Fixing Common jQuery Bugs

Events and Ajax Bugs

Elijah Manor
@elijahmanor
http://elijahmanor.com

pluralsight
hardcore developer training

Outline

False Start Bug	Crazy Context Bug	Tightly Bound Bug
Browser Madness Bug	Unintentional Destruction Bug	Secretive Publish Bug
Confusing Element Bug	Chicken Egg Bug	Security Access Bug

False Start Bug

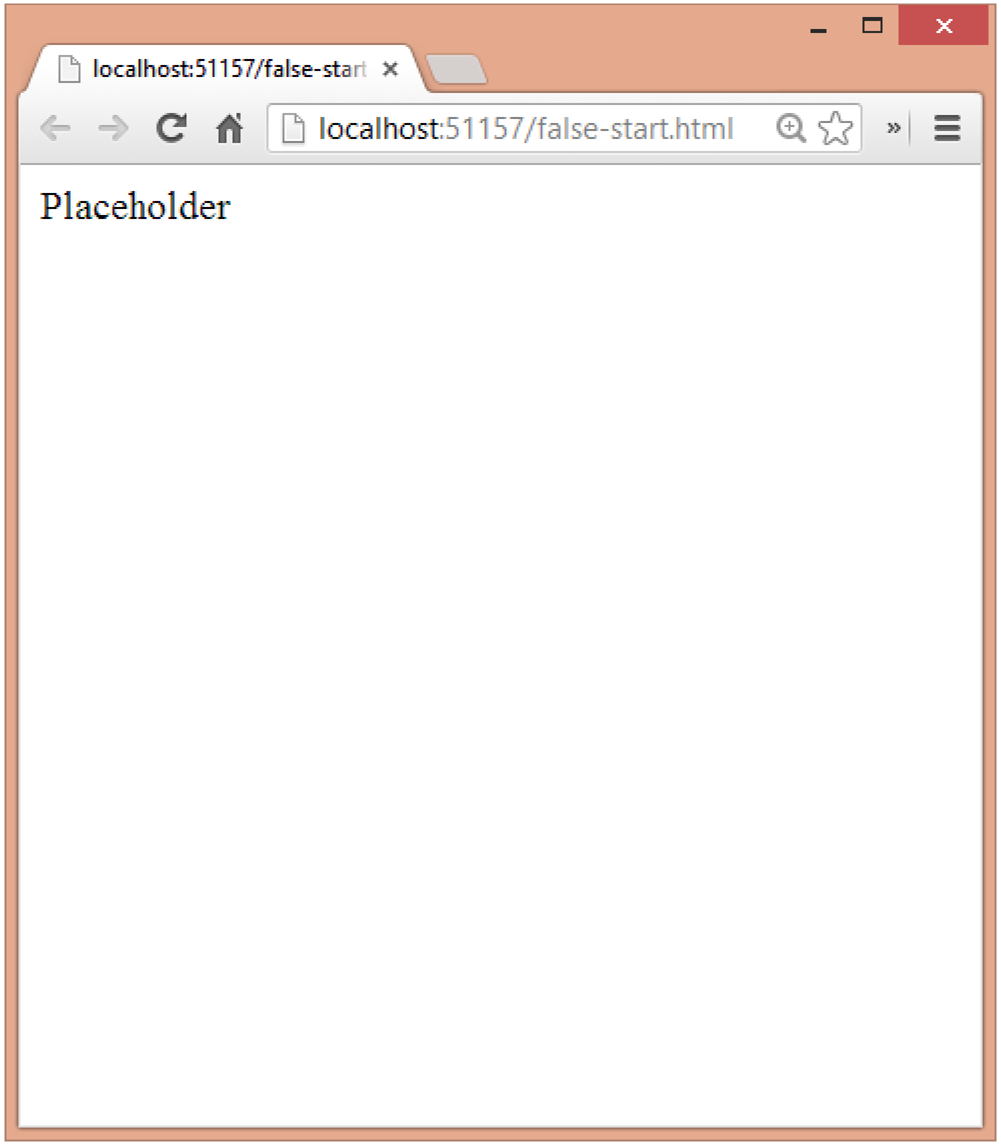
Examine

False Start Bug

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

<script>
$("header")
    .html("<h1>Hello</h1>")
    .on("click", function () { alert($(this).text()); })
    .append("<h2>World</h2>");
</script>

<header>Placeholder</header>
```

Expose

False Start Bug



localhost:51157/false-start ×

localhost:51157/false-start.html

Placeholder

Explain

False Start Bug

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

<script>
$("header")
    .html("<h1>Hello</h1>")
    .on("click", function () { alert($(this).text()); })
    .append("<h2>World</h2>
</script>

<header>Placeholder</header
```

The script is executed before the DOM is ready. The header element does not exist yet

Exterminate

False Start Bug

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

<script>
$(document).ready(function () {
    $("header")
        .html("<h1>Hello</h1>")
        .on("click", function() { alert($(this).text()); })
        .append("<h2>World</h2>");
});
</script>

<header>Placeholder</header>
```

Wrap the jQuery code in the DOM Ready event

Exterminate

False Start Bug

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

<script>
$(function () {
    $("header")
        .html("<h1>Hello</h1>")
        .on("click", function() { alert($(this).text()); })
        .append("<h2>World</h2>");
});
</script>

<header>Placeholder</header>
```

Or you can use the short-hand version of the DOM Ready event

Exterminate

False Start Bug

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

<script>
$(document).on("ready", function () {
    $("header")
        .html("<h
        .on("clic                              .text()); })
        .append("
});
</script>

<header>Placeholder</header>
```

Don’t use the .on(“ready”, handler) syntax to wire this up as it won’t call the handler if the “ready” event already occurred

Exterminate

False Start Bug

```
<body>
    <header>Placeholder</header>

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

    <script>
    $("header")
        .html("<h1>Hello</h1>")
        .on("click", function () { alert($(this).text()); })
        .append("<h2>World</h2>");
    </script>
</body>
```

Or you can move your jQuery code to the bottom of your body element

Exterminate

False Start Bug

