Fixing Common jQuery Bugs

Selectors, Traversing, & Manipulation Bugs

Elijah Manor
@elijahmanor
http://elijahmanor.com

pluralsight
hardcore developer training

Outline

Houdini Hieroglyph Bug

Superfluous Spawn Bug

Always Alive Bug

Brittle Blood Bug

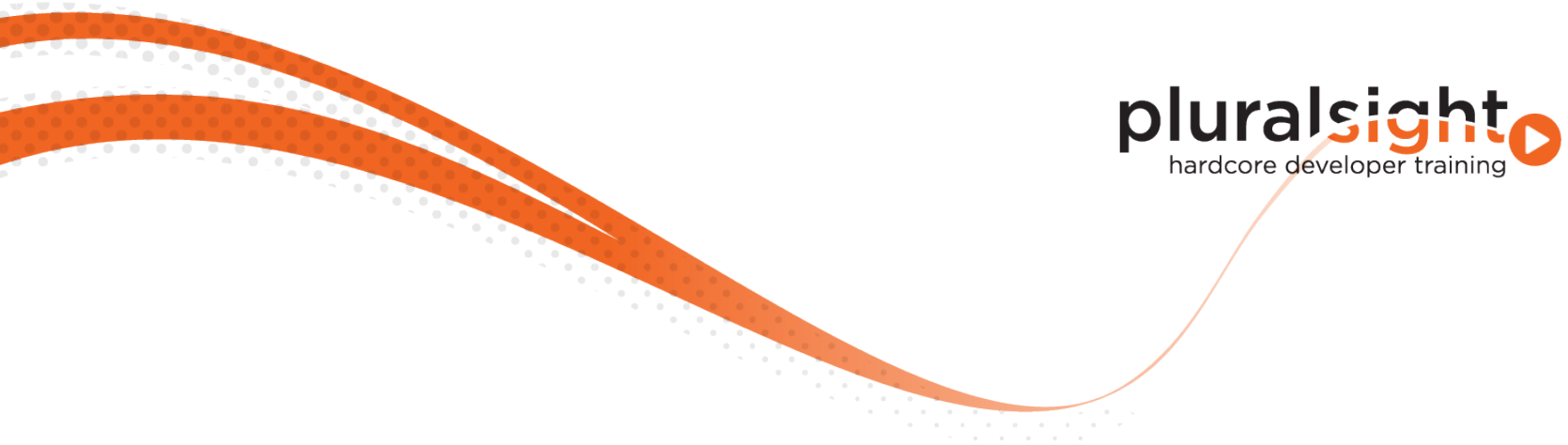
Forgetful Family Bug

Discarded Descendant Bug

Checking Condition Bug

Barbaric Behavior Bug

Houdini Hieroglyph Bug



pluralsight
hardcore developer training

Houdini Hieroglyph Bug

Examine

```
<form id="personForm" name="personForm" method="post"
action="/Demo/jsf/main.jsf;jsessionid=0596FB948C..."
enctype="application/x-www-form-urlencoded">
    <div class="field">
        <label>Last Name</label>
        <input id="personForm:lastname"
name="personForm:lastname" type="text" />
    </div>
    <!-- ... more HTML ... -->
</form>
<script>
$(document).ready(function () {
    $("#personForm:firstname").addClass("required");
});
</script>
```

Expose

```
<form id="pers                                    od="post"
action="/Demo/                                    948C..."
enctype="appli
    <div class
        <label
        <input id="personForm:lastname"
name="personForm:lastname" type="text" />
    </div>
    <!-- ... more HTML ... -->
</form>
<script>
$(document).ready(function () {
    $("#personForm:firstname").addClass("required");
});
</script>
```

Houdini Hieroglyph Bug

Explain

At the root of the problem is that JSF inserts a : delimiter inside of the id attribute. jQuery abides by the W3C CSS Specification Rules.

If your ID, name, or class contains one of the following meta-characters then you have a problem...
!"#$%&'()*+,./:;<=>?@[\]^`{|}~

```
// <input id="person.name" name="person.name" />
$( "#person.name" ).addClass( "highlight" ); // Fail

// <input id="person:name" name="person:name" />
$( "#person:name" ).addClass( "highlight" ); // Fail

// <input id="person$name" name="person$name" />
$( "#person$name" ).addClass( "highlight" ); // Fail
```

Houdini Hieroglyph Bug

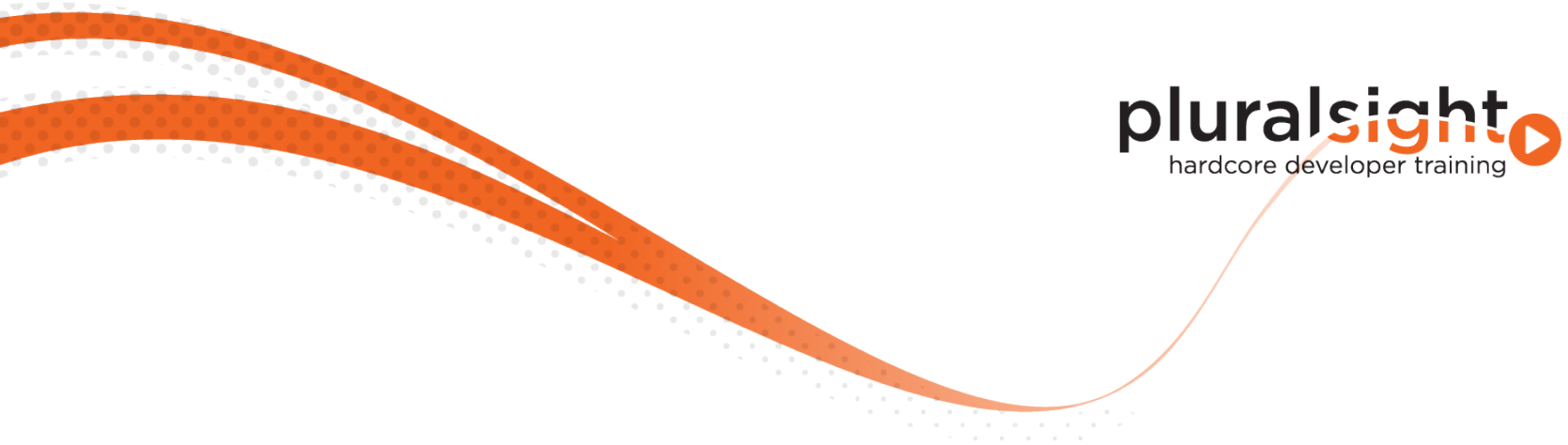
Exterminate

If you wish to use any of the meta-characters (such as !"#$%&'()*+,./:;<=>?@[\]^`{|}~) as a literal part of a name, you must escape the character with two backslashes: \\. For example, if you have an element with id="foo.bar", you can use the selector $("#foo\\.bar").

-- http://api.jquery.com/category/selectors/

```
$( document ).ready( function() {
    $("#personForm\\:firstname")
        .addClass( "required" );
});
```

Superfluous Spawn Bug



pluralsight
hardcore developer training

Superfluous Spawn Bug

Examine

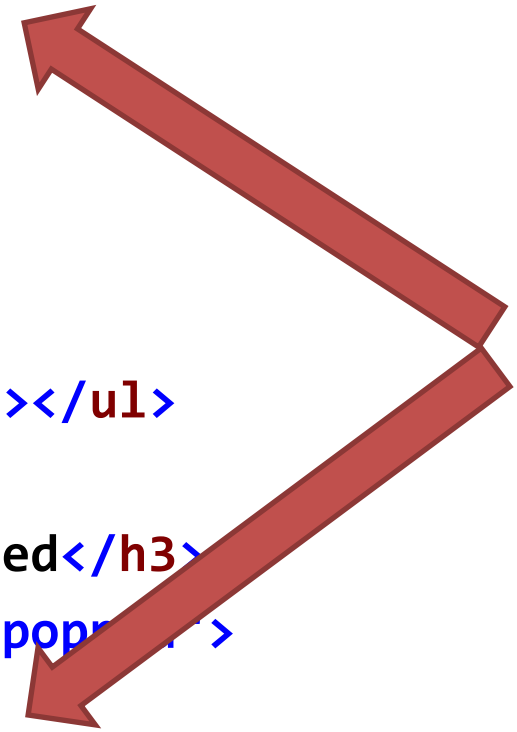
```
<h3>Items to Push</h3>
<ul id="items-to-push"><li>1</li><li>2</li></ul>
<h3>Stack</h3>
<ul id="stack"></ul>
<h3>Items Popped</h3>
<ul id="items-popped"></ul>
<button id="push">Push</button>
<button id="pop">Pop</button>

$("#push").on("click", function () {
    $("li:nth-child(1)").prependTo("#stack")
});
$("#pop").on("click", function () {
    $("#stack li:nth-child(1)").appendTo("#items-
popped");
});
```

