

# Fixing Common jQuery Bugs

Core Bugs

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



**pluralsight**   
hardcore developer training

# Outline

Security  
Scare Bug

Bank Robber  
Bug

Contorted  
Creation Bug

Chained  
Promises Bug

Preserved  
Phrase Bug

Punctuation  
Passage Bug

# Security Scare Bug



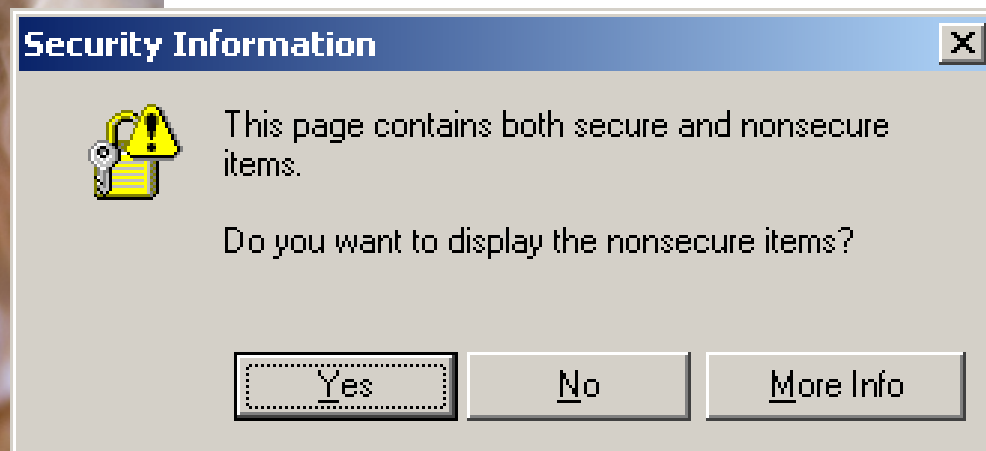
# Security Scare Bug

**<h1>**Can you spot the possible bug?**</h1>**

```
<script  
src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
```

```
<script>  
$(document).ready(function () {  
    $("h1").text("So, what is so wrong with this?");  
});  
</script>
```

# Security Scare Bug



# Security Scare Bug

**oldIE warns User if a Secure HTTPS site tries to load a non-secure (HTTP) resource**

- Setting can be changed, but it's default behavior
- Shouldn't expect the user to change this setting

**Common oversight when moving a website from HTTP to HTTPS**

- Common when moving from a development environment to testing, staging, or production
- How can we easily prevent this from happening?

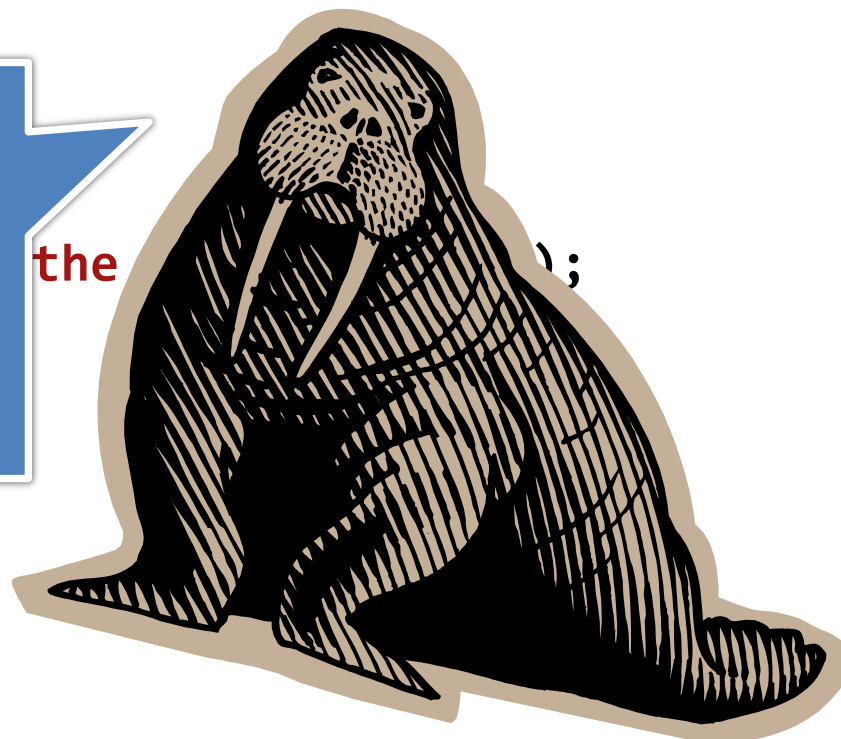
# Security Scare Bug

`<h1>Dynamically pull secure or non-secure script...</h1>`

```
<script
src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.
min.js"></script>
```

