# JQUERYUI - DIALOG

Dialog boxes are one of the nice ways of presenting information on an HTML page. A dialog box is a floating window with a title and content area. This window can be moved, resized, and of course, closed using "X" icon by default.

jQueryUI provides **dialog** method that transforms the HTML code written on the page into HTML code to display a dialog box.

## Syntax

The **dialog** method can be used in two forms:

- $(selector, context).dialog (options) Method
- $(selector, context).dialog ("action", [params]) Method

## $(selector, context).dialog (options) Method

The *dialog* $options$ method declares that an HTML element can be administered in the form of a dialog box. The *options* parameter is an object that specifies the appearance and behavior of that window.

## Syntax

```
$(selector, context).dialog(options);
```

You can provide one or more options at a time using Javascript object. If there are more than one options to be provided then you will separate them using a comma as follows:

```
$(selector, context).dialog({option1: value1, option2: value2..... });
```

Following table lists the different *options* that can be used with this method:

| Option | Description |
|---|---|
| appendTo | If this option is set to **false**, it will prevent the **ui-draggable** class from being added in the list of selected DOM elements. By default its value is **true**. |
| autoOpen | This option unless set to *false*, the dialog box is opened upon creation. When *false*, the dialog box will be opened upon a call to dialog$'open'$. By default its value is **true**. |
| buttons | This option adds buttons in the dialog box. These are listed as objects, and each property is the text on the button. The value is a callback function called when the user clicks the button. By default its value is **{}**. |
| closeOnEscape | Unless this option set to *false*, the dialog box will be closed when the user presses the Escape key while the dialog box has focus. By default its value is **true**. |
| closeText | This option contains text to replace the default of Close for the close button. By |

| | |
|---|---|
| | default its value is **"close"**. |
| dialogClass | If this option is set to **false**, it will prevent the **ui-draggable** class from being added in the list of selected DOM elements. By default its value is **""**. |
| draggable | Unless you this option to **false**, dialog box will be draggable by clicking and dragging the title bar. By default its value is **true**. |
| height | This option sets the height of the dialog box. By default its value is **"auto"**. |
| hide | This option is used to set the effect to be used when the dialog box is closed. By default its value is **null**. |
| maxHeight | This option sets maximum height, in pixels, to which the dialog box can be resized. By default its value is **false**. |
| maxWidth | This option sets the maximum width to which the dialog can be resized, in pixels. By default its value is **false**. |
| minHeight | This option is the minimum height, in pixels, to which the dialog box can be resized. By default its value is **150**. |
| minWidth | This option is the minimum width, in pixels, to which the dialog box can be resized. By default its value is **150**. |
| modal | If this option is 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. By default its value is **false**. |
| position | This option specifies the initial position of the dialog box. Can be one of the predefined positions: *center thedefault, left, right, top, or bottom*. Can also be a 2-element array with the left and top values *inpixels* as [left,top], or text positions such as ['right','top']. By default its value is **{ my: "center", at: "center", of: window }**. |
| resizable | This option unless set to **false**, the dialog box is resizable in all directions. By default its value is **true**. |
| show | This option is an effect to be used when the dialog box is being opened. By default its value is **null**. |
| title | This option specifies the text to appear in the title bar of the dialog box. By default its value is **null**. |

| | |
|---|---|
| width | This option specifies the width of the dialog box in pixels. By default its value is **300**. |

Following section will show you few working examples of dialog functionality.

## Default functionality

The following example demonstrates a simple example of dialog functionality passing no parameters to the **dialog** method .

```
<!doctype html>
<html lang="en">
   <head>
      <meta charset="utf-8">
      <title>jQuery UI Dialog functionality</title>
      <link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css"
rel="stylesheet">
      <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
      <script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
      <!-- CSS -->
      <style>
         .ui-widget-header,.ui-state-default, ui-button{
            background:#b9cd6d;
            border: 1px solid #b9cd6d;
            color: #FFFFFF;
            font-weight: bold;
         }
      </style>
      <!-- Javascript -->
      <script>
         $(function() {
            $( "#dialog-1" ).dialog({
               autoOpen: false,
            });
            $( "#opener" ).click(function() {
               $( "#dialog-1" ).dialog( "open" );
            });
         });
      </script>
   </head>
   <body>
      <!-- HTML -->
      <div >This my first jQuery UI Dialog!</div>
      <button >Open Dialog</button>
   </body>
</html>
```

Let's save above code in an HTML file **dialogexample.htm** and open it in a standard browser which supports javascript, you must see the following output. Now you can play with the result:

This my first jQuery UI Dialog!

