

# Resizable Widget

Categories: [Interactions \(//api.jqueryui.com/1.8/category/interactions/\)](http://api.jqueryui.com/1.8/category/interactions/)

Resizable Widget

version added: 1.0

Description: *Change the size of an element using the mouse.*

QuickNav

Options

alsoResize

animate

animateDuration

animateEasing

aspectRatio

autoHide

cancel

containment

delay

disabled

distance

ghost

grid

handles

helper

maxHeight

maxWidth

minHeight

minWidth

Methods

destroy

disable

enable

option

widget

Events

create

resize

start

stop

Examples

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.

Additional Notes:

This widget requires some functional CSS, otherwise it won't work. If you build a custom theme, use the widget's specific CSS file as a starting point.

Options

alsoResize

**Type:** [Selector \(http://api.jquery.com/Types/#Selector\)](http://api.jquery.com/Types/#Selector) or [jQuery \(http://api.jquery.com/Types/#jQuery\)](http://api.jquery.com/Types/#jQuery) or [Element \(http://api.jquery.com/Types/#Element\)](http://api.jquery.com/Types/#Element)

**Default:** false

One or more elements to resize synchronously with the resizable element.

---

## animate

**Type:** [Boolean \(http://api.jquery.com/Types/#Boolean\)](http://api.jquery.com/Types/#Boolean)

**Default:** false

Animates to the final size after resizing.

---

## animateDuration

**Type:** [Number \(http://api.jquery.com/Types/#Number\)](http://api.jquery.com/Types/#Number) or [String \(http://api.jquery.com/Types/#String\)](http://api.jquery.com/Types/#String)

**Default:** "slow"

How long to animate when using the [animate](#) option.

**Multiple types supported:**

**Number:** Duration in milliseconds.

**String:** A named duration, such as "slow" or "fast".

---

## animateEasing

**Type:** [String \(http://api.jquery.com/Types/#String\)](http://api.jquery.com/Types/#String)

**Default:** "swing"

Which [easing \(/easings/\)](#) to apply when using the [animate](#) option.

---

## aspectRatio

**Type:** [Boolean \(http://api.jquery.com/Types/#Boolean\)](http://api.jquery.com/Types/#Boolean) or [Number \(http://api.jquery.com/Types/#Number\)](http://api.jquery.com/Types/#Number)

**Default:** false

Whether the element should be constrained to a specific aspect ratio.

**Multiple types supported:**

**Boolean:** When set to `true`, the element will maintain its original aspect ratio.

**Number:** Force the element to maintain a specific aspect ratio during resizing.

---

## autoHide

**Type:** [Boolean \(http://api.jquery.com/Types/#Boolean\)](http://api.jquery.com/Types/#Boolean)

**Default:** false

Whether the handles should hide when the user is not hovering over the element.

---

## cancel

**Type:** [Selector \(http://api.jquery.com/Types/#Selector\)](http://api.jquery.com/Types/#Selector)

**Default:** "input,textarea,button,select,option"

Prevents resizing from starting on specified elements.

---

## containment

**Type:** [Selector \(http://api.jquery.com/Types/#Selector\)](http://api.jquery.com/Types/#Selector) or [Element \(http://api.jquery.com/Types/#Element\)](http://api.jquery.com/Types/#Element) or [String \(http://api.jquery.com/Types/#String\)](http://api.jquery.com/Types/#String)

**Default:** false

Constrains resizing to within the bounds of the specified element or region.

**Multiple types supported:**

**Selector:** The resizable element will be contained to the bounding box of the first element found by the selector. If no element is found, no containment will be set.

**Element:** The resizable element will be contained to the bounding box of this element.

**String:** Possible values: "parent" and "document".

---

## delay

**Type:** [Number \(http://api.jquery.com/Types/#Number\)](http://api.jquery.com/Types/#Number)

**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.

---

## disabled

**Type:** [Boolean \(http://api.jquery.com/Types/#Boolean\)](http://api.jquery.com/Types/#Boolean)

**Default:** false

Disables the resizable if set to true.

---

## distance

**Type:** Number (<http://api.jquery.com/Types/#Number>)

**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.

---

## ghost

**Type:** Boolean (<http://api.jquery.com/Types/#Boolean>)

**Default:** false

If set to `true`, a semi-transparent helper element is shown for resizing.

