

JavaScript

00110001	000000000	000000000
00110001	000000001	000000001
00110011	000000001	000000010
01010001	00001011	000000010
00100010	000000010	00001000
01000011	000000001	000000000
01000001	000000001	000000001
00010000	000000010	000000000
01100010	000000000	000000000

Set "total" to 0.

Set "count" to 1.

[loop]

Set "compare" to "count".

Subtract 11 from "compare".

If "compare" is zero, continue at [end].

Add "count" to "total".

Add 1 to "count".

Continue at [loop].

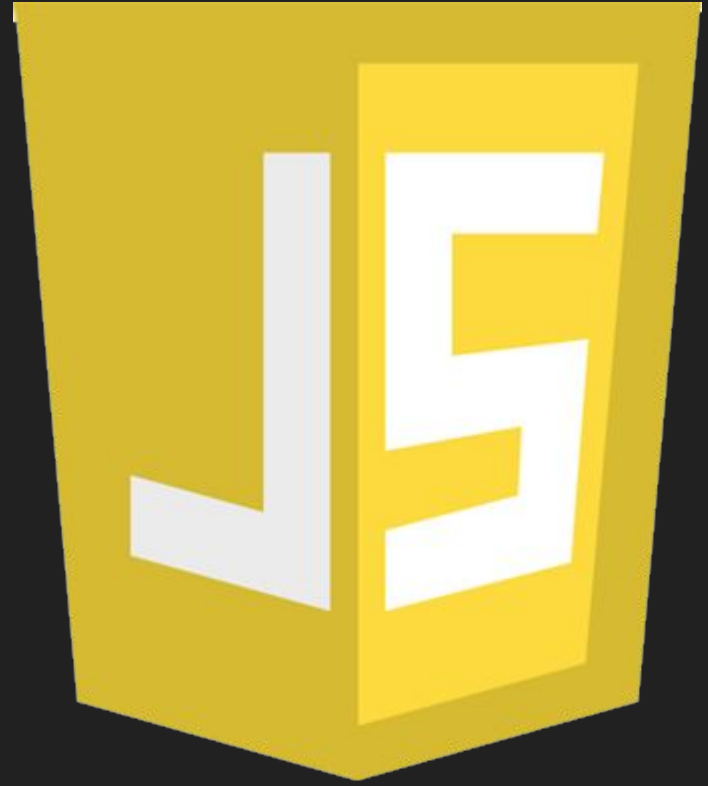
[end]

Output "total".

```
var total = 0, count = 1;  
while (count <= 10) {  
    total += count;  
    count += 1;  
}  
console.log(total);  
// → 55
```

Javascript

ES6



The story begins...