Use my sharp walrus teeth ("//") to dynamically get a resource to match your secure or non-secure page! 😊

--Thanks to Alex Sexton (@slexaxton) for the awesome terminology



# Bank Robber Bug





# Bank Robber Bug

```
<div id="content">What's Wrong?</div>
```

```
<script src="Scripts/prototype.js"></script>
```

```
<script src="Scripts/jquery-2.0.0.js"></script>
```

```
<script>
```

```
    $("content").style.display = "none";
```

```
</script>
```

# Bank Robber Bug

```
<div id="content">What's Wrong?</div>
```

```
<script src="Scripts/prototype.js"></script>
```

```
<script src="Scripts/jquery-2.0.0.js"></script>
```

```
<script>
```

```
  $("content").style.display = "none";
```

```
</script>
```

Hello, my name is Query, Jay Query.

I've just stolen all the \$ from the  
Prototype Bank. I'm in control of all  
the functionality now! 😊



# Bank Robber Bug

## JavaScript is a Dynamic Programming Language

- Whatever library assigns to a variable last will take precedence

## Ideally Wouldn't Use Libraries that Define Same Object

- Unfortunately in the real world this can happen

# Bank Robber Bug

```

<div id="content">
</div>
<script src="http://code.jquery.com/jquery-1.4.2.min.js"></script>
<script src="http://code.jquery.com/jquery-migrate-1.2.1.min.js"></script>
<script>
    jQuery.noConflict(); // Reverts $ back to what it was
    $("content").style.display = "none"; // Prototype
    jQuery("div").show(); // jQuery
</script>

```

Using the special .noConflict() method that jQuery provides to revert the \$ back to whatever it was beforehand

# Bank Robber Bug

// DOM Ready

```
jQuery(document).ready(function ($) {
    // Safe to use $ as jQuery
});
```

// DOM Ready Shortcut

```
jQuery(function ($) {
    // Safe to use $ as jQuery
});
```

// IIFE Wrapper

```
(function ($) {
    // Safe to use $ as jQuery
}(jQuery));
```

# Contorted Creation Bug



# Contorted Creation Bug

```
<section id="output"></section>
```

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

```
<script>
```

```
    $(document).ready(function () {  
        $("<ul><li>test 1</li><li>test  
2</li></ul></div>").appendTo("#output");  
    });
```

```
</script>
```

# Contorted Creation Bug

```
<section id="output"></section>
```

```
<script src="Scripts/jquery-1.4.2.js"></script>
```

IE8 silently fails and  
doesn't create any DOM  
elements whatsoever

```
<script>
```

```
$(document).ready(function () {  
    $("<ul><li>test 1</li><li>test  
2</li></ul></div>").appendTo("#output");  
});
```

```
</script>
```





# Contorted Creation Bug

It's easy to accidentally build invalid markup in a string.

```
"<ul><li>test 1</li><li>test 2</li></ul></div>"
```

... versus ...

```
<ul>  
    <li>test 1</li>  
    <li>test 2</li>  
</ul>  
</div>
```

It's easier to visually spot invalid markup when it is structured

# Contorted Creation Bug

```
<section id="output"></section>
```

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

```
<script>
```

```
    $(document).ready(function () {
```

```
        $("<div><ul><li>test 1</li><li>test  
2</li></ul></div>").appendTo("#output");
```

```
    });
```

```
</script>
```

# Contorted Creation Bug

```
<section id="output"></section>
<script src="Scripts/jquery.min.js"></script>
<script id="custom-list" type="text/template">
    <ul>
        <li>test 1</li>
        <li>test 2</li>
    </ul>
</script>
<script>
    $(document).ready(function () {
        $("#custom-list").html().append("#output");
    });
</script>
```

