

Thomas Fuchs

Combining advanced
JavaScript/DOM techniques
and Ajax to build better
User Interfaces

wol|zelle
mol|zelle

Thomas Fuchs

Or:

Create Happy Users

wollzelle
mollzelle

Thomas Fuchs

- **wol||zelle**
- Core team member of
Ruby on Rails
- *script.aculo.us*
- **mir.aculo.us**



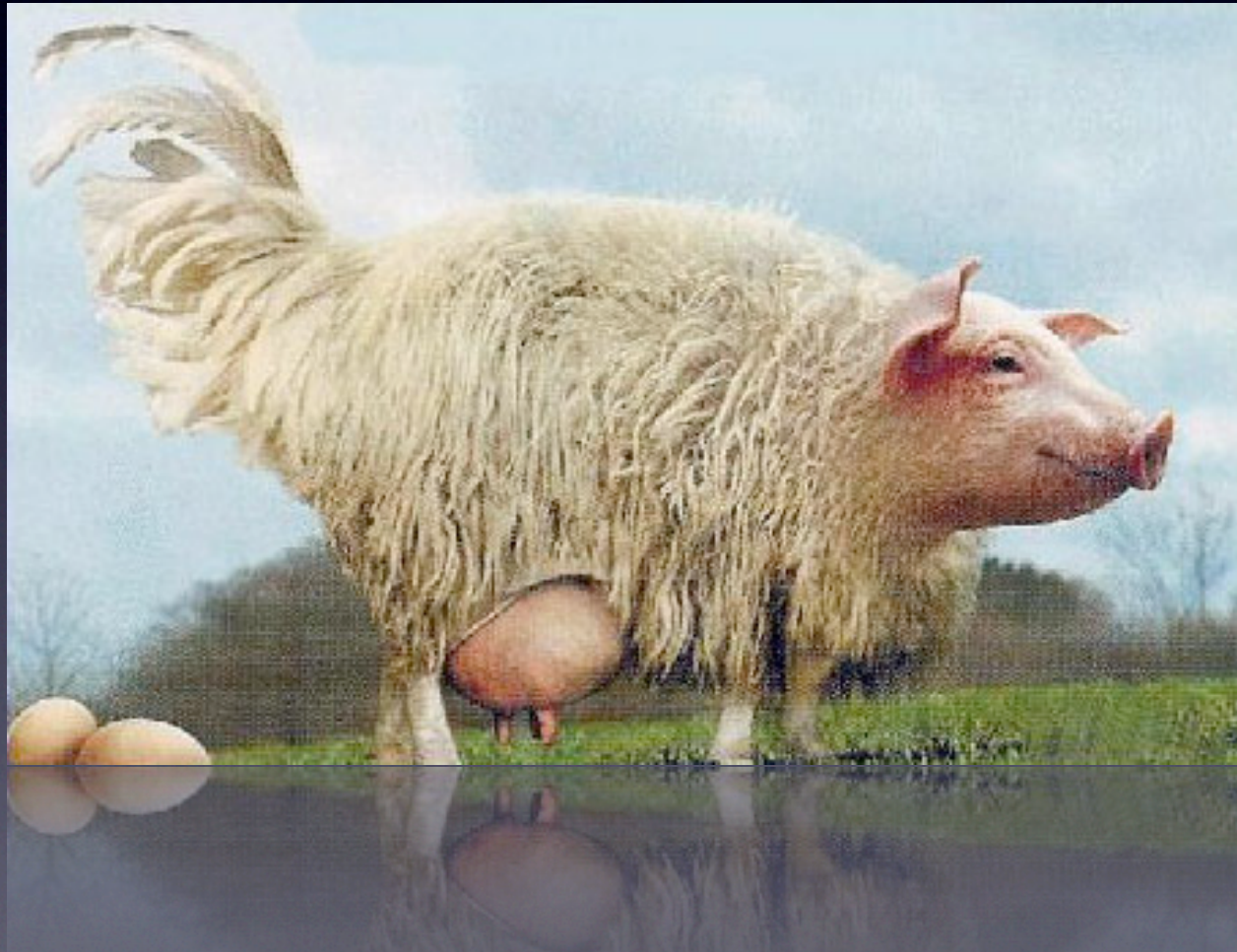
wollzelle

mollzelle

- Branding/Design/Web
- fluxiom Digital Asset Management
<http://www.fluxiom.com>
- <http://www.wollzelle.com/>

Tools

JavaScript, as seen by the browser



Source: <http://www.pollux.franken.de/KNF/>

JavaScript Frameworks
make you...

