* Overview
* Options
* Events
* Methods
* Theming

jQuery UI Resizable

## Overview

The jQuery UI Resizable plugin makes selected elements resizable (meaning they have draggable resize handles). You can specify one or more handles as well as min and max width and height.

All callbacks (start,stop,resize) receive two arguments: The original browser event and a prepared ui object. The ui object has the following fields:

* **ui.helper** - a jQuery object containing the helper element
* **ui.originalPosition** - {top, left} before resizing started
* **ui.originalSize** - {width, height} before resizing started
* **ui.position** - {top, left} current position
* **ui.size** - {width, height} current size

### Dependencies

* UI Core
* UI Widget
* UI Mouse

### Example

* Demo
* View Source

A simple jQuery UI Resizable.

$("#resizable").resizable();

<!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>  
 <style type="text/css">  
 #resizable { width: 100px; height: 100px; background: silver; }  
 </style>  
 <script>  
 $(document).ready(function() {  
 $("#resizable").resizable();  
 });  
 </script>  
</head>  
<body style="font-size:62.5%;">  
   
<div id="resizable"></div>  
  
</body>  
</html>

## Options

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

#### **alsoResize**Type: Selector, jQuery, Element Default: false Resize these elements synchronous when resizing. Code examplesInitialize a resizable with the alsoResize option specified. $( ".selector" ).resizable({ alsoResize: '.other' });Get or set the alsoResize option, after init. //getter var alsoResize = $( ".selector" ).resizable( "option", "alsoResize" ); //setter $( ".selector" ).resizable( "option", "alsoResize", '.other' );

#### **animate**Type: Boolean Default: false Animates to the final size after resizing. Code examplesInitialize a resizable with the animate option specified. $( ".selector" ).resizable({ animate: true });Get or set the animate option, after init. //getter var animate = $( ".selector" ).resizable( "option", "animate" ); //setter $( ".selector" ).resizable( "option", "animate", true );

#### **animateDuration**Type: Integer, String Default: 'slow' Duration time for animating, in milliseconds. Other possible values: 'slow', 'normal', 'fast'. Code examplesInitialize a resizable with the animateDuration option specified. $( ".selector" ).resizable({ animateDuration: 500 });Get or set the animateDuration option, after init. //getter var animateDuration = $( ".selector" ).resizable( "option", "animateDuration" ); //setter $( ".selector" ).resizable( "option", "animateDuration", 500 );

#### **animateEasing**Type: String Default: 'swing' Easing effect for animating. Code examplesInitialize a resizable with the animateEasing option specified. $( ".selector" ).resizable({ animateEasing: 'swing' });Get or set the animateEasing option, after init. //getter var animateEasing = $( ".selector" ).resizable( "option", "animateEasing" ); //setter $( ".selector" ).resizable( "option", "animateEasing", 'swing' );

#### **aspectRatio**Type: Boolean, Float Default: false If set to true, resizing is constrained by the original aspect ratio. Otherwise a custom aspect ratio can be specified, such as 9 / 16, or 0.5. Code examplesInitialize a resizable with the aspectRatio option specified. $( ".selector" ).resizable({ aspectRatio: .75 });Get or set the aspectRatio option, after init. //getter var aspectRatio = $( ".selector" ).resizable( "option", "aspectRatio" ); //setter $( ".selector" ).resizable( "option", "aspectRatio", .75 );

#### **autoHide**Type: Boolean Default: false If set to true, automatically hides the handles except when the mouse hovers over the element. Code examplesInitialize a resizable with the autoHide option specified. $( ".selector" ).resizable({ autoHide: true });Get or set the autoHide option, after init. //getter var autoHide = $( ".selector" ).resizable( "option", "autoHide" ); //setter $( ".selector" ).resizable( "option", "autoHide", true );

#### **cancel**Type: Selector Default: ':input,option' Prevents resizing if you start on elements matching the selector. Code examplesInitialize a resizable with the cancel option specified. $( ".selector" ).resizable({ cancel: ':input,option' });Get or set the cancel option, after init. //getter var cancel = $( ".selector" ).resizable( "option", "cancel" ); //setter $( ".selector" ).resizable( "option", "cancel", ':input,option' );

#### **containment**Type: String, Element, Selector Default: false Constrains resizing to within the bounds of the specified element. Possible values: 'parent', 'document', a DOMElement, or a Selector. Code examplesInitialize a resizable with the containment option specified. $( ".selector" ).resizable({ containment: 'parent' });Get or set the containment option, after init. //getter var containment = $( ".selector" ).resizable( "option", "containment" ); //setter $( ".selector" ).resizable( "option", "containment", 'parent' );

#### **delay**Type: Integer Default: 0 Tolerance, in milliseconds, for when resizing should start. If specified, resizing will not start until after mouse is moved beyond duration. This can help prevent unintended resizing when clicking on an element. Code examplesInitialize a resizable with the delay option specified. $( ".selector" ).resizable({ delay: 20 });Get or set the delay option, after init. //getter var delay = $( ".selector" ).resizable( "option", "delay" ); //setter $( ".selector" ).resizable( "option", "delay", 20 );

#### **distance**Type: Integer Default: 1 Tolerance, in pixels, for when resizing should start. If specified, resizing will not start until after mouse is moved beyond distance. This can help prevent unintended resizing when clicking on an element. Code examplesInitialize a resizable with the distance option specified. $( ".selector" ).resizable({ distance: 20 });Get or set the distance option, after init. //getter var distance = $( ".selector" ).resizable( "option", "distance" ); //setter $( ".selector" ).resizable( "option", "distance", 20 );

