Event

Event Tips and Tricks

pluralsight
hardcore developer training

Event

.on() Instead of .bind()

.on() Instead of .live() or .delegate()

Handling Event Delegation

Namespacing Your Events

Registering Events Before DOM Ready

Reusing an Existing Method as an Event Handler

Creating Custom Events

Determining User Input

Determine If Event was Triggered by the User

Just-in-Time Initialization

Event

Using the on() Method Instead of bind() or Shorthand Methods

pluralsight
hardcore developer training

What are the Shorthand methods?

- **The shorthand methods are just convenience wrappers around the 2 parameter on() method**
  - blur, change, click, dblclick, focus, focusin, focusout, hover, keydown, keypress, keyup, mousedown, mouseenter, mouseleave, mousemove, mouseover, mouseup, resize, scroll, select, submit

These 3 methods are all equivalent

```
$(".display").click(function() { });
$(".display").bind("click", function() { });
$(".display").on("click", function() { });
```

What Does bind() Really Do?

- Attaches event handlers to all DOM elements jQuery selected

```
$("#members li a").bind("click", function() { });

<ul id="members" data-role="listview" data-filter="true">
    <li>
        <a href="detail.html?id=5">Detail 5</a>
    </li>
    <li>
        <a href="detail.html?id=6">Detail 6</a>
    </li>
</ul>
```

Convert bind() to on()

- Use the 2 parameter version of on() to convert your shorthand and bind() method code

```
$(".display").bind("click", function() { });
$(".display").click(function() { });
```

Use 2 parameter .on()

```
$(".display").on("click", function() { });
```

Event

DEMO: .on() Instead of .bind()

pluralsight
hardcore developer training

Event

Using the on() Method Instead of live() or delegate()

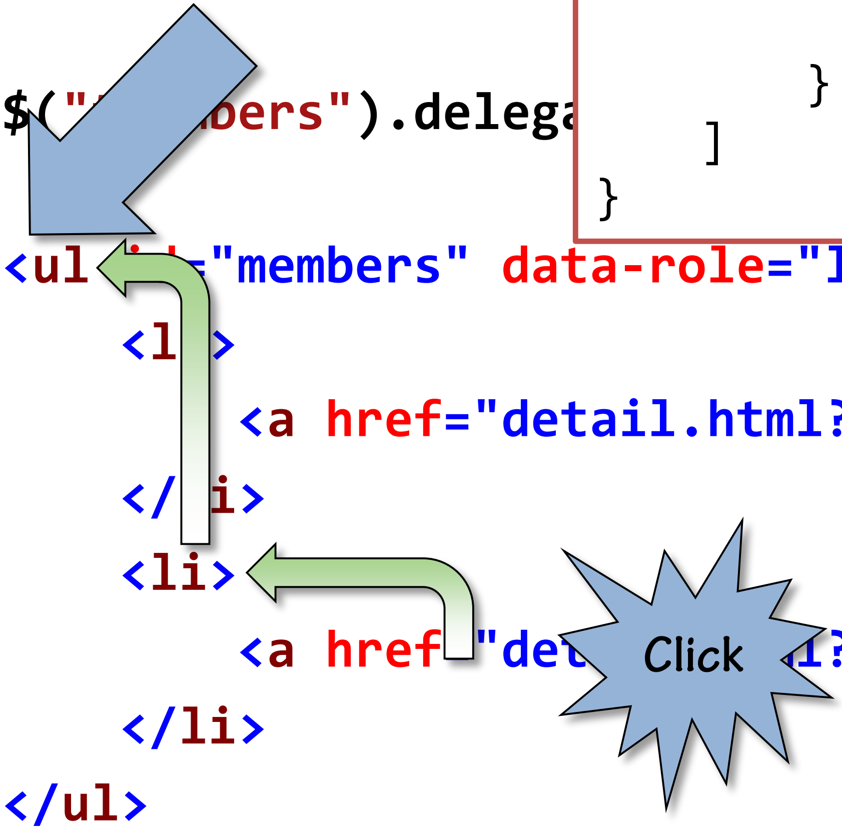
pluralsight
hardcore developer training

Event

What Do

- Attaches an even

```
{
    click: [
        {
            selector: "li a",
            handler: function() {},
            type: "click",
            ...
        }
    ]
}
```



```
$("  bers").delega                                         ;

<ul id="members" data-role="listview" data-filter="true">
    <li>
        <a href="detail.html?id=5">Detail 5</a>
    </li>
    <li>
        <a href="det          l?id=6">Detail 6</a>
    </li>
</ul>
```

