* Overview
* Options
* Events
* Methods
* Theming

jQuery UI Tabs

## Overview

Tabs are generally used to break content into multiple sections that can be swapped to save space, much like an accordion.

By default a tab widget will swap between tabbed sections onClick, but the events can be changed to onHover through an option. Tab content can be loaded via Ajax by setting an href on a tab.

*NOTE: Tabs created dynamically using .tabs( "add", ... ) are given an id of ui-tabs-NUM, where NUM is an auto-incrementing id. If you use this naming convention for your own elements, you may encounter problems.*

|  |
| --- |
| Contents  * [1 Events](#id.gjdgxs) * [2 Ajax mode](#id.30j0zll)   + [2.1 Back button and bookmarking](#id.1fob9te) * [3 How to...](#id.3znysh7)   + [3.1 ...retrieve the index of the currently selected tab](#id.2et92p0)   + [3.2 ...open links in the current tab instead of leaving the page](#id.tyjcwt)   + [3.3 ...select a tab from a text link instead of clicking a tab itself](#id.3dy6vkm)   + [3.4 ...prevent switching to the tab on click depending on form validation](#id.1t3h5sf)   + [3.5 ...immediately select a just added tab](#id.4d34og8)   + [3.6 ...follow a tab's URL instead of loading its content via ajax](#id.2s8eyo1)   + [3.7 ...prevent a FOUC (Flash of Unstyled Content) before tabs are initialized](#id.17dp8vu) * [4 Why does...](#id.3rdcrjn)   + [4.1 ...my slider, Google Map, sIFR etc. not work when placed in a hidden (inactive) tab?](#id.26in1rg) |

[[edit](http://docs.jquery.com/action/edit/Template:UIAPIPlugin?section=1)]

## Events

A series of events fire when interacting with a tabs interface:

* tabsselect, tabsload, tabsshow (in that order)
* tabsadd, tabsremove
* tabsenable, tabsdisable

Event binding example:

$('#example').bind('tabsselect', function(event, ui) {  
  
 // Objects available in the function context:  
 ui.tab // anchor element of the selected (clicked) tab  
 ui.panel // element, that contains the selected/clicked tab contents  
 ui.index // zero-based index of the selected (clicked) tab  
  
});

Note that if a handler for the tabsselect event returns false, the clicked tab will not become selected (useful for example if switching to the next tab requires a form validation).

[[edit](http://docs.jquery.com/action/edit/Template:UIAPIPlugin?section=2)]

## Ajax mode

Tabs supports loading tab content via Ajax in an unobtrusive manner.

The HTML you need is slightly different from the one that is used for static tabs: A list of links pointing to existing resources (from where the content gets loaded) and no additional containers at all (unobtrusive!). The containers' markup is going to be created on the fly:

<div id="example">  
 <ul>  
 <li><a href="ahah\_1.html"><span>Content 1</span></a></li>  
 <li><a href="ahah\_2.html"><span>Content 2</span></a></li>  
 <li><a href="ahah\_3.html"><span>Content 3</span></a></li>  
 </ul>  
</div>

Obviously this degrades gracefully - the links, e.g. the content, will still be accessible with JavaScript disabled.

Note that if you wish to reuse an existing container, you could do so by matching a title attribute and the container's id:

<li><a href="hello/world.html" title="Todo Overview"> ... </a></li>

and a container like:

<div id="Todo\_Overview"> ... </div>

(Note how white space is replaced with an underscore)

This is useful if you want a human readable hash in the URL instead of a cryptic generated one.

[[edit](http://docs.jquery.com/action/edit/Template:UIAPIPlugin?section=3)]

### Back button and bookmarking

Tabs 2 already supported this functionality, although the history plugin needs a rewrite first (it doesn't support Safari 3 and is in general a little inflexible) before it can be build back into the tabs. It is planned and Klaus is working on it whenever he finds the time. Actual bugs in the UI Tabs plugin itself always have higher priority though.

