Table of Contents

Behaviour Javascript framework Contrib....1
Introduction....1
Usage....2
Example....2
Leaking danger....4
Development....4
License....4
Links....4
Installation Instructions....4
Contrib Settings....5
Contrib Info....5

Behaviour Javascript framework Contrib

This contrib packages the third-party `Behaviour` Javascript event library, written by Ben Nolan.

Behaviour uses CSS selectors to subscribe to javascript event handlers. This allows to create clean code, separated from HTML (and well suited to create javascript based interaction that degrades nicely when javascript is not available).

Introduction

From the website:

> After all the work of WASP and others to promote clean markup, valid pages and graceful degradation via css - it sucks that we're going back to tag soup days by throwing javascript tags into our html.
>
> The better way to do javascript is to do it unobtrusively. PPK and Simon Willison have been recommending this approach for ages. And it's definitely the way to go. The only problem is that it's a bit of a pain in the ass.
>
> That's why I came up with Behaviour - my solution to unobtrusive javascript behaviours.
>
> **How does it work?**
>
> Behaviour lets you use CSS selectors to specify elements to add javascript events to. This means that instead of writing:

```
<li>
    <a onclick="this.parentNode.removeChild(this)" href="#">
        Click me to delete me
    </a>
</li>
```

> You can use:

```
<ul id="example">
    <li>
        <a href="/someurl">Click me to delete me</a>
    </li>
</ul>
```

> And then use css selectors to select that element and add javascript functions to it.

```
var myrules = {
    '#example li' : function(el){
        el.onclick = function(){
            this.parentNode.removeChild(this);

        }
```

```
        }
    };

    Behaviour.register(myrules);
```

Usage

Include the javascript file:

```
<script type="text/javascript" src="%PUBURL%/%SYSTEMWEB%/BehaviourContrib/behaviour.js"></
```

In your code you create a "rules" object, with sub-objects for each html element class name or id:

```
var myrules = {
    '.classname' : function(element) {
        // element event
        element.onclick = function() {
            // code here
        }
    },

    '#id' : function(element) {
        // element event
        element.onclick = function() {
            // code here
        }
    }
};
```

Or use nested identifiers:

```
var myrules = {
    '.menu li a' : function(element) {
        element.onclick = function() {
            // code here
        }
    }
};
```

Apply the rules with:

```
Behaviour.register(myrules);
```

Example

If we have a 'normal' link to a web hometopic: Web Home, we can use javascript to make it open a popup window. When javascript is not available the link behaviour defaults to opening the page in the current window.

```
<div id="demoblock" style="padding:1em; width:100px; text-align:center;">
MOUSE OVER ME
</div>

<script type="text/javascript">
// <![CDATA[
var myrules = {
    '#demoblock' : function(el) {
        var defaultColor = '#A3D6F8';
        var highlightColor = '#4A7FB5';

        el.style.backgroundColor = defaultColor;

        el.onmouseover = function() {
            this.style.backgroundColor = highlightColor;
            return false;
        }
        el.onmouseout = function() {
            this.style.backgroundColor = defaultColor;
            return false;
        }
    },
    '#demoblock span' : function(el) {

        var text = el.innerHTML;

        var fisherYates = function (inArray) {
          var i = inArray.length;
          if ( i == 0 ) return false;
          while ( --i ) {
             var j = Math.floor( Math.random() * ( i + 1 ) );
             var tempi = inArray[i];
             var tempj = inArray[j];
             inArray[i] = tempj;
             inArray[j] = tempi;
           }
        }

        var randomize = function(inText) {
            var letters = inText.split('');
            fisherYates(letters);
            return letters.join('');
        }
        el.onmouseover = function() {
            this.innerHTML = randomize(text);
            return false;
        }
        el.onmouseout = function() {
            this.innerHTML = text;
            return false;
        }
    }
};
Behaviour.register(myrules);
// ]]>
</script>
```

Creates:

MOUSE OVER ME

//

Leaking danger

Behaviour code leaks memory on Windows Explorer prior to version 7. To prevent leaking, set the element variable to `null`:

```
var myrules = {
    'table.test td' : function(element) {
        element.onmouseover = function() {
            this.style.backgroundColor = highlightColor;
            return false;
        }
        element = null; // by setting this IE will not leak
    }
};
Behaviour.register(myrules);
```

Development

- Google Groups: Behaviour Javascript Library
- Nabble - Behaviour Javascript Library forum & mailing list archive
- Behaviour2 - update in the making, since 2006

License

Behaviour is freely distributable under the terms of an BSD license. For details see the Behaviour website.

Links

- Behaviour Google Group

Installation Instructions

You do not need to install anything in the browser to use this extension. The following instructions are for the administrator who installs the extension on the server.

Open configure, and open the "Extensions" section. Use "Find More Extensions" to get a list of available extensions. Select "Install".

If you have any problems, or if the extension isn't available in `configure`, then you can still install manually from the command-line. See http://foswiki.org/Support/ManuallyInstallingExtensions for more help.

Contrib Settings

- Set SHORTDESCRIPTION = `Behaviour` Javascript event library to create javascript based interactions that degrade well when javascript is not available

You can also set the global preference settings BEHAVIOURCONTRIB_DEBUG to 1 to make the contrib use uncompressed javascript sources, in the event of problems.

Contrib Info

Author:	Foswiki:Main.ArthurClemens
Copyright:	Code: `behaviour.js` version 1.1 - Copyright (c) Ben Nolan and Simon Willison. Foswiki distribu Foswiki:Main.ArthurClemens.
License:	BSD
Version:	5016 (2009-09-20)
Release:	1.5
Change History:	
20 Sep 2009	1.5 Minor documentation updates for Foswiki 1.0.7. No functional changes
03 Dec 2008	1.4 Foswiki release version.
17 Oct 2007	1.3 Replaced "faster code" by other code from Dean Edwards, [[packed by http://groups.google.com/group/behaviour/browse_thread/thread/85137977bedf5ed/3cf3ba8065d41a8 Irving]].
02 Jul 2007	1.2 Integrated other faster code by Dean Edwards: faster onload (again).
08 Mar 2007	1.1 Integrated code by Dean Edwards (see Code update version 1.1 with faster DOM queries).
04 Jun 2006	1.0 First Version. Included Behaviour version: 1.1.
Dependencies:	None
Home:	http://foswiki.org/Extensions/BehaviourContrib
Support:	http://foswiki.org/Support/BehaviourContrib

Edit | Attach | Print version | History: %REVISIONS% | Backlinks | Raw View | More topic actions

Topic revision: r1 - 08 Jul 2007 - 18:54:51 - ProjectContributor

- . System

- Log In

- **Toolbox**
- Users
- Groups
- Index
- Search
- Changes
- Notifications
- RSS Feed
- Statistics
- Preferences

- **User Reference**
- BeginnersStartHere
- TextFormattingRules
- Macros
- FormattedSearch
- QuerySearch
- DocumentGraphics
- SkinBrowser
- InstalledPlugins

- **Admin Maintenance**
- Reference Manual
- AdminToolsCategory
- InterWikis
- ManagingWebs
- SiteTools
- DefaultPreferences
- WebPreferences

- **Categories**
- Admin Documentation
- Admin Tools
- Developer Doc
- User Documentation
- User Tools

- **Webs**
- Public
- System

Copyright © by the contributing authors. All material on this site is the property of the contributing authors.
Ideas, requests, problems regarding Wiki? Send feedback