```
require(["jquery"], function($) {
    $(document).ready(function() {
        /* DOM Ready */
    });
});
```

You can use the domReady module for DOM Ready if you ant to use jQuery's

```
require(["domReady"], fu          Ready) {
    domReady(function() {          Ready */ });
});
```

```
require(["domReady!"], function(doc) {
    /* DOM Ready */
});
```

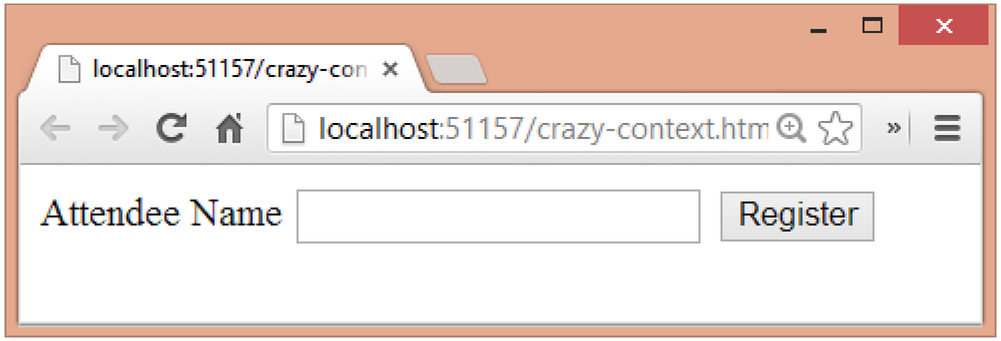
Append ! to force the require callback to wait for DOM Ready before executing

Crazy Context Bug

Examine

Crazy Context Bug

```
<label for="attendee-name">Attendee Name</label>

<input id="attendee-name" type="text"></input>

<button id="register" data-target="#attendee-
name">Register</button>
```



Examine

Crazy Context Bug

```
var Conference = function(name) {
    this.name = name; this.attendees = [];
};

Conference.prototype.register = function(e) {
    var $attendee = $($(e.target).data("target"));
    this.attendees.push($attendee.val());
    alert("Registered for " + this.name + ". " +
        this.attendees.length + " registered so far." );
    $attendee.val("");
};

var conf = new Conference("JavaScript Code Camp");
$("#register").on("click", conf.register);
```

Expose

ntext Bug

```
ie) {
tendees = [];


                        {
                     get"));
ndee.val());
 this.name + ". " +
 + " registered so far." );




vaScript Code Camp");
nf.register);
```

Uncaught TypeError: Cannot call method 'push' of undefined

Explain

Crazy Context Bug

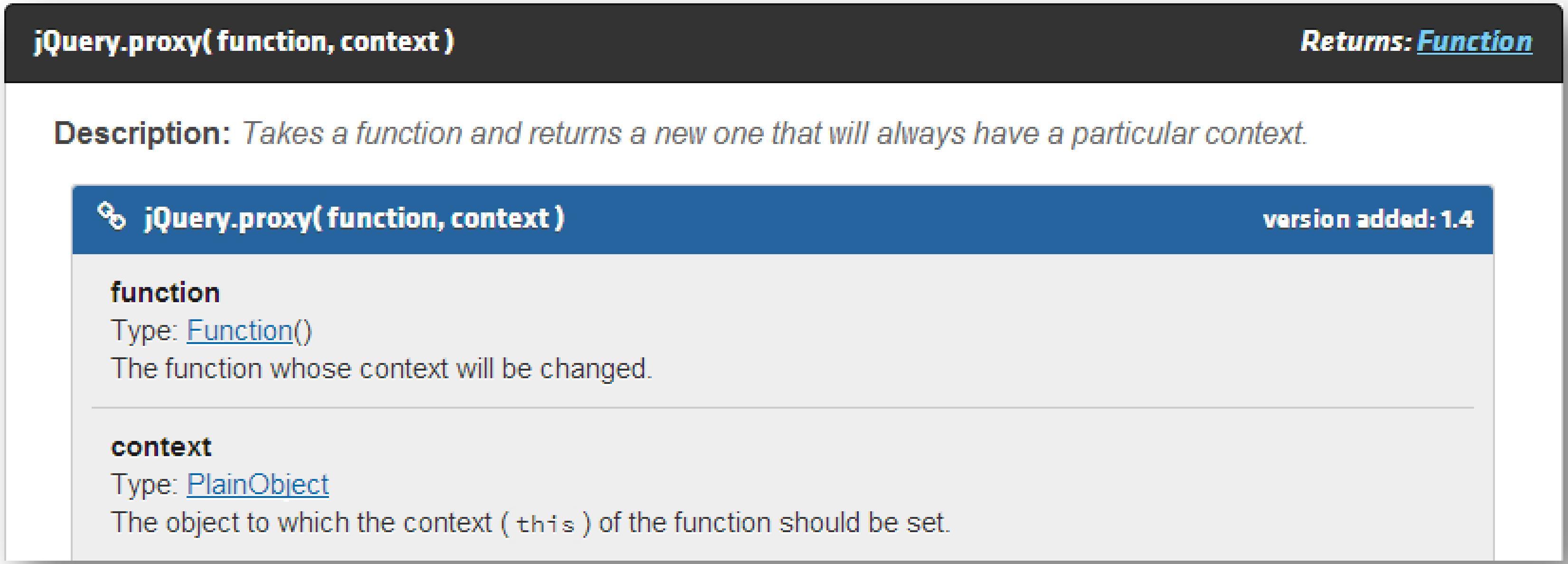
```
<label for="attendee-name">Attendee Name</label>
<input id="attendee-name" type="text"></input>
<button id="register" data-target="#attendee-
name">Register</button>

Conference.prototype.register = function(e) {
```

jQuery sets the `this` implicit argument to the DOM element in question. In this case it is the raw DOM button that was clicked