[[edit](http://docs.jquery.com/action/edit/Template:UIAPIPlugin?section=4)]

## How to...

[[edit](http://docs.jquery.com/action/edit/Template:UIAPIPlugin?section=5)]

### ...retrieve the index of the currently selected tab

var $tabs = $('#example').tabs();  
var selected = $tabs.tabs('option', 'selected'); // => 0

[[edit](http://docs.jquery.com/action/edit/Template:UIAPIPlugin?section=6)]

### ...open links in the current tab instead of leaving the page

"Hijax" links after tab content has been loaded:

$('#example').tabs({  
 load: function(event, ui) {  
 $('a', ui.panel).click(function() {  
 $(ui.panel).load(this.href);  
 return false;  
 });  
 }  
});

[[edit](http://docs.jquery.com/action/edit/Template:UIAPIPlugin?section=7)]

### ...select a tab from a text link instead of clicking a tab itself

var $tabs = $('#example').tabs(); // first tab selected  
  
$('#my-text-link').click(function() { // bind click event to link  
 $tabs.tabs('select', 2); // switch to third tab  
 return false;  
});

[[edit](http://docs.jquery.com/action/edit/Template:UIAPIPlugin?section=8)]

### ...prevent switching to the tab on click depending on form validation

Returning false in the tabs select handler prevents the clicked tab from becoming selected.

$('#example').tabs({  
 select: function(event, ui) {  
 var isValid = ... // form validation returning true or false  
 return isValid;  
 }  
});

[[edit](http://docs.jquery.com/action/edit/Template:UIAPIPlugin?section=9)]

### ...immediately select a just added tab

var $tabs = $('#example').tabs({  
 add: function(event, ui) {  
 $tabs.tabs('select', '#' + ui.panel.id);  
 }  
});

[[edit](http://docs.jquery.com/action/edit/Template:UIAPIPlugin?section=10)]

### ...follow a tab's URL instead of loading its content via ajax

Note that opening a tab in a new window is unexpected, e.g. inconsistent behaviour exposing a usablity problem (<http://www.useit.com/alertbox/tabs.html>).

$('#example').tabs({  
 select: function(event, ui) {  
 var url = $.data(ui.tab, 'load.tabs');  
 if( url ) {  
 location.href = url;  
 return false;  
 }  
 return true;  
 }  
});

[[edit](http://docs.jquery.com/action/edit/Template:UIAPIPlugin?section=11)]

### ...prevent a FOUC (Flash of Unstyled Content) before tabs are initialized

Add the necessary classes to hide an inactive tab panel to the HTML right away - note that this will **not** degrade gracefully with JavaScript being disabled:

<div id="example" class="ui-tabs">  
 ...  
 <div id="a-tab-panel" class="ui-tabs-hide"> </div>  
 ...  
</div>

[[edit](http://docs.jquery.com/action/edit/Template:UIAPIPlugin?section=12)]

## Why does...

[[edit](http://docs.jquery.com/action/edit/Template:UIAPIPlugin?section=13)]

### ...my slider, Google Map, sIFR etc. not work when placed in a hidden (inactive) tab?

Any component that requires some dimensional computation for its initialization won't work in a hidden tab, because the tab panel itself is hidden via display: none so that any elements inside won't report their actual width and height (0 in most browsers).

There's an easy workaround. Use the *off-left technique* for hiding inactive tab panels. E.g. in your style sheet replace the rule for the class selector ".ui-tabs .ui-tabs-hide" with

.ui-tabs .ui-tabs-hide {  
 position: absolute;  
 left: -10000px;  
}

For Google maps you can also resize the map once the tab is displayed like this:

$('#example').bind('tabsshow', function(event, ui) {  
 if (ui.panel.id == "map-tab") {  
 resizeMap();  
 }  
});

resizeMap() will call Google Maps' checkResize() on the particular map.

### Dependencies

* UI Core
* UI Widget
* **Optional:** [jquery.cookie.js](http://plugins.jquery.com/project/cookie) for use with [cookie option](http://jqueryui.com/demos/tabs/#option-cookie) (also included in development-bundle>external folder.)