Convert live() & delegate() to on()

- Use the 3 parameter version of on() to convert your live() and delegate() code

```
$("#members").delegate("li a", "click", function() { });
$("#members").on("click", "li a", function() { });
```

Use 3 parameter .on()

.live() has been deprecated & removed

```
$("#members li a").live("click", function() { });
$(document).on("click", "#members li a", function() { });
```

Event

DEMO: .on() Instead of .live() or .delegate()

pluralsight
hardcore developer training

Event

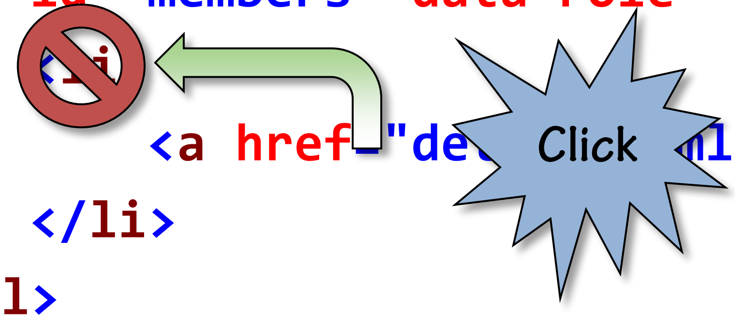
Handling Event Delegation

Stop the Propagation of Events

- e.stopPropagation() stops an event from bubbling up the DOM

```
$("li").on("click", function(e) { e.stopPropagation() });

<ul id="members" data-role="listview" data-filter="true">
    <li>
        <a href="de[illegible]ml?id=5">Detail 5</a>
    </li>
</ul>
```



Don’t Tell Your

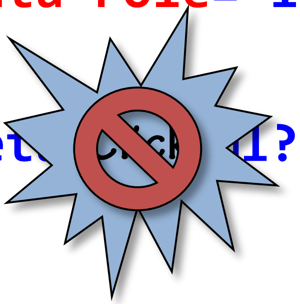
Event

Stop the Immediate Propagation of Events

- e.stopImmediatePropagation() stops any other events from firing on the element and stops bubbling

```
$("a").on("click", function(e) {
    e.stopImmediatePropagation();
}).on("click", function(e) { console.log("Hello") });

<ul id="members" data-role="listview" data-filter="true">
    <li>
        <a href="detail.html?id=5">Detail 5</a>
    </li>
</ul>
```



Don’t Tell Your

Stop Returning False

- A common mistake some developers make is to `return false` thinking it only prevents default behavior

```
$("a").on("click", function(e) { return false; });

<ul id="members" data-role="listview" data-filter="true">
    <li>
        <a href="de[illegible]l?id=5">Detail 5</a>
    </li>
</ul>
```

Click

return false is e.preventDefault() and e.stopPropagation()

Event

DEMO: Handling Event Delegation

Namespacing Your Events

Namespacing Events

- You can namespace any event type by appending a period and a unique term

.myplugin namespace

```
$("input").on("focus.myplugin", function() { });
$("input").on("focus", function() { });
$("input").on("keypress.myplugin", function() { });

$("#members").off("focus.myplugin");
$("#members").off(".myplugin");
```

Remove all .myplugin event handlers

DEMO: Namespacing Your Events

Event

Registering Events Before DOM Ready

pluralsight
hardcore developer training

Event

Running Code Before DOM Ready

- Even before the DOM is ready you can delegate events to the `document`
- This technique utilizes wasted time between when jQuery is loaded and the DOM is ready

```
$(document).ready(function () {
    //What are you waiting for?
});
```

DOM Ready not needed

```
$(document).on("click", "#members li a", function() { });
```

DEMO: Registering Events Before DOM Ready

Event

Reusing Existing Method as Event Handler

pluralsight
hardcore developer training

Reusing a Method as an Event Handler

- The following code outputs “undefined has been clicked!”

```
var Person = function( name ) {
    this.name = name;
};
Person.prototype.clicked = function(e) {
    console.log( this.name + " has been clicked!" );
};
```

undefined has been clicked!

