* Overview
* Options
* Events
* Methods
* Theming

jQuery UI Dialog

## Overview

A dialog is a floating window that contains a title bar and a content area. The dialog window can be moved, resized and closed with the 'x' icon by default.

If the content length exceeds the maximum height, a scrollbar will automatically appear.

A bottom button bar and semi-transparent modal overlay layer are common options that can be added.

A call to $(foo).dialog() will initialize a dialog instance and will auto-open the dialog by default. If you want to reuse a dialog, the easiest way is to disable the "auto-open" option with: $(foo).dialog({ autoOpen: false }) and open it with $(foo).dialog('open'). To close it, use $(foo).dialog('close'). A more in-depth explanation with a full demo is available on [the Nemikor blog](http://blog.nemikor.com/2009/04/08/basic-usage-of-the-jquery-ui-dialog/).

### Dependencies

* UI Core
* UI Position
* UI Widget
* UI Mouse (Optional; only needed if using UI Draggable or UI Resizable)
* UI Draggable (Optional)
* UI Resizable (Optional)

### Example

* Demo
* View Source

A simple jQuery UI Dialog.

$("#dialog").dialog();

<!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() {  
 $("#dialog").dialog();  
 });  
 </script>  
</head>  
<body style="font-size:62.5%;">  
   
<div id="dialog" title="Dialog Title">I'm in a dialog</div>  
  
</body>  
</html>

## Options

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

#### **autoOpen**Type: Boolean Default: true When *autoOpen* is *true* the dialog will open automatically when *dialog* is called. If *false* it will stay hidden until *.dialog("open")* is called on it. Code examplesInitialize a dialog with the autoOpen option specified. $( ".selector" ).dialog({ autoOpen: false });Get or set the autoOpen option, after init. //getter var autoOpen = $( ".selector" ).dialog( "option", "autoOpen" ); //setter $( ".selector" ).dialog( "option", "autoOpen", false );

#### **buttons**Type: Object Default: { } Specifies which buttons should be displayed on the dialog. The property key is the text of the button. The value is the callback function for when the button is clicked. The context of the callback is the dialog element; if you need access to the button, it is available as the target of the event object. Code examplesInitialize a dialog with the buttons option specified. $( ".selector" ).dialog({ buttons: { "Ok": function() { $(this).dialog("close"); } } });Get or set the buttons option, after init. //getter var buttons = $( ".selector" ).dialog( "option", "buttons" ); //setter $( ".selector" ).dialog( "option", "buttons", { "Ok": function() { $(this).dialog("close"); } } );

#### **buttons**Type: Array Default: [ ] Specifies which buttons should be displayed on the dialog. Each element of the array must be an Object defining the properties to set on the button. Code examplesInitialize a dialog with the buttons option specified. $( ".selector" ).dialog({ buttons: [ { text: "Ok", click: function() { $(this).dialog("close"); } } ] });Get or set the buttons option, after init. //getter var buttons = $( ".selector" ).dialog( "option", "buttons" ); //setter $( ".selector" ).dialog( "option", "buttons", [ { text: "Ok", click: function() { $(this).dialog("close"); } } ] );

#### **closeOnEscape**Type: Boolean Default: true Specifies whether the dialog should close when it has focus and the user presses the esacpe (ESC) key. Code examplesInitialize a dialog with the closeOnEscape option specified. $( ".selector" ).dialog({ closeOnEscape: false });Get or set the closeOnEscape option, after init. //getter var closeOnEscape = $( ".selector" ).dialog( "option", "closeOnEscape" ); //setter $( ".selector" ).dialog( "option", "closeOnEscape", false );

#### **closeText**Type: String Default: 'close' Specifies the text for the close button. Note that the close text is visibly hidden when using a standard theme. Code examplesInitialize a dialog with the closeText option specified. $( ".selector" ).dialog({ closeText: 'hide' });Get or set the closeText option, after init. //getter var closeText = $( ".selector" ).dialog( "option", "closeText" ); //setter $( ".selector" ).dialog( "option", "closeText", 'hide' );

#### **dialogClass**Type: String Default: '' The specified class name(s) will be added to the dialog, for additional theming. Code examplesInitialize a dialog with the dialogClass option specified. $( ".selector" ).dialog({ dialogClass: 'alert' });Get or set the dialogClass option, after init. //getter var dialogClass = $( ".selector" ).dialog( "option", "dialogClass" ); //setter $( ".selector" ).dialog( "option", "dialogClass", 'alert' );

#### **draggable**Type: Boolean Default: true If set to true, the dialog will be draggable will be draggable by the titlebar. Code examplesInitialize a dialog with the draggable option specified. $( ".selector" ).dialog({ draggable: false });Get or set the draggable option, after init. //getter var draggable = $( ".selector" ).dialog( "option", "draggable" ); //setter $( ".selector" ).dialog( "option", "draggable", false );

