MODULES

LEVEL UP YOUR JAVASCRIPT

BY CODY BARRUS & RYAN EWING

CLASSES IN JAVASCRIPT?

- YUP!
- WELL. SORT OF.
- SYNTACTIC SUGAR ON TOP OF PROTOTYPICAL INHERITANCE

PROTOYPAL INHERITANCES (A REFRESHER)

```
function Bond() {
  this.kills = 0;
  this.killRange = 15;
  this.puns = 0;
}

Bond.prototype.sneakAttack = function(badGuy) {
  if (badGuy.distance < this.killRange){
    this.kills += 1;
    this.puns += 1;
  }
}
```

PROTOYPAL INHERITANCE CNT'D

```
var jamesBond = new Bond();
jamesBond.sneakAttack();
```

ES2015 CLASS SYNTAX

- NO COMMAS. UNLIKE OBJECT LITERALS

```
class Bond {
  constructor () {
    this.kills = 0;
    this.killRange = 15;
    this.puns = 0;
  }
  sneakAttack () {
    if (badGuy.distance < this.killRange){
      this.kills += 1;
      this.puns += 1;
    }
  }
}
```

STATIC METHODS

- SUGAR ON TOP OF ES5 STATIC METHODS

IN ES5

```
function Bond(){
  this.kills = 0;
}
Bond.hasMoreKills = (bondOne, bondTwo){
  return bondOne.kills > bondTwo.kills;
}
```

STATIC METHODS IN ES2015

```
class Bond {
  constructor() {
    this.kills = 0;
  }
  static hasMoreKills (bondOne, bondTwo) {
    return bondOne.kills > bondTwo.kills;
  }
}
```

EXTENDS

- ENABLES EASY INHERITANCE
- `super` CALLS THE METHOD ON THE BASE CLASS. SO WE DON'T NEED TO REPEAT OURSELVES.

```
class RogerMoore extends Bond {
  sneakAttack(){
    super.sneakAttack();
    this.puns += 3;
  }
}
```

MORE ON SUPER

- OFTEN USED TO OVERRIDE CONSTRUCTOR METHOD

```
class Bond {
  constructor(kills) {
    this.kills = kills;
  }
}
class DanielCraig extends Bond {
  constructor(kills){
    super(kills * 7);
  }
}
```