## Use of buttons, title and position

The following example demonstrates the usage of three options **buttons**, **title** and **position** in the dialog widget of JqueryUI.

```
<!doctype html>
<html lang="en">
   <head>
      <meta charset="utf-8">
      <title>jQuery UI Dialog functionality</title>
      <link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css"
rel="stylesheet">
      <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
      <script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
      <!-- CSS -->
      <style>
         .ui-widget-header,.ui-state-default, ui-button{
            background:#b9cd6d;
            border: 1px solid #b9cd6d;
            color: #FFFFFF;
            font-weight: bold;
         }
      </style>
      <!-- Javascript -->
      <script>
         $(function() {
            $( "#dialog-2" ).dialog({
               autoOpen: false,
               buttons: {
                  OK: function() {$(this).dialog("close");}
               },
               title: "Success",
               position: {
                  my: "left center",
                  at: "left center"
               }
            });
            $( "#opener-2" ).click(function() {
               $( "#dialog-2" ).dialog( "open" );
            });
         });
      </script>
   </head>
   <body>
      <!-- HTML -->
      <div >This my first jQuery UI Dialog!</div>
      <button >Open Dialog</button>
   </body>
</html>
```

Let's save above code in an HTML file **dialogexample.htm** and open it in a standard browser which supports javascript, you must see the following output. Now you can play with the result:

This my first jQuery UI Dialog!

## Use of hide, show and height

The following example demonstrates the usage of three options **hide**, **show** and **height** in the dialog widget of JqueryUI.

```
<!doctype html>
<html lang="en">
   <head>
      <meta charset="utf-8">
      <title>jQuery UI Dialog functionality</title>
      <link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css"
rel="stylesheet">
      <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
      <script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
      <!-- CSS -->
      <style>
         .ui-widget-header,.ui-state-default, ui-button{
            background:#b9cd6d;
            border: 1px solid #b9cd6d;
            color: #FFFFFF;
            font-weight: bold;
         }
      </style>
      <!-- Javascript -->
      <script>
         $(function() {
            $( "#dialog-3" ).dialog({
               autoOpen: false,
               hide: "puff",
               show : "slide",
               height: 200
            });
            $( "#opener-3" ).click(function() {
               $( "#dialog-3" ).dialog( "open" );
            });
         });
      </script>
   </head>
   <body>
      <!-- HTML -->
      <div >This my first jQuery UI Dialog!</div>
      <button >Open Dialog</button>
   </body>
</html>
```

Let's save above code in an HTML file **dialogexample.htm** and open it in a standard browser which supports javascript, you must see the following output. Now you can play with the result:

This my first jQuery UI Dialog!

## Use of modal

The following example demonstrates the usage of three options **buttons**, **title** and **position** in the dialog widget of JqueryUI.

```
<!doctype html>
<html lang="en">
   <head>
      <meta charset="utf-8">
      <title>jQuery UI Dialog functionality</title>
      <link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css"
rel="stylesheet">
      <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
      <script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
      <!-- CSS -->
      <style>
         .ui-widget-header,.ui-state-default, ui-button{
            background:#b9cd6d;
            border: 1px solid #b9cd6d;
            color: #FFFFFF;
            font-weight: bold;
         }
      </style>
      <!-- Javascript -->
      <script>
         $(function() {
            $( "#dialog-4" ).dialog({
               autoOpen: false,
               modal: true,
               buttons: {
                  OK: function() {$(this).dialog("close");}
               },
            });
            $( "#opener-4" ).click(function() {
               $( "#dialog-4" ).dialog( "open" );
            });
         });
      </script>
   </head>
   <body>
      <!-- HTML -->
      <div >This my first jQuery UI Dialog!</div>
      <button >Open Dialog</button>
      <p>
         Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod 
tempor incididunt
         ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud 
exercitation ullamco
         laboris nisi ut aliquip ex ea commodo consequat.    </p>
      <label for="textbox">Enter Name: </label>
      <input type="text">
   </body>
</html>
```

Let's save above code in an HTML file **dialogexample.htm** and open it in a standard browser which supports javascript, you must see the following output. Now you can play with the result:

This my first jQuery UI Dialog!