```
    this.attendees.push($attendee.val());
};

var conf = new Conference("JavaScript Code Camp");
$("#register").on("click", conf.register);
```

Explain

Crazy Context Bug



jQuery.proxy(function, context) Returns: Function

Description: *Takes a function and returns a new one that will always have a particular context.*

jQuery.proxy(function, context) version added: 1.4

function
Type: Function()
The function whose context will be changed.

context
Type: PlainObject
The object to which the context (`this`) of the function should be set.

.proxy() allows you to control what the `this` implicit parameter will be in your event handler

Exterminate

Crazy Context Bug

```
var Conference = function(name) {
    this.name = name; this.attendees = [];
};

Conference.prototype.register = function(e) {
    var $attendee = $($(e.target).data("target"));
    this.attendees.push($attendee.val());
    alert("Registered for " + this.name + ". " +
        this.attendees.length
    $attendee.val("");
};

var conf = new Conference("JavaScr      Code Camp");
$("#register").on("click", $.proxy(conf.register, conf));
```

Let jQuery know that when calling the conf.register event handler have the `this` implicit parameter equal to conf

Exterminate

Crazy Context Bug

```
$("#register").on("click", function (e) {
    conf.register(e);
});

$("#register").on("click", function (e) {
    conf.register.call(conf, e);
});

$("#register").on("click", function (e) {
    conf.register.apply(conf, [e]);
});

$("#register").on("click", conf.register.bind(conf));
```

Tightly Bound Bug

pluralsight
hardcore developer training

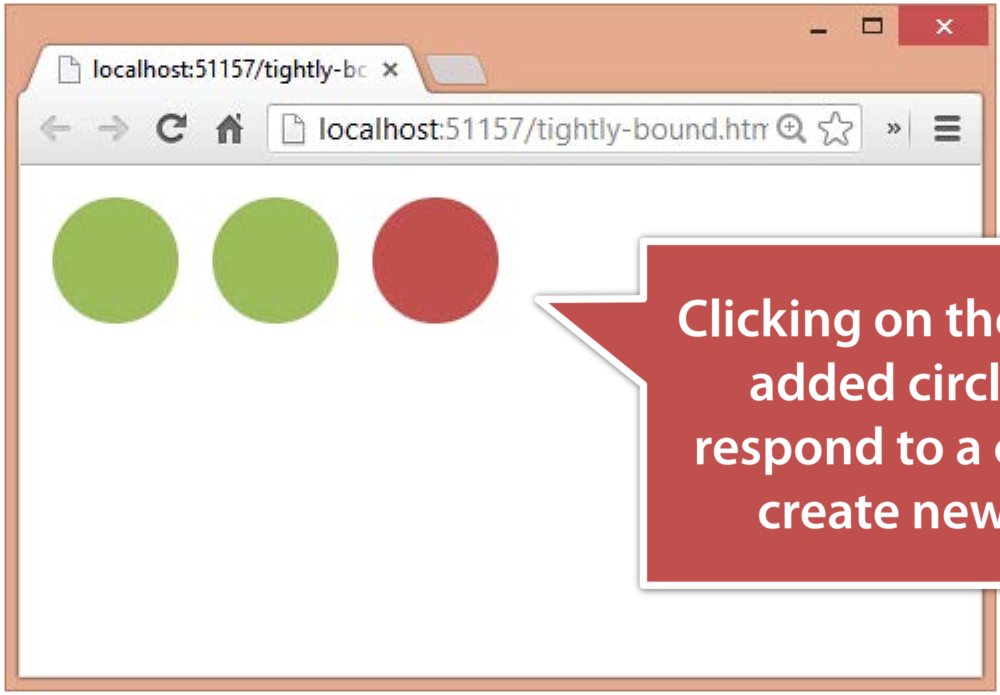
Examine

Tightly Bound Bug

```
<div id="shapes">
    <div class="circle"></div>
    <div class="circle"></div>
</div>