Superfluous Spawn Bug

Expose

Superfluous Spawn Bug

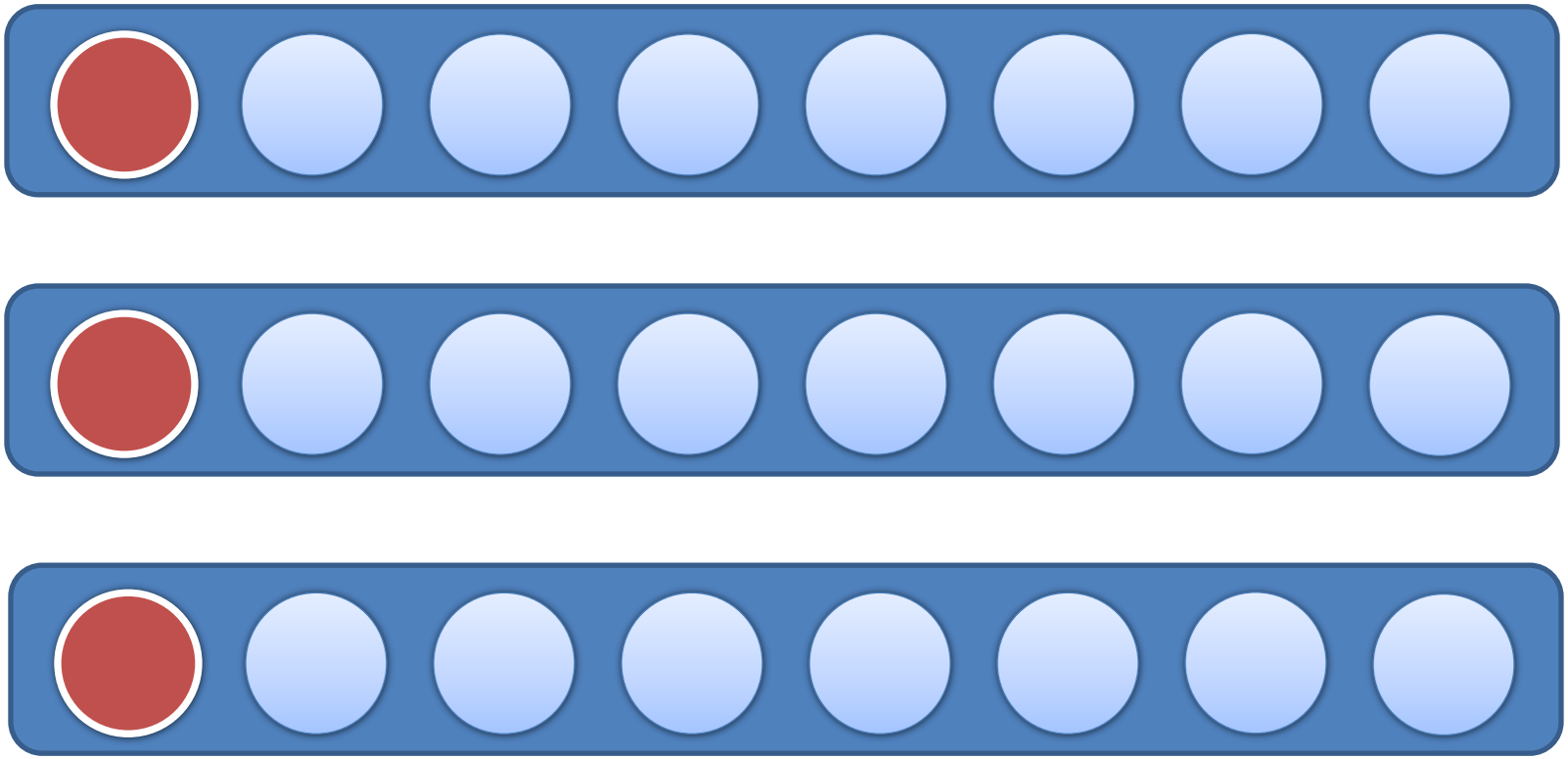

```
<h3>Items to Push</h3>
<ul id="items-to-push">
    <li>3</li>
    <li>4</li>
</ul>

<h3>Stack</h3>
<ul id="stack"></ul>

<h3>Items Popped</h3>
<ul id="items-popped">
    <li>2</li>
    <li>1</li>
</ul>
```



```
$("li:nth-child(1)")
```

