

Tackling JavaScript

Allen Pike, Steam Clock Software

Me

- JavaScript @ Apple
- Steam Clock (HP/Palm, Time)
- Founded VanJS

JavaScript

- Most widely deployed language
- Worst named language
- Highly dynamic and powerful

Why Now

- Browser landscape
- jQuery
- Node.js

Syntax

```
function enterPortal(player) {  
    var remark = "Welcome, " + player.name;  
  
    if (OrangePortal.isOpen()) {  
        player.location = OrangePortal.location;  
    }  
    console.log(remark); // Welcome, Chell.  
}
```

Like PHP

- Dynamic and loose
- C-like syntax
- Curly braces
- if, else, for, etc.
- Garbage collection
- Post-web

Not like PHP

- Everything is an object
- Objects and inheritance
- scope and this
- batteries not included
- . VS ->
- + VS .

Functions

```
function multiply(x, y) {  
    return x * y;  
};  
  
console.log(multiply(2,3)); // 6
```


First Class

```
var multiply = function (x, y) {  
    return x * y;  
};  
  
console.log(multiply.length); // 2  
  
var plugIn2and3 = function(fn) {  
    return fn(2, 3);  
};  
  
console.log(  
    plugIn2and3(multiply)  
); // 6
```

Anonymous

```
var colors = ["blue", "green"];
```

```
colors.forEach(function(c) {  
    console.log(c);  
});
```

```
// blue  
// green
```

Closures

```
var portals = ["orange", "blue"];  
var colors = ["blue", "green"];  
  
colors.forEach(function(c) {  
    if (portals.indexOf(c) == -1) {  
        console.log(c + " is not a valid portal.");  
    }  
}); // green is not a valid portal.
```

Objects

```
var cube = {  
  kind: "plain",  
  sides: 4,  
  speak: function() {  
    console.log("Cubes do not speak.");  
  }  
};
```

```
console.log(cube.kind); // plain
```

```
cube["kind"] = "companion";  
console.log(cube.kind); // companion
```


new

```
function Portal(c) {  
  this.color = c;  
}
```

```
Portal.prototype.open = function() {  
  console.log("Opening ", this.color);  
};
```

```
orangePortal = new Portal("orange");  
bluePortal = new Portal("blue");
```

```
bluePortal.open(); // Opening blue
```

extend

```
var Turret = Class.extend({  
  init: function(isStanding) {  
    this.standing = isStanding;  
  },  
  canShoot: function() {  
    return this.standing;  
  }  
});
```

```
var agatha = new Turret(false);  
agatha.canShoot(); // false
```

<http://ejohn.org/blog/simple-javascript-inheritance/>

Classical Inheritance

```
var SuperTurret = Turret.extend({  
    canShoot: function() {  
        return true;  
    }  
});  
  
var chloe = new SuperTurret(false);  
chloe.canShoot(); // true
```

<http://ejohn.org/blog/simple-javascript-inheritance/>

JSON

```
var foo = {  
  "name": "GlaDOS",  
  "occupation": "evil supercomputer",  
  "cpus": 72000,  
  "interests": [  
    "testing",  
    "test subjects",  
    "revenge"  
  ],  
  "creator": {  
    "name": "Aperture Science Labs",  
    "founder": "Cave Johnson"  
  }  
}
```

Scope

```
var global = true; // global scope
if (global) {
  var block = true; // global scope
}
```

```
function demo() {
  var fn = true; // function scope
  novar = true; // global scope
  console.log(global, block, novar, fn);
}
```

```
demo(); // true true true true
console.log(global, block, novar);
console.log(fn); // Error: fn is not defined
```

this

```
var bluePortal = {  
  color: "blue",  
  open: function() {  
    console.log("I am ", this.color);  
  }  
};
```

```
bluePortal.open(); // I am blue
```


oh no

```
var bluePortal = {  
  color: "blue",  
  open: function() {  
    console.log("I am ", this.color);  
  }  
};
```

```
var orangeThing = { color: "orange" };
```

```
orangeThing.open = bluePortal.open;  
orangeThing.open(); // I am orange
```

```
var strayOpen = bluePortal.open;  
strayOpen(); // I am undefined
```

this

this refers to the calling with, not the
object you're writing

Lint

- Bad Parts and simple mistakes
- JSLint, JavaScript Lint, JSHint
- Auto lint

Wordpress

- wp_enqueue_script
- jQuery, scriptaculous, cropper, jQUI, Tiny MCE, Prototype, JSON
- Your own scripts or libraries

Batteries

- DOM (jQuery, Prototype)
- MVC (Backbone, JSMVC, Spine)
- Everything (ExtJS, YUI, SproutCore)
- microjs.com

JS Console

- `console.log`
- stack traces, step debugging
- wonderful, wonderful things

Learning More

- Learning Advanced JavaScript:
ejohn.org/apps/learn/
- Douglas Crockford:
javascript.crockford.com
- VanJS: vanjs.com

Thanks

www.antipode.ca

www.steamclocksw.com

@apike