---

## grid

**Type:** Array (<http://api.jquery.com/Types/#Array>)

**Default:** false

Snaps the resizing element to a grid, every x and y pixels. Array values: [ x, y ].

---

## handles

**Type:** String (<http://api.jquery.com/Types/#String>) or Object (<http://api.jquery.com/Types/#Object>)

**Default:** "e, s, se"

Which handles can be used for resizing.

### Multiple types supported:

**String:** A comma delimited 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.

### 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.

*Note: When generating your own handles, each handle must have the `ui-resizable-handle` class, as well as the appropriate `ui-resizable-{direction}` class, .e.g., `ui-resizable-s`.*

---

## helper

**Type:** String (<http://api.jquery.com/Types/#String>)

**Default:** false

A class name 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.

---

## maxHeight

**Type:** [Number \(http://api.jquery.com/Types/#Number\)](http://api.jquery.com/Types/#Number)

**Default:** null

The maximum height the resizable should be allowed to resize to.

---

## maxWidth

**Type:** [Number \(http://api.jquery.com/Types/#Number\)](http://api.jquery.com/Types/#Number)

**Default:** null

The maximum width the resizable should be allowed to resize to.

---

## minHeight

**Type:** [Number \(http://api.jquery.com/Types/#Number\)](http://api.jquery.com/Types/#Number)

**Default:** 10

The minimum height the resizable should be allowed to resize to.

---

## minWidth

**Type:** [Number \(http://api.jquery.com/Types/#Number\)](http://api.jquery.com/Types/#Number)

**Default:** 10

The minimum width the resizable should be allowed to resize to.

---

## Methods

### destroy()

**Returns:** [jQuery \(http://api.jquery.com/Types/#jQuery\)](http://api.jquery.com/Types/#jQuery) *(plugin only)*  
[\(http://learn.jquery.com/jquery-ui/widget-factory/widget-method-invocation/\)](http://learn.jquery.com/jquery-ui/widget-factory/widget-method-invocation/)

Removes the resizable functionality completely. This will return the element back to its pre-init state.

This method does not accept any arguments.

### Code examples:

Invoke the destroy method:

```
1 $( ".selector" ).resizable( "destroy" );
```

---

## **disable()**

**Returns:** *jQuery (<http://api.jquery.com/Types/#jQuery>) (plugin only (<http://learn.jquery.com/jquery-ui/widget-factory/widget-method-invocation/>))*

Disables the resizable.

This method does not accept any arguments.

### **Code examples:**

Invoke the disable method:

```
1 $( ".selector" ).resizable( "disable" );
```

---

## **enable()**

**Returns:** *jQuery (<http://api.jquery.com/Types/#jQuery>) (plugin only (<http://learn.jquery.com/jquery-ui/widget-factory/widget-method-invocation/>))*

Enables the resizable.

This method does not accept any arguments.

### **Code examples:**

Invoke the enable method:

```
1 $( ".selector" ).resizable( "enable" );
```

---

## **option( optionName )**

**Returns:** *Object (<http://api.jquery.com/Types/#Object>)*

Gets the value currently associated with the specified `optionName`.

### **optionName**

Type: String (<http://api.jquery.com/Types/#String>)

The name of the option to get.

### **Code examples:**

Invoke the method:

```
1 $( ".selector" ).resizable( "option" );
```

---

## **option()**

**Returns:** *PlainObject (<http://api.jquery.com/Types/#PlainObject>)*

Gets an object containing key/value pairs representing the current resizable options hash.

This signature does not accept any arguments.

#### Code examples:

Invoke the method:

```
1 $( ".selector" ).resizable( "option" );
```

---

### **option( optionName, value )**

**Returns:** *jQuery* (<http://api.jquery.com/Types/#jQuery>) (plugin only  
(<http://learn.jquery.com/jquery-ui/widget-factory/widget-method-invocation/>))

Sets the value of the resizable option associated with the specified `optionName`.

#### **optionName**

Type: String (<http://api.jquery.com/Types/#String>)

The name of the option to set.

#### **value**

Type: Object (<http://api.jquery.com/Types/#Object>)

A value to set for the option.