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Enter Name:

## $(selector, context).dialog ("action", [params]) Method

The *dialog* action, params method can perform an action on the dialog box, such as closing the box. The **action** is specified as a string in the first argument and optionally, one or more **params** can be provided based on the given action.

> *Basically, here actions are nothing but they are jQuery methods which we can use in the form of string.*

## Syntax

```
$(selector, context).dialog ("action", [params]);
```

The following table lists the actions for this method:

| Action | Description |
|---|---|
| close | This action closes the dialog box. This method does not accept any arguments. |
| destroy | This action removes the dialog box competely. This will return the element back to its pre-init state. This method does not accept any arguments. |
| isOpen | This action checks if the dialog box is open. This method does not accept any arguments. |
| moveToTop | This action positions the corresponding dialog boxes to the foreground ontopoftheothers. This method does not accept any arguments. |

| | |
|---|---|
| open | This action opens the dialog box. This method does not accept any arguments. |
| option*optionName* | This action gets the value currently associated with the specified optionName. Where *optionName* is the name of the option to get. |
| option | This action gets an object containing key/value pairs representing the current dialog options hash. This method does not accept any arguments. |
| option*optionName, value* | This action sets the value of the dialog option associated with the specified optionName. |
| option*options* | This action sets one or more options for the dialog. |
| widget | This action returns the dialog box's widget element; the element annotated with the ui-dialog class name. This method does not accept any arguments. |

## Example

Now let us see an example using the actions from the above table. The following example demonstrates the use of *isOpen*, *open* and *close* methods.

```
<!doctype html>
<html lang="en">
   <head>
      <meta charset="utf-8">
      <title>jQuery UI Dialog functionality</title>
      <link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css"
rel="stylesheet">
      <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
      <script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
      <!-- CSS -->
      <style>
         .ui-widget-header,.ui-state-default, ui-button{
            background:#b9cd6d;
            border: 1px solid #b9cd6d;
            color: #FFFFFF;
            font-weight: bold;
         }
      </style>
      <!-- Javascript -->
      <script type="text/javascript">
         $(function(){
            $("#toggle").click(function() {

               ($("#dialog-5").dialog("isOpen") == false) ? $("#dialog-
5").dialog("open") : $("#dialog-5").dialog("close") ;
            });
            $("#dialog-5").dialog({autoOpen: false});
         });
      </script>
   </head>
   <body>
      <button >Toggle dialog!</button>
      <div >
         Click on the Toggle button to open and cose this dialog box.
      </div>
   </body>
</html>
```

Let's save above code in an HTML file **dialogexample.htm** and open it in a standard browser which supports javascript, you must see the following output:

Click on the Toggle button to open and cose this dialog box.

## Event Management on Dialog Box

In addition to the dialog *options* method we saw in the previous sections, JqueryUI provides event methods as which gets triggered for a particular event. These event methods are listed below:

| Event Method | Description |
|---|---|
| beforeClose*event, ui* | This event is triggered when the dialog box is about to close. Returning *false* prevents the dialog box from closing. It is handy for dialog boxes with forms that fail validation. Where *event* is of type *Event*, and *ui* is of type *Object*. |
| close*event, ui* | This event is triggered when the dialog box is closed. Where *event* is of type *Event*, and *ui* is of type *Object*. |
| create*event, ui* | This event is triggered when the dialog box is created. Where *event* is of type *Event*, and *ui* is of type *Object*. |
| drag*event, ui* | This event is triggered repeatedly as a dialog box is moved about during a drag. Where *event* is of type *Event*, and *ui* is of type *Object*. |
| dragStart*event, ui* | This event is triggered when a repositioning of the dialog box commences by dragging its title bar. Where *event* is of type *Event*, and *ui* is of type *Object*. |
| dragStop*event, ui* | This event is triggered when a drag operation terminates. Where *event* is of type *Event*, and *ui* is of type *Object*. |
| focus*event, ui* | This event is triggered when the dialog gains focus. Where *event* is of type *Event*, and *ui* is of type *Object*. |
| open*event, ui* | This event is triggered when the dialog box is opened. Where *event* is of type *Event*, and *ui* is of type *Object*. |
| resize*event, ui* | This event is triggered repeatedly as a dialog box is resized. Where *event* is of type *Event*, and *ui* is of type *Object*. |

| | |
|---|---|
| resizeStart*event, ui* | This event is triggered when a resize of the dialog box commences. Where *event* is of type *Event*, and *ui* is of type *Object*. |
| resizeStop*event, ui* | This event is triggered when a resize of the dialog box terminates. Where *event* is of type *Event*, and *ui* is of type *Object*. |

Following example demonstrates the use of the events listed in the above table.

