CORNELL NYC TECH

Java Script

adam@honestbuildings.com

OOP Basics

1. JavaScript does NOT have inheritance
2. It does not really have classes
3. It has some mechanisms that can approximate these
4. Many frameworks have developed more sophisticated OOP.

   Unclear if that is really helpful

OOP Basics

```
// Class Definition
function SimpleClass() {
  this.data = "...";
  this.method = function(){};
}

// Class Instantiation
var y = new SimpleClass;
```

```
// Object Literal
var x = {
  data: "...",
  method: function(){},
};
```

OOP Idioms

```
function Vehicle() {              // This defines a class (Yes it actually a function)
  var privateData = "foo";        // Local variable
  this.publicData = "bar";        // Public Instance variable
  this.publicMethod = function () {   // Public Instance function (re-declared for each intance)
    alert(privateData);           // Note that 'privateData' bound from outer scope
  };
} ;


// Public Static variable.  Value visible to all instances, is just a field on the Vehicle fn-obj
Vehicle.staticData = "baz";

// Instance method be available to all instances
Vehicle.prototype.instanceMethod = function () {console.log(this.publicData)};
```

The Type System

What is the difference between ‘x = new X()’ and ‘x = {}’?

1. An Object is created **{}**
2. Special member **{}.__proto__** set to **X.prototype**
3. Function X() is invoked, with ‘this’ bound to the object **{}**
4. Calling **x**.<anything> searches prototype.<anything>

Understanding (2) and (4) are the key to mastering JavaScript
Commit this to memory: http://stackoverflow.com/questions/9959727

OOP Caveats

```
function ClickCounter() {
  this.counter = 1;
  this.countIt = function () {
    this.counter++;
  };
} ;
```

➔ `countIt()` is just a function. NO Relationship to ClickCounter

➔ **this** depends on the calling context

➔ https://jsfiddle.net/669Ld043/3/

The Missing Libraries

underscore.js [link]

solid, fast utilities for collections + function operators

moment.js [link]

time manipulation, conversions, printing and math

---

jQuery [link]

DOM selection + manipulation, ajax + promises

handlebars.js [link]

simple + straightforward templating