

**JavaScript
Refresher**

Let's get us all on the same page

Advanced
JavaScript

JS

1

tl;dr

- How JavaScript runs
- Where it runs
- Major differences between client-side and server-side
- How it changes over time
- Reminder of the basic syntax

3

It's interpreted not compiled

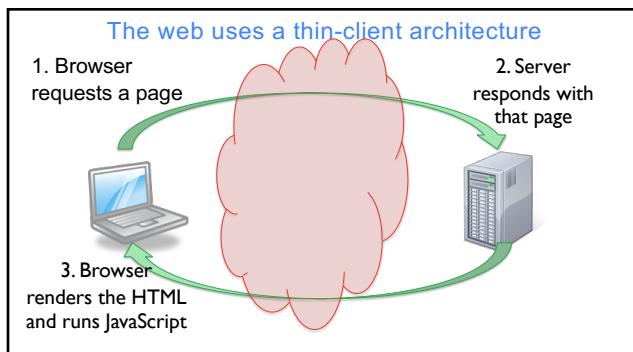
Compiled	Interpreted
<ul style="list-style-type: none"> • Dart • C • Objective-C • C++ • Rust • C# 	<ul style="list-style-type: none"> • COBOL • Java • Fortran • Scala • Kotlin
	<ul style="list-style-type: none"> • bash • MATLAB • Perl • PHP • Ruby • Smalltalk
	<ul style="list-style-type: none"> • Powershell • Python • Winbatch • JavaScript

You just write it in an editor and run it in an interpreter

4



5



6

What it can and cannot do in Node/web	
Web	Node
<ul style="list-style-type: none"> • Has a window object • Uses libraries • Manipulate the DOM • Read/Write cookies • Filesystem?!? No way! • There's no command line 	<ul style="list-style-type: none"> • Has a global object • Uses libraries • There's no DOM • Cookies don't exist • Read/write files • Can read arguments

7



8



9

JavaScript is single-threaded

- But it can borrow threads from its host
- There is no interrupting a task! Once a process runs, it runs to completion.
- No swapping!
- Best use ... "Ooh! I may have to wait for a response to this request. I'll let you know when I get an answer. You can do other things."
- Great for waiting on I/O. Not great for time-swapping.

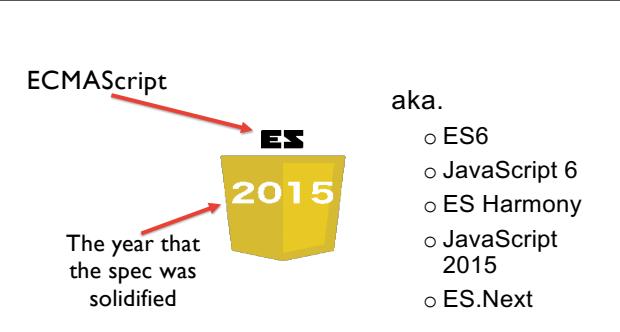
10

Technology improves but ...



"The web is an odd place where breaking changes can't just be hidden behind a version."
- Jared Faris

11



ECMAScript

The year that the spec was solidified

ES

2015

aka.

- o ES6
- o JavaScript 6
- o ES Harmony
- o JavaScript 2015
- o ES.Next

12



We're adding new features every year!

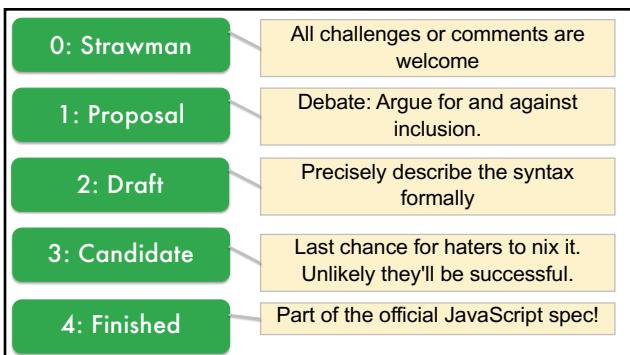
Best to stick with "ES20XX" naming

ecma
INTERNATIONAL

13



14



15

The best features of ES2015					
	Edge	Firefox	Chrome	Safari	Opera
Arrow functions	13	43	45	10	35
Block scoping (let, const)	13	44	45	10	35
Default parameters	14	45	49	10	36
Classes	13	45	49	9	36
Promises	12	38	49	9	36
New collections (Map, Set)	12	38	49	9	36
New iterators (for-of)	12	38	49	9	36
String templates (`\$`)	12	38	49	9	36
Destructuring	14	38	49	9	36
Modules (export, import)	38	60	61	10.1	47
Spread & rest	13	38	49	10	39

16

The best features of ES2016

	Edge	Firefox	Chrome	Safari	Opera
Exponentiation Operator	14+	52+	54+	10.1+	43+
Array.prototype.includes	14+	40+	54+	10+	43+