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

<script>
$(".circle").on("click", function () {
    $("<div class='circle dynamic'></div>")
        .appendTo("#shapes");
});
</script>
```

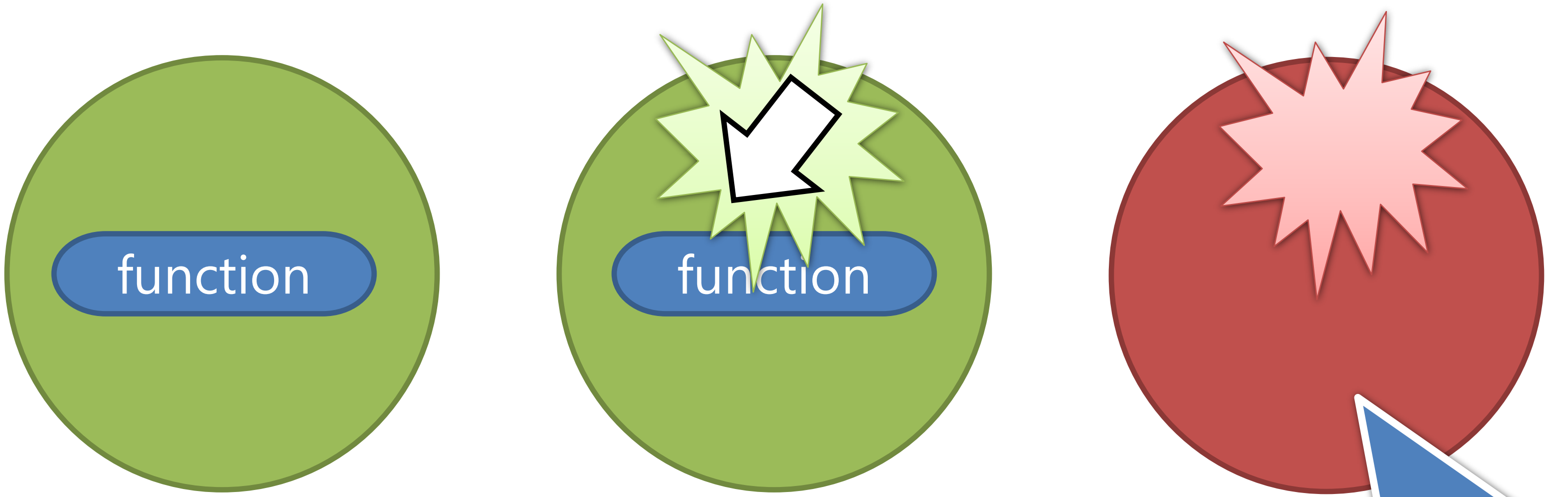
Expose

Tightly Bound Bug



Clicking on the dynamically added circle does not respond to a click event to create new elements

Tightly Bound Bug

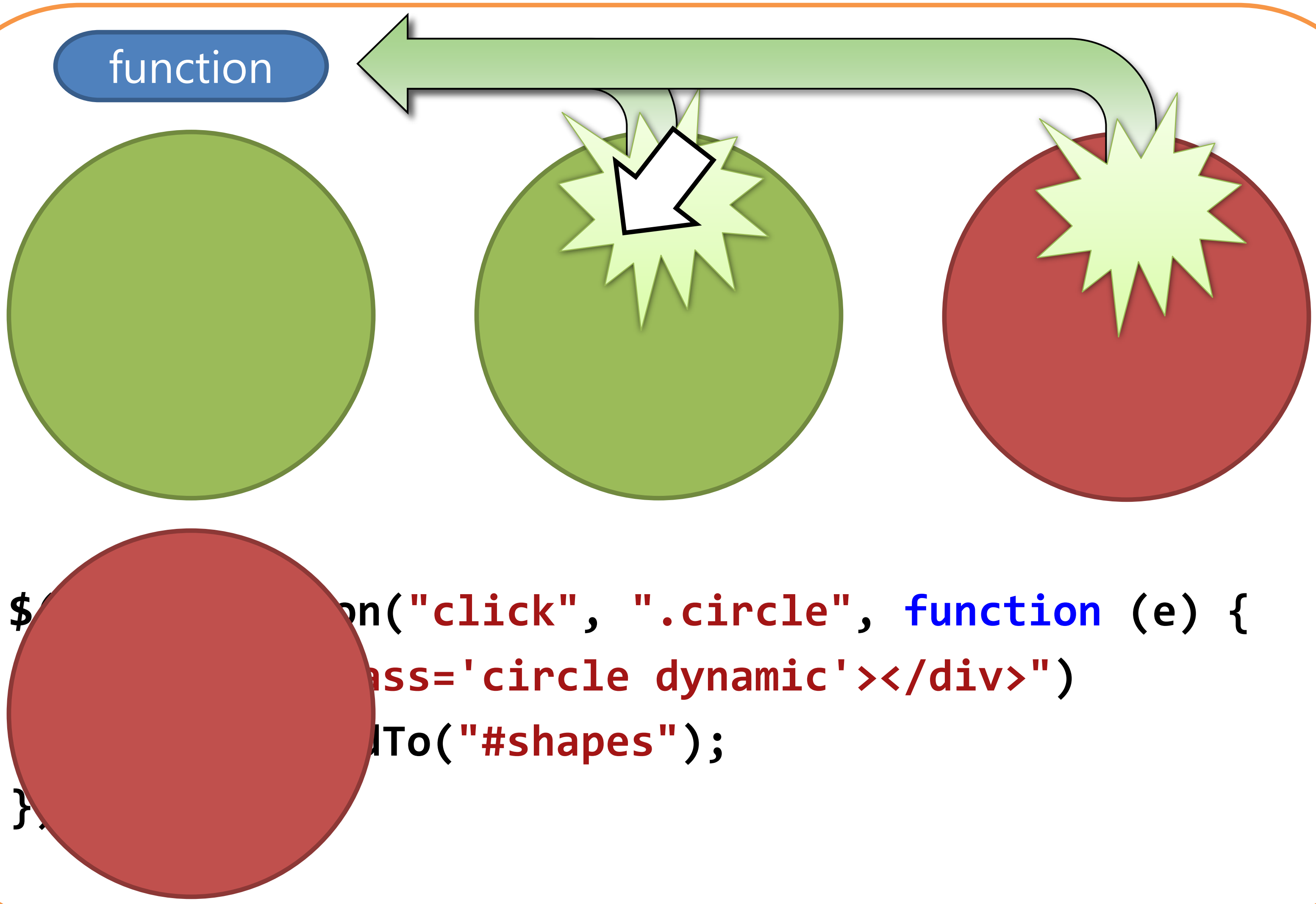


```
$(".circle").on("click",
    function () { /* ... */ });
```

Since this is a dynamic element there is no click event handler like the static elements

Explain

Tightly Bound Bug



```
$(          on("click", ".circle", function (e) {
              ass='circle dynamic'></div>")
              dTo("#shapes");
}
```

Exterminate

Tightly Bound Bug

```
<div id="shapes">
    <div class="circle"></div>
    <div class="circle"></div>
</div>

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

<script>
$("#shapes").on("click", ".circle", function (e) {
    $("<div class='circle dynamic'></div>")
        .appendTo("#shapes");
});
</script>
```

Event delegation instead of traditional event binding

Browser Madness Bug

Examine

Browser Madness Bug

```
<form>
    <input id="age" type="text" />
    <input type="submit" value="Save" />
</form>

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

<script>
$("#age").on("keypress", function (e) {
    var char = String.fromCharCode(e.charCode);

    if (!~"0123456789".indexOf(char)) { return false; }
});
</script>
```

Expose

adness Bug

```
xt" />
ue="Save" />

nin.js"></script>

ction (e)
arCode(e.charCode);

Of(char)) { return false; }
```

charCode in IE8 is undefined ☹

Explain

Browser Madness Bug

```
// jQuery Source Code to Normalize the Madness
if (event.which == null) {
    event.which = original.charCode != null ?
        original.charCode :
        original.keyCode;
}

var name;
if (name == null) {}
if (name === null || name === undefined) {}

if (name != null) {}
if (name !== null && name !== undefined) {}
```

As a side note you may be thinking… jQuery is using `==` and `!=`? Isn't it best practice to use `===` and ‘!==`?

Exterminate

Browser Madness Bug

