

Table of Contents

JQueryMetadata.....	1
Usage.....	1
Examples.....	1

JQueryMetadata

Homepage: <http://plugins.jquery.com/project/metadata>

Author(s): John Resig, Yehuda Katz, Joern Zaefferer, Paul McLanahan

Version: 3630

This plugin is capable of extracting metadata from classes, random attributes, and child elements.

Usage

A lot of jQuery plugins are initialized and configured using *metadata* embedded into the markup thus eliminating the need to explicitly write JavaScript. A behavior is assigned to an html element by giving it the pertinent class name that the plugin listens to. Parameters to the plugin itself are embedded as a JSON object inside the class attribute. JQueryMetadata can also be encoded differently. Having it inside the class attribute as well is most common. One exception is JQueryCorner for example, that uses an explicit `data-corner` attribute, which of course isn't standard conform markup. However browsers happily ignore attributes that they don't understand. So that's not a problem in real live.

Examples

Load the plugin as required for the current page:

```
%JQREQUIRE{"mynewplugin"}%
```

This is how the markup looks like:

```
<div class="jqMyNewPlugin {key1:'value1', key2:'value2', key3:'value3'}">  
</div>
```










This is the plugin's initializer reading the metadata:

```
jQuery(function($) {  
  
    var defaults {  
        key1: 'default1',  
        key2: 'default2',  
        key3: 'default3'  
    };  
  
    // find all elements tagged .jqMyNewPlugin that aren't init'ed yet  
    // ... using livequery instead of each to trigger initialisation of async content  
  
    $(".jqMyNewPlugin:not(jqInitedMyNewPlugin)").livequery(function() {  
  
        // create a jQuery object for this  
        var $this = $(this);  
  
        // prevent the markup to be init'ed multiple times  
        $this.addClass("jqInitedMyNewPlugin");  
  
        // get plugin options by merging defaults and current json objs  
        var opts = $.extend({}, defaults, $this.metadata());  
  
        // call the plugin handler  
        $this.myNewPlugin(opts);  
    });  
});
```

- ☐ 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