17

The best features of ES2017

	Edge	Firefox	Chrome	Safari	Opera
Object static methods	15	51	56	10.1	43
String padding	15	51	57	10	44
Trailing commas in function	14	52	58	10	45
Async functions	15	52	56	10.1	43
Atomics	X	57	X	10.1	X

18

Basic Syntax

20

Just a couple of things you need to know so we can get hands-on immediately

- To make a variable,

```
let x = "foo";  
let y = 10;
```

21

JavaScript is case-sensitive

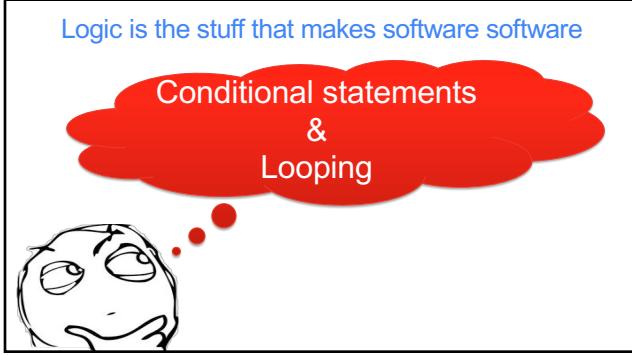
```
alert("This works");  
Alert("This does not work");  
ALERT("This doesn't either");
```

22

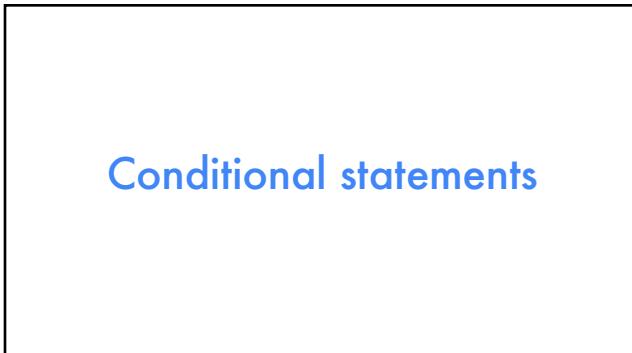
```
var x; // Everything to the right of slashes  
/*  
Anything between slash-splat and  
splat-slash are comments.  
*/
```

You can comment your code

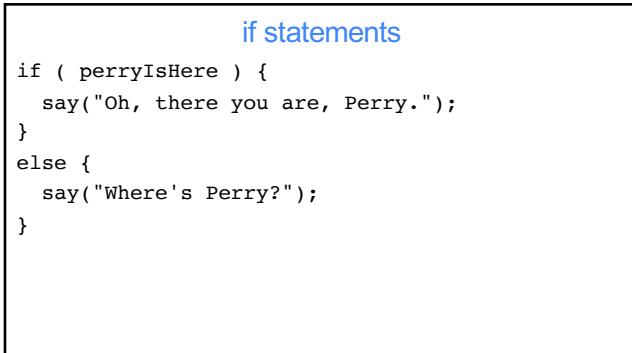
23



24



25



26

JavaScript has no elseif or elsif or elif

```
if ( boolean ) {
  doStuff();
}
else if ( anotherBoolean ) {
  doOtherStuff();
}
else {
  doSomeOtherStuff();
}
```

27

Switch statements are more abstract ways of doing multiple else/ifs

```
switch ( catchPhrase ) {
  case "Whatcha doin?":
    buildTodaysProject();
    break;
  case "Aren't you too young?":
    alert("Yes. Yes I am.");
    break;
  case "Curse you Perry!":
    evilPlotFail();
    break;
  default:
    tellMom();
```

28

```
switch ( studentGrade ) {
  case 'A':
  case 'B':
    addToHonorRoll(student);
  case 'C':
    reimburseFees(student);
  case 'D':
    pass(student);
    break;
  case 'F':
    fail(student);
    break;
  default:
    throw `${studentGrade} isn't valid`
```

Switches fall through
without *break*



29

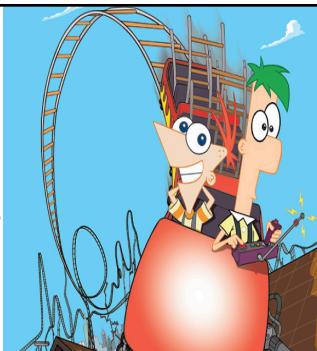
Loops

30

There are two kinds

of loops ...

1. Continuous loops
 - o Done with *while* loops
2. Discrete loops
 - o Done with *for* loops



31

while loops

```
while (somethingTrueOrFalse) {  
    doStuff();  
}
```

32

for loops are just like in Java, C, C++, C#, perl, etc. etc.

```
for (initializer; continue criteria; incrementer) {  
    doStuff();  
}  
• For example  
for (let i = 1 ; i < 10 ; i++) {  
    console.log("Counter is " + i + ".");  
}
```

34

tl;dr

- How JavaScript runs
- Where it runs
- Major differences between client-side and server-side
- How it changes over time
- Reminder of the basic syntax

35