#### Code examples:

Invoke the method:

```
1 $( ".selector" ).resizable( "option" );
```

---

### **option( options )**

**Returns:** *jQuery* (<http://api.jquery.com/Types/#jQuery>) (plugin only  
(<http://learn.jquery.com/jquery-ui/widget-factory/widget-method-invocation/>))

Sets one or more options for the resizable.

#### **options**

Type: Object (<http://api.jquery.com/Types/#Object>)

A map of option-value pairs to set.

#### Code examples:

Invoke the method:

```
1 $( ".selector" ).resizable( "option" );
```

---

### **widget()**

**Returns:** *jQuery* (<http://api.jquery.com/Types/#jQuery>)

Returns a `jQuery` object containing the resizable element.

This method does not accept any arguments.

### Code examples:

Invoke the widget method:

```
1 $( ".selector" ).resizable( "widget" );
```

## Events

### create( event, ui )

**Type:** *resizablecreate*

Triggered when the resizable is created.

**event**

Type: [Event \(http://api.jquery.com/Types/#Event\)](http://api.jquery.com/Types/#Event)

**ui**

Type: [Object \(http://api.jquery.com/Types/#Object\)](http://api.jquery.com/Types/#Object)

*Note: The `ui` object is empty but included for consistency with other events.*

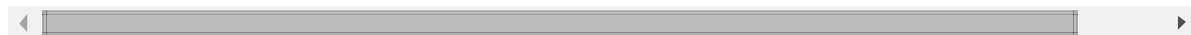
### Code examples:

Initialize the resizable with the create callback specified:

```
1 $( ".selector" ).resizable({  
2   create: function( event, ui ) {}  
3 });
```

Bind an event listener to the resizablecreate event:

```
1 $( ".selector" ).on( "resizablecreate", function( event, ui ) {
```



### resize( event, ui )

**Type:** *resizableresize*

This event is triggered during the resize, on the drag of the resize handler.

**event**

Type: [Event \(http://api.jquery.com/Types/#Event\)](http://api.jquery.com/Types/#Event)

**ui**

Type: [Object \(http://api.jquery.com/Types/#Object\)](http://api.jquery.com/Types/#Object)

**element**



Type: [jQuery \(http://api.jquery.com/Types/#jQuery\)](http://api.jquery.com/Types/#jQuery)

The jQuery object representing the element to be resized

#### **helper**

Type: [jQuery \(http://api.jquery.com/Types/#jQuery\)](http://api.jquery.com/Types/#jQuery)

The jQuery object representing the helper that's being resized

#### **originalElement**

Type: [jQuery \(http://api.jquery.com/Types/#jQuery\)](http://api.jquery.com/Types/#jQuery)

The jQuery object representing the original element before it is wrapped

#### **originalPosition**

Type: [Object \(http://api.jquery.com/Types/#Object\)](http://api.jquery.com/Types/#Object)

The position represented as { left, top } before the resizable is resized

#### **originalSize**

Type: [Object \(http://api.jquery.com/Types/#Object\)](http://api.jquery.com/Types/#Object)

The size represented as { width, height } before the resizable is resized

#### **position**

Type: [Object \(http://api.jquery.com/Types/#Object\)](http://api.jquery.com/Types/#Object)

The current position represented as { left, top }

#### **size**

Type: [Object \(http://api.jquery.com/Types/#Object\)](http://api.jquery.com/Types/#Object)

The current size represented as { width, height }

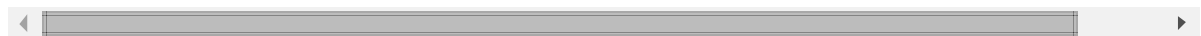
### **Code examples:**

Initialize the resizable with the resize callback specified:

```
1 $( ".selector" ).resizable({  
2   resize: function( event, ui ) {}  
3 });
```

Bind an event listener to the `resizableresize` event:

```
1 $( ".selector" ).on( "resizableresize", function( event, ui )
```



---

## **start( event, ui )**

**Type:** *resizablestart*

This event is triggered at the start of a resize operation.

