

# jQuery Behaviours

bind, live, livequery

Richard Paul, Sep 2009

# Unobtrusive Javascript

Clean separation of

- Data (HTML)
- Style (CSS)
- Behaviour (Javascript)

```
...
<form id="myForm" onsubmit="doSomething()">
  ...
</form>
...
<script type="text/javascript">
  $('#myForm').submit(function() { /* do something */ });
</script>
```

# Javascript Events

In Javascript, events bubble up.

```
<body>
  <div>
    <button>A button</button>
  </div>
</body>
```

Clicking the button bubbles the event up (button > div > body)

However IE doesn't play nice with form events (*select, change, submit, reset*), they don't bubble <sup>(1)</sup>.

[http://www.quirksmode.org/js/events\\_order.html](http://www.quirksmode.org/js/events_order.html)

(1) [http://blogs.sun.com/greimer/entry/event\\_delegation\\_for\\_submit\\_change](http://blogs.sun.com/greimer/entry/event_delegation_for_submit_change)

# Event Delegation

Event delegation is the practice of letting events bubble up to a handling node.

```
$('#table').click(function(event) {  
    alert(event.target.innerHTML);  
});
```

r1c1	r1c2	r1c3
r2c1	r2c2	r2c3

As the target always points to the element the event happened on, clicking on any cell will result in its contents being alerted.

# Adding new rows

When adding new cells, no extra work is required.  
Clicking a new row will bubble up to the table's click handler.

## [Demo](#)

In practice you would want to check the element is of the correct type/class etc.

# Binding Mechanisms in jQuery

There are a number of ways of binding behaviour in jQuery

bind	<code>\$('#form').bind('submit', function(){})</code>
live	<code>\$('#form').live('submit', function(){})</code>
liveQuery	<code>\$('#form').liveQuery('submit', function(){})</code>

Commonly when using `bind` you would use the shortcut method:  
`$('#form').submit(function(){})`

# Comparison Grid

	<b>Dynamic</b>	<b>Event Delegation</b>	<b>IE6 Form Events</b>	<b>jQuery Core</b>
<b>bind</b>	No	No	Yes	Yes
<b>live</b>	Yes	Yes	No	Yes
<b>liveQuery</b>	Yes	No	Yes	No (plugin)

# Facebook Use Case



[Facebook](#) uses jQuery's clone method to copy the required content into its modal dialog.

Any behaviours initialised with `bind` aren't applied to the facebook content.



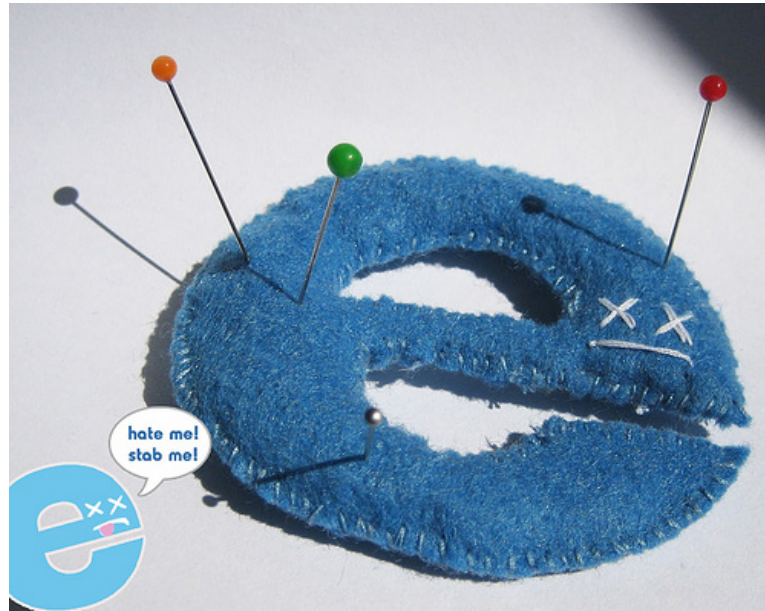
# Facebook with jQuery.live

Solution, use jQuery.live to apply behaviours using the event delegation style.

```
$( 'form' ).live( 'submit', function() {  
    // submit form via ajax  
    return false;  
});
```

**Simple!**

# What about Internet Explorer?



Recall that IE doesn't bubble form events... like submit.

Using `jQuery.live` doesn't work for IE as the event doesn't bubble so the handler is never invoked.

<http://www.chigarden.com/2007/10/tutorial-making-the-ie-voodoo-doll/>

# liveQuery

We can't use `jQuery.bind` as the form is dynamically created (by facebook's clone call).

Enter [liveQuery](#)

This plugin adds the handler directly to the form element.

No event delegation, and it **works in IE6!**

[Demo](#)

liveQuery uses some [fancy tricks](#) (async binding etc) to keep your website performing.

Comments, queries, suggestions or theories?