ecmascript



Microsoft®

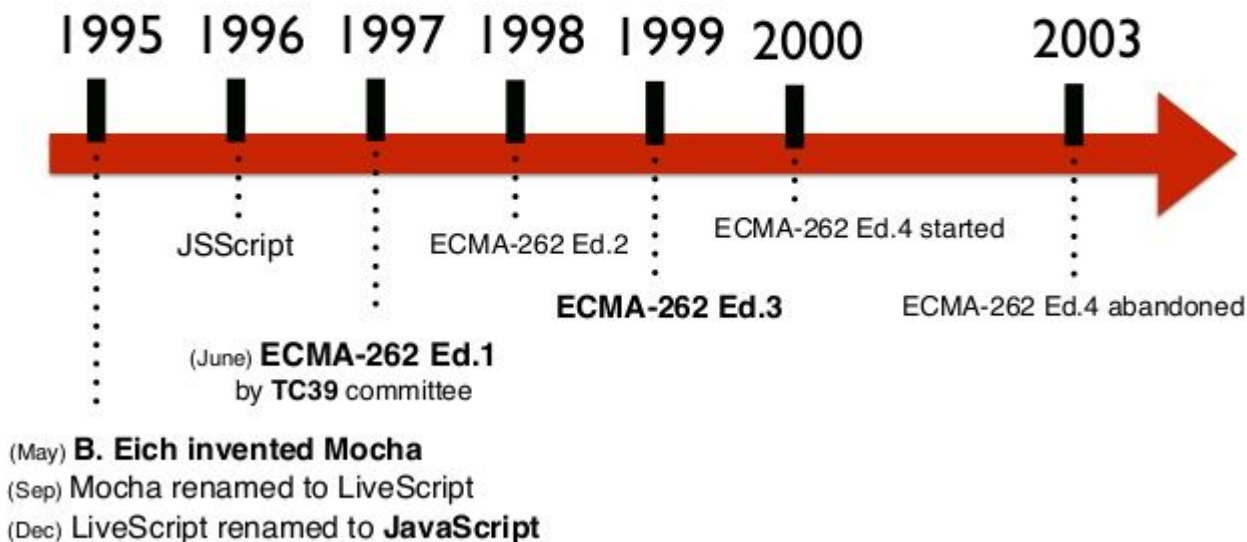
.NET

ECMAScript

- JScript
- Javascript
- ActionScript
- Opera interpreter

ECMAScript 6

History

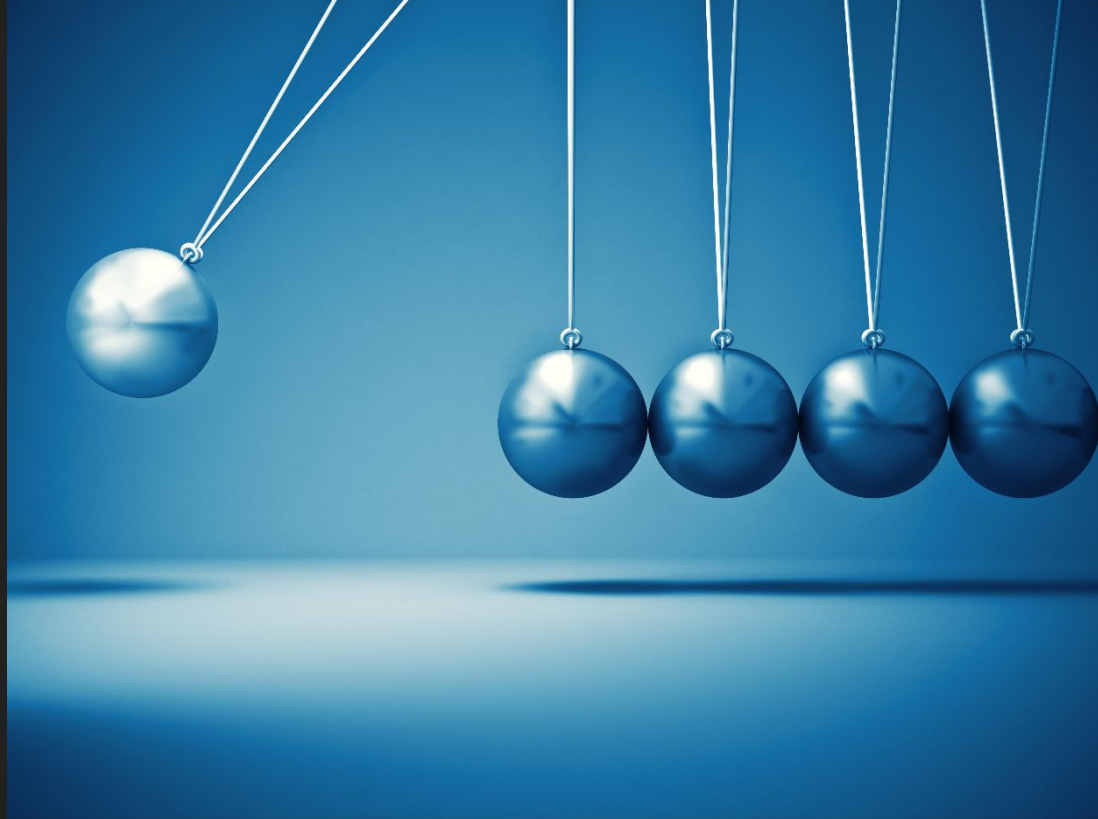






COMPAT

ES



ES 6

ES 2015

TypeScript



CoffeeScript

Desktop browsers															
11%	83%	93%	86%	92%	92%	94%	94%	97%	97%	97%	54%	100%	100%	100%	5%
IE 11	Edge 13 ^[4]	Edge 14 ^[4]	FF 45 ESR	FF 49	FF 50 Beta	FF 51 Aurora	FF 52 Nightly	CH 53, OP 40 ^[1]	CH 54, OP 41 ^[1]	CH 55, OP 42 ^[1]	SF 9	SF 10	SF TP	WK	KQ 4.14 ^[5]
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	0/2
0/7	0/7	7/7	4/7	4/7	4/7	6/7	6/7	7/7	7/7	7/7	0/7	7/7	7/7	7/7	0/7
0/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	0/5	5/5	5/5	5/5	0/5
0/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	9/15	15/15	15/15	15/15	0/15
0/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	5/6	6/6	6/6	6/6	0/6
0/9	7/9	7/9	7/9	7/9	7/9	7/9	7/9	9/9	9/9	9/9	8/9	9/9	9/9	9/9	0/9
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	0/4
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	0/5
0/5	5/5	5/5	2/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	0/5	5/5	5/5	5/5	0/5
0/22	0/22	21/22	19/22	21/22	21/22	21/22	21/22	22/22	22/22	22/22	19/22	22/22	22/22	22/22	0/22
0/24	0/24	23/24	21/24	23/24	23/24	23/24	23/24	24/24	24/24	24/24	21/24	24/24	24/24	24/24	0/24
0/23	0/23	22/23	18/23	19/23	19/23	20/23	20/23	23/23	23/23	23/23	18/23	23/23	23/23	23/23	0/23
0/2	2/2	2/2	1/2	1/2	1/2	1/2	1/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	0/2
0/2	1/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	0/2	2/2	2/2	2/2	0/2