...write less code...

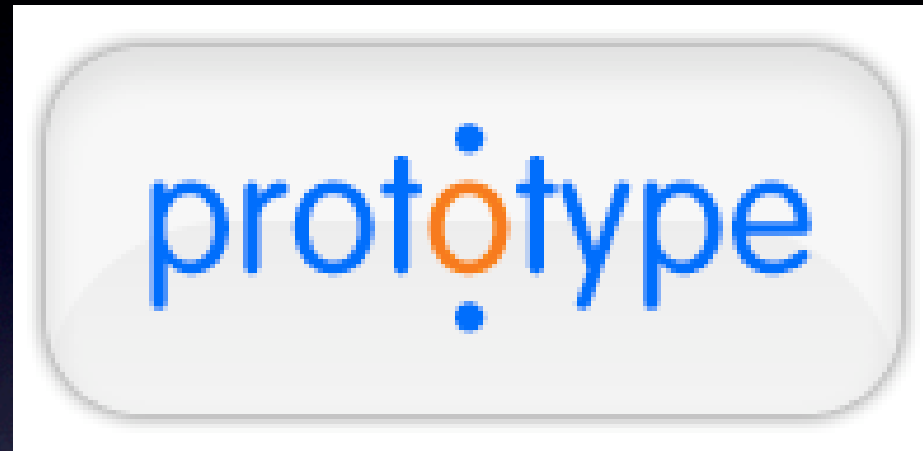
...code more elegantly...

...deal less with
browser bugs...

...and ultimately stay happy.

Concentrate on
your application...

...and not on
browser bugs.



1.5



Prototype is
JavaScript, reloaded



Ajax



Basic Ajax calls (HTML snippets)



Ajax Forms



The hot topic
of 2006!

Ajax behaviours

Server-side JavaScript generation:
Prototype automatically evaluates JavaScript
when Content-type: text/javascript is set



Hooks for integration
with UI enhancements
(like activity indicators)



Extensions to built-in object types

prototype

```
alert('This is a test'.gsub(/s/, '(s)'));  
  
alert('This is a long string'.truncate(10));  
  
alert('String with <tag/>'.stripTags());  
  
var i = 3;  
i.times(function(v){ alert(v) });
```



Enumerables

The logo for 'prototype' is centered at the top of the slide. It consists of the word 'prototype' in a lowercase, sans-serif font. The letter 'o' is stylized with a dot above it and a dot below it, resembling a vertical ellipsis. The text is white and is contained within a white rounded rectangle with a subtle drop shadow.

```
[1,2,3].each( function(n){  
  alert(n);  
});
```

```
alert(['Prototype', 'script.aculo.us', 'Ruby on Rails'].join(', '));
```

```
alert([1,2,[3,4,[5,6]]].flatten().collect( function(v) {  
  return v*v;  
}).join('/'));
```




\$ and \$\$

The logo for the Prototype JavaScript framework, featuring the word "prototype" in a blue, lowercase, sans-serif font. The letter "o" is stylized with a dot above it and a dot below it. The logo is contained within a white rounded rectangle with a subtle drop shadow.

```
if($('blah').visible()) { ... }
```

```
$$('p.hint').each(Element.hide);
```

```
$$('p.hint').each(function(e){  
  e.hide();  
});
```



Direct Element extensions



Prototype 1.4

```
if(Element.visible('blah'))  
  Element.hide('blah');
```

VS

Prototype 1.5

```
if($('blah').visible())  
  $('blah').hide();
```



`.update()`

`.replace()`

`.remove()`

`.visible()`

`.setStyle()`

`.addClassName()`

`(...and lots more)`



Events (without cross-browser pain)

prototype

```
Event.observe('some_input', 'blur',  
  function(){ alert('blur!') }  
);
```


Of course,
just think of it as magic.