```
<form>
    <input id="age" type="text" />
    <input type="submit" value="Save" />
</form>

<script src="Scripts/jquery

<script>
$("#age").on("keypress", function (e) {
    var char = String.fromCharCode(e.which);

    if (!~"0123456789".indexOf(char)) { return false; }
});
</script>
```

Used the jQuery normalized `which` property to get around cross-browser inconsistencies

Unintentional Destruction Bug

Examine

Unintentional Destruction Bug

```
<div id="shps"><div class="circle"></div></div>
<script src="Scripts/jquery.min.js"></script>
<script>
$("#shps").on("click", function() {
    $(this).css({ backgroundColor: '#' + Math.floor(
Math.random() * 16777215).toString(16) });
});
$("#shps").on("click dblclick", ".circle", function(e) {
    $(this).clone(true).appendTo("#shps");
    if (e.type === "dblclick") {
        $(e.delegateTarget).off("click");
    }
});
</script>
```

Expose

Destruction Bug

```
:ircle"></div></div>
.min.js"></script>

tion() {
dColor: '#' + Math.floor(
oString(16) });

ck", ".cir
endTo("#sh
<") {
.off("click");
```

By removing the "click" handler you are removing ALL the "click" handlers!

Explain

Unintentional Destruction Bug

```
// Removes all event handlers
$("#shapes").off();

// Removes all click event handlers
$("#shapes").off("click");

// Removes just this click event handler (traditional or
delegated)
function changeColor() { /* ... */ }
$("#shapes").on("click", changeColor);
$("#shapes").off("click", changeColor);
```

Explain

Unintentional Destruction Bug

```
// Removes just the shape namespaced click event handler
(traditional & delegated)
$("#shapes").on("click.shape", changeColor);
$("#shapes").off("click.shape");

// Removes any shape namespaced event handlers
(traditional & delegated)
$("#shapes").off(".shape");

// Removes all shape namespaced delegated event handlers,
but keeps traditional handlers intact
$("#shapes").off("click.shape", "**");
```

Explain

Unintentional Destruction Bug

```
// Removes all delegated click event handlers
$("#shapes").off("click", ".circle");

// Removes just this delegated click event handler
$("#shapes").off("click", ".circle", cloneShape);

// Removes just shape namespaced delegated click event
handlers
$("#shapes").off("click.shape", ".circle");

// Removes all the shape namespaced delegated event
handlers
$("#shapes").off(".shape", ".circle");
```

Explain

Unintentional Destruction Bug

```
// Removes event handlers from the event types listed
(traditional & delegated)
$("#shapes").off({
    "click" : changeColor,
    ".shape": highlightShape
});
```

Exterminate

Unintentional Destruction Bug

```
<div id="shps"><div class="circle"></div></div>
<script src="Scripts/jquery.min.js"></script>
<script>
$("#shps").on("click", function() {
    $(this).css({ /* ... */ });
});
$("#shps").on("click.shape dblclick", ".circle",
    function(e) {
        $(this).clone(true
        if (e.type === "db
            $(e.delegateTarget).off("click.shape");
        }
    });
</script>
```

Introduced a namespace onto the click event for removal later

Exterminate

Unintentional Destruction Bug

```
<div id="shps"><div class="circle"></div></div>
<script src="Scripts/jquery.min.js"></script>
<script>
$("#shps").on("click", function() {
    $(this).css({/* ... */ });
});
$("#shps").on("click dblclick", ".circle",
    function cloneShape(e) {
        $(this).clone(true).appendTo("#shps");
        if (e.type === "dblclick") {
            $(e.delegateTarget).off("click", cloneShape);
        }
    });
</script>
```

Tell the off method exactly which event handler to remove

Secretive Publish Bug

Examine

Secretive Publish Bug

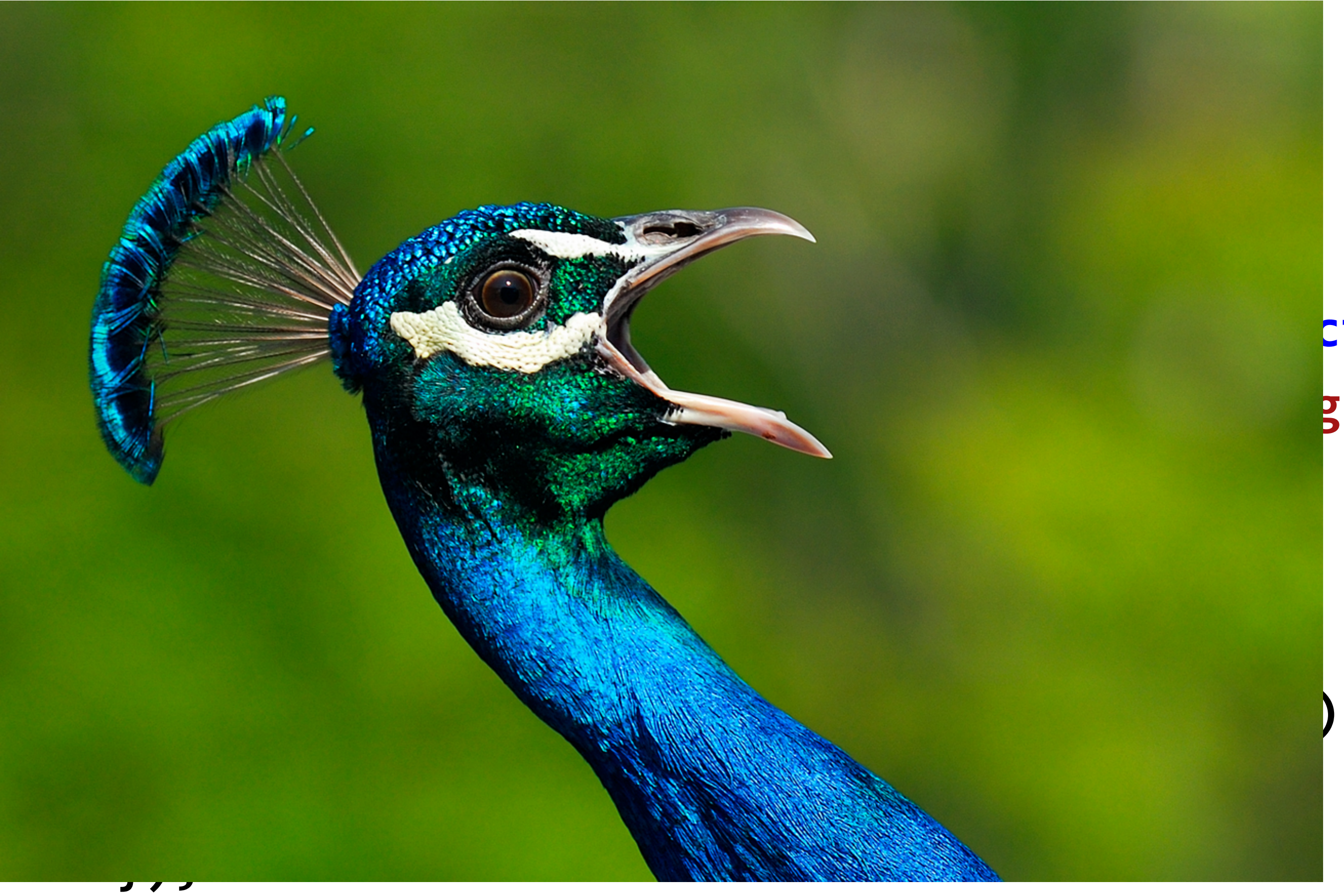
```
<ul id="items">
    <li>
        <input type="checkbox" />
        <div class="content">Testing 1</div>
    </li>
    <li>
        <input type="checkbox" />
        <div class="content">Testing 2</div>
    </li>
    <!-- ... more markup ... -->
</ul>