BABEL

You put JavaScript in

```
[1,2,3].map(n => n + 1);
```

And get JavaScript out

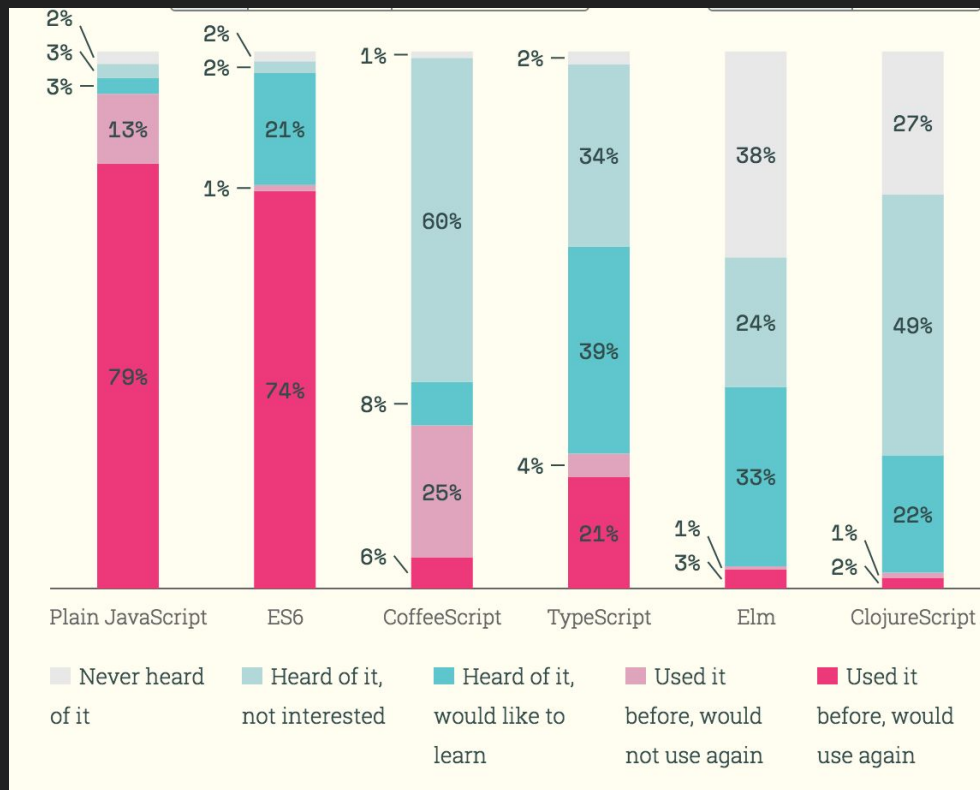
```
[1,2,3].map(function(n) {  
  return n + 1;  
});
```

So....JavaScript

- Flexible - Good or bad?
- You need to know how to **really** use it
- Learn ES5 or ES6

Javascript Flavors

- CoffeeScript
- TypeScript
- Elm
- ClojureScript



WARNING!

30,000,000,000

Variables

- Numbers
- Strings
- Booleans
- Objects
- Functions
- Undefined



Numeric Types

- 64 bits to represent any number.
- 18 000 000 000 000 000 000
- Ok...nonwhole numbers are a little harder
- 9 000 000 000 000 000

Arithmetic

- $10 + 4 * 2;$
- $(10 + 4) * 2;$
- $8 \% 2;$
- $8 \% 3;$
- $8 \% 5;$

Special numbers

- Infinity
- -Infinity
- NaN

Strings



HELLO

WORLD!

