

Fixing Common jQuery Bugs

Effects, Utilities, and Internal Bugs

Elijah Manor
@elijahmanor
<http://elijahmanor.com>



pluralsight 
hardcore developer training

Outline

Never-ending
Story Bug

Cross-
Pollinating
Bug

Continuation
Bug

Decrepit
Deduction
Bug

Bind Breaking
Bug

Overwriting
Option Bug

Forgetful
Plugin Bug

Never-ending Story Bug



Never-ending Story Bug

```
<ul id="menu">
  <li><a href="#">Browsers</a>
    <ul>
      <li><a href="#">Firefox</a></li>
      <li><a href="#">Google Chrome</a></li>
      <li><a href="#">Internet Explorer</a></li>
      <li><a href="#">Opera</a></li>
      <li><a href="#">Safari</a></li>
    </ul>
  </li>
  <!-- ... -->
</ul>
```

Never-ending Story Bug

```
<script src="Scripts/jquery.min.js"></script>
```

```
<script>
```

```
$("#menu li").hover(  
    function () {  
        $(this).find("ul").slideDown();  
    },  
    function () {  
        $(this).find("ul").slideUp();  
    });
```

```
</script>
```

Story Bug

```
js"></script>
```

The animations are going
on & on & on & on...

```
Down();
```

```
Up();
```

Never-ending Story Bug

Each element has
it's own internal
animation queue

Animation methods
add to the default
'fx' queue attached
to the element


As each effect
completes jQuery
moves on to the
next effect

Never-ending Story Bug

.stop([clearQueue] [, jumpToEnd])

Returns: [jQuery](#)

Description: *Stop the currently-running animation on the matched elements.*

 **.stop([clearQueue] [, jumpToEnd])**

version added: 1.2

clearQueue

Type: [Boolean](#)

A Boolean indicating whether to remove queued animation as well. Defaults to `false`.

jumpToEnd

Type: [Boolean](#)

A Boolean indicating whether to complete the current animation immediately. Defaults to `false`.

Never-ending Story Bug

```

$("#menu li").hover(
  function () {
    $(this).find("ul")
      .stop(true, true)
      .slideDown();
  },
  function () {
    $(this).find("ul")
      .stop(true, true)
      .slideUp();
  });

```

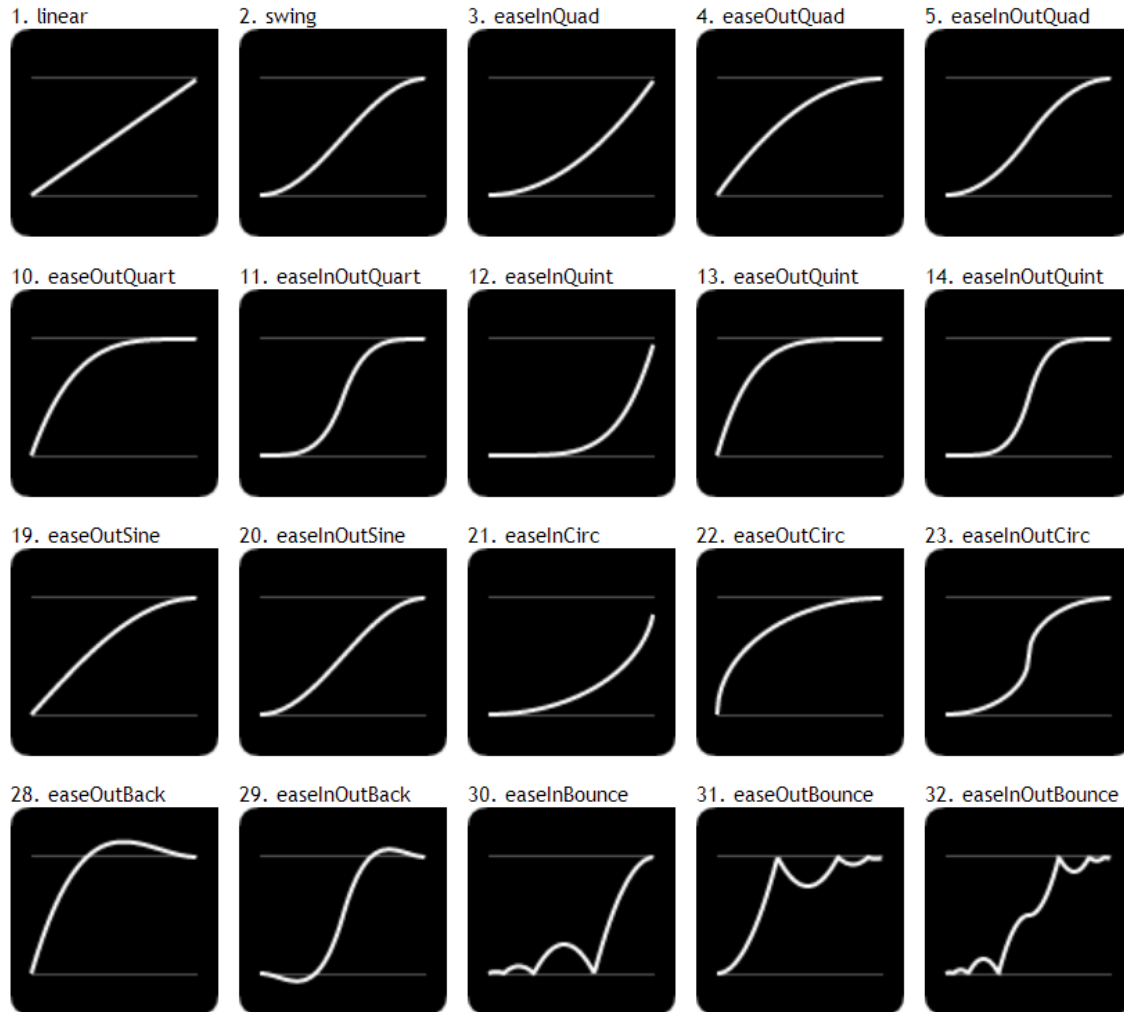
Clear the animation queue and jump the animation to the end as if it has finished

