



**Wintellect®**  
Know how.

## Authoring jQuery Plugins


Jeff Prossie  
jeffpro@wintellect.com

**Wintellect NOW** Consulting/Training 

1

## jQuery Plugins


- The jQuery ecosystem includes thousands of extensions called *plugins*, many of them free
  - <http://plugins.jquery.com/>
  - <http://www.jqueryplugins.com/>
  - <http://www.templateite.com/100-popular-jquery-plugins/>
  - <http://www.instantshift.com/2012/03/26/80-useful-jquery-plugins-to-enrich-your-sites-user-experience/>
- You can author plugins of your own

**Wintellect NOW** 


2

## Retina Plugin

- Popular plugin that adds "retina effect" to images



<http://tutorialzine.com/2010/06/apple-like-retina-effect-jquery-css/>


**Wintellect NOW** 

3

## Authoring jQuery Plugins

- jQuery object is extensible
- Most jQuery plugins are scripts that add methods to the jQuery object via \$.fn
  - \$.fn == jQuery.prototype
- Enclose plugin code in a self-executing anonymous function
  - Makes plugin code self-contained
  - Aliases \$ to the jQuery function inside the plugin
  - Avoids polluting the global JavaScript namespace


<http://docs.jquery.com/Plugins/Authoring>

**Wintellect NOW** 

4

## Structuring a jQuery Plugin


```
(function ($) {
  $.fn.pluginname = function () {
    // TODO: Write plugin code here
  };
})(jQuery);
```

**Wintellect NOW** 

5

## Plugin Authoring

- When plugin is invoked, *this* refers to jQuery object upon which it was invoked
- Use `.each()` to iterate over elements in wrapped set, invoking plugin code on each element
- return *this* to support chainability
  - Exception: Plugins that return scalar values

**Wintellect NOW** 

6

## Writing a Highlight Plugin

```
(function ($) {
  $.fn.highlight = function () {
    // Set each element's background color to yellow
    this.each(function () {
      $(this).css("background-color", "yellow");
    });

    // Return jQuery object for chainability
    return this;
  };
})(jQuery);
```



7

## Invoking the Highlight Plugin

```
// Highlight all DIVs in the document
$("div").highlight();
```



8

## Plugins and Events

- jQuery plugins can register event handlers
- Always scope events using custom namespaces to prevent collisions

```
// Register a handler for click events fired by <img> elements
// and place them in the "wintellect" namespace
$("img").bind("click.wintellect", function(e) {
  alert("Click!");
});
```



9

## Using .bind() in a Plugin

```
(function ($) {
  $.fn.highlight = function () {
    // Set each element's background color to yellow
    // when the element is clicked
    this.each(function () {
      $(this).bind("click.wintellect", function (e) {
        $(this).css("background-color", "yellow");
      });
    });

    // Return jQuery object for chainability
    return this;
  };
})(jQuery);
```



10

# DEMO

jQuery Mousewheel Plugin



Consulting/Training



11

## Plugins and Parameters

- Plugins can accept function parameters
- Include parameters in function prototype

```
(function ($) {
  $.fn.pluginname = function (param1, param2) {

    // TODO: Write plugin code here and reference
    // param1 and param2 as needed

  };
})(jQuery);
```



12

## Parameterizing the Highlight Plugin

```
(function ($) {
  $.fn.highlight = function (color) {
    // Set each element's background color to the
    // specified color when the element is clicked
    this.each(function() {
      $(this).bind("click.wintellect", function (e) {
        $(this).css("background-color", color);
      });
    });

    // Return jQuery object for chainability
    return this;
  };
})(jQuery);
```



13

## Invoking the Plugin

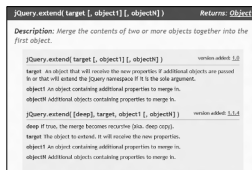
```
// Highlight all DIVs in the document
$("div").highlight("green");
```



14

## jQuery.extend()

- Used to pass "options" to jQuery plugins in lieu of many discrete parameters



<http://api.jquery.com/jquery.extend/>



15

## Using jQuery.extend()

```
(function ($) {
  $.fn.highlight = function (options) {
    var settings = $.extend({
      param1: "yellow", param2: "red"
    }, options);

    this.each(function() {
      $(this).bind("click.wintellect", function (e) {
        $(this).css("background-color", settings.param1);
      });
    });

    return this;
  };
})(jQuery);
```



16

## Invoking the Plugin

```
// Invoke the plugin with default options
$("div").highlight();

// Invoke the plugin with explicit options
$("div").highlight({ param1: "green", param2: "blue" });
```



17

# DEMO

jQuery Image Plugin



Consulting/Training



18