Week 4

OOP JAVASCRIPT AND ANIMATION

SOME EXTRA BITS ABOUT JAVASCRIPT

VARIABLE IS REFERENCE

```
var a = { test: 123 }
var b = a
var c = b

a.test // 123
b.test // 123
c.test // 123

a === c // true
```

One single underlying object: `{ test: 123 }`

`a, b, c` are just pointers to the same thing.

TRUTHY AND FALSY

```
var a = 0
if (a) {
    // things here will not happen
}

var b = '' // an empty string
if (b) {
    // will not happen either
}

var c // c is undefined
if (c) {
    // still not happening!
}
```

TRUTHY AND FALSY

"Falsy" values:

```
0,
'',
null,
undefined,
false,
NaN
```

All others are considered "Truthy"

```
var a = {}
if (a) {
    // this will happen!
}
```

TYPE COERCION

JavaScript sometimes performs sneaky type-conversions under the hood:

```
// adding a string with a number:
var a = 'a string' + 123 // 'a string123'

// can't use * with strings though:
var b = 'lol' * 5 // NaN

// comparing stuff:
7 == '7' // true
0 == false // true
'' == false // true
0 == '' // true
```

CRAZY!

What if we don't want that fuzzy-ness?
Use deep equal:

```
7 === '7'
'' === false
0 === false
```

Much better.

BASIC OOP IN JAVASCRIPT

"OBJECT ORIENTED"

```
var dog = {
    name: 'Pepper',
    owner: 'Jeff',
    bark: function () {
        console.log('Woof!')
    }
}

dog.bark() // Woof!
```

UNDERSTANDING "THIS"

```
var dude = {
    firstName: 'John',
    lastName: 'Doe',
    getFullName: function () {
        console.log(this.firstName + ' ' + this.lastName)
    }
}

dude.getFullName() // John Doe
```

WARNING

`this` points to the object that "invokes" the function, in this case, `dude`. If the function is not invoked by an object, or `this` is used outside of a function, it will point to the global object!!!

USING CONSTRUCTORS

```
function Dude (name, age) {
    this.name = name
    this.age = age
}

var dude1 = new Dude('John', 21)
dude1.name // John
dude1.age // 21

var dude2 = new Dude('Mike', 25)
dude2.name // Mike
dude2.age // 25
```

THE PROTOTYPE

```
Dude.prototype.intro = function () {
    console.log('Hi, my name is ' + this.name)
    console.log('I am ' + this.age + ' years old.')
}

dude1.intro()
// Hi, my name is John
// I am 21 years old.

dude2.intro()
// Hi, my name is Mike
// I am 25 years old.
```

THE PROTOTYPE