### Example

* Demo
* View Source

A simple jQuery UI Tabs.

$("#tabs").tabs();

<!DOCTYPE html>  
<html>  
<head>  
 <link href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/base/jquery-ui.css" rel="stylesheet" type="text/css"/>  
 <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.5/jquery.min.js"></script>  
 <script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/jquery-ui.min.js"></script>  
   
 <script>  
 $(document).ready(function() {  
 $("#tabs").tabs();  
 });  
 </script>  
</head>  
<body style="font-size:62.5%;">  
   
<div id="tabs">  
 <ul>  
 <li><a href="#fragment-1"><span>One</span></a></li>  
 <li><a href="#fragment-2"><span>Two</span></a></li>  
 <li><a href="#fragment-3"><span>Three</span></a></li>  
 </ul>  
 <div id="fragment-1">  
 <p>First tab is active by default:</p>  
 <pre><code>$('#example').tabs();</code></pre>  
 </div>  
 <div id="fragment-2">  
 Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.  
 Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.  
 </div>  
 <div id="fragment-3">  
 Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.  
 Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.  
 Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.  
 </div>  
</div>  
</body>  
</html>

## Options

#### **disabled**Type: Boolean Default: false Disables (true) or enables (false) the tabs. Can be set when initialising (first creating) the tabs. Code examplesInitialize a tabs with the disabled option specified. $( ".selector" ).tabs({ disabled: true });Get or set the disabled option, after init. //getter var disabled = $( ".selector" ).tabs( "option", "disabled" ); //setter $( ".selector" ).tabs( "option", "disabled", true );

#### **ajaxOptions**Type: Options Default: null Additional Ajax options to consider when loading tab content (see $.ajax). Code examplesInitialize a tabs with the ajaxOptions option specified. $( ".selector" ).tabs({ ajaxOptions: { async: false } });Get or set the ajaxOptions option, after init. //getter var ajaxOptions = $( ".selector" ).tabs( "option", "ajaxOptions" ); //setter $( ".selector" ).tabs( "option", "ajaxOptions", { async: false } );

#### **cache**Type: Boolean Default: false Whether or not to cache remote tabs content, e.g. load only once or with every click. Cached content is being lazy loaded, e.g once and only once for the first click. Note that to prevent the actual Ajax requests from being cached by the browser you need to provide an extra cache: false flag to ajaxOptions. Code examplesInitialize a tabs with the cache option specified. $( ".selector" ).tabs({ cache: true });Get or set the cache option, after init. //getter var cache = $( ".selector" ).tabs( "option", "cache" ); //setter $( ".selector" ).tabs( "option", "cache", true );

#### **collapsible**Type: Boolean Default: false Set to true to allow an already selected tab to become unselected again upon reselection. Code examplesInitialize a tabs with the collapsible option specified. $( ".selector" ).tabs({ collapsible: true });Get or set the collapsible option, after init. //getter var collapsible = $( ".selector" ).tabs( "option", "collapsible" ); //setter $( ".selector" ).tabs( "option", "collapsible", true );

#### **cookie**Type: Object Default: null Store the latest selected tab in a cookie. The cookie is then used to determine the initially selected tab if the *selected* option is not defined. Requires [cookie plugin](http://plugins.jquery.com/project/cookie), which can also be found in the development-bundle>external folder from the download builder. The object needs to have key/value pairs of the form the cookie plugin expects as options. Available options (example): { expires: 7, path: '/', domain: 'jquery.com', secure: true }. Since jQuery UI 1.7 it is also possible to define the cookie name being used via *name* property. Code examplesInitialize a tabs with the cookie option specified. $( ".selector" ).tabs({ cookie: { expires: 30 } });Get or set the cookie option, after init. //getter var cookie = $( ".selector" ).tabs( "option", "cookie" ); //setter $( ".selector" ).tabs( "option", "cookie", { expires: 30 } );

#### **deselectable**Type: Boolean Default: false deprecated in jQuery UI 1.7, use collapsible. Code examplesInitialize a tabs with the deselectable option specified. $( ".selector" ).tabs({ deselectable: true });Get or set the deselectable option, after init. //getter var deselectable = $( ".selector" ).tabs( "option", "deselectable" ); //setter $( ".selector" ).tabs( "option", "deselectable", true );

#### **disabled**Type: Array<Number> Default: [] An array containing the position of the tabs (zero-based index) that should be disabled on initialization. Code examplesInitialize a tabs with the disabled option specified. $( ".selector" ).tabs({ disabled: [1, 2] });Get or set the disabled option, after init. //getter var disabled = $( ".selector" ).tabs( "option", "disabled" ); //setter $( ".selector" ).tabs( "option", "disabled", [1, 2] );

#### **event**Type: String Default: 'click' The type of event to be used for selecting a tab. Code examplesInitialize a tabs with the event option specified. $( ".selector" ).tabs({ event: 'mouseover' });Get or set the event option, after init. //getter var event = $( ".selector" ).tabs( "option", "event" ); //setter $( ".selector" ).tabs( "option", "event", 'mouseover' );

#### **fx**Type: Options, Array<Options> Default: null Enable animations for hiding and showing tab panels. The duration option can be a string representing one of the three predefined speeds ("slow", "normal", "fast") or the duration in milliseconds to run an animation (default is "normal"). Code examplesInitialize a tabs with the fx option specified. $( ".selector" ).tabs({ fx: { opacity: 'toggle' } });Get or set the fx option, after init. //getter var fx = $( ".selector" ).tabs( "option", "fx" ); //setter $( ".selector" ).tabs( "option", "fx", { opacity: 'toggle' } );

#### **idPrefix**Type: String Default: 'ui-tabs-' If the remote tab, its anchor element that is, has no title attribute to generate an id from, an id/fragment identifier is created from this prefix and a unique id returned by $.data(el), for example "ui-tabs-54". Code examplesInitialize a tabs with the idPrefix option specified. $( ".selector" ).tabs({ idPrefix: 'ui-tabs-primary' });Get or set the idPrefix option, after init. //getter var idPrefix = $( ".selector" ).tabs( "option", "idPrefix" ); //setter $( ".selector" ).tabs( "option", "idPrefix", 'ui-tabs-primary' );

#### **panelTemplate**Type: String Default: '<div></div>' HTML template from which a new tab panel is created in case of adding a tab with the add method or when creating a panel for a remote tab on the fly. Code examplesInitialize a tabs with the panelTemplate option specified. $( ".selector" ).tabs({ panelTemplate: '<li></li>' });Get or set the panelTemplate option, after init. //getter var panelTemplate = $( ".selector" ).tabs( "option", "panelTemplate" ); //setter $( ".selector" ).tabs( "option", "panelTemplate", '<li></li>' );

#### **selected**Type: Number Default: 0 Zero-based index of the tab to be selected on initialization. To set all tabs to unselected pass -1 as value. Code examplesInitialize a tabs with the selected option specified. $( ".selector" ).tabs({ selected: 3 });Get or set the selected option, after init. //getter var selected = $( ".selector" ).tabs( "option", "selected" ); //setter $( ".selector" ).tabs( "option", "selected", 3 );

#### **spinner**Type: String Default: '<em>Loading…</em>' The HTML content of this string is shown in a tab title while remote content is loading. Pass in empty string to deactivate that behavior. An span element must be present in the A tag of the title, for the spinner content to be visible. Code examplesInitialize a tabs with the spinner option specified. $( ".selector" ).tabs({ spinner: 'Retrieving data...' });Get or set the spinner option, after init. //getter var spinner = $( ".selector" ).tabs( "option", "spinner" ); //setter $( ".selector" ).tabs( "option", "spinner", 'Retrieving data...' );