Never-ending Story Bug

```
$("#menu li").hover(function (e) {  
    $(this).find("ul")  
        .stop(true, true)  
        .slideToggle(400, "easeOutBounce");  
});
```

Using one method for both mouseenter and mouseleave events & adding an easing algorithm to spice up the animation 😊

Never-ending Story Bug



<http://jqueryui.com/resources/demos/effect/easing.html>

Never-ending Story Bug

```

$("#menu li").hoverIntent(
    function () {
        $(this).find("ul")
            .stop(true, true)
            .slideDown();
    },
    function () {
        $(this).find("ul")
            .stop(true, true)
            .slideUp();
    });

```

Anticipate intention of user based on their mouse movement with the jQuery hoverIntent plugin

Never-ending Story Bug

```
function toggleMenu(type) {
    var dir = type === "mouseenter" ? "Down" : "Up";
    $(this).find("ul")
        .stop(true, true)["slide" + dir]();
}

$("#menu li").on("mouseenter mouseleave", function (e) {
    if (e.type === "mouseenter") {
        $.doTimeout("delayToggle", 500,
            toggleMenu.bind(this, e.type));
    } else {
        toggleMenu.call(this, e.type);
        $.doTimeout("delayToggle");
    }
});
```

Handle your own
timer logic and
cancel as needed

Cross-Pollinating Bug



Cross-Pollinating Bug

```
<header>What Is Wrong?</header>
```

```
<script src="Scripts/jquery.min.js"></script>
```

```
<script>
```

```
$("#header").fadeOut("slow").remove();
```

```
</script>
```

Cross-Pollinating Bug

```
<header>What Is Wrong?</header>
```

```
<script src="Scripts/jquery.min.js">
```

Doesn't fadeOut over 600 milliseconds, but appears to be removed immediately

```
</script>
```

```
$("#header").fadeOut("slow").remove();
```

```
</script>
```



Cross-Pollinating Bug

Animation methods
add to the default
`fx` queue attached
to the element

Non-animation
methods don't
know about the `fx`
queue

```
$(elem).animationMethod(  
    [duration ]  
    [, easing ]  
    [, complete ]  
)
```

Cross-Pollinating Bug

```
<header>What Is Wrong?</header>
```

```
<script src="Scripts/jquery.min.js"></script>
```

```
<script>
```

```
$("#header").fadeOut("slow", function () {
```

```
    $(this).remove();
```

```
});
```

```
</script>
```