*Sam Stephenson,
author of Prototype*



script.aculo.us

1.6

script.aculo.us

Visual effects

script.aculo.us

Drag and Drop

sortable lists demo

this is the first list

- Item 1 from first list.
- Item 2 from first list.
- Item 3 from first list.

and now the second list

- DRAG HERE Item 1 from second list.
- DRAG HERE Item 2 from second list.
- DRAG HERE Item 3 from second list.

2 lines of code!

script.aculo.us

Ready-to-use controls

script.aculo.us

Real-time search

Autocompleting

Drag and Drop

Inplace editor

Slider

script.aculo.us

Plus some extras

JavaScript unit testing, DOM Builder

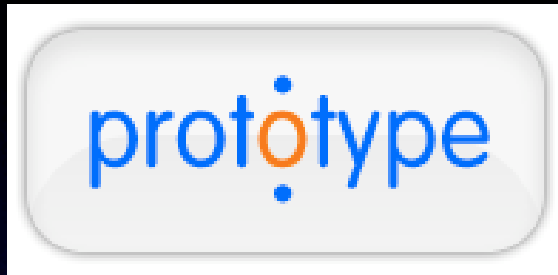
script.aculo.us



script.aculo.us

Docs & Help

- **Wiki:**
<http://wiki.script.aculo.us/>
- **Mailing list:**
<http://lists.rubyonrails.org/mailman/listinfo/rails-spinoffs>
- **IRC channel:**
#prototype on freenode.net