<button id="invert">Invert All</button>
```

Examine

Secretive Publish Bug

```
$("#items").find("input").on("click", function () {
    $(this).closest("li").toggleClass("highlight");
});

$("#items").on("click", "input", function (e) {
    $(this).closest("li").find(".content")
        .html(function (i, html) { return html + "."; });
});

$("#invert").on("click", function () {
    $("#items input:checkbox")
        .triggerHandler("click")
        .fadeOut("fast").fadeIn("slow");
});
```

Expose



```
ction () {
ghlight");

 (e) {
)
 html + "."; });
});

$("#invert").on("click", function () {
    $("#items input:checkbox")
        .triggerHandler("click")
        .fadeOut("fast").fadeIn("slow");
});
```

Uncaught TypeError: Cannot call method 'fadeOut' of undefined

Explain

Secretive Publish Bug

Default behavior does not happen

Only matches 1st jQuery element

Event does not bubble up the DOM

Not chainable. Returns value from handler

triggerHandler()

Exterminate

Secretive Publish Bug

```
$("#items").find("input").on("click", function () {
    $(this).closest("li").toggleClass("highlight");
});

$("#items").on("click", "input", function (e) {
    $(this).closest("li").find(".content")
        .html(function (i, html) { return html + "."; });
});

$("#invert").on("click", function
    $("#items input:checkbox")
        .trigger("click")
        .fadeOut("fast").fadeIn("slow");
});
```

Use .trigger() instead. Chainable, matches all elements, bubble up DOM, performs default behavior

Confusing Element Bug

Examine

Confusing Element Bug

```
<div id="container">
    <div class="widget">
        <div class="content">placeholder</div>
    </div>
</div>

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

Examine

Confusing Element Bug

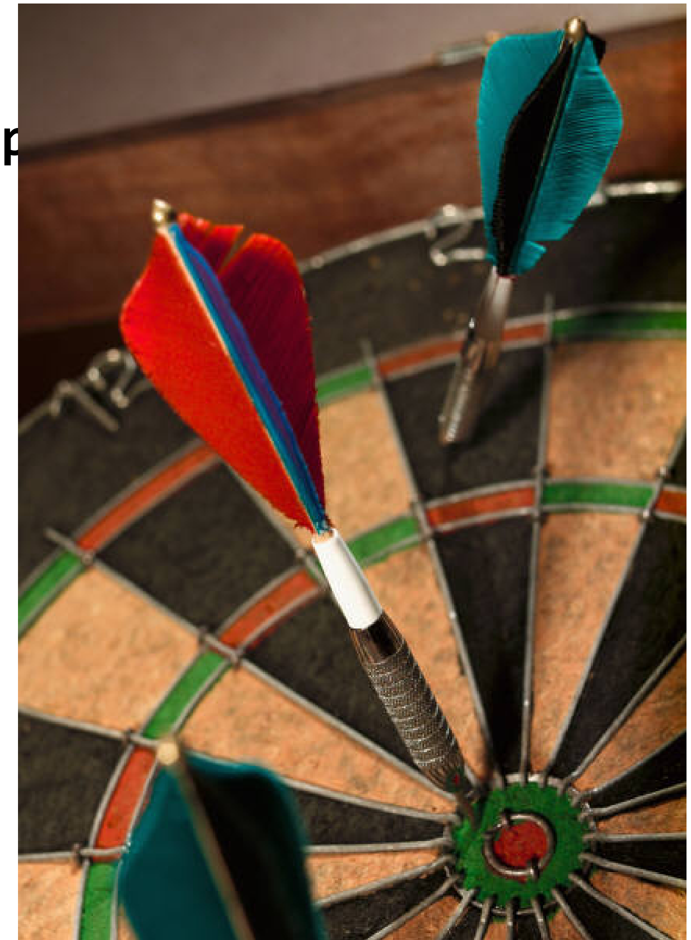
```
var app = {
    name: "Fixing Common jQuery Bugs",
    handler: function (e) {
        var $elem = $(e.target);
        $elem.find(".content").text(app.name);
    }};

$("#container").on("click", ".widget", app.handler);
```