Add a complete callback
to be invoked when the
animation is complete

Continuation Bug



Continuation Bug

```
<ul id="numberList">
  <li data-value="1">One</li>
  <li data-value="2">Two</li>
  <li data-value="3">Three</li>
  <li data-value="4">Four</li>
  <li data-value="5">Five</li>
  <li data-value="6">Six</li>
  <li data-value="7">Seven</li>
  <li data-value="8">Eight</li>
  <li data-value="9">Nine</li>
  <li data-value="10">Ten</li>
</ul>
```

Continuation Bug

```
function listHasNumber($list, numberToFind) {  
    $list.find("li").each(function () {  
        if ($(this).data("value") === numberToFind) {  
            return true;  
        }  
    });  
    return false;  
}  
  
var $numberList = $("#numberList");  
console.log(listHasNumber($numberList, 0)); // false  
console.log(listHasNumber($numberList, 1)); // false  
console.log(listHasNumber($numberList, 5)); // false  
console.log(listHasNumber($numberList, 10)); // false  
console.log(listHasNumber($numberList, 15)); // false
```

Continuation Bug

```
function listHasNumber($list, numberToFind) {  
    $list.find("li").each(function () {  
        if ($(this).data("value") === numberToFind) {  
            return true;  
        }  
    });  
}
```

We are returning
`true` from inside
of the jQuery.each()
method



```
List");  
umberList, 0)); // false  
umberList, 1)); // false  
umberList, 5)); // false  
umberList, 10)); // false  
umberList, 15)); // false
```

Continuation Bug

``return false`` has a special meaning within a `.each()` method... It breaks out of the loop

We can stop the loop from within the callback function by returning false. --<http://api.jquery.com/each>

In a similar way, ``return true`` has side effect of continuing to the next item in the loop

Continuation Bug

```
function listHasNumber( $list, numberToFind ) {
    var hasNumber = false;
    $list.find("li").each(function() {
        if ($(this).data("value") === numberToFind) {
            hasNumber = true; return false;
        }
    });
    return hasNumber;
}

var $numberList = $( "#numberList" );
console.log( listHasNumber( $numberList, 0 ) ); // false
console.log( listHasNumber( $numberList, 1 ) ); // true
console.log( listHasNumber( $numberList, 10 ) ); // true
console.log( listHasNumber( $numberList, 15 ) ); // false
```

Set hasNumber to `true` if found item
and then `return false` to exit loop

\$.grep iterates over every item in the array. There is no break mechanism

```
function listHasNumber($list, numberToFind) {
    return !!$.grep($list.find("li").get(),
        function (element) {
            var value = $(element).data("value");
            return value === numberToFind;
        }).length;
}
```

Use \$.grep to filter DOM array of li elements and then turn length of array into a boolean

```
var $numberList = $("#numberList");
console.log(listHasNumber($numberList, 0)); // false
console.log(listHasNumber($numberList, 1)); // true
console.log(listHasNumber($numberList, 5)); // true
console.log(listHasNumber($numberList, 10)); // true
console.log(listHasNumber($numberList, 15)); // false
```

Decrepit Deduction Bug



Decrepit Deduction Bug

```
<script src="Scripts/jquery-1.9.0.min.js"></script>
```

```
<script>
```

```
if ($.browser.msie &&  
    parseInt($.browser.version, 10) <= 8) {  
    $(".widget").corner(); // Shim  
}
```

```
</script>
```

Decrepit Deduction Bug

```
<script src="Scripts/jquery-1.9.0.min.js"></script>
```

```
<script>
```

```
if ($.browser.msie &&  
    parseInt($.browser.version, 10) <= 8) {  
    $(".widget").corner();  
}
```

```
</script>
```

\$.browser has been deprecated and has finally been removed from jQuery 1.9+



Decrepit Deduction Bug

Browser
Detection is
brittle and
cumbersome

Feature Detection
is the preferred
technique instead

Decrepit Deduction Bug

```

<head>
  <title></title>
  <script src="Scripts/modernizr.min.js"></script>
</head>
<body>
  <script src="Scripts/modernizr.min.js"></script>
  <script>
    if (!Modernizr.borderradius) {
      $(".widget").corner(); // Shim
    }
  </script>
</body>

```

Detect for the feature instead
of the browser version

<http://html5please.com>

Decrepit Deduction Bug

However, you
don't always
control your
whole codebase

What if you use an
older 3rd party
plugin that uses
\$.browser?