script.aculo.us

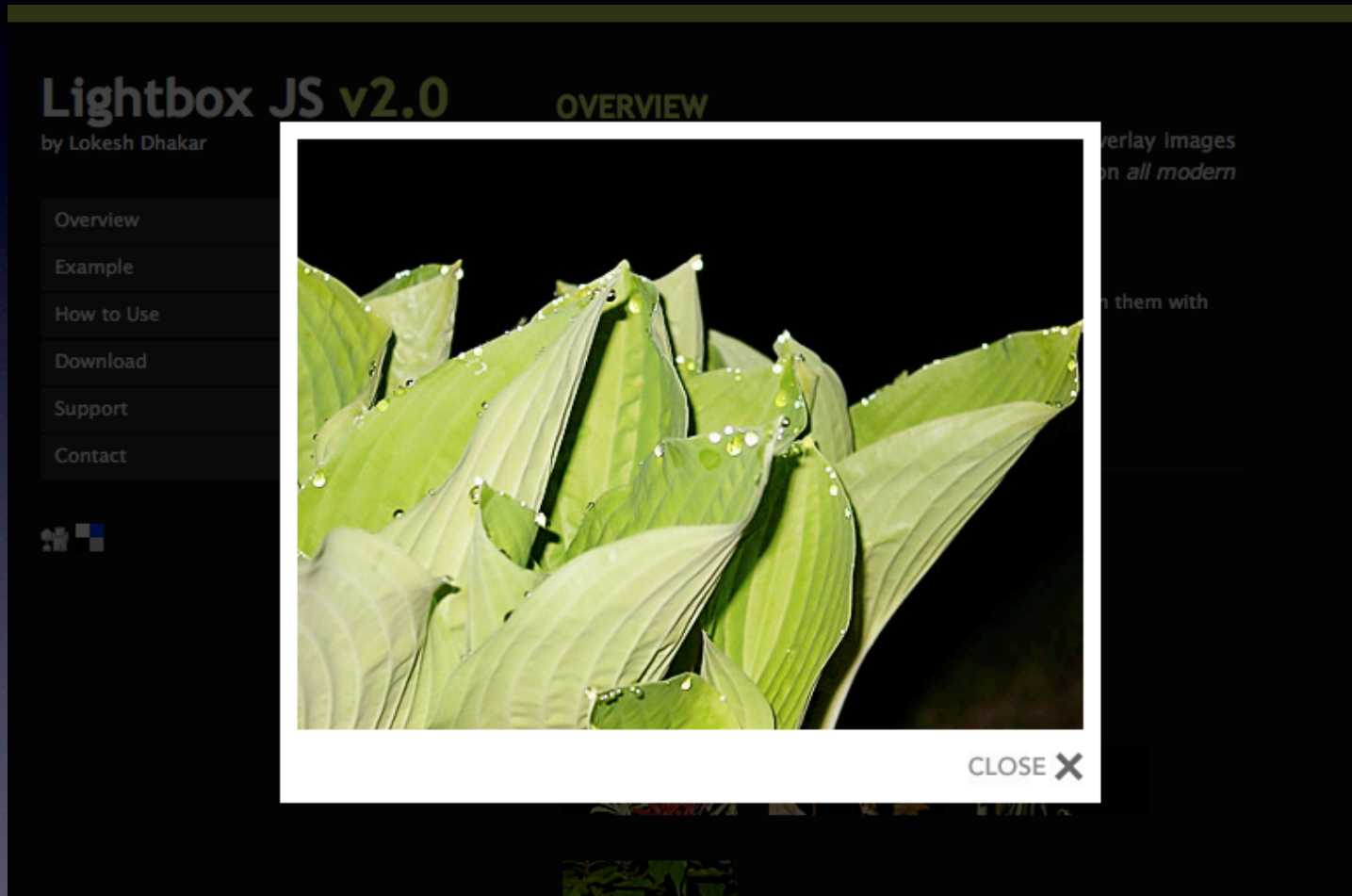
Add-ons

event:Selectors

```
var Rules = {  
  '#icons a:mouseover': function(element) {  
    var app = element.id;  
    new Effect.BlindDown(app + '-content',  
      {queue: 'end', duration: 0.2});  
  },  
  
  '#icons a:mouseout': function(element) {  
    var app = element.id;  
    new Effect.BlindUp(app + '-content',  
      {queue: 'end', duration: 0.2});  
  }  
}
```

<http://encytemedia.com/event-selectors/>

Lightbox JS



<http://www.huddletogether.com/projects/lightbox2/>

Windows

```
<link href="/stylesheets/themes/default.css" r  
  
win = new Window('dialog2', {top:100,left:350  
win.getContent().innerHTML = "Hello world !!";  
win.show();  
  
Dialog.confirm("<center>Are you sure to remove  
    { windowParameters: {width:300,  
      ok:function(win) {  
        // Remove it here !  
      }  
    }  
  });
```

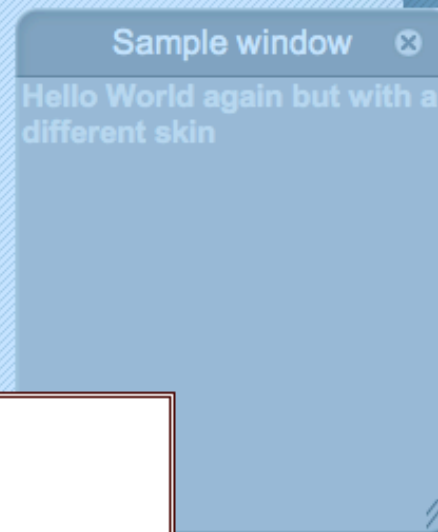
Example

Click on the links below to pop-up windows

[Open a window](#)
[Open another](#)
[Open a window](#)
[position/size](#)
[you close it.\)](#)
[Open a modal](#)

I have also written
javascript :). It opens a window and you can write
messages in this window with the `debug(message)`
function.