Superfluous Spawn Bug

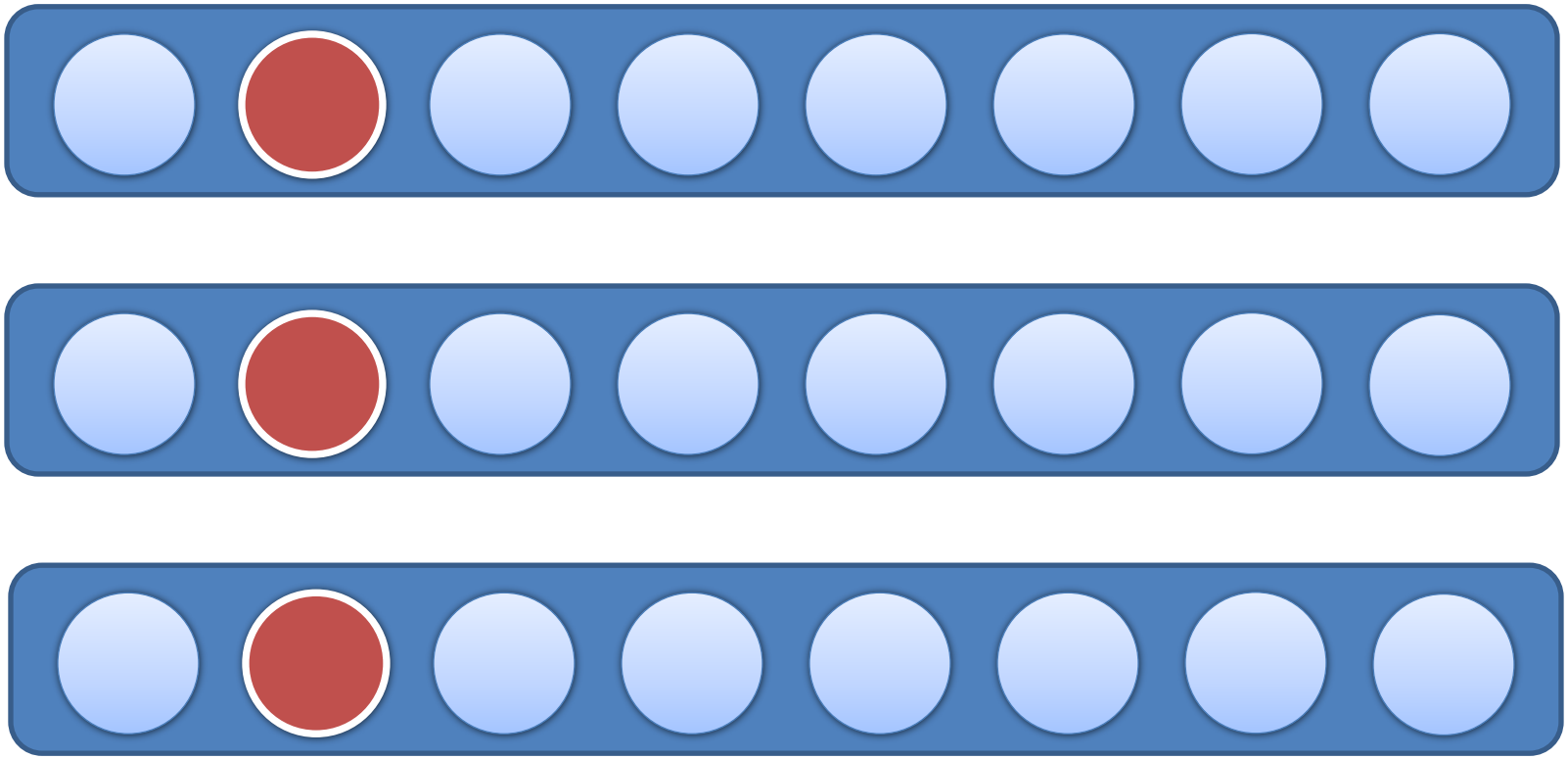
Explain



```
$("li:nth-child(1)")
```

Superfluous Spawn Bug

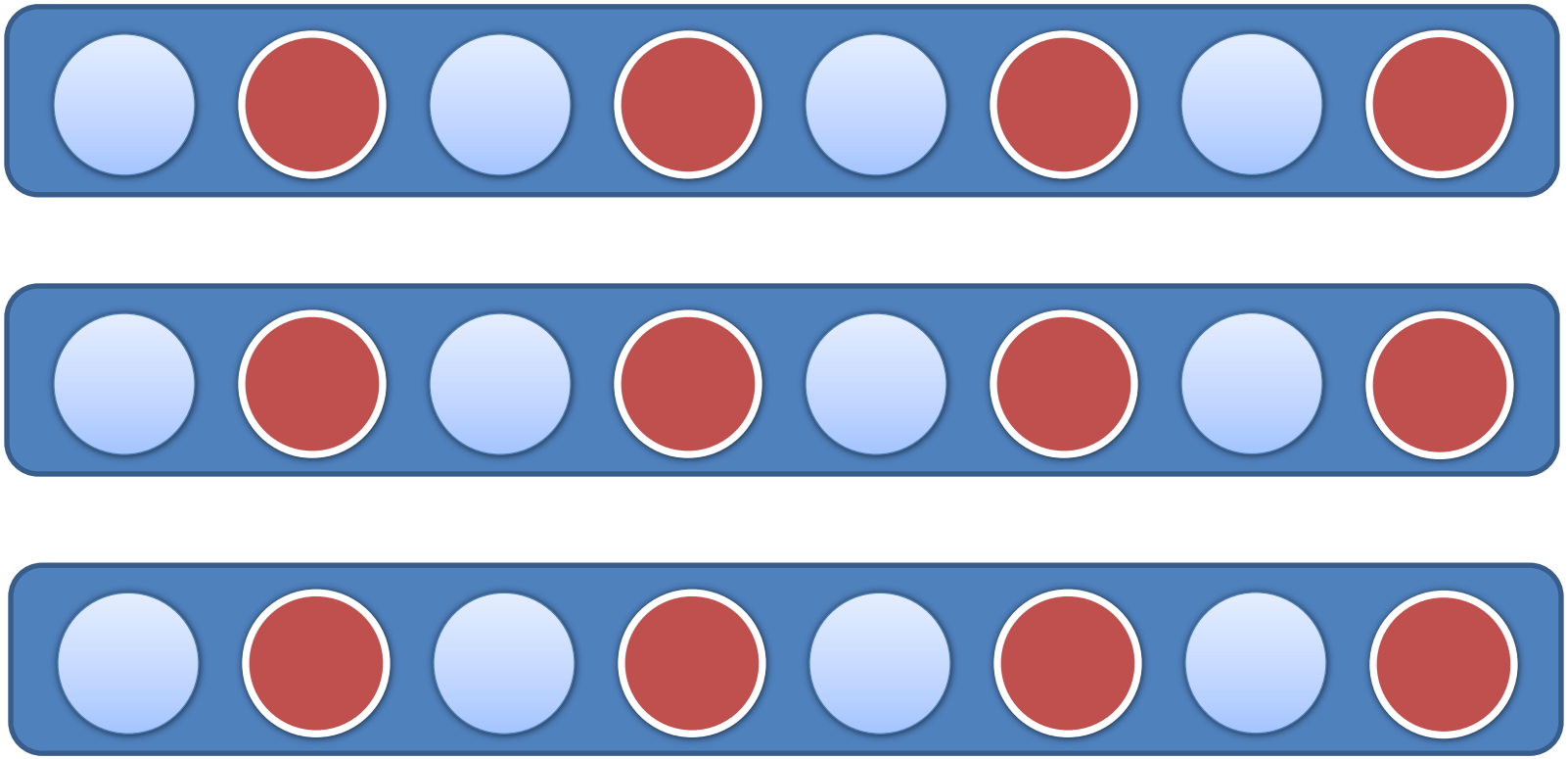
Explain



```
$("li:nth-child(2)")
```

Superfluous Spawn Bug

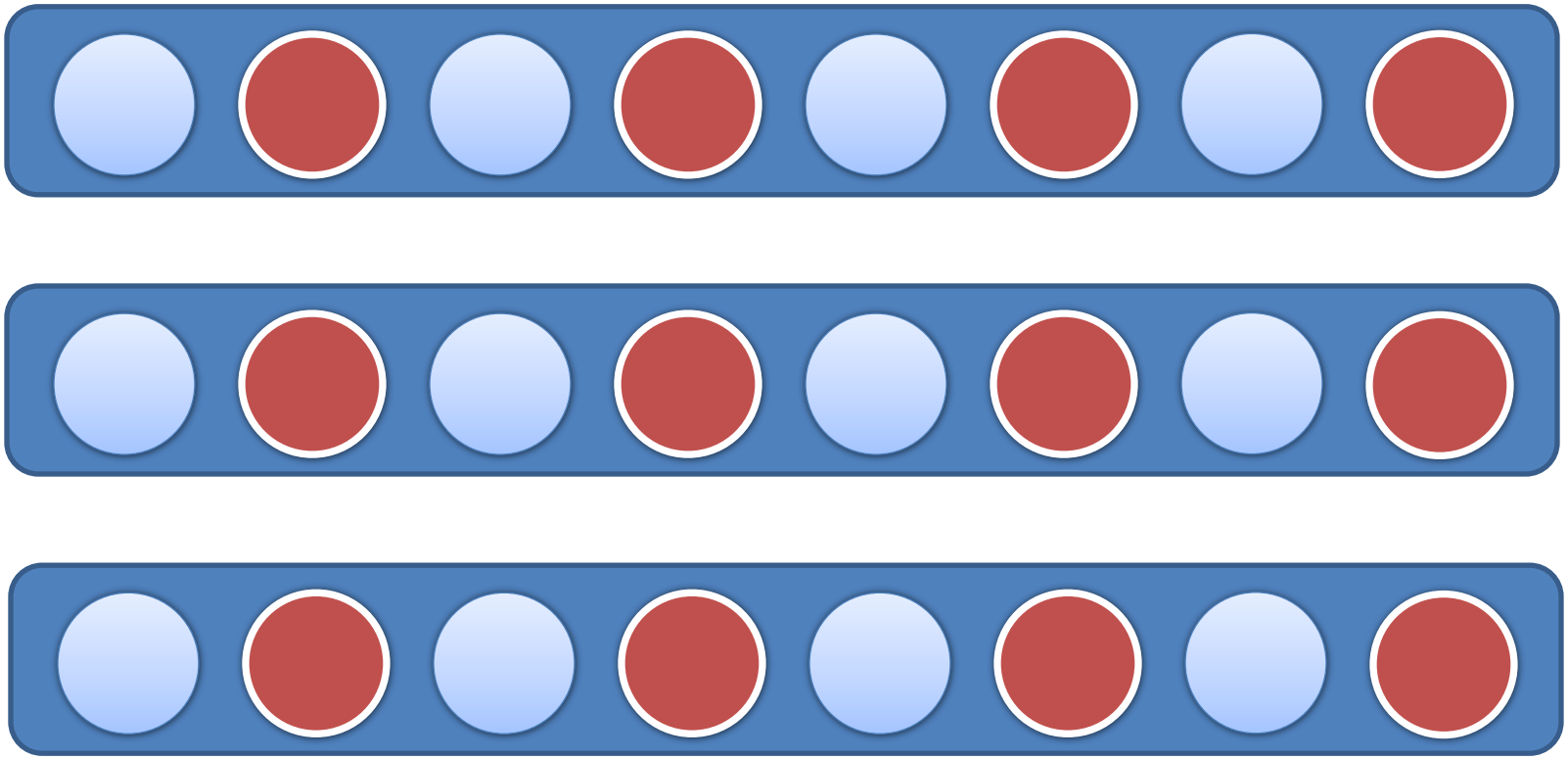
Explain



```
$("li:nth-child(2n)")
```