#### **tabTemplate**Type: String Default: '<li><a href="#{href}"><span>#{label}</span></a></li>' HTML template from which a new tab is created and added. The placeholders #{href} and #{label} are replaced with the url and tab label that are passed as arguments to the add method. Code examplesInitialize a tabs with the tabTemplate option specified. $( ".selector" ).tabs({ tabTemplate: '<div><a href="#{href}"><span>#{label}</span></a></div>' });Get or set the tabTemplate option, after init. //getter var tabTemplate = $( ".selector" ).tabs( "option", "tabTemplate" ); //setter $( ".selector" ).tabs( "option", "tabTemplate", '<div><a href="#{href}"><span>#{label}</span></a></div>' );

## Events

#### **create**Type: tabscreate This event is triggered when tabs is created. Code examplesSupply a callback function to handle the create event as an init option. $( ".selector" ).tabs({ create: function(event, ui) { ... } });Bind to the create event by type: tabscreate. $( ".selector" ).bind( "tabscreate", function(event, ui) { ... });

#### **select**Type: tabsselect This event is triggered when clicking a tab. Code examplesSupply a callback function to handle the select event as an init option. $( ".selector" ).tabs({ select: function(event, ui) { ... } });Bind to the select event by type: tabsselect. $( ".selector" ).bind( "tabsselect", function(event, ui) { ... });

#### **load**Type: tabsload This event is triggered after the content of a remote tab has been loaded. Code examplesSupply a callback function to handle the load event as an init option. $( ".selector" ).tabs({ load: function(event, ui) { ... } });Bind to the load event by type: tabsload. $( ".selector" ).bind( "tabsload", function(event, ui) { ... });

#### **show**Type: tabsshow This event is triggered when a tab is shown. Code examplesSupply a callback function to handle the show event as an init option. $( ".selector" ).tabs({ show: function(event, ui) { ... } });Bind to the show event by type: tabsshow. $( ".selector" ).bind( "tabsshow", function(event, ui) { ... });

#### **add**Type: tabsadd This event is triggered when a tab is added. Code examplesSupply a callback function to handle the add event as an init option. $( ".selector" ).tabs({ add: function(event, ui) { ... } });Bind to the add event by type: tabsadd. $( ".selector" ).bind( "tabsadd", function(event, ui) { ... });

#### **remove**Type: tabsremove This event is triggered when a tab is removed. Code examplesSupply a callback function to handle the remove event as an init option. $( ".selector" ).tabs({ remove: function(event, ui) { ... } });Bind to the remove event by type: tabsremove. $( ".selector" ).bind( "tabsremove", function(event, ui) { ... });

#### **enable**Type: tabsenable This event is triggered when a tab is enabled. Code examplesSupply a callback function to handle the enable event as an init option. $( ".selector" ).tabs({ enable: function(event, ui) { ... } });Bind to the enable event by type: tabsenable. $( ".selector" ).bind( "tabsenable", function(event, ui) { ... });

#### **disable**Type: tabsdisable This event is triggered when a tab is disabled. Code examplesSupply a callback function to handle the disable event as an init option. $( ".selector" ).tabs({ disable: function(event, ui) { ... } });Bind to the disable event by type: tabsdisable. $( ".selector" ).bind( "tabsdisable", function(event, ui) { ... });

## Methods

### **destroy**Signature: .tabs( "destroy" ) Remove the tabs functionality completely. This will return the element back to its pre-init state.

### **disable**Signature: .tabs( "disable" ) Disable the tabs.

### **enable**Signature: .tabs( "enable" ) Enable the tabs.

### **option**Signature: .tabs( "option" , optionName , [value] ) Get or set any tabs option. If no value is specified, will act as a getter.

### **option**Signature: .tabs( "option" , options ) Set multiple tabs options at once by providing an options object.

### **widget**Signature: .tabs( "widget" ) Returns the .ui-tabs element.

### **add**Signature: .tabs( "add" , url , label , [index] ) Add a new tab. The second argument is either a URL consisting of a fragment identifier only to create an in-page tab or a full url (relative or absolute, no cross-domain support) to turn the new tab into an Ajax (remote) tab. The third is the zero-based position where to insert the new tab. Optional, by default a new tab is appended at the end.