Decrepit Deduction Bug

```
<a title="A tooltip with default settings, the href is
displayed below the title" href="http://google.de">Link
to google</a>
```

```
<script src="Scripts/jquery-1.9.0.min.js"></script>
<script src="Scripts/jquery.tooltip.js"></script>
<script src="Scripts/jquery-migrate.js"></script>
<script>
$("[title]").tooltip();
</script>
```

The jquery-migrate plugin adds back deprecated features such as \$.browser, .live(), etc...

<https://github.com/jquery/jquery-migrate/>

Bind Breaking Bug

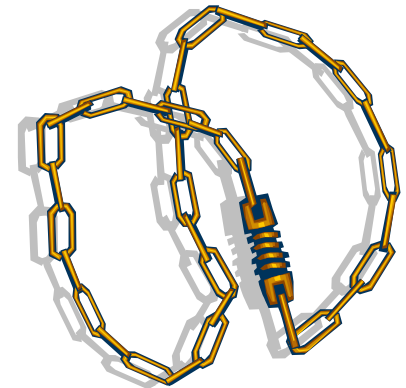


Bind Breaking Bug

```
<a href="http://pluralsight.com">Hello</a>
<a href="http://elijahmanor.com">World</a>
<script src="Scripts/jquery.min.js"></script>
<script>
(function ($) {
    $.fn.valentines = function () {
        this.queue(function (next) {
            $(this).css("color", "red");
            next();
        });
    };
})(jQuery));
$("a").fadeOut("slow").valentines().fadeIn("slow");
</script>
```

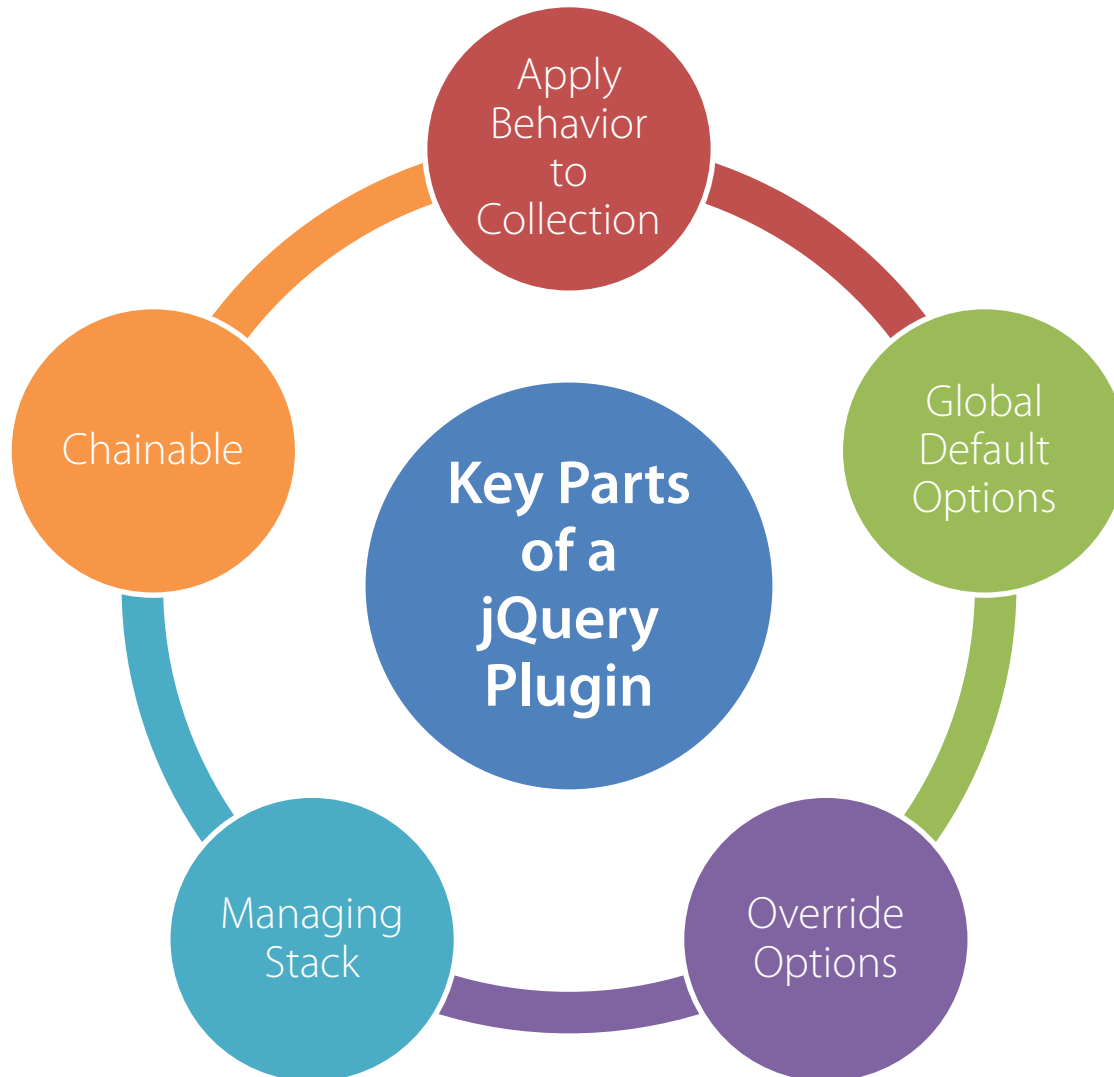
Bind Breaking Bug

```
<a href="http://pluralsight.com">Hello</a>
<a href="http://elijahmanor.com">World</a>
<script src="Scripts/jquery.min.js"></script>
<script>
(function ($) {
    $.fn.valentines = function () {
        this.queue(function (next) {
            $(this).css("color", "red");
            next();
        });
    };
})(jQuery));
$("a").fadeOut("slow").valentines().fadeIn("slow");
</script>
```