Superfluous Spawn Bug

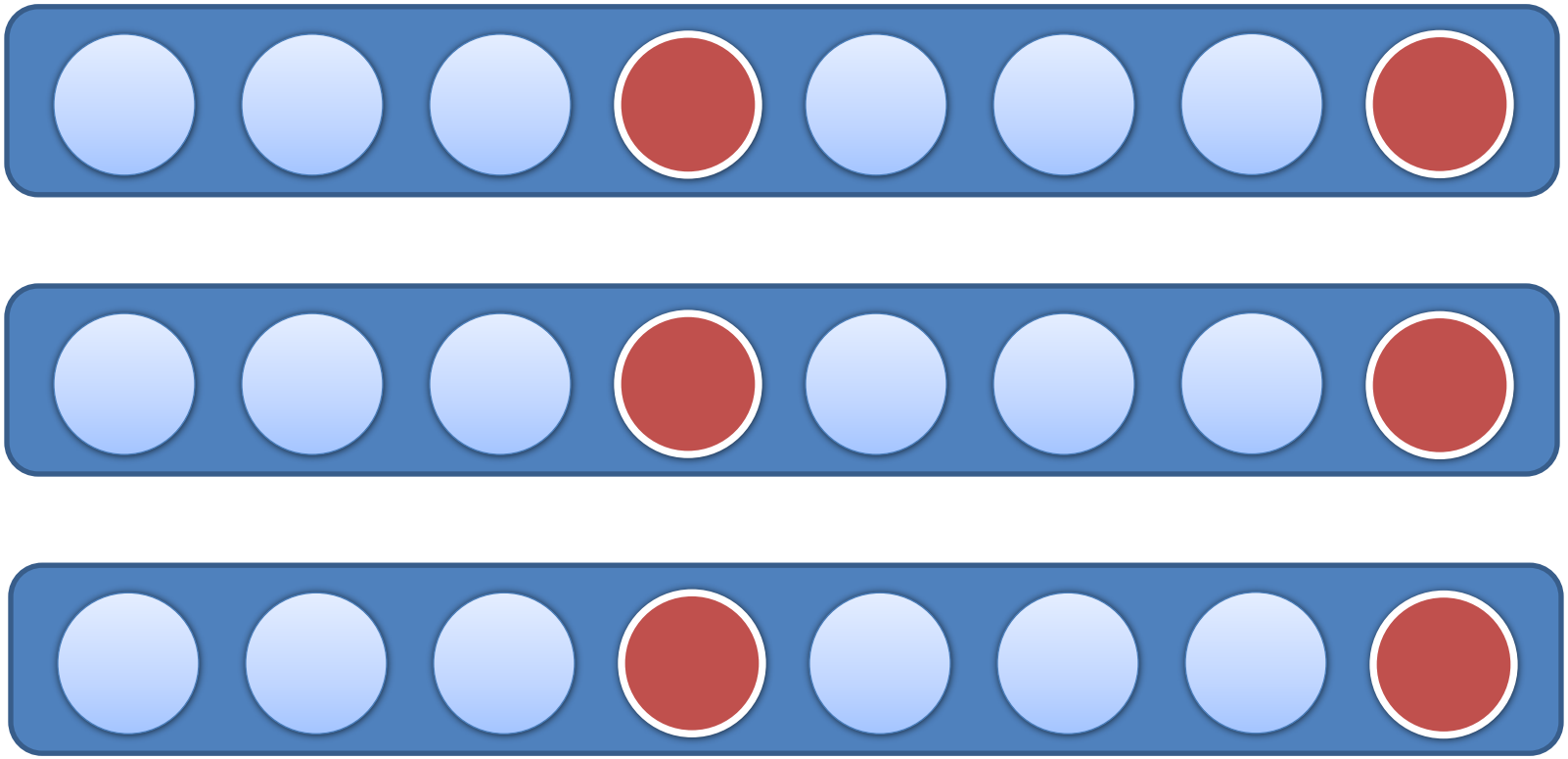
Explain



```
$("li:nth-child(even)")
```

Superfluous Spawn Bug

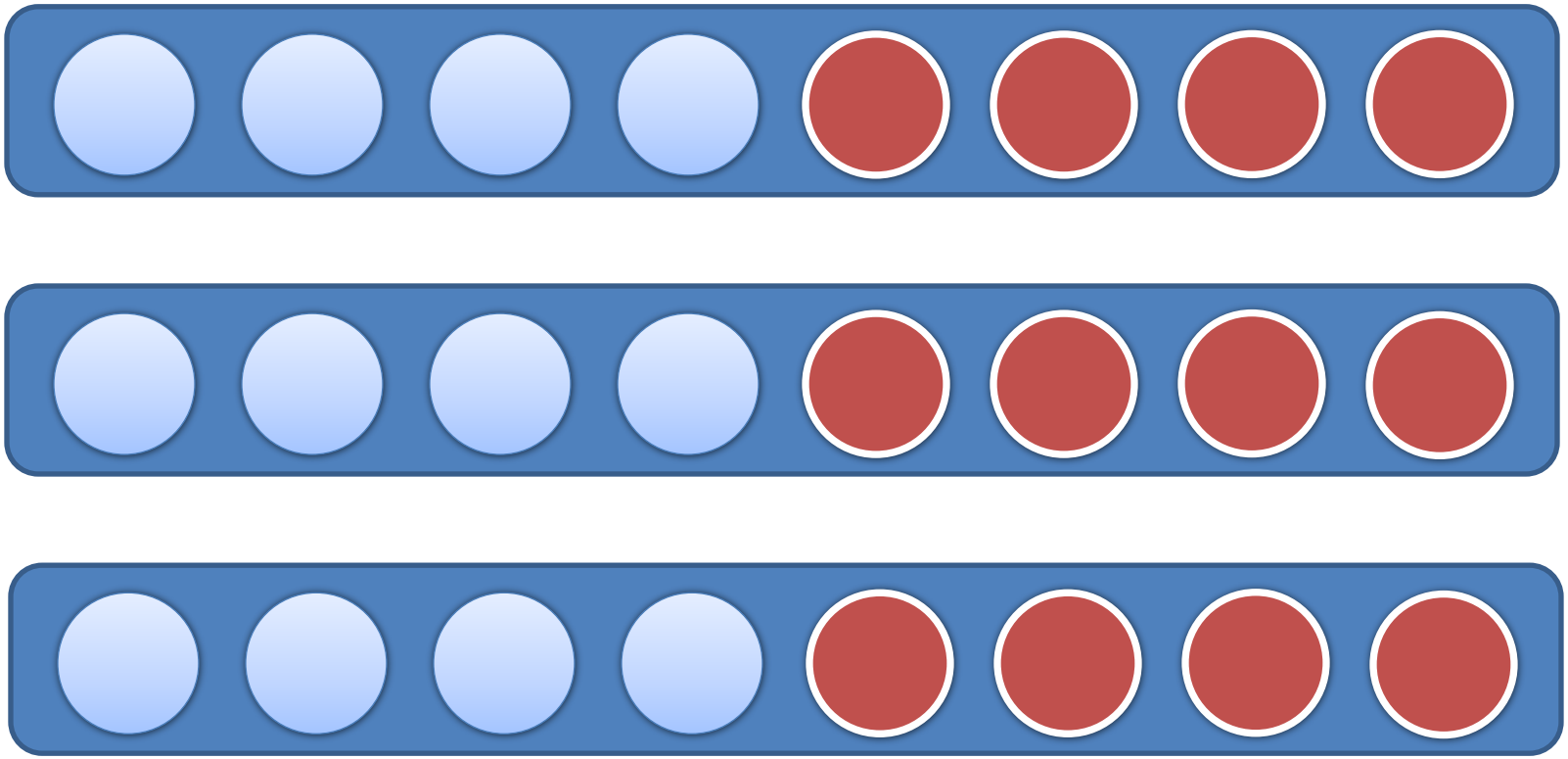
Explain



```
$("li:nth-child(4n)")
```