For dynamic templates  
you can use  
underscore.js or  
handlebars.js

# Chained Promises Bug



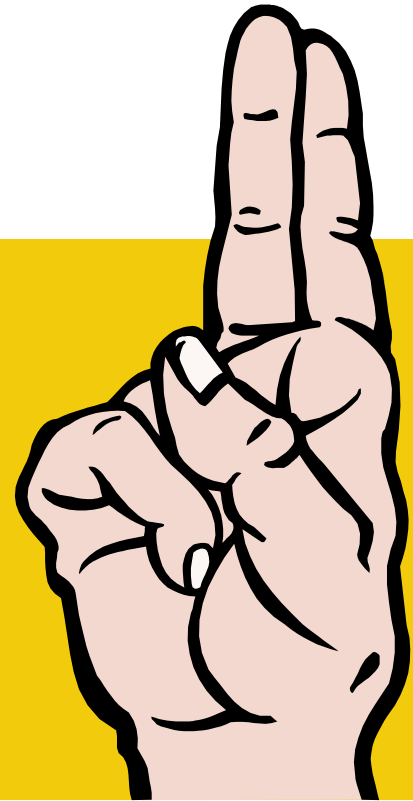
# Chained Promises Bug

```
var promises = [  
    $.getJSON("/contacts.js"),  
    $("#output").fadeIn("slow"),  
    new $.Deferred(function (dfd) {  
        setTimeout(dfd.resolve, 5000);  
        return dfd.promise;  
    });  
];  
$.when(promises).then(  
    function (xhr, faded) {  
        faded.html(xhr[0].length + " Contact(s) Found");  
    },  
    function (xhr, status) { console.log("Error"); }  
);
```


# Chained Promises Bug

```
var promises = [  
    $.getJSON("/contacts.js"),  
    $("#output").fadeIn("slow"),  
    new $.Deferred(function (dfd) {  
        setTimeout(dfd.resolve, 5000);  
        return dfd.promise;  
    });  
];  
$.when(promises).done(function (xhtml, faded) {  
    function (xhtml, faded) {  
        faded.html(xhtml[0].length + " Contact");  
    },  
    function (xhtml, status) { console.log("Error");  
});
```

Uncaught TypeError: Cannot  
call method 'html' of undefined



# Chained Promises Bug

 **jQuery.when(deferreds)** version added: 1.5

**deferreds**  
Type: [Deferred](#)  
One or more Deferred objects, or plain JavaScript objects.

At first glance it might seem the API accepts an array of deferreds...

```
$.when([$.get("/contact"), $("#content").fadeIn("slow")])
```

...but in reality it accepts N number of deferreds like the following...

```
$.when($.get("/contact"), $("#content").fadeIn("slow"))
```

# Chained Promises Bug

If an argument is passed to `jQuery.when` and it is not a `Deferred` or a `Promise`, it will be treated as a resolved `Deferred` and any `doneCallbacks` attached will be executed immediately.

```
$.when({ lookAtMe: "auto-resolved immediately" }).then(  
    function (message) {  
        console.log(message.lookAtMe);  
    }  
);
```

```
$.when([ { name: "John" }, { name: "Susan" } ]).then(  
    function (names) {  
        console.log("names.length: " + names.length);  
    }  
);
```



# Chained Promises Bug

```
var promises = [
    $.getJSON("/contacts.js"),
    $("#output").fadeIn("slow"),
    new $.Deferred(function (dfd) {
        setTimeout(dfd.resolve, 5000);
        return dfd.promise;
    });
];
$.when.apply($, promises).then(
    function (xhr, faded) {
        faded.html(xhr[0].length + " Contact(s) Found");
    },
    function (xhr, status) { console.log("Error"); }
);
```

# Preserved Phrase Bug



# Preserved Phrase Bug

```
<style>
```

```
.freshman { color: red; } .sophomore { color: orange; }
```

```
.junior { color: green; } .senior { color: blue; }
```

```
</style>
```

```
<script>
```

```
var johnSmith = {
```

```
  name: "John Smith",
```

```
  class: "freshman",
```

```
  gpa: 3.5
```

```
};
```

```
$("#student").html(johnSmith.name + " " + johnSmith.gpa)
```

```
  .addClass(johnSmith.class);
```

```
</script>
```

