

JavaScript Frameworks

Prototype, jQuery, Mootools

Ben Hughes
Matt Bango
Brandon Dow

Why use a JS framework?

- Solves Browser Inconsistencies
- Abstracts Ajax Functionality
- Makes working with the DOM fun!
- Clear Succinct, and Clear Code
- In general, minimizes the bullshit!

Prototype

Ben Hughes

Overview

- Developed with and heavily integrated with Ruby on Rails
- Influenced architecturally from the Ruby programming language.
- 120KB raw, 30KB gzipped

Class-Based Object-Orientation

- JavaScript is Prototypal OO, not Class-Based OO
- Prototype lets you work in a Class-OO fashion:

```
var Person = Class.create({
  initialize: function(first, last, location) {
    this.first = first;
    this.last = last;
    this.location = location;
  },
  fullName: function() {
    return (this.first + ' ' + this.last).strip();
  }
});
```

```
var Cat = Class.create(Animal, {
  eat: function($super, food) {
  }
});
```

The \$ Function

- Alias for document.getElementById
- Pervasive in Prototype
- Used similarly in other frameworks as well

```
document.getElementById('navbar').style.display = 'none';  
  
// vs:  
  
$('navbar').hide();
```

Selecting Elements with \$\$

- Use CSS3 selectors to get a collection of elements
- Eliminates a lot of otherwise tedious DOM traversal

```
$$('li.featured');

$$('table tbody > tr:nth-child(2n)').invoke('addClassName', 'alternate');

$$('#navbar li', '#sidebar li');

$$('a:not([rel=~nofollow])');
```

Useful String Extensions

- Prototype extends the String class with some useful methods:

```
strip();  
stripTags();  
truncate(50, '...');  
gsub(/aeiouy/, '*');  
escapeHTML();  
unescapeHTML();  
startsWith(str);  
endsWith(str);  
include(str);  
  
camelize();  
capitalize();  
underscore();  
dasherize();
```

Enhanced Arrays with \$A

- Turns “array-like things” (such as NodeList, HTMLCollection) into full Arrays
- Gains the full features of Array and Prototype’s Enumerable extension

```
arr.clear();
arr.compact(); // Without nil and undefined
arr.intersect(otherArray);
arr.uniq();
arr.without(value);
arr.first();
arr.last(); // Instead of arr[arr.length - 1]
```

Easy Form Value Calculation

- Gets the actual value as sent to the server in a form; dealing with the differences in field types

```
$F('name') // text field  
  
$F('status') // select list  
  
$F('active') // checkbox
```

Enumerable

```
var people = [
  new Person('John', 'Coltrane', 'Detroit', 21);
  new Person('Oscar', 'Peterson', 'Toronto', 19);
  new Person('Clifford', 'Brown', 'New York', 12);
];

items.each(function(item) {
  alert(item.toString());
});

havingParty = people.any(function(person) {
  return (person.age >= 21);
});
```

Functional Programming Examples with Enumerable

```
people.pluck('first').sort().join(', ');
// => 'Clifford, John, Oscar'

people.invoke('poke');

[1, 2, 3, 5, 8, 13].inject(0, function(acc, n) {
  return acc + n;
});
// => 32 (sum)

people.map(function(person) { return person.last; });
// => ['Coltrane', 'Peterson', 'Brown']
```

Other Interesting Things

```
(10).times(function() {  
  alert("Isn't this annoying?");  
});
```

```
document.observe('dom:loaded', function() {  
});
```

Higher-Order Functions & Partial Function Application

```
func.bind(obj);
func.bind(obj, 1, 2);

func.bindAsEventListener(obj);
```

```
func.wrap(function(proceed, data) {
    doSomethingBefore(data);
    proceed(data);
    doSomethingAfter(data);
});
```

```
Math.round(3.23); // => 3
Number.prototype['abs'] = Math['abs'].methodize();
(3.23).round(); => 3
```

Event-Handling

- Unified Model for All Browsers

```
$element.observe(eventName, handler, [useCapture = false]);  
  
$('clickme').observe('click', function(event) {  
    alert('You Clicked: ' + event.element().toString());  
});  
  
event.element();  
event.findElement(selector);
```

```
event.stopPropagation();  
event.preventDefault();  
event.stop();
```

Custom Events

```
$('container').observe('title:changed', function(event) {  
    this.highlight({ duration: 0.5 });  
});  
  
$('title').fire('title:changed');
```

Navigating the DOM

```
// Finding:  
$(element).down();  
$(element).next();  
$(element).previous();  
$(element).up();  
  
// Altering  
$(element).remove();  
$(element).next('li').remove();  
  
$(element).replace('<h3>My Content</h3>');  
$(element).update('<h3>My Content</h3>');
```

```
addClassName(className);  
hasClassName(className);  
removeClassName(className);  
toggleClassName(className);  
  
hide();  
show();  
toggle();  
visible();  
  
getStyle('font-weight');  
setStyle({ color: 'black' });
```

Ajax!

- Abstract Ajax.Request class
- Lots of options and features

```
new Ajax.Request(url [, options]);  
  
new Ajax.Request('/events', {  
  method: 'GET',  
  asynchronous: true,  
  contentType: ,  
  encoding: ,  
  evalJS: ,  
  evalJSON: ,  
  parameters: ,  
  requestHeaders ,  
  
  onCreate: callback,  
  onFailure: callback,  
  onSuccesss: callback,  
  onComplete: callback  
});
```

Abstraction: Ajax.Updater

- Easy class for the common use case of updating part of the DOM with what is returned via Ajax

```
<div id="results"></div>

new Ajax.Updater('results', '/results');
```

Abstraction: Ajax.PeriodicUpdater

```
new Ajax.PeriodicUpdater('userlist', '/users', { frequency: 3 });
```

Script.aculo.us

Effects & More for Prototype...

Modules

- Animated Effects
- Drag & Drop
- Autocompleters & In-Place Editing
- Sliders
- DOM Builder
- Sound

Animated Effects

```
new Effect.EffectName(element [, options]);  
  
myElement.visualEffect('EffectName' [, options]);
```

- Effect.Highlight
- Effect.Move
- Effect.Opacity
- Effect.Scale
- Effect.Morph

Combined Effects

- Effect.Fade / Effect.Appear
- Effect.BlindUp / Effect.BlindDown
- Effect.SlideUp / Effect.SlideDown
- Effect.Puff
- Effect.Fold
- Effect.Squish
- Effect.Pulsate

Easy Sortables

- Given a list with 5 li children:

```
Sortable.create('tasks');

// Or with Ajax:

Sortable.create('tasks', {
  onUpdate: function(list) {
    new Ajax.Request('/tasklist/1', { parameters: Sortable.serialize(list); });
  }
});
```

Other Cool Stuff (Not Covering)

- Drag & Drop
- Auto Completer Controls
- Builder
- In-Place Editing
- Slider

Learn More...