#### **height**Type: Number Default: 'auto' The height of the dialog, in pixels. Specifying 'auto' is also supported to make the dialog adjust based on its content. Code examplesInitialize a dialog with the height option specified. $( ".selector" ).dialog({ height: 530 });Get or set the height option, after init. //getter var height = $( ".selector" ).dialog( "option", "height" ); //setter $( ".selector" ).dialog( "option", "height", 530 );

#### **hide**Type: String, Object Default: null The effect to be used when the dialog is closed. Code examplesInitialize a dialog with the hide option specified. $( ".selector" ).dialog({ hide: 'slide' });Get or set the hide option, after init. //getter var hide = $( ".selector" ).dialog( "option", "hide" ); //setter $( ".selector" ).dialog( "option", "hide", 'slide' );

#### **maxHeight**Type: Number Default: false The maximum height to which the dialog can be resized, in pixels. Code examplesInitialize a dialog with the maxHeight option specified. $( ".selector" ).dialog({ maxHeight: 400 });Get or set the maxHeight option, after init. //getter var maxHeight = $( ".selector" ).dialog( "option", "maxHeight" ); //setter $( ".selector" ).dialog( "option", "maxHeight", 400 );

#### **maxWidth**Type: Number Default: false The maximum width to which the dialog can be resized, in pixels. Code examplesInitialize a dialog with the maxWidth option specified. $( ".selector" ).dialog({ maxWidth: 600 });Get or set the maxWidth option, after init. //getter var maxWidth = $( ".selector" ).dialog( "option", "maxWidth" ); //setter $( ".selector" ).dialog( "option", "maxWidth", 600 );

#### **minHeight**Type: Number Default: 150 The minimum height to which the dialog can be resized, in pixels. Code examplesInitialize a dialog with the minHeight option specified. $( ".selector" ).dialog({ minHeight: 300 });Get or set the minHeight option, after init. //getter var minHeight = $( ".selector" ).dialog( "option", "minHeight" ); //setter $( ".selector" ).dialog( "option", "minHeight", 300 );

#### **minWidth**Type: Number Default: 150 The minimum width to which the dialog can be resized, in pixels. Code examplesInitialize a dialog with the minWidth option specified. $( ".selector" ).dialog({ minWidth: 400 });Get or set the minWidth option, after init. //getter var minWidth = $( ".selector" ).dialog( "option", "minWidth" ); //setter $( ".selector" ).dialog( "option", "minWidth", 400 );

#### **modal**Type: Boolean Default: false If set to true, the dialog will have modal behavior; other items on the page will be disabled (i.e. cannot be interacted with). Modal dialogs create an overlay below the dialog but above other page elements. Code examplesInitialize a dialog with the modal option specified. $( ".selector" ).dialog({ modal: true });Get or set the modal option, after init. //getter var modal = $( ".selector" ).dialog( "option", "modal" ); //setter $( ".selector" ).dialog( "option", "modal", true );

#### **position**Type: String, Array Default: 'center' Specifies where the dialog should be displayed. Possible values: 1) a single string representing position within viewport: 'center', 'left', 'right', 'top', 'bottom'. 2) an array containing an *x,y* coordinate pair in pixel offset from left, top corner of viewport (e.g. [350,100]) 3) an array containing *x,y* position string values (e.g. ['right','top'] for top right corner). Code examplesInitialize a dialog with the position option specified. $( ".selector" ).dialog({ position: 'top' });Get or set the position option, after init. //getter var position = $( ".selector" ).dialog( "option", "position" ); //setter $( ".selector" ).dialog( "option", "position", 'top' );

#### **resizable**Type: Boolean Default: true If set to true, the dialog will be resizeable. Code examplesInitialize a dialog with the resizable option specified. $( ".selector" ).dialog({ resizable: false });Get or set the resizable option, after init. //getter var resizable = $( ".selector" ).dialog( "option", "resizable" ); //setter $( ".selector" ).dialog( "option", "resizable", false );

#### **show**Type: String, Object Default: null The effect to be used when the dialog is opened. Code examplesInitialize a dialog with the show option specified. $( ".selector" ).dialog({ show: 'slide' });Get or set the show option, after init. //getter var show = $( ".selector" ).dialog( "option", "show" ); //setter $( ".selector" ).dialog( "option", "show", 'slide' );

#### **stack**Type: Boolean Default: true Specifies whether the dialog will stack on top of other dialogs. This will cause the dialog to move to the front of other dialogs when it gains focus. Code examplesInitialize a dialog with the stack option specified. $( ".selector" ).dialog({ stack: false });Get or set the stack option, after init. //getter var stack = $( ".selector" ).dialog( "option", "stack" ); //setter $( ".selector" ).dialog( "option", "stack", false );