### **remove**Signature: .tabs( "remove" , index ) Remove a tab. The second argument is the zero-based index of the tab to be removed.

### **enable**Signature: .tabs( "enable" , index ) Enable a disabled tab. To enable more than one tab at once reset the disabled property like: $('#example').tabs("option","disabled",[]);. The second argument is the zero-based index of the tab to be enabled.

### **disable**Signature: .tabs( "disable" , index ) Disable a tab. The selected tab cannot be disabled. To disable more than one tab at once use: $('#example').tabs("option","disabled", [1, 2, 3]); The second argument is the zero-based index of the tab to be disabled.

### **select**Signature: .tabs( "select" , index ) Select a tab, as if it were clicked. The second argument is the zero-based index of the tab to be selected or the id selector of the panel the tab is associated with (the tab's href fragment identifier, e.g. hash, points to the panel's id).

### **load**Signature: .tabs( "load" , index ) Reload the content of an Ajax tab programmatically. This method always loads the tab content from the remote location, even if cache is set to true. The second argument is the zero-based index of the tab to be reloaded.

### **url**Signature: .tabs( "url" , index , url ) Change the url from which an Ajax (remote) tab will be loaded. The specified URL will be used for subsequent loads. Note that you can not only change the URL for an existing remote tab with this method, but also turn an in-page tab into a remote tab. The second argument is the zero-based index of the tab of which its URL is to be updated. The third is a URL the content of the tab is loaded from.

### **length**Signature: .tabs( "length" ) Retrieve the number of tabs of the first matched tab pane.

### **abort**Signature: .tabs( "abort" ) Terminate all running tab ajax requests and animations.

### **rotate**Signature: .tabs( "rotate" , ms , [continuing] ) Set up an automatic rotation through tabs of a tab pane. The second argument is an amount of time in milliseconds until the next tab in the cycle gets activated. Use 0 or null to stop the rotation. The third controls whether or not to continue the rotation after a tab has been selected by a user. Default: false.

## Theming

The jQuery UI Tabs plugin uses the jQuery UI CSS Framework to style its look and feel, including colors and background textures. We recommend using the ThemeRoller tool to create and download custom themes that are easy to build and maintain.

If a deeper level of customization is needed, there are widget-specific classes referenced within the jquery.ui.tabs.css stylesheet that can be modified. These classes are highlighed in bold below.

### Sample markup with jQuery UI CSS Framework classes

<div class="**ui-tabs** ui-widget ui-widget-content ui-corner-all" id="tabs">

   <ul class="**ui-tabs-nav** ui-helper-reset ui-helper-clearfix ui-widget-header ui-corner-all">

     <li class="ui-state-default ui-corner-top ui-tabs-selected ui-state-active"><a href="#tabs-1">Nunc tincidunt</a></li>

      <li class="ui-state-default ui-corner-top"><a href="#tabs-2">Proin dolor</a></li>

   <div class="**ui-tabs-panel** ui-widget-content ui-corner-bottom" id="tabs-1">

      <p>Tab one content goes here.</p>

   </div>

    ...

</div>

**Note: This is a sample of markup generated by the tabs plugin, not markup you should use to create a tabs. The only markup needed for that is**

**<div id="tabs">**

**<ul>**

**<li><a href="#tabs-1">Nunc tincidunt</a></li>**

**<li><a href="#tabs-2">Proin dolor</a></li>**

**<li><a href="#tabs-3">Aenean lacinia</a></li>**

**</ul>**

**<div id="tabs-1">**

**<p>Tab 1 content</p>**

**</div>**

**<div id="tabs-2">**

**<p>Tab 2 content</p>**

**</div>**

**<div id="tabs-3">**

**<p>Tab 3 content</p>**

**</div>**

**</div>.**