Expose

Confusing Element Bug

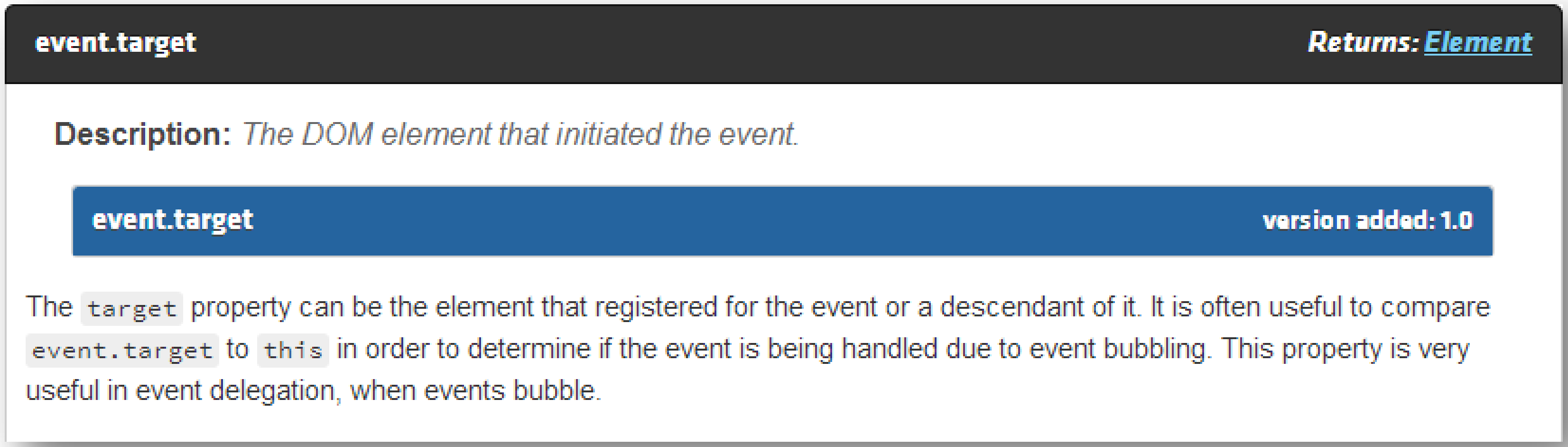
```
var app = {
    name: "Fixing Commo
    handler: function (e) {
        var $elem = $(e.target);
        $elem.find(".content").text(app
    }};

$("#container").on("click", ".widget",
```

The target could be the .content or the .widget depending on where the user clicked



Explain

Confusing Element Bug

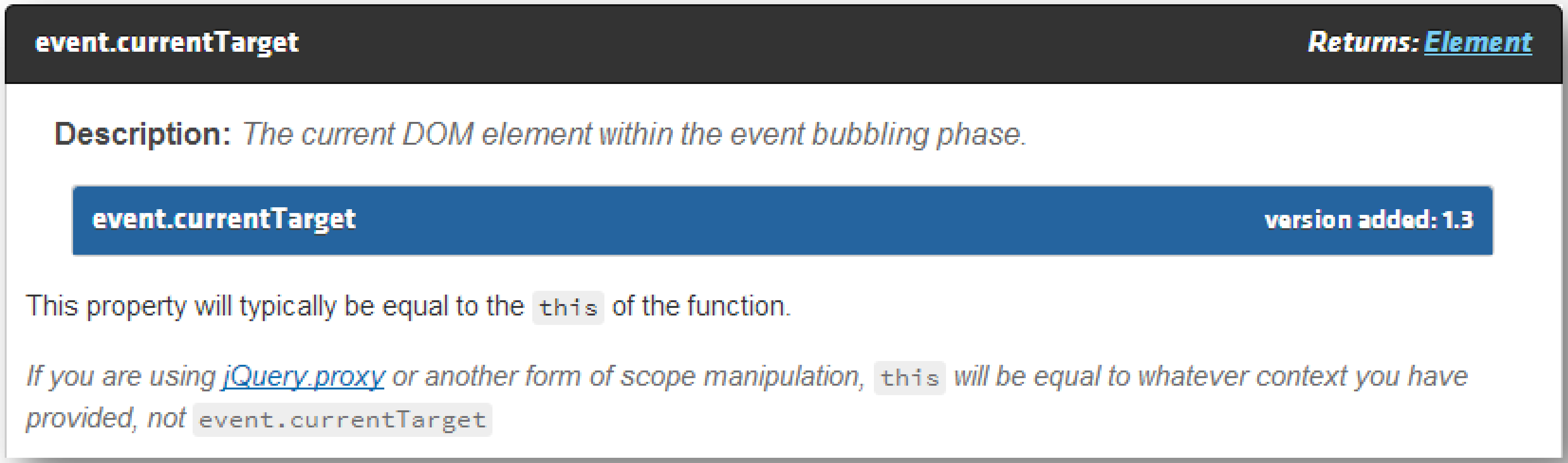


event.target *Returns: Element*

Description: *The DOM element that initiated the event.*

event.target version added: 1.0

The `target` property can be the element that registered for the event or a descendant of it. It is often useful to compare `event.target` to `this` in order to determine if the event is being handled due to event bubbling. This property is very useful in event delegation, when events bubble.



event.currentTarget *Returns: Element*

Description: *The current DOM element within the event bubbling phase.*

event.currentTarget version added: 1.3

This property will typically be equal to the `this` of the function.

