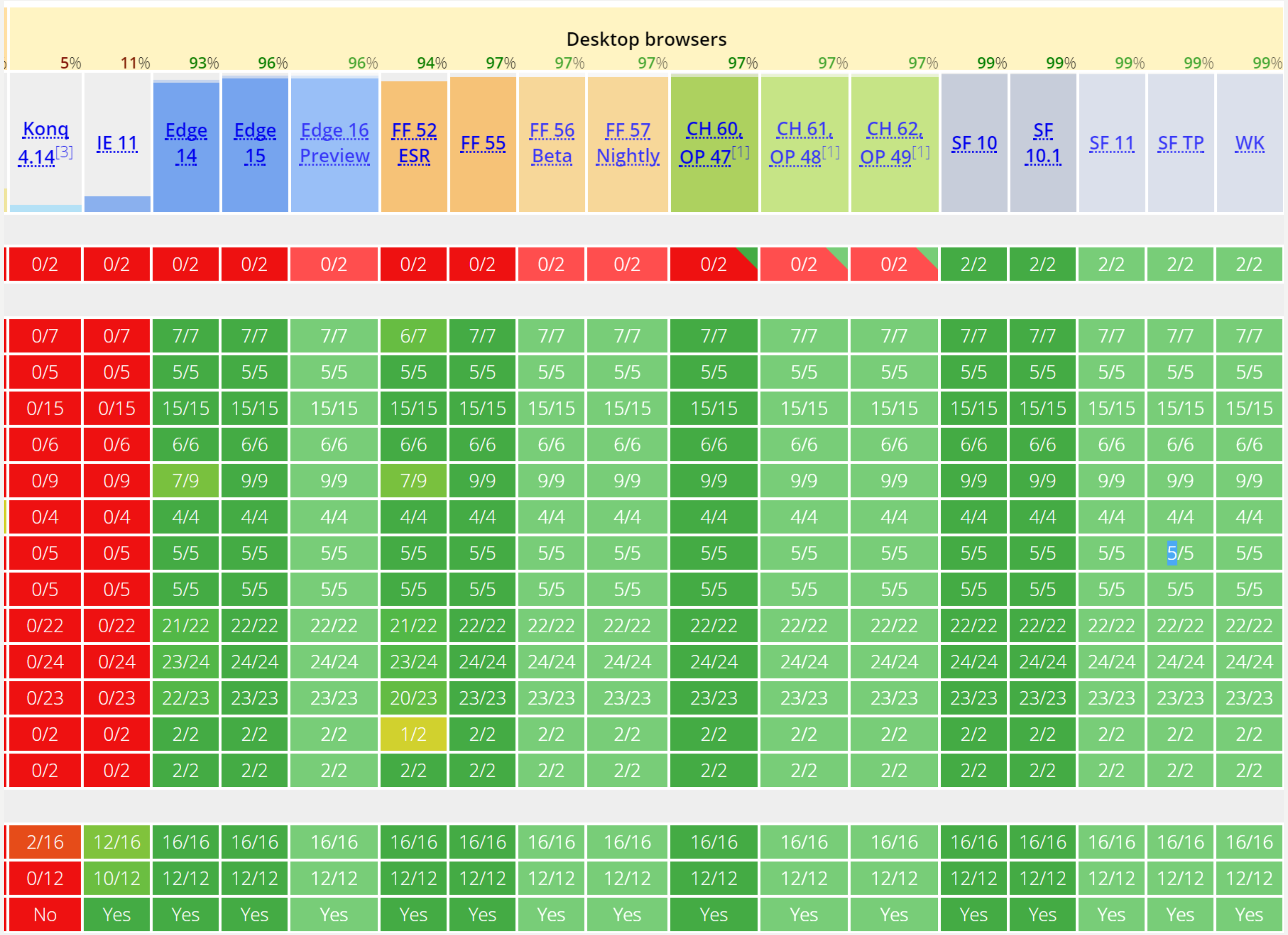
- ES6: http://www.ecma-international.org/ecma-262/6.0/
- ES7 (ES2016): https://www.ecma-international.org/ecma-262/7.0/
- ES8 (ES2017): https://www.ecma-international.org/ecma-262/8.0/

READ THE BLOGS/BOOKS

- Dr. Axel Rauschmayer
  - http://2ality.com/
  - http://exploringjs.com/es6/
  - http://exploringjs.com/es2016-es2017/
- Wes Bos
  - ES6 for Everyone (https://es6.io/)
- Eric Elliott
  - https://medium.com/javascript-scene/how-to-learn-es6-47d9a1ac2620

BUT CAN I ACTUALLY USE IT?