Output

- What does typeof do?

```
console.log(typeof 3);
```

```
console.log(typeof "h");
```

```
console.log(typeof "3");
```

Booleans

- How are strings compared?
- `console.log("abcd" > "bcd");`
- `console.log("ABC" > "BCD");`
- `console.log("ABC" > "abc");`
- `console.log("AbC" > "ADC");`

Undefined values

- null
- undefined

Weird Javascript

- `8 * null`
- `"5" - 1`
- `"5" + 1`
- `"five" * 2`
- `false == 0`

Strict Comparison

- `false == 0`
- `false == "0"`
- `false == false`

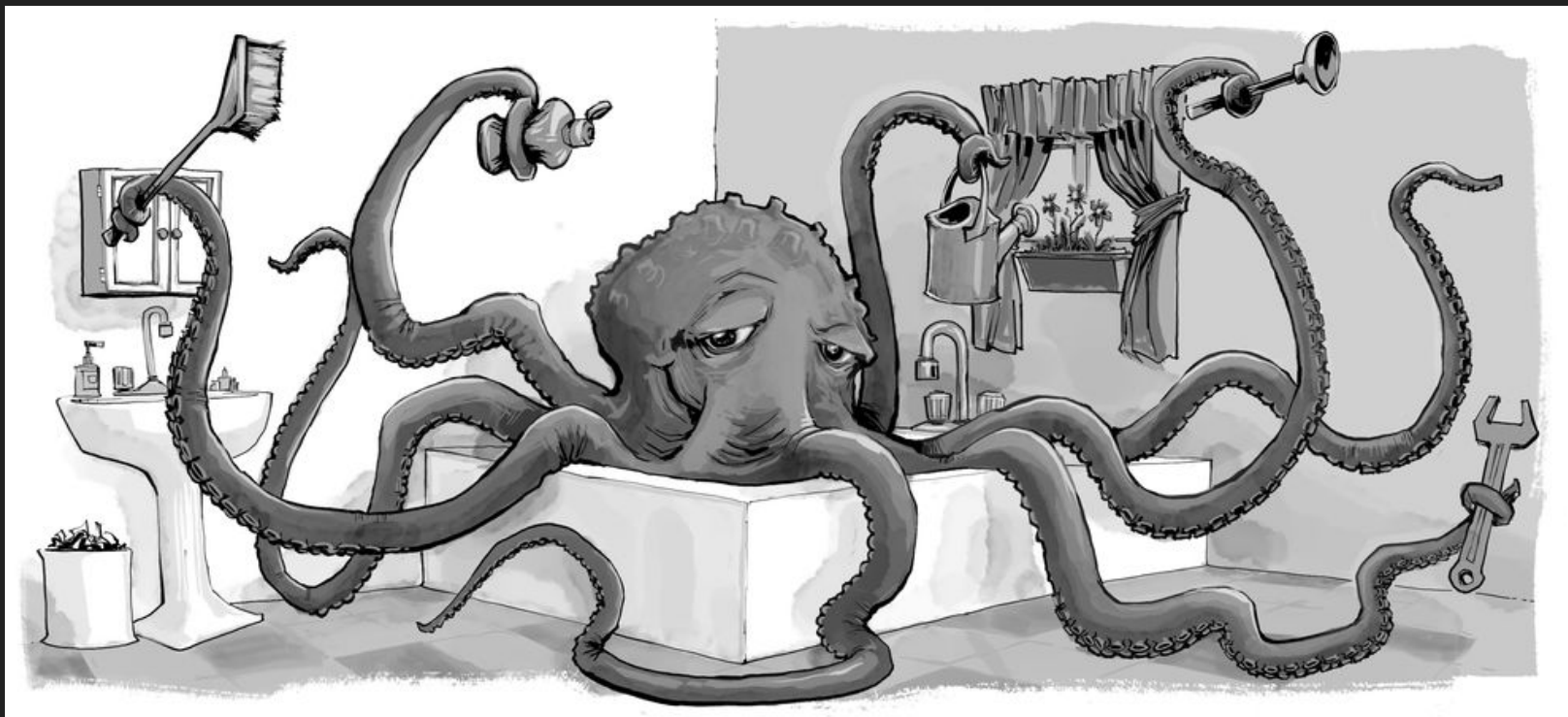
- `false === 0`
- `false === "0"`
- `false === false`

Variable

```
var mood = "light";  
console.log(mood);
```

```
mood = "dark";  
console.log(mood);
```

Variable



Variable

```
var one = 1, two = 2;  
console.log(one + two);
```

Keywords

- var
- break
- class
- const
- true
- throw
- exports
- extends
- let
- new

Functions



Returning values

- Math max
- Math min
- Confirm
- Prompt

Training Program: <https://www.freecodecamp.com>

Day 1: Comment your JavaScript Code up to Divide one Number by Another

Day 2: Increment a Number with JavaScript up to Convert Celsius to Fahrenheit

Day 3: Declare String Variables up to Word Blanks