# Preserved Phrase Bug

```
<style>
```

```
.freshman { color: red; } .sophomore { color: orange; }
```

```
.junior { color: green; } .senior { color: blue; }
```

```
</style>
```

```
<script>
```

```
var johnSmith = {  
  name: "John Smith",  
  class: "freshman",  
  gpa: 3.5
```

```
};
```

```
$("#student").html(johnSmith.name + " " + johnSmith.gpa)  
  .addClass(johnSmith.class);
```

```
</script>
```

SCRIPT1028: Expected identifier,  
string or number

# Preserved Phrase Bug

In ECMAScript 3 reserved words can not be used as identifier names or identifiers.

// Identifier Names

~~a.import~~

~~a["import"]~~

~~a = { import: "test" }~~

// Identifier

~~function import() {}~~

~~var import = "test";~~

In ECMAScript 5 reserved words can be used as identifier names, but not identifiers.

// Identifier Names

a.import

a["import"]

a = { import: "test" }

// Identifier

~~function import() {}~~

~~var import = "test";~~

# Preserved Phrase Bug

```
<style>
```

```
.freshman { color: red; } .sophomore { color: orange; }
```

```
.junior { color: green; } .senior { color: blue; }
```

```
</style>
```

```
<script>
```

```
var johnSmith = {
```

```
  name: "John Smith",
```

```
  cssClass: "freshman", //or `klass`
```

```
  gpa: 3.5
```

```
};
```

```
$("#student").html(johnSmith.name + " " + johnSmith.gpa)
```

```
  .addClass(johnSmith.cssClass);
```

```
</script>
```

# Preserved Phrase Bug

```
<style>
```

```
.freshman { color: red; } .sophomore { color: orange; }
```

```
.junior { color: green; } .senior { color: blue; }
```

```
</style>
```

```
<script>
```

```
var johnSmith = {  
  name: "John Smith",  
  "class": "freshman",  
  gpa: 3.5
```

```
};
```

```
$("#student").html(johnSmith.name + " " + johnSmith.gpa)  
  .addClass(johnSmith["class"]);
```

```
</script>
```

# Punctuation Passage Bug





# Punctuation Passage Bug

```
<div id="output">What is wrong with me?</div>
<script src="Scripts/jquery.min.js"></script>
<script>
var options = {
    color: "orange",
    backgroundColor: "blue",
    opacity: 0.5,
    padding: "5px",
};

$(document).ready(function () {
    $("#output").css(options);
});
</script>
```

# Punctuation Passage Bug

```
<div id="output">What is wrong with  
<script src="Scripts/jquery.min.js":  
<script>  
var options = {  
    color: "orange",  
    backgroundColor: "blue",  
    opacity: 0.5,  
    padding: "5px",  
};
```



Error: Expected identifier, string or number

```
$(document).ready(function () {  
    $("#output").css(options);  
});  
</script>
```

# Punctuation Passage Bug

## Internet Explorer 7

```
var opts = { color: "orange", padding: "5px", }; // Error  
var numbers = [ 1, 2, 3, ]; // Error
```

## Internet Explorer 8

```
var opts = { color: "orange", padding: "5px", }; // Works  
var numbers = [ 1, 2, 3, ]; // length 4
```

## Internet Explorer 9

```
var opts = { color: "orange", padding: "5px", }; // Works  
var numbers = [ 1, 2, 3, ]; // length 3
```

# Punctuation Passage Bug

```
<div id="output">What is wrong with me?</div>
<script src="Scripts/jquery.min.js"></script>
<script>
var options = {
    color: "orange",
    backgroundColor: "blue",
    opacity: 0.5,
    padding: "5px"
};

$(document).ready(function () {
    $("#output").css(options);
});
</script>
```

# Conclusion

- **Use walrus teeth to protect yourself from a secure website**
- **Use `jQuery.noConflict()` if you need to use conflicting libraries**
- **Make sure your markup is valid when creating DOM elements**
- **Use `$.when.apply` to pass an array of Deferreds**
- **Don't use reserved words as your object keys**
- **Watch out for trailing commas in your objects and arrays**