You can view and/or download it here : [debug.js](#) or [view](#)
it in live



<http://blogus.xilinus.com/pages/javawin>

Visual effects

Sliding panels and homegrown effects

script.aculo.us

Advanced
engine for visual effects

Time-based

Complex effects easily
possible

script.aculo.us

More than just „eye candy“

Visual feedback

Loading times „feel shorter“

Can help move away from
proprietary plugins

script.aculo.us

On board

Often-used effects

„Yellow Fade Technique“

Fade in and out

Sliding panels

Puff, Shake, etc.

script.aculo.us

```
<div id="message" style="display:none">  
  Hello world!  
</div>  
  
<script type="text/javascript">  
  // <br/>    new Effect.Appear('message');<br/>  // ]]&gt;<br/>&lt;/script&gt;</pre></div>
```

script.aculo.us

```
<style type="text/css" media="screen">
```

```
  #panel {
```

```
    background: #fcc;
```

```
    position: absolute;
```

```
    left: 200px;
```

```
    width: 400px;
```

```
  }
```

```
  #panel div {
```

```
    border: 2px solid #ebb;
```

```
    padding: 20px;
```

```
  }
```

```
</style>
```

```
<div id="panel" style="display:none"><div>
```

```
  Hello world!
```

```
</div></div>
```

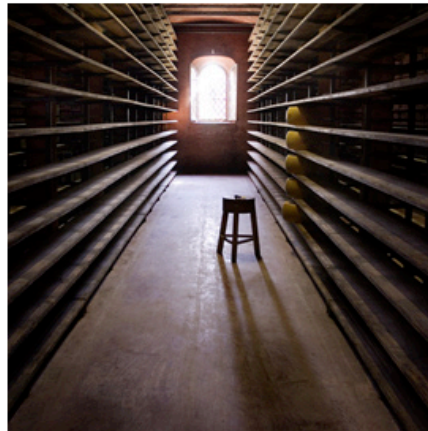
```
<a href="#" onclick="Effect.toggle('panel','slide'); return false">
```

```
  Show/hide panel
```

```
</a>
```

christof wagner

[portfolio 1](#) / [portfolio 2](#) / [portfolio 3](#) / [portfolio 4](#)
[publikationen](#) / [biografie](#) / [referenzen](#) / [kontakt](#)



zollergasse 15/12a / 1070 wien / österreich /
+43 664 185 71 46 / foto@christofwagner.com

copyright © 2006 christof wagner

script.aculo.us

cheat sheet!

a field guide to **scriptaculous**

Combination Effects

the basic syntax

```
Effect.EffectName('dom_id')
```

effects require just a DOM ID

you can use options for customizing
options are optional!

with options

```
Effect.EffectName('dom_id', {option:value, option:value})
```

highlight syntax

optional, if you don't want yellow fade to white

```
new Effect.Highlight('dom_id', {startcolor:'hex', endcolor:'hex'})
```

this bit's critical

default options

Default options are available for all fx.

Some fx have custom options (not shown except for Highlight)

duration: seconds, in decimals

delay: seconds, in decimals

delay before starting effect

from: 0.0 to 1.0 **to:** 0.0 to 1.0

stages of the effect, 0 being no effect;
a Fade set to to: .5 would fade half-way

transition: transition method

<http://www.slash7.com/articles/2006/04/22/scriptaculous-cheat-sheet-1>

script.aculo.us

Do-it-yourself effects

Very easy to do

Automatically use all the
features from the visual
effects engine

script.aculo.us

Complete tutorial &
demo code:

[http://www.thinkvitamin.com/features/
ajax/create-your-own-ajax-effects](http://www.thinkvitamin.com/features/ajax/create-your-own-ajax-effects)

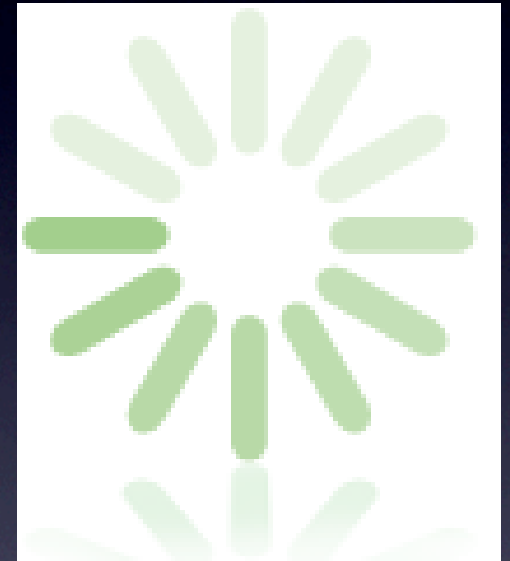
Activity indicators

Show what's going on

„There’s something going
on, wait a sec“

Revival of the
animated GIF

Both „per use“ and
„global“ indicators possible



„Local“ indicator

```
new Ajax.Request('some url',{
  onLoading: function() { $('busy').show() },
  onComplete: function() { $('busy').hide() }
});
```