#### **title**Type: String Default: '' Specifies the title of the dialog. Any valid HTML may be set as the title. The title can also be specified by the title attribute on the dialog source element. Code examplesInitialize a dialog with the title option specified. $( ".selector" ).dialog({ title: 'Dialog Title' });Get or set the title option, after init. //getter var title = $( ".selector" ).dialog( "option", "title" ); //setter $( ".selector" ).dialog( "option", "title", 'Dialog Title' );

#### **width**Type: Number Default: 300 The width of the dialog, in pixels. Code examplesInitialize a dialog with the width option specified. $( ".selector" ).dialog({ width: 460 });Get or set the width option, after init. //getter var width = $( ".selector" ).dialog( "option", "width" ); //setter $( ".selector" ).dialog( "option", "width", 460 );

#### **zIndex**Type: Integer Default: 1000 The starting z-index for the dialog. Code examplesInitialize a dialog with the zIndex option specified. $( ".selector" ).dialog({ zIndex: 3999 });Get or set the zIndex option, after init. //getter var zIndex = $( ".selector" ).dialog( "option", "zIndex" ); //setter $( ".selector" ).dialog( "option", "zIndex", 3999 );

## Events

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

#### **beforeClose**Type: dialogbeforeclose This event is triggered when a dialog attempts to close. If the beforeClose event handler (callback function) returns false, the close will be prevented. Code examplesSupply a callback function to handle the beforeClose event as an init option. $( ".selector" ).dialog({ beforeClose: function(event, ui) { ... } });Bind to the beforeClose event by type: dialogbeforeclose. $( ".selector" ).bind( "dialogbeforeclose", function(event, ui) { ... });

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

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

#### **dragStart**Type: dialogdragstart This event is triggered at the beginning of the dialog being dragged. Code examplesSupply a callback function to handle the dragStart event as an init option. $( ".selector" ).dialog({ dragStart: function(event, ui) { ... } });Bind to the dragStart event by type: dialogdragstart. $( ".selector" ).bind( "dialogdragstart", function(event, ui) { ... });

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

#### **dragStop**Type: dialogdragstop This event is triggered after the dialog has been dragged. Code examplesSupply a callback function to handle the dragStop event as an init option. $( ".selector" ).dialog({ dragStop: function(event, ui) { ... } });Bind to the dragStop event by type: dialogdragstop. $( ".selector" ).bind( "dialogdragstop", function(event, ui) { ... });

#### **resizeStart**Type: dialogresizestart This event is triggered at the beginning of the dialog being resized. Code examplesSupply a callback function to handle the resizeStart event as an init option. $( ".selector" ).dialog({ resizeStart: function(event, ui) { ... } });Bind to the resizeStart event by type: dialogresizestart. $( ".selector" ).bind( "dialogresizestart", function(event, ui) { ... });

#### **resize**Type: dialogresize This event is triggered when the dialog is resized. [demo](http://www.jsfiddle.net/Jp7TM/18/) Code examplesSupply a callback function to handle the resize event as an init option. $( ".selector" ).dialog({ resize: function(event, ui) { ... } });Bind to the resize event by type: dialogresize. $( ".selector" ).bind( "dialogresize", function(event, ui) { ... });

#### **resizeStop**Type: dialogresizestop This event is triggered after the dialog has been resized. Code examplesSupply a callback function to handle the resizeStop event as an init option. $( ".selector" ).dialog({ resizeStop: function(event, ui) { ... } });Bind to the resizeStop event by type: dialogresizestop. $( ".selector" ).bind( "dialogresizestop", function(event, ui) { ... });

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

## Methods

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

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

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

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

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

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

### **close**Signature: .dialog( "close" ) Close the dialog.

### **isOpen**Signature: .dialog( "isOpen" ) Returns true if the dialog is currently open.

### **moveToTop**Signature: .dialog( "moveToTop" ) Move the dialog to the top of the dialogs stack.

### **open**Signature: .dialog( "open" ) Open the dialog.

## Theming

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

### Sample markup with jQuery UI CSS Framework classes

<div class="**ui-dialog** ui-widget ui-widget-content ui-corner-all ui-draggable ui-resizable">

   <div class="**ui-dialog-titlebar** ui-widget-header ui-corner-all ui-helper-clearfix">

      <span id="**ui-dialog-title-dialog**" class="ui-dialog-title">Dialog title</span>

      <a class="**ui-dialog-titlebar-close** ui-corner-all" href="#"><span class="ui-icon ui-icon-closethick">close</span></a>

   </div>

   <div style="height: 200px; min-height: 109px; width: auto;" class="**ui-dialog-content** ui-widget-content" id="dialog">

      <p>Dialog content goes here.</p>

   </div>

</div>

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