Superfluous Spawn Bug

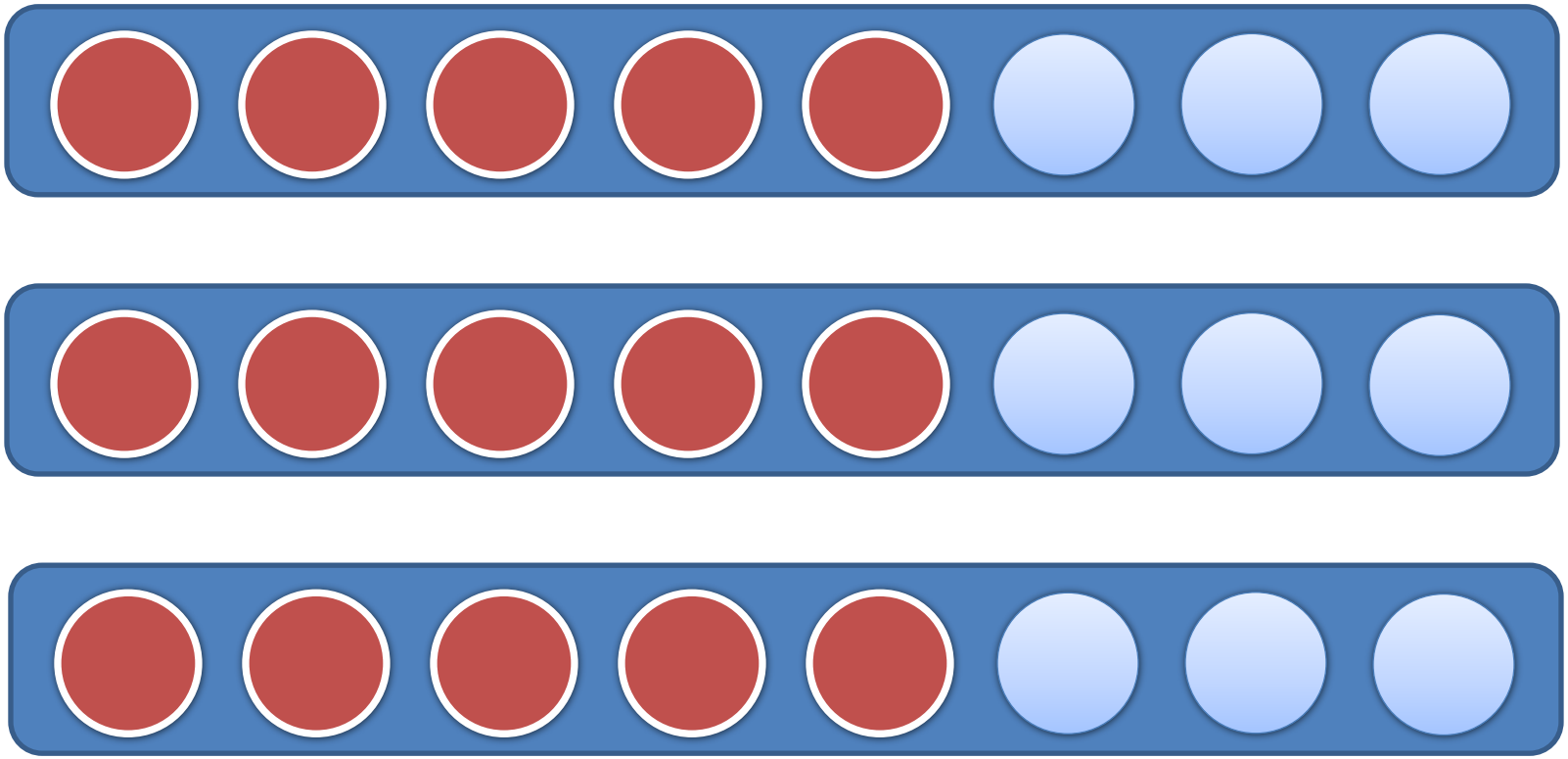
Explain



```
$("li:nth-child(n+5)")
```

Superfluous Spawn Bug

Explain



```
$("li:nth-child(-n+5)")
```

Superfluous Spawn Bug

Exterminate

Let jQuery know you only want list items that have an ancestor of "#items-to-push"

```
$("#push").on("click", function () {
    $("#items-to-push li:nth-child(1)")
        .prependTo("#stack")
});

$("#pop").on("click", function () {
    $("#stack li:nth-child(1)")
        .appendTo("#items-popped");
});
```

Superfluous Spawn Bug

Exterminate

:eq filters the matches and gives you an item by index

```
$("#push").on("click", function () {
    $("li:eq(0)")
        .prependTo("#stack")
});

$("#pop").on("click", function () {
    $("#stack li:eq(0)")
        .appendTo("#items-popped");
});
```

Superfluous Spawn Bug

Exterminate

:first is just a shorthand way of saying :eq(0)

```
$("#push").on("click", function () {
    $("li:first")
        .prependTo("#stack")
});

$("#pop").on("click", function () {
    $("#stack li:first")
        .appendTo("#items-popped");
});
```

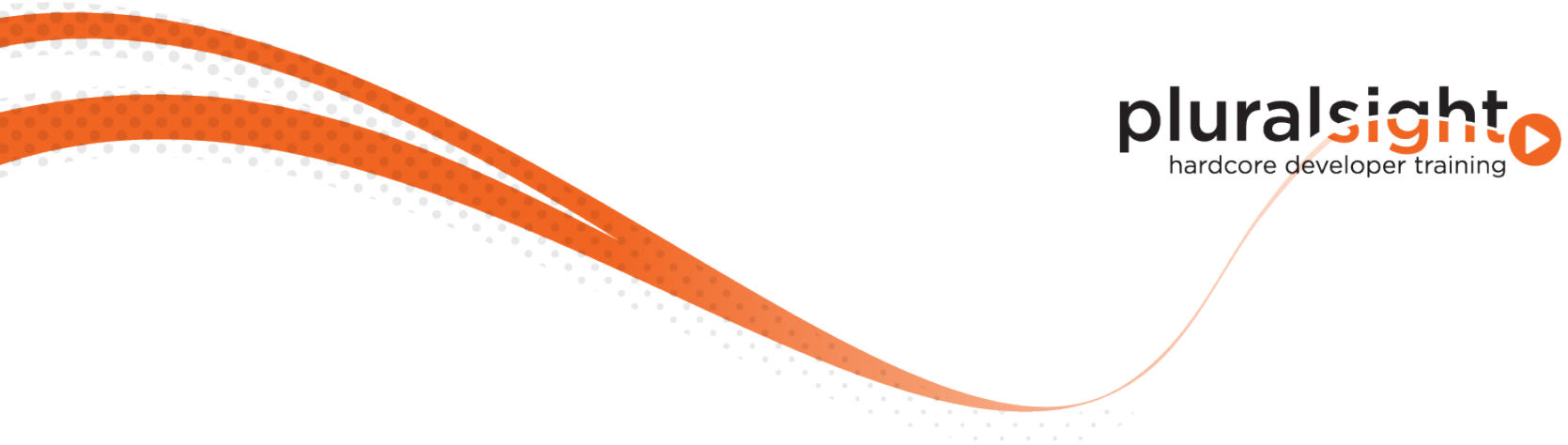
Superfluous Spawn Bug

Exterminate

Providing a context to start searching from and focus on finding the 1st list item

```
$("#push").on("click", function () {
    $("#items-to-push")
        .find("li:first")
        .prependTo("#stack")
});

$("#pop").on("click", function () {
    $("#stack")
        .find("li:first")
        .appendTo("#items-popped");
});
```

Always Alive Bug



pluralsight
hardcore developer training

Always Alive Bug

Examine

```
function notify(message) {
    var $notify = $("#notify");
    if (!$notify) {
        $("<div />", {
            id: "notify",
            html: message
        }).appendTo("body");
    } else {
        $notify.html(message);
    }
}

notify("My 1st Message");
notify("My 2nd Message");
```

Always Alive Bug

Expose

```
function notify(message) {
    var $notify = $("#notify");
    if (!$notify) {
        $("<div />", {
            id: "notify",
            html: message
        }).appendTo("body");
    } else {
        $notify.html(message);
    }
}

notify("My 1st Message");
notify("My 2nd Message");
```

The if statement always resolves to something truthy

Always Alive Bug

Explain

jQuery(selector [, context]) *Returns: jQuery*

Description: *Accepts a string containing a CSS selector which is then used to match a set of elements.*

```
var $notify = $("#notify");
```

Always returns the jQuery object. If selection is found then it's stored in jQuery's internal collection

```
if (jQuery) {
    console.log("True!");
}
```

Objects are always "truthy" based on truthy/falsey rules in JavaScript

FALSEY: `false, 0, -0, null, undefined, NaN, ""`

TRUTHY: `true, {}, [], 5, etc...`

Always Alive Bug

Exterminate

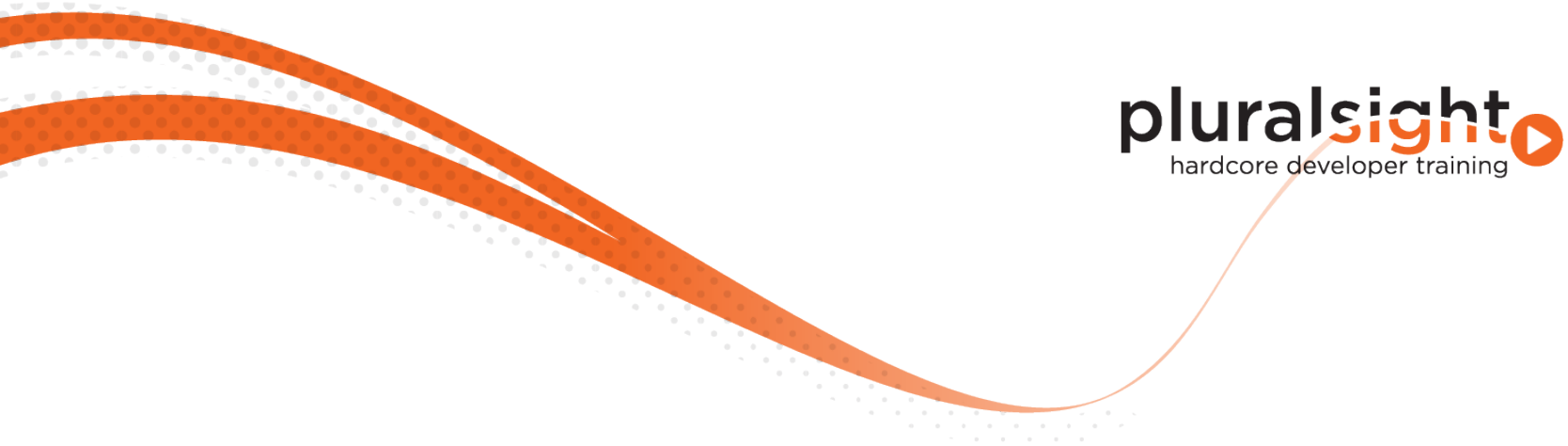
```
function notify(message) {
    var $notify = $("#notify");
    if (!$notify.length) {
        $("<div />", {
            id: "notify",
            html: message
        }).appendTo("body");
    } else {
        $notify.html(message);
    }
}

notify("My 1st Message");
notify("My 2nd Message");
```