- http://kangax.github.io/compat-table/es6/

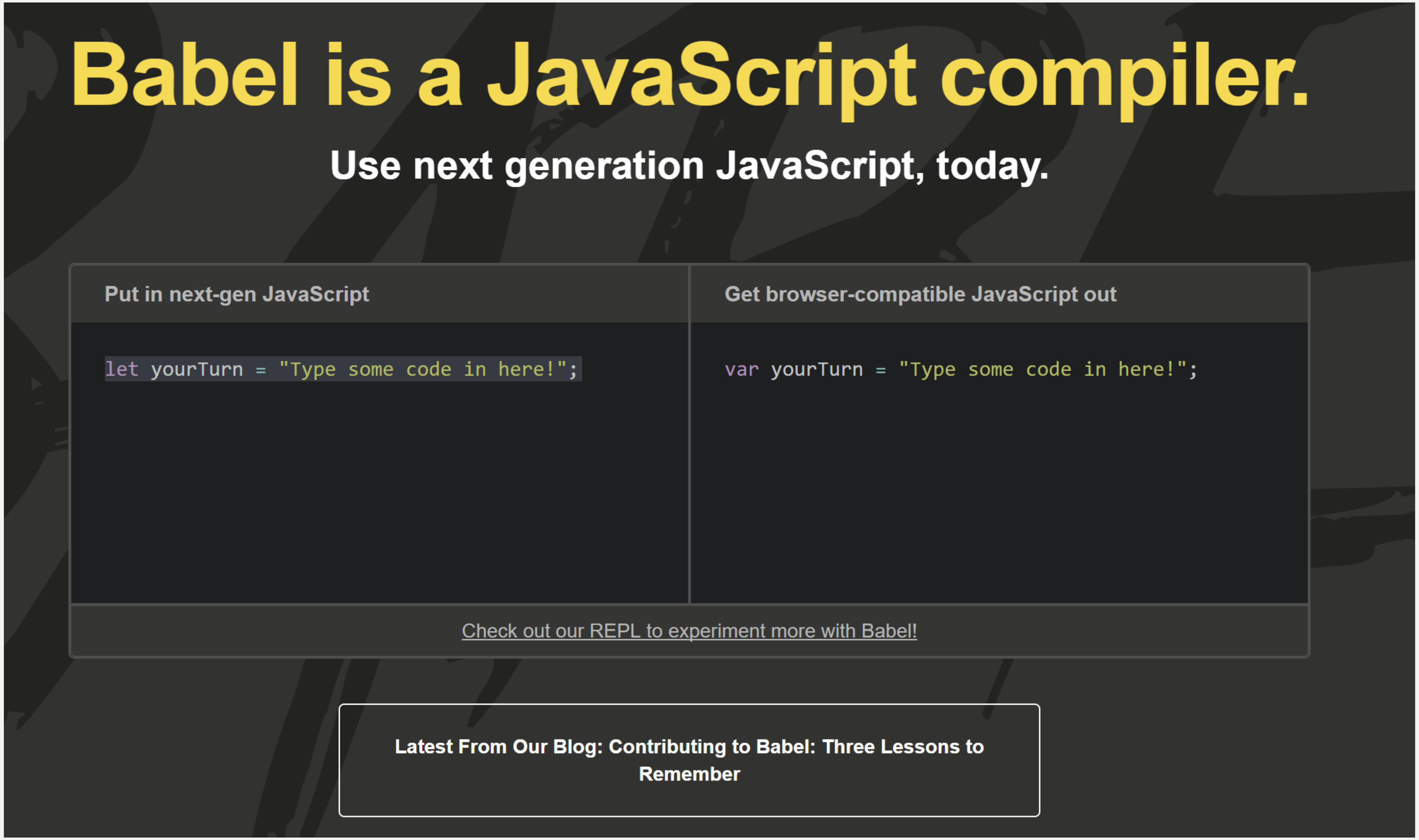


Desktop browsers

Konq 4.14[3]	IE 11	Edge 14	Edge 15	Edge 16 Preview	FF 52 ESR	FF 55	FF 56 Beta	FF 57 Nightly	CH 60, OP 47[1]	CH 61, OP 48[1]	CH 62, OP 49[1]	SF 10	SF 10.1	SF 11	SF TP	WK
5%	11%	93%	96%	96%	94%	97%	97%	97%	97%	97%	97%	99%	99%	99%	99%	99%
0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	2/2	2/2	2/2	2/2	2/2
0/7	0/7	7/7	7/7	7/7	6/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7
0/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5
0/15	0/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15
0/6	0/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6
0/9	0/9	7/9	9/9	9/9	7/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9
0/4	0/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4
0/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5
0/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5
0/22	0/22	21/22	22/22	22/22	21/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22
0/24	0/24	23/24	24/24	24/24	23/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24
0/23	0/23	22/23	23/23	23/23	20/23	23/23	23/23	23/23	23/23	23/23	23/23	23/23	23/23	23/23	23/23	23/23
0/2	0/2	2/2	2/2	2/2	1/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2
0/2	0/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2
2/16	12/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16
0/12	10/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12
No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

BUT CAN I ACTUALLY USE IT?

- Checking Support: http://kangax.github.io/compat-table/es6/
- Transpiling: https://babeljs.io/



Babel is a JavaScript compiler.

Use next generation JavaScript, today.

Put in next-gen JavaScript

```
let yourTurn = "Type some code in here!";
```

Get browser-compatible JavaScript out

```
var yourTurn = "Type some code in here!";
```

Check out our REPL to experiment more with Babel!

Latest From Our Blog: Contributing to Babel: Three Lessons to Remember

BUT CAN I ACTUALLY USE IT?

- Checking Support: http://kangax.github.io/compat-table/es6/
- Transpiling: https://babeljs.io/
- Checking Support (2): http://caniuse.com/