## Use of beforeClose Event method

The following example demonstrates the use of **beforeClose** event method.

```
<!doctype html>
<html lang="en">
   <head>
      <meta charset="utf-8">
      <title>jQuery UI Dialog functionality</title>
      <link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css"
rel="stylesheet">
      <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
      <script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
      <!-- CSS -->
      <style>
         .ui-widget-header,.ui-state-default, ui-button{
            background:#b9cd6d;
            border: 1px solid #b9cd6d;
            color: #FFFFFF;
            font-weight: bold;
         }
         .invalid { color: red; }
         textarea {
            display: inline-block;
            width: 100%;
            margin-bottom: 10px;
         }
      </style>
      <!-- Javascript -->
      <script type="text/javascript">
         $(function(){
            $( "#dialog-6" ).dialog({
        autoOpen: false,
               buttons: {
                  OK: function() {
                     $( this ).dialog( "close" );
                  }
               },
               beforeClose: function( event, ui ) {
                  if ( !$( "#terms" ).prop( "checked" ) ) {
                     event.preventDefault();
                     $( "[for=terms]" ).addClass( "invalid" );
                  }
               },
               width: 600
            });
            $( "#opener-5" ).click(function() {
               $( "#dialog-6" ).dialog( "open" );
            });
         });
      </script>
   </head>
   <body>
      <div >
         <p>You must accept these terms before continuing.</p>
         <textarea>This Agreement and the Request constitute the entire agreement of the
         parties with respect to the subject matter of the Request. This Agreement shall
be
         governed by and construed in accordance with the laws of the State, without
giving
         effect to principles of conflicts of law.</textarea>
         <div>
            <label for="terms">I accept the terms</label>
            <input type="checkbox" >
         </div>
      </div>
      <button >Open Dialog</button>
   </body>
</html>
```

Let's save above code in an HTML file **dialogexample.htm** and open it in a standard browser which supports javascript, you must see the following output:

You must accept these terms before continuing.

This Agreement and the Request constitute the entire agreement of the parties with respect to the subject matter of the Request. This Agreement shall be governed by and construed in

I accept the terms

## Use of resize Event method

The following example demonstrates the use of **resize** event method.

```
<!doctype html>
<html lang="en">
   <head>
      <meta charset="utf-8">
      <title>jQuery UI Dialog functionality</title>
      <link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css"
rel="stylesheet">
      <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
      <script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
      <!-- CSS -->
      <style>
         .ui-widget-header,.ui-state-default, ui-button{
            background:#b9cd6d;
            border: 1px solid #b9cd6d;
            color: #FFFFFF;
            font-weight: bold;
         }
      </style>
      <!-- Javascript -->
      <script type="text/javascript">
         $(function(){
            $( "#dialog-7" ).dialog({
               autoOpen: false,
               resize: function( event, ui ) {
                  $( this ).dialog( "option", "title",
          ui.size.height + " x " + ui.size.width );
               }
            });
            $( "#opener-6" ).click(function() {
               $( "#dialog-7" ).dialog( "open" );
            });
         });
      </script>
   </head>
   <body>
      <div >Resize this dialog to see the dialog co-ordinates in the title!</div>
      <button >Open Dialog</button>
   </body>
</html>
```

Let's save above code in an HTML file **dialogexample.htm** and open it in a standard browser which supports javascript, you must see the following output:

Resize this dialog to see the dialog co-ordinates in the title!

## Extension Points

The dialog widget is built with the widget factory and can be extended. To extend widgets, we can either override or add to the behavior of existing methods. Following method provides as extension point with the same API stability as the dialog methods listed in the above table.

| Method | Description |
|---|---|
| _allowInteraction *event* | This method allows the user to interact with a given target element by whitelisting elements that are not children of the dialog but allow the users to be able to use. Where *event* is of type *Event*. |

> *The jQuery UI Widget Factory is an extensible base on which all of jQuery UI's widgets are built. Using the widget factory to build a plugin provides conveniences for state management, as well as conventions for common tasks like exposing plugin methods and changing options after instantiation.*

Processing math: 100%