#### **event**

Type: [Event \(http://api.jquery.com/Types/#Event\)](http://api.jquery.com/Types/#Event)

**ui**

Type: [Object](http://api.jquery.com/Types/#Object) (<http://api.jquery.com/Types/#Object>)

**element**

Type: [jQuery](http://api.jquery.com/Types/#jQuery) (<http://api.jquery.com/Types/#jQuery>)

The jQuery object representing the element to be resized

**helper**

Type: [jQuery](http://api.jquery.com/Types/#jQuery) (<http://api.jquery.com/Types/#jQuery>)

The jQuery object representing the helper that's being resized

**originalElement**

Type: [jQuery](http://api.jquery.com/Types/#jQuery) (<http://api.jquery.com/Types/#jQuery>)

The jQuery object representing the original element before it is wrapped

**originalPosition**

Type: [Object](http://api.jquery.com/Types/#Object) (<http://api.jquery.com/Types/#Object>)

The position represented as { left, top } before the resizable is resized

**originalSize**

Type: [Object](http://api.jquery.com/Types/#Object) (<http://api.jquery.com/Types/#Object>)

The size represented as { width, height } before the resizable is resized

**position**

Type: [Object](http://api.jquery.com/Types/#Object) (<http://api.jquery.com/Types/#Object>)

The current position represented as { left, top }

**size**

Type: [Object](http://api.jquery.com/Types/#Object) (<http://api.jquery.com/Types/#Object>)

The current size represented as { width, height }

### Code examples:

Initialize the resizable with the start callback specified:

```
1 $( ".selector" ).resizable({  
2   start: function( event, ui ) {}  
3 });
```

Bind an event listener to the resizablestart event:

```
1 $( ".selector" ).on( "resizablestart", function( event, ui ) {
```



---

**stop( event, ui )**

**Type:** *resizablestop*

This event is triggered at the end of a resize operation.

**event**

Type: [Event \(http://api.jquery.com/Types/#Event\)](http://api.jquery.com/Types/#Event)

**ui**

Type: [Object \(http://api.jquery.com/Types/#Object\)](http://api.jquery.com/Types/#Object)

**element**

Type: [jQuery \(http://api.jquery.com/Types/#jQuery\)](http://api.jquery.com/Types/#jQuery)

The jQuery object representing the element to be resized

**helper**

Type: [jQuery \(http://api.jquery.com/Types/#jQuery\)](http://api.jquery.com/Types/#jQuery)

The jQuery object representing the helper that's being resized

**originalElement**

Type: [jQuery \(http://api.jquery.com/Types/#jQuery\)](http://api.jquery.com/Types/#jQuery)

The jQuery object representing the original element before it is wrapped

**originalPosition**

Type: [Object \(http://api.jquery.com/Types/#Object\)](http://api.jquery.com/Types/#Object)

The position represented as { left, top } before the resizable is resized

**originalSize**

Type: [Object \(http://api.jquery.com/Types/#Object\)](http://api.jquery.com/Types/#Object)

The size represented as { width, height } before the resizable is resized

**position**

Type: [Object \(http://api.jquery.com/Types/#Object\)](http://api.jquery.com/Types/#Object)

The current position represented as { left, top }

**size**

Type: [Object \(http://api.jquery.com/Types/#Object\)](http://api.jquery.com/Types/#Object)

The current size represented as { width, height }

**Code examples:**

Initialize the resizable with the stop callback specified:

```
1 $( ".selector" ).resizable({  
2   stop: function( event, ui ) {}  
3 });
```

Bind an event listener to the resizablestop event:

```
1 $( ".selector" ).on( "resizablestop", function( event, ui ) {}
```



## Example:

A simple jQuery UI Resizable.

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>resizable demo</title>
6   <link rel="stylesheet" href="//code.jquery.com/ui/1.8.24/themes/base
7   <style>
8     #resizable {
9       width: 100px;
10      height: 100px;
11      background: #ccc;
12    }
13    </style>
14    <script src="//code.jquery.com/jquery-1.8.2.js"></script>
15    <script src="//code.jquery.com/ui/1.8.24/jquery-ui.js"></script>
16  </head>
17  <body>
18    <div id="resizable"></div>
19
20    <script>
21      $( "#resizable" ).resizable();
22    </script>
23
24  </body>
25 </html>
```

### Demo:

