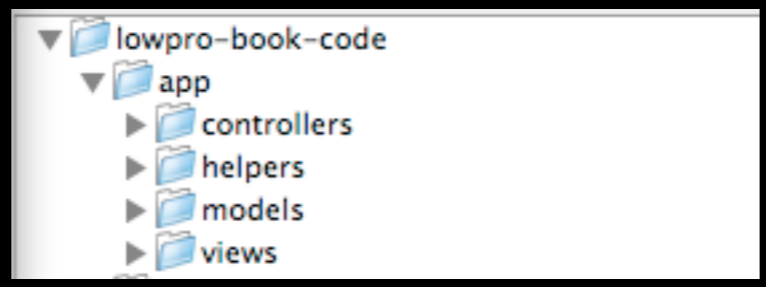


# ACCESSIBLE AJAX ON RAILS

**Jarkko Laine**  
with Geoffrey Grosenbach



```
r.resources :categories do |cat|
  cat.resources :products
  cat.resources :companies
  cat.resources :subcategories do |sub|
    sub.resources :products
    sub.resources :companies
  end
end
```

```
//<![CDATA[
new Form.Element.EventObserver('undone_box_1',
    function(element, value) {
        new Ajax.Request('/items/1',
            {asynchronous:true,
              evalScripts:true,
              method:'put',
              parameters:value + '&authenticity_token=' +
                encodeURIComponent('8d829cfcccd4d2b494891ef47cc95893faa361e')}}))
//]]>
</script>
</li>
<li id="undone_2">
<input id="undone_box_2" name="item[2][done]" type="checkbox" value="1" />
<input name="item[2][done]" type="hidden" value="0" />
<label for="undone_box_2">
    Return bottles to recycling
</label>
<script type="text/javascript">
//<![CDATA[
new Form.Element.EventObserver('undone_box_2',
    function(element, value) {
        new Ajax.Request('/items/2',
            {asynchronous:true,
              evalScripts:true, method:'put',
              parameters:value + '&authenticity_token=' +
                encodeURIComponent('8d829cfcccd4d2b494891ef47cc95893faa361e')}}))
//]]>
</script>
</li>
<li id="undone_3">
<input id="undone_box_3" name="item[3][done]" type="checkbox" value="1" />
<input name="item[3][done]" type="hidden" value="0" />
<label for="undone_box_3">
    Return bottles to recycling
</label>
<script type="text/javascript">
//<![CDATA[
new Form.Element.EventObserver('undone_box_3',
    function(element, value) {
        new Ajax.Request('/items/3',
            {asynchronous:true,
              evalScripts:true, method:'put',
              parameters:value + '&authenticity_token=' +
                encodeURIComponent('8d829cfcccd4d2b494891ef47cc95893faa361e')}}))
//]]>
```

# ACCESSIBILITY

*“the degree to which a product (e.g., device, service, environment) is accessible by as many people as possible.”*  
*-Wikipedia*



# ACCESSIBILITY

*The ultimate goal*



# PROGRESSIVE ENHANCEMENT

*A methodology for producing  
accessible web content*

# PROGRESSIVE ENHANCEMENT

*Roots: graceful degradation*

*hardly ever happened in real  
world*

# PROGRESSIVE ENHANCEMENT TURNS GRACEFUL DEGRADATION ON ITS HEAD

*build the essential, universal  
first*

*then gradually enhance the  
experience for those capable of  
digesting it*

# UNOBTRUSIVE JAVASCRIPT

*part of the progressive  
enhancement process*

# UNOBTRUSIVE JAVASCRIPT

*Separation of concerns*

# STRUCTURE

```
<div id="wrapper">  
  <ul id="my_stuff">  
    <li id="hide_me">  
      I am Iron Man  
    </li>  
  </ul>  
</div>
```

# PRESENTATION

```
#wrapper {  
  width: 100%;  
  overflow: hidden;  
}  
#my_stuff {  
  list-style: none;  
}  
#my_stuff li {  
  padding: 1.5em;  
}
```

# STRUCTURE

```
<div id="wrapper">
  <ul id="my_stuff">
    <li id="hide_me">
      I am Iron Man
    </li>
  </ul>
</div>
```

# PRESENTATION

```
#wrapper {
  width: 100%;
  overflow: hidden;
}
#my_stuff {
  list-style: none;
}
#my_stuff li {
  padding: 1.5em;
}
```