*for a specific request,
show/hide indicator image*

„Global“ indicator

```
Ajax.Responders.register({
  onCreate: function() {
    if($('busy') && Ajax.activeRequestCount>0)
      Effect.Appear('busy',{duration:0.5,queue:'end'});
  },
  onComplete: function() {
    if($('busy') && Ajax.activeRequestCount==0)
      Effect.Fade('busy',{duration:0.5,queue:'end'});
  }
});
```

*for a all Ajax requests,
appear/fade an indicator image*

Need an indicator image?

hey! this service is Web 2.0!

AjaxLoad

Generator

Indicator type:

Background color: Transparent background





Foreground color:

Generate it

Preview

Press "Generate it"

Top 10

1.		2508
2.		1919
3.		639
4.		597

<http://www.ajaxload.info/>

Annonyances & Debugging

Lessons learned from a year of Ajax

Not all Browsers are
created equal.

CSS

CSS can eliminate or save on
server accesses

HTML sizes are reduced

Be creative, CSS is extremely
capable

JavaScript performance

DOM-API can be slow

Firefox is the slowest (better in 1.5)

Caching references in JavaScript vars

`.innerHTML` is your friend

Venkman `profiling`

Ajax Security

Ajax is as secure as any
browser request

So: Don't trust it

Always verify user submitted
data on the server

Which data format?

Trivial updates: HTML

Complex updates: JavaScript

Use HTTP caching

Don't overengineer

be pragmatic!

Some helpful hints



Local calls only

- Ajax can only call originating server
- Also works with file:// URLs
- Use server-side proxying if you want to tie in other services



The Back button

- The browser back button works by returning the user to the last „non-ajax“ loaded page
- Keep that in mind, especially for public sites
- Workarounds available, but aren't working with all browsers



display: none

- Don't use `display:none` in external CSS
- Browser can't determine the original (built-in) display value of an element (like „block“ for DIVs, and „inline“ for SPANs)
- Use the style attribute instead:
`<div style="display:none">...</div>`



Internet Explorer image cache

- Internet Explorer doesn't use its cache when HTML page fragments that are inserted via Ajax are loading images (IMG tag)
- Some proprietary HTTP headers exist to work around this (hey, it's Microsoft!):
- <http://mir.aculo.us/articles/2005/08/28/internet-explorer-and-ajax-image-caching-woes>



The verboden IDs of Internet Explorer

- IE 6 has some bugs regarding naming ID attributes
- Don't use: „length“, „item“, „namedItem“, „tags“ or „urns“
- <http://meyerweb.com/eric/thoughts/2005/08/29/reserved-id-values/>



JavaScript objects

```
new Ajax.Updater('test', 'update.php', {  
  onComplete: function() {},  
  insertion: Insertion.Top,  
});
```

This code works in Firefox,
but doesn't in Safari.

evil comma
of browser
death

Always test in all browsers you
want to support!