\`this\` is DOM element, not person object

```
var person = new Person( "John Doe" );
$( ".person" ).on( "click", person.clicked );
```

Reusing a Method as an Event Handler

- You can easily fix the issue using the jQuery.proxy method

```
var Person = function(name) {
    this.name = name;
};
Person.prototype.clicked = function(e) {
    console.log(this.name + " has been clicked!");
};


var person = new Person("John Doe");
$(".person").on("click",
    $.proxy(person.clicked, person));
```

Use $.proxy() to set value of `this`

DEMO: Reusing Existing Method as Event Handler

Creating Custom Events

Creating Custom Events

```
<input type="text" value="Hello" />
<input type="text" value="World" />
<button>Clear</button>

$(document).ready(function () {
    $("input").on("clear", function () {
        $(this).val("");
    });

    $("button").on("click", function (e) {
        $("input").trigger("clear");
    });
});
```

New “clear” event type

Trigger custom event type

Event

Creating Custom Events

Use document as a mediator

```
$(document).on("updated", function (e, fruits) {
    console.log(fruits.join(", "));
});

var items = ["apples", "oranges", "grapes"];

$(document).trigger("updated", [items]);
```

Use postal.js for a cleaner & feature rich message

DEMO: Creating Custom Events

Event

Determining User Input

pluralsight
hardcore developer training

Event

Determining User Input

```
$("input").on("keypress", function (e) {
    console.log(e.which);
    console.log(e.altKey, e.ctrlKey, e.shiftKey);
});
```

e.which normalizes cross-browser issues

```
$("img").on("click", function (e) {
    console.log(e.pageX, e.pageY);
    console.log(e.offsetX, e.offsetY);
    console.log(e.screenX, e.screenY);
});
```

Access mouse position

Event

DEMO: Determining User Input

Event

Determine If Event Triggered by the User

Who Invoked an Event?

- Sometimes you need to know who invoked an event

```
$( ".person" ).on( "click", function(e) {
    console.log("I was clicked " +
        (e.originalEvent ?
            "by the user" :
            "programmatically"));
});

setTimeout( function() {
    $( ".person" ).trigger( "click" );
}, 5000 );
```

undefined if manually triggered

trigger manually

Event

DEMO: Determine If Event Triggered by the User

pluralsight
hardcore developer training

Event

Just-in-Time Initialization

pluralsight
hardcore developer training

Event

Just-in-Time Initialization

```
<!-- More HTML... -->
<div class="control-group">
    <label for="birthday">Birthday</label>
    <div class="controls">
        <input id="birthday" type="text" class="date">
    </div>
</div>
<div class="control-group">
    <label for="anniversary">Anniversary</label>
    <div class="controls">
        <input id="anniversary" type="text" class="date">
    </div>
</div>
<!-- More HTML... -->
```

Event

Just-in-Time Initialization

Requires DOM ready

initializing all date fields

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

    $("input.date").datepicker({
        minDate: moment().subtract("months", 1).toDate(),
        maxDate: moment().add("months", 1).toDate(),
        dateFormat: "d M, y",
        constrainInput: true,
        beforeShowDay: $.datepicker.noWeekends
    });

});
```

Event

Just-in-Time Initialization

DOM ready not needed

Only uninitialized elements match

```
$(document).on("focus",
    "input.date:not(.hasDatepicker)", function () {
    toastr.info("Initializing " + this.id);

    $(this).datepicker({
        minDate: moment().subtract("months", 1).toDate(),
        maxDate: moment().add("months", 1).toDate(),
        dateFormat: "d M, y",
        constrainInput: true,
        beforeShowDay: $.datepicker.noWeekends
    });
});
```

http://code.dougneiner.com/speaking/

Event

DEMO: Just-in-Time Initialization

Summary

- **Several jQuery techniques can be applied when working with Events:**
  - Switching to the new on() methods
  - Handling Event Delegation
  - Namespacing Your Events
  - Registering Events before DOM Ready
  - Reusing an Existing Method as an Event Handler
  - Determine If an Event was Triggered by the User
  - Creating a Custom Event Type
  - Just-in-Time Initialization

Event

Summary