The length property exposes how many items are in jQuery's internal collection

FALSEY: 0

TRUTHY: 1 - Number.MAX_VALUE

Brittle Blood Bug



Brittle Blood Bug

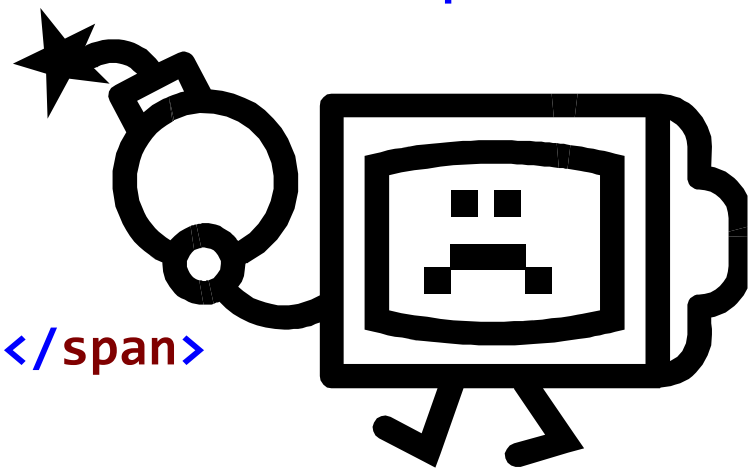
Examine

```
<ul id="mailbox" data-role="listview" data-count-
theme="c" data-inset="true">
    <li><a href="/inbox">Inbox <span class="ui-li-
count">12</span></a></li>
    <!-- ... more markup ... -->
</ul>

$(document).on("click", "#mailbox a", function (e) {
    var $this = $(this),
        $listItem = $this.parent()

    e.preventDefault();
    $listItem.addClass("highlight");
});
```

Brittle Blood Bug

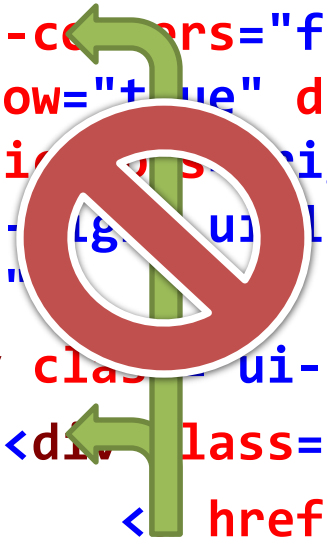
Expose

```
<li data-corners="false" data-shadow="false" data-
iconshadow="true" data-wrapperels="div" data-icon="arrow-
r" data-iconpos="right" data-theme="c" class="ui-btn ui-
btn-icon-right ui-li-has-arrow ui-li ui-li-has-count ui-
btn-up-c">
    <div class="ui-btn-inner ui-li">
        <div class="ui-btn-text">
            <a href="/drafts" class="ui-link-
inherit">Drafts <span class="ui-li-count ui-btn-up-c ui-
btn-corner-all">4</span></a>
        </div>
        <span class="ui-icon
            ui-icon-arrow-r
            ui-icon-shadow"> </span>
    </div>
</li>
```



Brittle Blood Bug

Explain



Exterminate

.closest(selector)

Returns: jQuery

Description: For each e … testing the element itself and traver…

.closest(selector)

version added: 1.3

selector

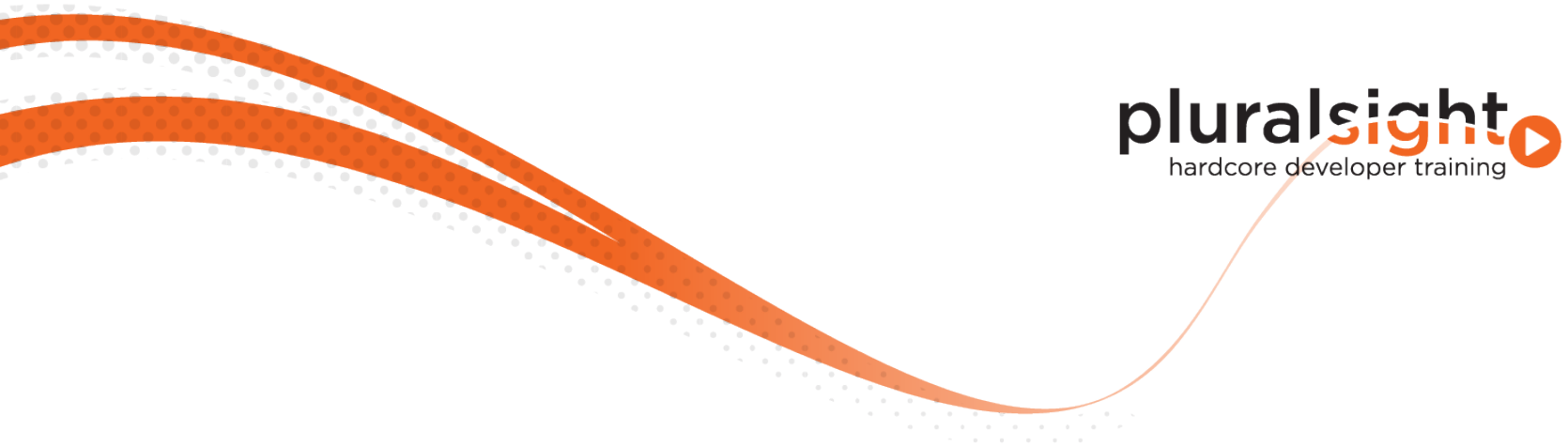
Type: Selector

A string containing a s…

```
$(document).on("click", "#mailbox a", function (e) {
    var $this = $(this),
        $listItem = $this.closest("li");

    e.preventDefault();
    $listItem.addClass("highlight");
});
```

Traverse ancestors (starting with itself) until it finds an element matching passed selector