Debugging JavaScript

No rocket surgery here

The first
rule of
AJAX
debugging:
Use Firefox



The second
rule of
AJAX
debugging:
Use Firefox

+Extensions

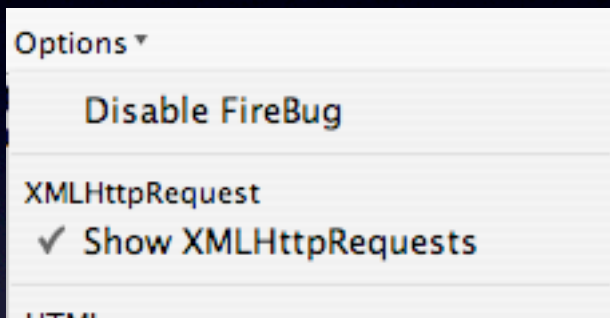


Firebug

The screenshot shows the Firebug console interface. At the top, there are tabs for 'Clear', 'Inspect', 'Options', 'Console', and 'Inspector'. The 'Console' tab is active. The input field contains the command `>>> $$("button")`. The output is a jQuery object array: `[<button onclick="log('Don\'t be a Lert! We have enough Lerts.')" id="button">, <button onclick="log(document.documentElement)" id="button1">, <button onclick="logxml(document.documentElement)" id="button2">, <button onclick="logbox(document.getElementById('button'))" id="button3">, <button onclick="logjs(document.body)" id="button4">, <button onclick="2BeOrNot2Be(40)" id="button5">, <button onclick="foops()" id="button5">, <button onclick="loadStylesheet()" id="button5">, <button onclick="loadXML()" id="button5">, <button onclick="explode()" id="button5">]`. Below the output, a red error icon is followed by the message 'missing ; before statement' and the code snippet `2BeOrNot2Be(40)`. The error is located in 'test.html (line 1)'. Below the error, there is a log entry for a GET request to `http://www13.meebo.com/mcmd/start`. At the bottom of the console, the input field contains `>>> $$("a[href]` and the status bar shows 'Done' and '1 Error'.

```
Clear Inspect Options ▾ Console Inspector 🔍
>>> $$("button")
[<button onclick="log('Don\'t be a Lert! We have enough Lerts.')" id="button">, <button
onclick="log(document.documentElement)" id="button1">, <button
onclick="logxml(document.documentElement)" id="button2">, <button
onclick="logbox(document.getElementById('button'))" id="button3">, <button
onclick="logjs(document.body)" id="button4">, <button onclick="2BeOrNot2Be(40)"
id="button5">, <button onclick="foops()" id="button5">, <button
onclick="loadStylesheet()" id="button5">, <button onclick="loadXML()"
id="button5">, <button onclick="explode()" id="button5"> ]
❌ missing ; before statement test.html (line 1)
  2BeOrNot2Be(40)
  ▼ GET http://www13.meebo.com/mcmd/start
    Response Headers
>>> $$("a[href]
Done ❌ 1 Error
```

Firebug



```
post http://localhost:3000/logtest/update Post Response
try {
$("list").replace("<ul id=\"list\">\n <li id=\"i_1\">item 1</li>\n
$$("p.hint").each(function(value, index) {
value.visualEffect("highlight");
});
var h = $$("p.hint").findAll(function(value, index) {
return value.visible();
});
$$("p.hint").first().hide();
new Draggable("blah", {});
```

Version 0.4 will include a
lightweight JavaScript debugger

Web Developer Extension

The screenshot shows a web browser window with the Web Developer extension menu open. The browser's address bar shows the URL `script.aculo.us`. The page content includes a green header with the text `script.aculo.us`, a navigation bar with `back to site` and `demos` links, and a main content area with a shopping cart interface. The shopping cart interface has a section titled `silly easy shopping` with a text input field containing `Drag products to the cart to fill it:` and two product images: a mug and a t-shirt. Below this is a section titled `your cart:` showing a list of items with a `T-Shirt (2)` entry. A dashed box below the cart contains the text `Drop items here to remove them from the cart.` The Web Developer extension menu is open, displaying a list of options: `Clear Radio Buttons`, `Convert GETs to POSTs`, `Convert POSTs to GETs`, `Display Form Details`, `Enable Auto Completion`, `Make Form Fields Writable`, `Populate Form Fields`, `Remove Maximum Lengths`, `Show Passwords`, and `View Form Information`. The browser's toolbar includes icons for `Disable`, `CSS`, `Forms`, `Images`, `Information`, `Miscellaneous`, `Outline`, `Resize`, `Tools`, `View Source`, and `Options`. A sidebar on the right contains a list of links including `rails 0.13`, `script.aculo.us`, `Autocomp`, `Autocomp`, `customize`, `Shopping`, `Sortable el`, `New AJAX f`, `Error hand`, and `Update ele`. At the bottom right, there is a `RAIL` logo and a text block that reads: `still working of what's in y framework y for you -- do but Ruby on`.