If you are using jQuery.proxy or another form of scope manipulation, `this` *will be equal to whatever context you have provided, not* `event.currentTarget`

Exterminate

Confusing Element Bug

You can use `this` instead of e.target & it will typically be the element referenced from your selector

```
var
    handler: functio  (e) {
        var $elem = $(this);
        $elem.find(".content").text(app.name);
    }
};

$("#container").on("click", ".widget", app.handler);
```

Exterminate

Confusing Element Bug

```
var app = {
    name: "Fixing Co
    handler: function (e
        var $elem = $(this);
        $elem.find(".content").text(app.name);
    }
};
$("#container").on(
    "click",
    ".widget",
    $.proxy(app.handler, app)
);
```

… then `this` will now be the `app` object and not the `.widget` ☹

If someone happens to use $.proxy() to manipulate the context of the handler…

Exterminate

Confusing Element Bug

```
var app = {
    name: "Fixing Common jQuery Bugs",
    handler: function (e) {
        var $elem = $(e.currentTarget);
        $elem.find(".content").text(this.name);
    }
};

$("#container").on(
    "click",
    ".widget",
    $.proxy(app.handler, app)
);
```

Here we can use `e.currentTarget` to reference the DOM element and `this` to reference the `app` object

Chicken Egg Bug

pluralsight
hardcore developer training

Examine

Chicken Egg Bug

```
<div>
    Today's Temperature: <span id="temperature"></span>
</div>

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

<script>
var weather;
$.getJSON("/weather", function (data) {
    weather = data.forecast;
});
$("#forecast").html(weather.temperature);
</script>
```

Expose

Chicken Egg Bug

```
erature: <span id="temperature"></span>

cri   rc="Scripts/jquery.js"></script>

<script>
var weather;
$.getJSON("/weather'
    weather = data.f
});
$("#forecast").html(weather.temperature);
</script>
```

Using `weather` before the response from Ajax has returned from the server

Explain

Chicken Egg Bug

```
1   $.ajax({
        url: "/weather",
        dataType: "json",
        success: function(data, status, xhr) {
3           console.log("Ajax was successful");
|       },
or
|       error: function(xhr, status, error) {
3           console.log("Ajax failed");
        },
        complete: function(xhr, status) {
4           console.log("Ajax is done");
        }
    });
2   console.log("Ajax is still requesting…");
```

Exterminate

Chicken Egg Bug

```
<div>
    Today's Temperature: <span id="temperature"></span>
</div>

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

<script>
$.getJSON("/weather", function (data) {
    var weather = data.forecast;
    $("#temperature").html(weather.temperature);
});
</script>
```

Moved the DOM manipulation inside the callback

Exterminate

Chicken Egg Bug

```
$.ajax({
    url: "/weather",
    dataType: "json",
    success: function (data) {
        var weather = data.forecast;
        $("#forecast").html(weather.temperature);
    }
});
```

Moved the DOM manipulation inside the `success` callback

Exterminate

Chicken Egg Bug

```
$.getJSON("/weather").done(function(data) {
    var weather = data.forecast;
    $("#temperature").html(weather.temperature);
});
```

Use the new Promise that Ajax returns and hook into when it's done

```
$.ajax({
    url: "/weather",
    dataType: "json"
}).done(function (data) {
    var weather = data.forecast;
    $("#forecast").html(weather.temperature);
});
```

Security Access Bug

Examine

Security Access Bug

```
<div>
    Today's Temperature: <span id="temperature"></span> K
</div>

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

<script>
$.getJSON("http://api.openweathermap.org/data/2.5/weather
?q=Nashville,TN", function (data) {
    var weather = data.main;
    $("#temperature").html(weather.temp);
});
</script>
```

Examine

ccess Bug

```
an id="temperature"></span> K
```

XMLHttpRequest cannot load
http://api.openweathermap.org/data/2.5/...

Origin http://elijahmanor.com is not
allowed by Access-Control-Allow-Origin.

```
athermap.org/data/2.5/weather
data) {

eather.temp);
```

Expose

Security Access Bug

The same-origin policy requires that the request matches the same domain, protocol, and port number

http://elijahmanor.com

URL	Outcome	Reason
http://elijahmanor.com/dir/page2.html	Success	Same
http://elijahmanor.com/dir2/other.html	Success	Same
https://elijahmanor.com/dir1/test.html	Failure	Different Protocol
http://en.elijahmanor.com/dir/other.html	Failure	Different Host
http://www.elijahmanor.com/dir3/test.html	Failure	Different Host
http://elijahmanor.com:88/dir4/test.html	Failure	Different Port

Explain

Security Access Bug

JSONP is a simple way to get around the same-origin policy. What does that mean?

Script tags don't follow the same rules, which is how we can do this…

```
<script
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.1/jquery
.min.js"></script>
```

JSONP uses this "technique" to get around the same-origin policy.

Security Access Bug

Exterminate

```
<div>
    Today's Temperature: <span id="temperature"></span> K
</div>

<script src="Scripts/jq

<script>
$.getJSON("http://api.openweathermap.org/data/2.5/weather
?q=Nashville,TN&callback=?", function (data) {
    var weather = data.main;
    $("#temperature").html(weather.temp);
});
</script>
```

Provide an additional &callback=? URL parameter

Exterminate

Security Access Bug

```
$.getJSON("http://api.openweathermap.org/data/2.5/weather
?q=Nashville,TN&callback=?", function (data) {});
```

```
<script
src="http://api.openweathermap.org/data/2.5/weather?q=Nas
hville,TN&callback=jQuery19103622920857742429_13709242552
02&_=1370924255203"></script>
```

```
function jQuery19103622920857742429_1370924255202(data) {
    /* ... Wires into Ajax Callback ... */
}
```

```
jQuery19103622920857742429_1370924255202({ "main": {
"temp": 294.01, "humidity": 95, "pressure": 1013 }, "id":
4644585 });
```

Exterminate

Security Access Bug

```
$.ajax({
    url: "http://api.openweathermap.org/data/2.5/weather",
    dataType: "jsonp",
    data: { q: "Nashville, TN" },
    success: functio
        var weather
        $("#temperat
    }
});
```

jQuery will auto add the &callback=? With $.ajax when using dataType of "jsonp". Also you can use the data property that will be added to the URL

Conclusion

- Wait for the DOM to be Ready
- Make sure `this` is what you think it is
- Use delegated events when it make sense
- Use event.which to avoid cross-browser issues
- The trigger method is probably what you want
- The event.currentTarget is probably what you want
- Get comfortable with asynchronous code
- Don’t forget to tell Ajax that you want JSONP