# BEHAVIOUR

```
document.observe('dom:loaded',
function() {
  $('hide_me').hide();
  $('my_stuff').observe('click',
function() {
  // do something
  })
});
```

# UNOBTRUSIVE JAVASCRIPT BENEFITS

*clean and maintainable code*  
*—just like MVC in Rails*

# UNOBTRUSIVE JAVASCRIPT BENEFITS

*fits naturally in the progressive  
enhancement process*

# UNOBTRUSIVE JAVASCRIPT BENEFITS

*makes it easy for designers  
and programmers to work on  
the same code base*

**BUT**

*Only (part of) a  
means*

# BUT

*Does **not** guarantee  
accessibility*

```
<a href="#">Get bacon!</a>
```

```
$( 'mylink' ).observe( 'click', function( e ) {  
    window.location = 'http://google.fi';  
} );
```

```
<a href="#">Get bacon!</a>
```

```
$( 'mylink' ).observe( 'click', function( e ) {  
    window.location = 'http://google.fi';  
});
```

# FULLY UNOBTRUSIVE

**NOT ACCESSIBLE  
AT ALL**

BRIEF HISTORY OF

JAVASCRIPT ON RAILS

# BRIEF HISTORY OF JAVASCRIPT ON RAILS

*link\_to\_remote*  
*form\_remote\_tag*

# BRIEF HISTORY OF JAVASCRIPT ON RAILS

*link\_to\_remote*  
*form\_remote\_tag*

ABSOLUTELY FABULOUS

**BUT...**

**2 problems**

# 2 problems

```
<form action="/items" id="add_form" method="post"
onsubmit="new Ajax.Request('/items',
{asynchronous:true,
  evalScripts:true, parameters:Form.serialize(this)});
return false;"
style="display: none;">
```

# 2 problems

```
<a href="#"  
  onclick="$('add_form').toggle(); return false;">  
  Add new item  
</a>
```

# reaction: UJS4RAILS

*Dan Webb and Luke Redpath*

*made Rails js helpers unobtrusive*

*provided a method for attaching  
behaviours to elements*

# reaction: UJS4RAILS

*heavy*

*didn't encourage people towards  
progressive enhancement*

# take two: LOW PRO

*underlying JS framework behind  
UJS4Rails*

*better to just use Low Pro than to  
mess with the Rails internals*

# LOW PRO

*Event.addBehaviour()*

*Behaviour “classes”*

*DOM builder*

```
Event.addBehavior({
  '#add_form' : function() {
    this.hide();
  },
  '#add_new_link:click' : function(e) {
    $('add_form').toggle();
    e.stop();
  }
});
```

# BEHAVIOURS

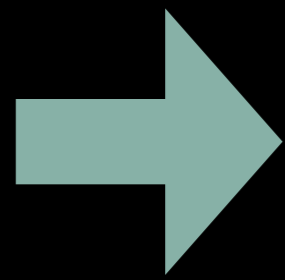
```
Event.addBehavior({  
  '#add_form' : Remote.Form  
});
```

# BEHAVIOURS

```
Event.addBehavior({  
  '#add_form' : Remote.Form({  
    onComplete: doSomething  
  })  
});
```

# DOM BUILDER

```
$div({ id: 'run-1'},  
    $p('A',  
        $em("marathon"),  
        '!')  
    );
```



```
<div id="run-1">  
  <p>A<em>marathon</em>!</p>  
</div>
```

# THE BASTARD SON

```
var item = DOM.Builder.fromHTML(  
    '<li>Remember to recover!</li>');  
list.append(item);
```

**CODE!**

EVENT

DELEGATION

# THE PROBLEMS

*event handlers aren't automatically  
assigned to dynamically added  
elements*

*performance inversely related to the  
amount of elements*

# THE PROBLEMS

```
Event.addBehavior({  
  'td:click' : function(e) {  
    // execute some code for table cells  
  }  
});
```

*What if there are thousands of cells?*

*The solution lies in event  
propagation*

TABLE

\*CLICK\*

TR

\*CLICK\*

TD

\*CLICK\*

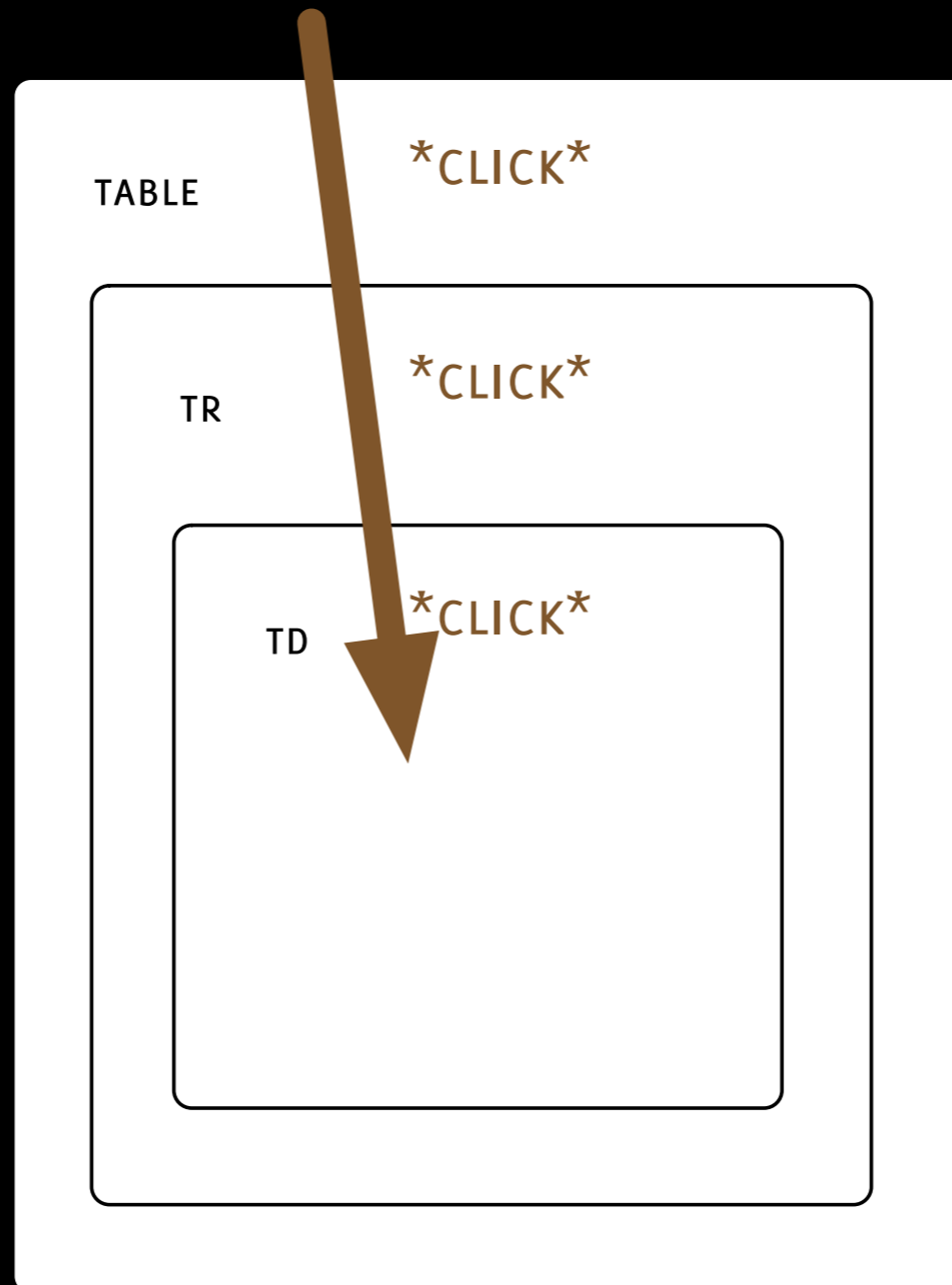


*Two event propagation  
modes*

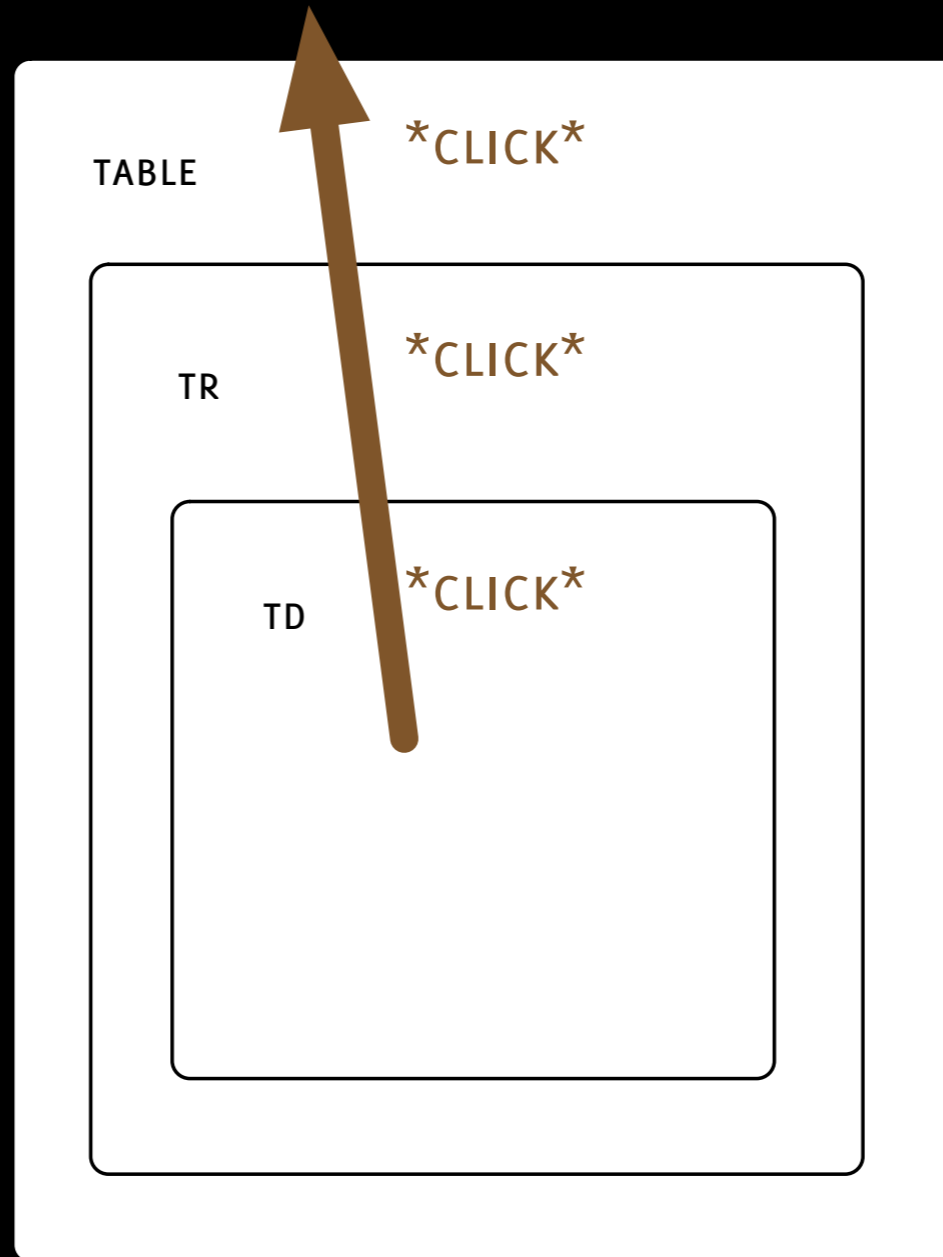
*event capture  
event bubbling*

```
Event.addBehavior({
  'table:click' : function(e) {
    // do something upon a table click
  },
  'td:click' : function(e) {
    // execute some code for table cells
  }
});
```

# *event capture*



# *event bubbling*



*current W<sub>3</sub>C DOM spec  
supports both modes*

*as do all modern browsers*

*alas, not Internet Explorer*

*and thus, not Prototype  
either*

*so we'll rely on event  
bubbling*

*(which is totally fine)*

# SO WHAT?

*given that every event knows its target element (**Event.element()**)*

*why not observe a higher-level element and only then work according to the original target?*

*enter*

# EVENT DELEGATION

*made "famous" by Christian  
Heilmann with Yahoo! UI*

*now "natively" supported by Low  
Pro, with `Event.delegate()`*

```
Event.addBehavior({
  'table:click' : Event.delegate({
    'td' : function(e) {
      var el = e.element();
      // ...
    },
    'a' : function(e) {
      e.stop();
      // handle link clicks
    }
  })
});
```

**CODE!**

# CAVEATS

*with event delegation*

*not all events bubble up (most notably **focus** and **blur**)*

*can kill performance in some cases  
(**onmousemove**)*

# LOW PRO BEHAVIOURS

# LOW PRO BEHAVIOURS

*“Behaviours are an object orientated mechanism by which you can handle events and maintain the state of an element”*

