

Today

- JS Review
- JS Continued

**JavaScript Is
Ubiquitous**

It is THE way
to get interactivity into your pages

ES6

Originally proposed in 2011; official in 2015

ES6 is a Superset

Always backwards compatible

Compatibility Table

ES6

Introduced classes, modules, lexical block scoping, iterators, generators, native language support for promises

ES6 -> ES5

Requires Transpilers

Babel

Popular transpiler

The language itself

```
for (let i = 0; i < 5; i++) {  
    ...}
```

```
while (true) { ... }
```

```
if (thing) { ... } else { ... }
```

```
function add(one ,two)  
{ return one+two }
```

Example function, params not typed

Hoisting?

Ex: pull functions to the top

Dynamic Typing

Do not need to declare what type of info is stored
in the variable

Today

- ~~JS Review~~
- JS Continued

var / let / const

Example js 4

const+let vs var

block scope keeps things clean

Blocks

Group of statements {} - not BEM

Primitives

- boolean
- number
- string
- symbol
- null
- undefined

Boolean

true / false

Number

doubles: 10.22

64 bit floating point

String

'one' or "Two"

Oddities

Precision errors

L to R concat

Symbol

Kinda like unique strings.

Null

assigned no value

Undefined

no value assigned

Objects

- Array
- Date

Loose Equality

==

```
console.log(num == str);
```

Strong Equality

===

Model for understanding

**JavaScript is event
driven**

JS -> DOM <- HTML


```
let button =  
document.querySelector  
or('.button');
```

Example add event

Attributes

via dot syntax (ex: .src)

innerHTML vs textContent

What happens when you parse the content?

Classes are special

via `.classList.add/remove`

Elements are Nodes

HTMLImageElement

Node Interface

MDN Docs

of vs in

Example enumerations

Event.currentTarget vs target

Sometimes what you click is not what is listening

Example

currentTarget

object event is attached to

target

element you clicked