MY FAVORITE FEATURES*

- Template Literals (aka template strings)
- Arrow Functions
- Async/Await

* Subject to change!

TEMPLATE LITERALS

- A new way to define strings in JavaScript
- Help solve the problem of creating dynamic strings

TEMPLATE LITERALS

```
var name = "ray";
var age = 44;
var status = "cool";
var desc = "My name is <b>"+name+"</b>";
desc += " and I'm <i>"+age+"</i> years old and ";
desc += " and I'm currently <blink>" + status +
"</blink>!";
```

TEMPLATE LITERALS

```
var name = "ray";
var age = 44;
var status = "cool";
var desc = `
My name is <b>${name}</b> and I'm
<i>${age}</i> years old and and I'm
currently <blink>${status}</blink>!
`;
```

ARROW FUNCTIONS

- Shorter (simpler?) syntax
- Properly handle "this" inside

ARROW FUNCTIONS

```
hello = function(name) {
  return `Hello, ${name}`;
}

hello = (name) => `Hello, ${name}`;
```

ARROW FUNCTIONS

```
hello = (name, greeting) => {
   let result = `${greeting}, ${name}`;
   return result;
};
```

ARROW FUNCTIONS

```
function Person() {
  this.age = 0;

  setInterval(function growUp() {
  this.age++;
  }, 1000);
}

var p = new Person();
```

Source: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions

ARROW FUNCTIONS

```
function Person() {
  this.age = 0;

  setInterval(() => {
  this.age++;
  }, 1000);
}

var p = new Person();
```

Source: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions

ASYNC/AWAIT

- Working with Async is hard (everyone knows this)
- Callbacks, Promises, Observables

ASYNC/AWAIT

```
function slow1() {
  return new Promise((resolve, reject)=> {
    window.setTimeout(() => {
      resolve(1)
    }, 1000);
  });
}

function slow2() {
  return new Promise((resolve, reject)=> {
    window.setTimeout(() => {
      resolve(2)
    }, 1000);
  });
}
```

ASYNC/AWAIT

```
function doslow() {
    let total = 0;
    Promise.all([
        slow1(),slow2()
    ]).then(values => {
        total = values[0] + values[1];
        console.log(`total is ${total}`);
  });
}
```

ASYNC/AWAIT

```
async function doslow2() {
    let total = 0;
    total += await slow1();
    total += await slow2();
    console.log(`total is ${total}`);
}
```

THANK YOU!