Forgetful Family Bug



pluralsight
hardcore developer training

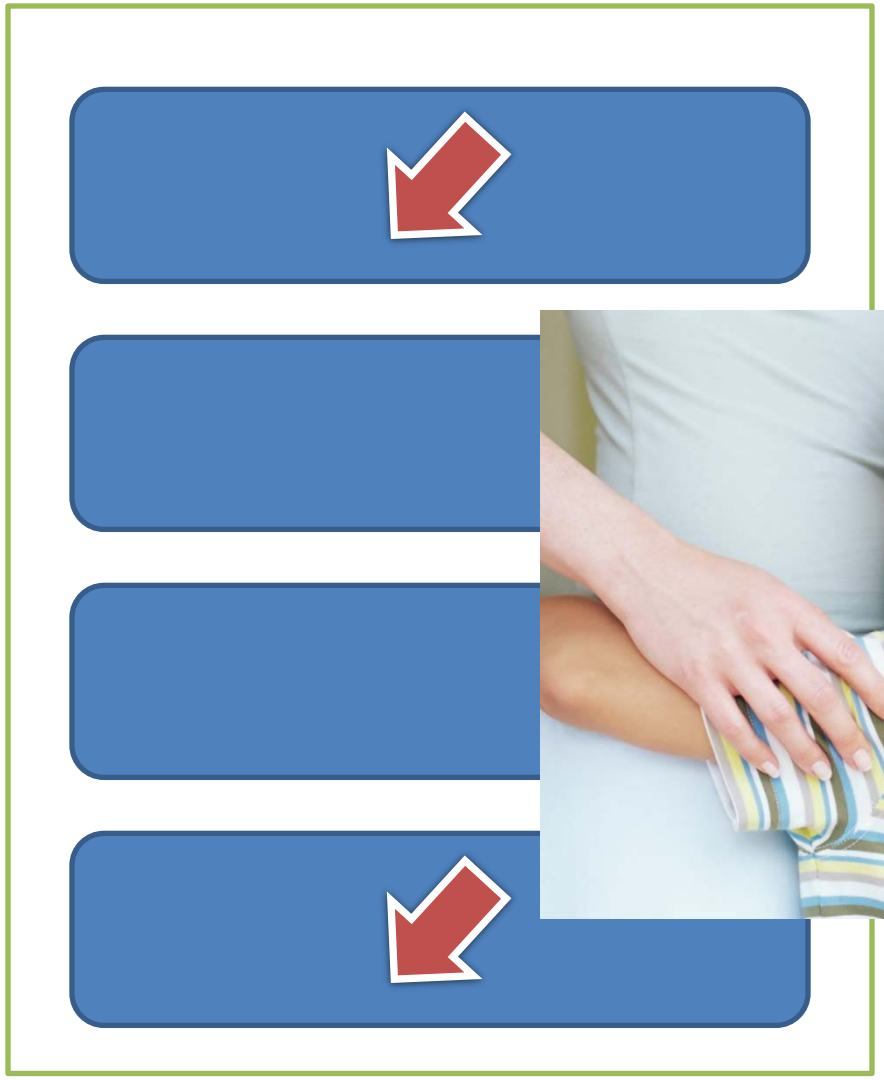
Forgetful Family Bug

Examine

```
<ul id="items">
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
    <li>Item 4</li>
    <!-- ... more markup ... -->
</ul>

$(document).on("mouseenter", "#items li", function (e) {
    var $this = $(this);

    $this.addClass("highlight");
    $this.nextAll().removeClass("highlight");
});
```

Expose

Forgetful Family Bug



Forgetful Family Bug

Explain

.nextAll([selector]) — **Returns:** jQuery

Description: *Get all following siblings of each element in the set of matched elements, optionally filtered by a selector.*

.prevAll([selector]) — **Returns:** jQuery

Description: *Get all preceding siblings of each element in the set of matched elements, optionally filtered by a selector.*

.siblings([selector]) — **Returns:** jQuery

Description: *Get the siblings of each element in the set of matched elements, optionally filtered by a selector.*

Forgetful Family Bug

Exterminate

```
$(document).on("mouseenter", "#items li", function (e) {
    $(this).addClass("highlight")
        .prevAll().removeClass("highlight").end()
        .nextAll().removeClass("highlight");
});

// OR

$(document).on("mouseenter", "#items li", function (e) {
    $(this).addClass("highlight")
        .siblings().removeClass("highlight");
});
```

Make sure to remove the highlight class from previous & next elements

The siblings() methods gets both previous and next elements

Forgetful Family Bug

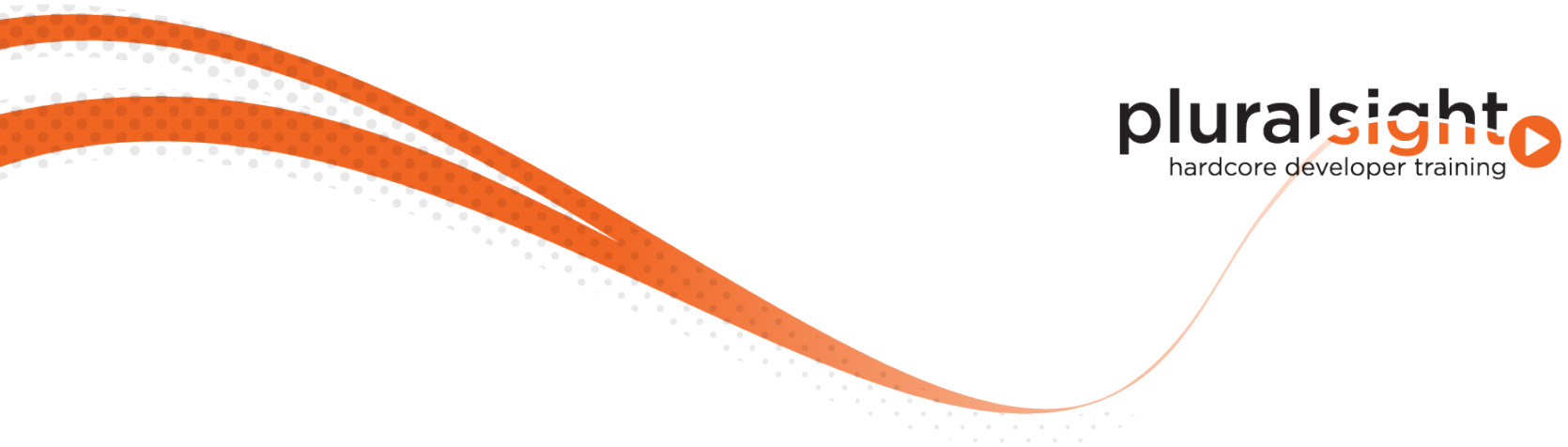
Exterminate

```
$(document).on("mouseenter mouseleave", "#items li",
    function (e) {
        $(this).toggleClass("highlight");
    });

// OR

$(document).on("mouseenter mouseleave", "#items li",
    function (e) {
        $(this).toggleClass("highlight",
            e.type === "mouseenter");
    });
```

Instead of messing with siblings, listen to mouseleave & remove class

Pass boolean to toggleClass() to guarantee if adding or removing

Discarded Descendant Bug



pluralsight
hardcore developer training

Discarded Descendant Bug

Examine

```
<ul id="source" style="float: left;">
    <li><input type="checkbox" /> <div
class="content">Item 1</div></li>
    <li><input type="checkbox" /> <div
class="content">Item 2</div></li>
    <li><input type="checkbox" /> <div
class="content">Item 3</div></li>
</ul>

<ul id="destination" style="float: right;"></ul>

<button>Add</button>
```

Discarded Descendant Bug

Examine

```
$(document).on("click", "#source input", function () {
    var $this = $(this),
        $item = $this.closest("li");

    $item.toggleClass("highlight", $this.is(":checked"));
});
```

Discarded Descendant Bug

Examine

```
$(document).on("click", "button", function (e) {
    var $srcItems = $("#src li"), $dest = $("#dest");

    $srcItems.add("#destination li");
    $srcItems.removeClass("highlight");

    $srcItems.find("input:checked").each(function () {
        var $this = $(this), $item = $this.closest("li");
        $("<li class='highlight'>" +
            $item.find(".content").html() + "</li>")
           .appendTo($dest);
        $this.prop("checked", false);
    });
});
```

Discarded Descendant Bug

Expose

```
$(document).on("click", "button", function (e) {
    var $srcItems = $("#src li"), $dest = $("#dest");

    $srcItems.add("#destination li");
    $srcItems.removeClass("highlight");

    $srcItems.find("input:checked").each(function () {
        var $this = $(this), $item = $this.closest("li");
        $("<li class='highlight'>" +
            $item.find(".content").html() + "</li>")
           .appendTo($dest);
        $this.prop("checked", false);
    });
});
```

The highlight class isn't being removed from the destination list items

Discarded Descendant Bug

Explain

```
var pdiv = $("p");
pdiv.add("div");
pdiv.fadeIn(); // fadeIn only p elements
```

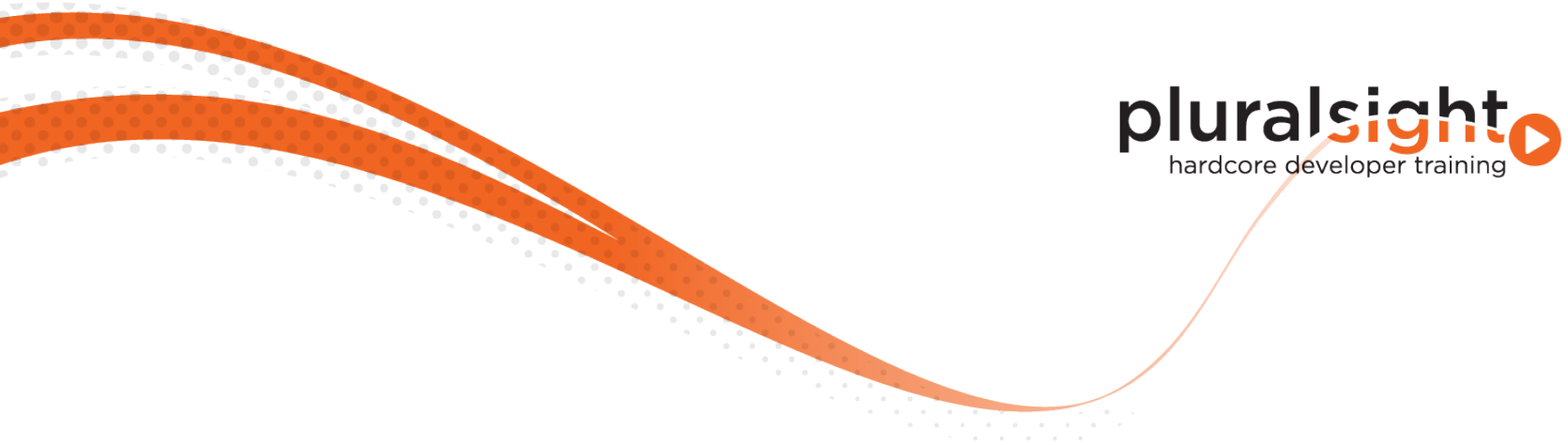
> ".add() creates a new set and leaves the original set unchanged" –jQuery Docs

```
var pdiv = $("p");
pdiv = pdiv.add("div");
pdiv.fadeIn(); // fadeIn p & div elements

// OR

$("p").add("div").fadeIn(); // fadeIn p & div elements
```