Uncaught TypeError: Cannot call method 'fadeIn' of undefined

Bind Breaking Bug



Bind Breaking Bug

```
(function ($) {
    $.fn.valentines = function () {
        this.queue(function (next) {
            $(this).css("color", "red");
            next();
        });

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

Return `this`, which is jQuery in order to support chaining another method

```
$("a").fadeOut("slow").valentines().fadeIn("slow");
```

Bind Breaking Bug

```
(function ($) {
    $.fn.valentines = function () {
        return this.queue(function (next) {
            $(this).css("color", "red");
            next();
        });
    };
})(jQuery);
```

Move the `return` to before `this.queue()`
since it already returns jQuery

```
$("a").fadeOut("slow").valentines().fadeIn("slow");
```

Overwriting Option Bug

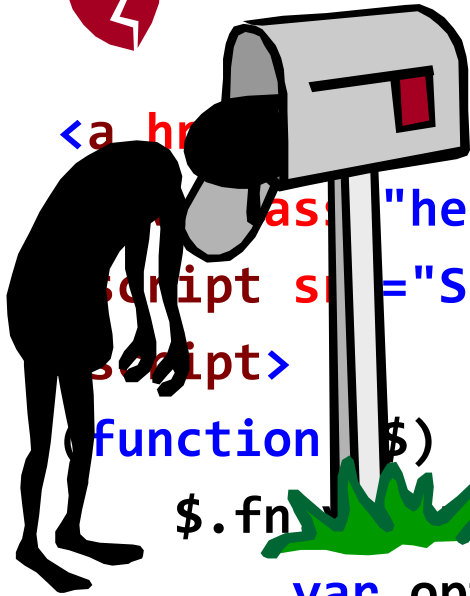


Overwriting Option Bug

```
<a href="http://pluralsight.com">Hello</a>
<div class="heart">Heart</div>
<script src="Scripts/jquery.min.js"></script>
<script>
(function ($) {
    $.fn.valentines = function (options) {
        var opts =
            $.extend($.fn.valentines.defaults, options);
        this.css("color", opts.color);
    };
    $.fn.valentines.defaults = { color: "red" };
})(jQuery));
$("a").valentines({ color: "#B00000" });
$(".heart").valentines();
</script>
```

The color of the .heart element is `#B00000` instead of the defaults.color `red`!

Overwriting Option Bug