*-Dan Webb*

```
var myForm = $( 'add_form' );  
var ajaxForm = new Remote.Form(  
    myForm,  
    { method: 'post' }  
);
```

```
Event.addBehavior({  
    '#add_form': Remote.Form,  
    '#ajax_link': Remote.Link  
});
```

```
Event.addBehavior({
  '#add_form': Remote.Form({
    method : 'put'
  })
});
```

# BEHAVIOURS BUNDLED WITH LOW PRO

*Remote.Form, Remote.Link*

*Drag'n'drop*

*Auto-completer*

*In-place editor*

*...but the **real benefits** come  
from*

*...but the real benefits come  
from*

*writing your own  
Behaviours to  
structure your  
JavaScript code*

```
var Hover = Behavior.create();
```

```
var Hover = Behavior.create();
```

```
Object.extend(Hover.prototype, {  
  initialize: function(className) {  
    this.className = className || 'over';  
  },  
  onmouseover: function() {  
    this.element.addClassName(this.className);  
  },  
  onmouseout: function() {  
    this.element.removeClassName(this.className);  
  }  
});
```

```
var Hover = Behavior.create({
  initialize: function(className) {
    this.className = className || 'over';
  },
  onmouseover: function() {
    this.element.addClassName(this.className);
  },
  onmouseout: function() {
    this.element.removeClassName(this.className);
  }
});
```

```
Event.addBehavior({  
  '.dongle': Hover('hover')  
});
```

# EVENT DELEGATION WITH BEHAVIOURS

```
var Calendar = Behavior.create({
  // other methods hidden for clarity
  onclick : function(e) {
    var source = e.element();
    e.stop();

    if ($(source.parentNode).hasClass('day'))
      return this._setDate(source);
    if ($(source.parentNode).hasClass('back'))
      return this._backMonth();
    if ($(source.parentNode).hasClass('forward'))
      return this._forwardMonth();
  }
});
```

```
var Calendar = Behavior.create({
  // other methods hidden for clarity
  onclick : Event.delegate({
    '.day a' : function(e) {
      // set date
    },
    'back a' : function(e) {
      // go back a month
    },
    '.forward a' : function(e) {
      // go forward
    }
  })
});
```

LINKING BEHAVIOURS  
IN MORE COMPLEX  
SITUATIONS

```
Draggable = Behavior.create({
  initialize : function(options) {
    // code hidden for clarity
    this.handle = this.options.handle || this.element;
    Draggable.Handle.attach(this.handle, this);
  }
  // code hidden for clarity
});
```

```
Draggable.Handle = Behavior.create({
  initialize : function(draggable) {
    this.draggable = draggable;
  },
  onmousedown : function(e) {
    // code hidden for clarity
  }
});
```

**CODE!**

```
var TodoList = Behavior.create({
  initialize: function() {
    // initialization code
  },
  onclick: function() {
    // do something upon a click
  }
});
```

JQUERY

# JQUERY

```
$(document).ready(function() {  
    $("a").click(function() {  
        alert("Hello world!");  
    });  
});
```

```
jQuery.fn.check = function() {  
    return this.each(function() {  
        this.checked = true;  
    });  
};
```

```
$( "input[@type='checkbox'] " ).check();
```

# LOW PRO FOR JQUERY

```
Hover = $.klass({
  initialize: function(hoverClass) {
    this.hoverClass = hoverClass;
  },
  onmouseover: function() {
    this.element.addClass(this.hoverClass);
  },
  onmouseout: function() {
    this.element.removeClass(this.hoverClass);
  }
});

jQuery(function($) {
  $('span.name').attach(Hover, 'myClassName');
});
```

**EVEN EVENT  
DELEGATION!**

```
$( '#thing' ).click( $.delegate( {  
  '.quit': function() {  
    /* do quit stuff */  
  },  
  '.edit': function() {  
    /* do edit stuff */  
  }  
} ) );
```

```
DateSelector = $.klass({
  onclick: $.delegate({
    '.close': function() {
      this.close();
    },
    '.day': function(e) {
      this.selectDate(e.target);
    }
  })
  selectDate: function(dayElement) {
    // code ...
  },
  close: function() {
    // code ...
  }
});
```



THANK YOU



\$9

# Unobtrusive Prototype

*Robust, Organized Javascripting*

by Jarkko Laine

Use code “UNOBTRUSIVE” in Google  
Checkout to get 50% off of any ebook  
or screencast