Discarded Descendant Bug


```
$(document).on("click", "button", function (e) {
    var $srcItems = $("#src li"), $dest = $("#dest");

    $srcItems = $srcItems.add("#destination li");
    $srcItems.removeClass("highlight");

    $srcItems.find("input:checked").each(function () {
        var $this = $(this), $item = $this.closest("li");
        $("<li class='highlight'>" +
            $item.find(".content").html() + "</li>")
           .appendTo($dest);
        $this.prop("checked", false);
    });
});
```

Checking Condition Bug



pluralsight
hardcore developer training

Checking Condition Bug

Examine

```
<div id="toggle"><input type="checkbox" /> Toggle</div>
<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>
    <li><input type="checkbox" /> <div
class="content">Testing 3</div></li>
</ul>

$(document).on("click", "#toggle input",
    function (e) {
        $("#items").find("input").attr("checked",
            $(this).attr("checked"));
});
```

Checking Condition Bug

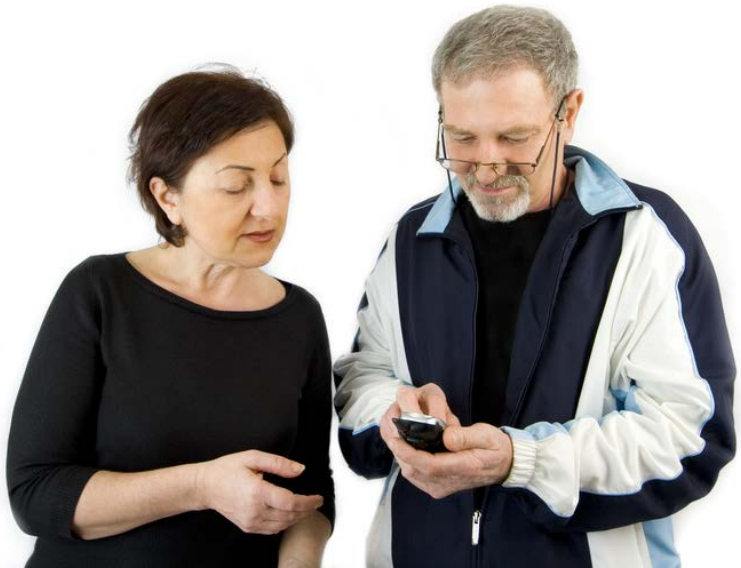

```
$(document).on(
    "click",
    "#toggle input",
    function (e) {
        $("#items").find("input").attr(
            "checked",
            $(this).attr("checked")
        );
    }
);
```

.attr("checked") always returns undefined

Checking Condition Bug

Explain

```
$(elem).attr("checked") // 1.6.0-      true      Boolean
// Changes with checkbox state

$(elem).attr("checked") // 1.6.0       "checked" String
// Initial state of the checkbox;
// does not change

$(elem).attr("checked") // 1.6.1-1.6.2 "checked" String
// Changes with checkbox state

$(elem).attr("checked") // 1.6.3+
// Initial state of the checkbox;
// does not change
```



Checking Condition Bug

Explain

```
elem.checked                // true Boolean
// Changes with checkbox state

$(elem).prop("checked") // true Boolean
// Changes with checkbox state

$(elem).is(":checked")  // true Boolean
// Changes with checkbox state
```

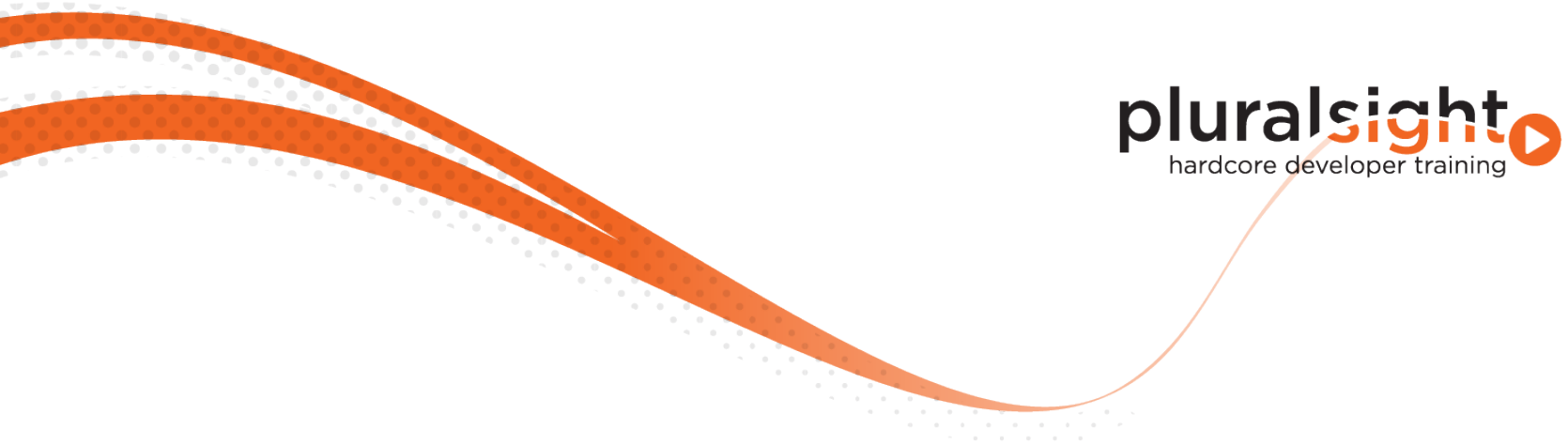
Checking Condition Bug

Exterminate

```
$(document).on(
    "click",
    "#toggle input",
    function (e) {
        $("#items").find("input").prop(
            "checked",
            $(this).prop("checked")
        );
    }
);
```

.prop("checked") changes with state of checkbox

Barbaric Behavior Bug



pluralsight
hardcore developer training

Barbaric Behavior Bug

Examine

```
var contacts = [
    { name: "John Doe", birthday: "1/1/1970" },
    { name: "Jane Doe", birthday: "2/2/1975" },
    { name: "Baby Doe", birthday: "3/3/2010" }
];

addContacts(contacts);

$("button").on("click", function () {
    addContacts([
        { name: "Yoda Doe", birthday: "4/4/2013" }
    ]);
});
```

Barbaric Behavior Bug

Examine

```
var $container = $("#container");
var $contacts = $("#contacts");
var addContacts = function (contacts) {
    var list = $contacts.remove();
    $.each(contacts, function (index, contact) {
        $("<li />", { html: contact.name })
            .data("birthday", contact.birthday)
            .appendTo($contacts);
    });
    $container.append(list);
};
$container.on("click", "#contacts li", function () {
    alert("Birthday: " + $(this).data("birthday"));
});
```

Expose

```
var $co
var $co
var add
    var
    $.e
    });
    $co
};
$contai                                    {
    ale                                   ;
});
```

Barbaric Behavior Bug

Explain

```
var list = $contacts.remove();
```

.remove() removes elements from the DOM, all bound events, & jQuery data associated with the elements

```
var list = $contacts.detach();
```

.detach() removes elements from the DOM, but keeps all bound events & jQuery data associated to be reinserted

Barbaric Behavior Bug

Exterminate

```
var $container = $("#container");
var $contacts = $("#contacts");
var addContacts = function (contacts) {
    var list = $contacts.detach();
    $.each(contacts, function (index, contact) {
        $("<li />", { html: contact.name })
            .data("birthday", contact.birthday)
            .appendTo($contacts);
    });
    $container.append(list);
};
$container.on("click", "#contacts li", function () {
    alert("Birthday: " + $(this).data("birthday"));
});
```

Conclusion

- Make sure to escape your selectors if they have special characters
- Understand what the nth-child selector does before you use it
- Learn the truthy/falsey JavaScript rules
- Use .closest() instead of .parent() to traverse up the DOM
- Make use of .siblings() method or listen to multiple events
- Save off jQuery if you are making a new set and not chaining
- Use the .prop() method when getting/setting a property
- Use .detach() to temporarily remove DOM instead of .remove()