Venkman JavaScript Debugger

The screenshot displays the Venkman JavaScript Debugger interface. The background shows a web browser with a shopping cart page. The debugger window is titled "JavaScript Debugger" and includes a toolbar with buttons for Stop, Continue, Step Over, Step Into, Step Out, Profile, and Pretty Print. The "Loaded Scripts" panel lists various functions with their line numbers. The "Local Variables" panel shows the current scope, event, style, and this object. The "Source Code" panel displays the JavaScript code for the "dragdrop.js" file, with a breakpoint set at line 295. The "Interactive Session" panel shows the current execution context and a list of commands and their results.

Name	Line
[endeffect]	247
[destroy]	282
[currentLeft]	288
[currentTop]	291
[startDrag]	294
[finishDrag]	307
[keyPress]	333
[endDrag]	341

```
288     currentLeft: function() {
289         return parseInt(this.element.style.left || '0')
290     },
291     currentTop: function() {
292         return parseInt(this.element.style.top || '0')
293     },
294     startDrag: function(event) {
295         if(Event.isLeftClick(event)) {
296             this.active = true;
297         }
298     },
299     var style = this.element.style;
300     this.originalY = this.element.offsetTop - th
301     this.originalX = this.element.offsetLeft - th
302     this.offsetY = event.clientY - this.original
```

also does profiling!

Tamperdata

This site demonstrates the **Ruby on Rails 0.13** AJAX helpers, which use script.aculo.us effects, drag-and-drop and autocompletion.

silly easy shopping

Drag products to the cart to fill it:



your cart:



Drop items here to

[View source](#)

Tamper Data - Ongoing requests

Start Tamper Stop Tamper Clear

Time	Duration	Size	Method	Status	Content Type	URL
14:42:...	196 ms	-1	POST	200	text/html	http://script.aculo.us/demos/shop... ..

Request Header Name	Request Header Value	Response Header Name	Response Header Value
Host	script.aculo.us	Status	OK - 200
User-Agent	Mozilla/5.0 (Macintosh; U;...	Date	Mon, 07 Nov 2005 14:38:...
Accept	application/x-shockwave...	Server	Apache/2.0.52 (Gentoo/L...
Accept-Language	en-us,en;q=0.5	Set-Cookie	_session_id=6878c1b45...
Accept-Encoding	gzip,deflate	Cache-Control	no-cache
Accept-Charset	ISO-8859-1,utf-8;q=0.7,*...	Connection	close
Keep-Alive	300	Transfer-Encoding	chunked
Connection	close	Content-Type	text/html
X-Requested-With	XMLHttpRequest		
X-Prototype-Version	1.3.0		
Content-Type	application/x-www-form-...		
Content-Length	15		
Cookie	_session_id=6878c1b45...		
POSTDATA	ⓧid=product_2&_=&ⓧ		

The third
rule:

Test with all
browsers
you want to
support





Safari Web Inspector

The screenshot displays the Safari Web Inspector interface. At the top, the title bar reads "Web Inspector". Below it, a breadcrumb path shows the current selection: `<body>`. A search bar is located to the right of the breadcrumb. The main area shows a tree view of the DOM. The selected element is ``, which is highlighted with a blue border. Below the tree view, there are four tabs: "Node", "Style", "Metrics", and "Properties". The "Node" tab is active, showing the following information:

- Node Type:** Element
- Node Name:** IMG
- Namespace URI:** <http://www.w3.org/1999/xhtml>
- Element Attributes**
 - `src = "images/logo.gif"`
 - `alt = "fluxiom"`
- Markup & Content**
 - ``



Microsoft Script Debugger for IE

- <http://blogs.msdn.com/ie/archive/2004/10/26/247912.aspx>

„It's more like Web 3.0“

What can you actually do with all this?

fluxiom

Available now

www.fluxiom.com

Q & A