```

<a href="//pluralsight.com">Hello</a>
<div class="heart">Heart</div>
<script src="Scripts/jquery.min.js"></script>
<script>
function ($) {
    $.fn.valentines = function (options) {
        var opts =
            $.extend($.fn.valentines.defaults, options);
        this.css("color", opts.color);
    };
    $.fn.valentines.defaults = { color: "red" };
}(jQuery));
$("#a").valentines({ color: "#B00000" });
$("#heart").valentines();
</script>

```

Somehow the defaults
are getting overridden
with the merged result

Overwriting Option Bug

`jQuery.extend(target [, object1] [, objectN])`

Returns: [Object](#)

Description: Merge the contents of two or more objects together into the first object.

 `jQuery.extend(target [, object1] [, objectN])`

version added: 1.0

target

Type: [Object](#)

An object that will receive the new properties if additional objects are passed in or that will extend the jQuery namespace if it is the sole argument.

object1

Type: [Object](#)

An object containing additional properties to merge in.

objectN

Type: [Object](#)

Additional

Merged results get
returned from
`$.extend()` ...

... and results will get
stored into the 1st
argument

```
var opts = $.extend($.fn.valentines.defaults, options);
```

Overwriting Option Bug

```
(function ($) {
    $.fn.valentines = function
        var opts = $.extend(
            {},
            $.fn.valentines.defaults,
            options
        );
        this.css("color", opts.color);
    };
    $.fn.valentines.defaults = { color: "red" };
})(jQuery));
```

You'll commonly see the 1st argument passed to \$.extend() by an empty object

```
$("a").valentines({ color: "#B00000" });
$(".heart").valentines();
```

Forgetful Plugin Bug



Forgetful Plugin Bug

```
<div id="content">
  <div data-links="training">
    <a data-name="website"
href="http://pluralsight.com">Pluralsight</a>
  </div>
  <div data-name="description">
    <span >Online Training</span>
  </div>
  <div>
    <table></table>
  </div>
</div>
<script src="Scripts/jquery.min.js"></script>
```

Forgetful Plugin Bug

```
(function ($) {  
    $.fn.findWithData = function(selector, type, value) {  
        var selected = this.find(selector);  
        return selected.filter(function () {  
            return value !== null ?  
                $(this).data(type) === value :  
                $(this).data(type) !== null;  
        });  
    };  
})(jQuery);  
$("#content").findWithData("div", "links", "training")  
    .css("background-color", "red")  
    .end()  
    .css("background-color", "#ccc");
```

gin Bug



```
(selector, type, value) {  
  (selector);  
  unction () {  
    l ?  
    e) === value :  
    e
```

All the divs get set to a
background-color of #ccc

```
$("#content").findWithData("div", "links", "training")  
  .css("background-color", "red")  
  .end()  
.css("background-color", "#ccc");
```

Forgetful Plugin Bug

.pushStack(elements)

Returns: [jQuery](#)

Description: Add a collection of DOM elements onto the jQuery stack.

 **.pushStack(elements)**

version added: 1.0

elements

Type: [Array](#)

An array of elements to push onto the stack and make into a new jQuery object.

.pushStack()

3 items

5 items

3 items

5 items

.end()

Forgetful Plugin Bug

```
(function ($) {
    $.fn.findWithData = function(selector, type, value) {
        var selected = this.find(selector);
        selected = selected.filter(function () {
            return value !== null ?
                :
        });
        return this.pushStack(selected);
    };
})(jQuery));

$("#content").findWithData("div", "links", "training")
    .css("background-color", "red")
    .end()
    .css("background-color", "#ccc");
```

Make sure to the new set to
pushStack so that it works with .end()

Conclusion

- **Make sure your animations stop with `.stop()`**
- **Be careful when mixing animation and non-animation methods**
- **Remember that returning from `$.each` does something special**
- **If you need deprecated functionality then use jQuery Migrate plugin**
- **Make sure you return jQuery from your plugins to be chainable**
- **When using `$.extend` pass an empty object as the 1st parameter**
- **If you have a plugin that changes the stack then use `.pushStack()`**