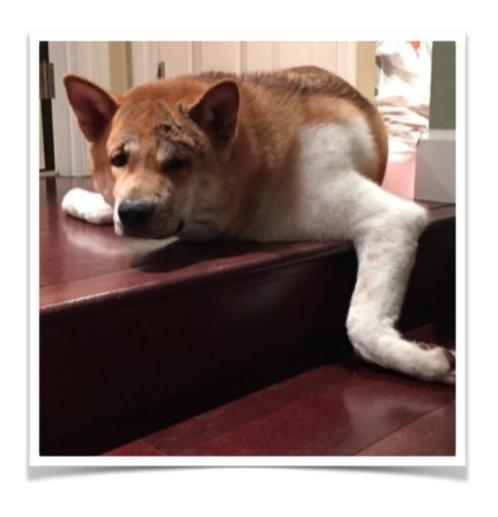
JSON, Objects & Scope



Matt Sexton

msexton@expedia.com



Class Overview

JSON

Objects & Constructors

Scope

Errors

homework solution

javascript object notation

```
"name": "some JSON",
"age": 56,
"isUseless": true,
"friends": [
  "dog",
  "cat",
  "tasseled wobbegong"
```



JSON Things To Remember

Curly & Square Brackets

Double-Quotes

jsonlint.com

Objects

```
// Object literals
var x = {};
var y = {
 prop1: 'one',
 myFunc: function (arg1) {
   return 'yo yo yo';
```

Constructor

Normal JavaScript function called with the 'new' keyword

```
// Object constructor

function Person (name, age) {
  this.name = name;
  this.age = age;
}

var ryan = new Person('Ryan', 22);
```



this

this

Current context of your code (usually the function).

```
// this examples
function MyConstructor (name) {
  this.name = name;
var x = new MyConstructor('Ryan');
var y = new MyConstructor('Matt');
x.name; // Ryan
y.name; // Matt
```

Function Scope

```
function someFunc () {
  var x = 3;
}
x; //undefined
```

Module Scope

```
var x = 3;
var y = 6;
module.exports = y;
```

No Block Scope in JavaScript

try catch

```
// try / catch example
try {
  JSON.parse('{not valid ]JSON');
} catch (err) {
  console.log(err);
```

Throw

Strings

```
Errors
```

```
throw 'You did something very bad';

// no stack trace or error line num
```

```
throw new Error('Whoa!');

// strack trace
```