#### **ghost**Type: Boolean Default: false If set to true, a semi-transparent helper element is shown for resizing. Code examplesInitialize a resizable with the ghost option specified. $( ".selector" ).resizable({ ghost: true });Get or set the ghost option, after init. //getter var ghost = $( ".selector" ).resizable( "option", "ghost" ); //setter $( ".selector" ).resizable( "option", "ghost", true );

#### **grid**Type: Array Default: false Snaps the resizing element to a grid, every x and y pixels. Array values: [x, y] Code examplesInitialize a resizable with the grid option specified. $( ".selector" ).resizable({ grid: [50, 50] });Get or set the grid option, after init. //getter var grid = $( ".selector" ).resizable( "option", "grid" ); //setter $( ".selector" ).resizable( "option", "grid", [50, 50] );

#### **handles**Type: String, Object Default: 'e, s, se' If specified as a string, should be a comma-split list of any of the following: 'n, e, s, w, ne, se, sw, nw, all'. The necessary handles will be auto-generated by the plugin. If specified as an object, the following keys are supported: { n, e, s, w, ne, se, sw, nw }. The value of any specified should be a jQuery selector matching the child element of the resizable to use as that handle. If the handle is not a child of the resizable, you can pass in the DOMElement or a valid jQuery object directly. Code examplesInitialize a resizable with the handles option specified. $( ".selector" ).resizable({ handles: 'n, e, s, w' });Get or set the handles option, after init. //getter var handles = $( ".selector" ).resizable( "option", "handles" ); //setter $( ".selector" ).resizable( "option", "handles", 'n, e, s, w' );

#### **helper**Type: String Default: false This is the css class that will be added to a proxy element to outline the resize during the drag of the resize handle. Once the resize is complete, the original element is sized. Code examplesInitialize a resizable with the helper option specified. $( ".selector" ).resizable({ helper: 'ui-state-highlight' });Get or set the helper option, after init. //getter var helper = $( ".selector" ).resizable( "option", "helper" ); //setter $( ".selector" ).resizable( "option", "helper", 'ui-state-highlight' );

#### **maxHeight**Type: Integer Default: null This is the maximum height the resizable should be allowed to resize to. Code examplesInitialize a resizable with the maxHeight option specified. $( ".selector" ).resizable({ maxHeight: 300 });Get or set the maxHeight option, after init. //getter var maxHeight = $( ".selector" ).resizable( "option", "maxHeight" ); //setter $( ".selector" ).resizable( "option", "maxHeight", 300 );

#### **maxWidth**Type: Integer Default: null This is the maximum width the resizable should be allowed to resize to. Code examplesInitialize a resizable with the maxWidth option specified. $( ".selector" ).resizable({ maxWidth: 250 });Get or set the maxWidth option, after init. //getter var maxWidth = $( ".selector" ).resizable( "option", "maxWidth" ); //setter $( ".selector" ).resizable( "option", "maxWidth", 250 );

#### **minHeight**Type: Integer Default: 10 This is the minimum height the resizable should be allowed to resize to. Code examplesInitialize a resizable with the minHeight option specified. $( ".selector" ).resizable({ minHeight: 150 });Get or set the minHeight option, after init. //getter var minHeight = $( ".selector" ).resizable( "option", "minHeight" ); //setter $( ".selector" ).resizable( "option", "minHeight", 150 );

#### **minWidth**Type: Integer Default: 10 This is the minimum width the resizable should be allowed to resize to. Code examplesInitialize a resizable with the minWidth option specified. $( ".selector" ).resizable({ minWidth: 75 });Get or set the minWidth option, after init. //getter var minWidth = $( ".selector" ).resizable( "option", "minWidth" ); //setter $( ".selector" ).resizable( "option", "minWidth", 75 );

## Events

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

#### **start**Type: resizestart This event is triggered at the start of a resize operation. Code examplesSupply a callback function to handle the start event as an init option. $( ".selector" ).resizable({ start: function(event, ui) { ... } });Bind to the start event by type: resizestart. $( ".selector" ).bind( "resizestart", function(event, ui) { ... });

#### **resize**Type: resize This event is triggered during the resize, on the drag of the resize handler. Code examplesSupply a callback function to handle the resize event as an init option. $( ".selector" ).resizable({ resize: function(event, ui) { ... } });Bind to the resize event by type: resize. $( ".selector" ).bind( "resize", function(event, ui) { ... });

#### **stop**Type: resizestop This event is triggered at the end of a resize operation. Code examplesSupply a callback function to handle the stop event as an init option. $( ".selector" ).resizable({ stop: function(event, ui) { ... } });Bind to the stop event by type: resizestop. $( ".selector" ).bind( "resizestop", function(event, ui) { ... });

## Methods

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

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

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

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

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

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

## Theming

The jQuery UI Resizable 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.resizable.css stylesheet that can be modified. These classes are highlighed in bold below.

### Sample markup with jQuery UI CSS Framework classes

<div class="**ui-resizable**">

   <div style="-moz-user-select: none;" unselectable="on" class="**ui-resizable-handle ui-resizable-e**"></div>

   <div style="-moz-user-select: none;" unselectable="on" class="**ui-resizable-handle ui-resizable-s**"></div>

   <div unselectable="on" style="z-index: 1001; -moz-user-select: none;" class="**ui-resizable-handle ui-resizable-se** ui-icon ui-icon-gripsmall-diagonal-se"></div>

</div>

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