* Overview
* Options
* Events
* Methods
* Theming

jQuery UI Draggable

## Overview

The jQuery UI Draggable plugin makes selected elements draggable by mouse.

Draggable elements gets a class of ui-draggable. During drag the element also gets a class of ui-draggable-dragging. If you want not just drag, but drag-and-drop, see the jQuery UI Droppable plugin, which provides a drop target for draggables.

All callbacks (start, stop, drag) receive two arguments: The original browser event and a prepared ui object, view below for a documentation of this object (if you name your second argument 'ui'):

* **ui.helper** - the jQuery object representing the helper that's being dragged
* **ui.position** - current position of the helper as { top, left } object, relative to the offset element
* **ui.offset** - current absolute position of the helper as { top, left } object, relative to page

To manipulate the position of a draggable during drag, you can either [use a wrapper as the draggable helper](http://jsbin.com/etako/edit) and position the wrapped element with absolute positioning, or you can correct internal values like so: $(this).data('draggable').offset.click.top -= x.

### Dependencies

* UI Core
* UI Widget
* UI Mouse

### Example

* Demo
* View Source

Initialize a draggable with default options.

$("#draggable").draggable();

<!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">  
 #draggable { width: 100px; height: 70px; background: silver; }  
 </style>  
 <script>  
 $(document).ready(function() {  
 $("#draggable").draggable();  
 });  
 </script>  
</head>  
<body style="font-size:62.5%;">  
   
<div id="draggable">Drag me</div>  
  
</body>  
</html>

## Options

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

#### **addClasses**Type: Boolean Default: true If set to false, will prevent the ui-draggable class from being added. This may be desired as a performance optimization when calling .draggable() init on many hundreds of elements. Code examplesInitialize a draggable with the addClasses option specified. $( ".selector" ).draggable({ addClasses: false });Get or set the addClasses option, after init. //getter var addClasses = $( ".selector" ).draggable( "option", "addClasses" ); //setter $( ".selector" ).draggable( "option", "addClasses", false );

#### **appendTo**Type: Element, Selector Default: 'parent' The element passed to or selected by the appendTo option will be used as the draggable helper's container during dragging. By default, the helper is appended to the same container as the draggable. Code examplesInitialize a draggable with the appendTo option specified. $( ".selector" ).draggable({ appendTo: 'body' });Get or set the appendTo option, after init. //getter var appendTo = $( ".selector" ).draggable( "option", "appendTo" ); //setter $( ".selector" ).draggable( "option", "appendTo", 'body' );

#### **axis**Type: String Default: false Constrains dragging to either the horizontal (x) or vertical (y) axis. Possible values: 'x', 'y'. Code examplesInitialize a draggable with the axis option specified. $( ".selector" ).draggable({ axis: 'x' });Get or set the axis option, after init. //getter var axis = $( ".selector" ).draggable( "option", "axis" ); //setter $( ".selector" ).draggable( "option", "axis", 'x' );

#### **cancel**Type: Selector Default: ':input,option' Prevents dragging from starting on specified elements. Code examplesInitialize a draggable with the cancel option specified. $( ".selector" ).draggable({ cancel: 'button' });Get or set the cancel option, after init. //getter var cancel = $( ".selector" ).draggable( "option", "cancel" ); //setter $( ".selector" ).draggable( "option", "cancel", 'button' );

#### **connectToSortable**Type: Selector Default: false Allows the draggable to be dropped onto the specified sortables. If this option is used (helper must be set to 'clone' in order to work flawlessly), a draggable can be dropped onto a sortable list and then becomes part of it. Note: Specifying this option as an array of selectors has been removed. Code examplesInitialize a draggable with the connectToSortable option specified. $( ".selector" ).draggable({ connectToSortable: 'ul#myList' });Get or set the connectToSortable option, after init. //getter var connectToSortable = $( ".selector" ).draggable( "option", "connectToSortable" ); //setter $( ".selector" ).draggable( "option", "connectToSortable", 'ul#myList' );

#### **containment**Type: Selector, Element, String, Array Default: false Constrains dragging to within the bounds of the specified element or region. Possible string values: 'parent', 'document', 'window', [x1, y1, x2, y2]. Code examplesInitialize a draggable with the containment option specified. $( ".selector" ).draggable({ containment: 'parent' });Get or set the containment option, after init. //getter var containment = $( ".selector" ).draggable( "option", "containment" ); //setter $( ".selector" ).draggable( "option", "containment", 'parent' );

#### **cursor**Type: String Default: 'auto' The css cursor during the drag operation. Code examplesInitialize a draggable with the cursor option specified. $( ".selector" ).draggable({ cursor: 'crosshair' });Get or set the cursor option, after init. //getter var cursor = $( ".selector" ).draggable( "option", "cursor" ); //setter $( ".selector" ).draggable( "option", "cursor", 'crosshair' );

#### **cursorAt**Type: Object Default: false Sets the offset of the dragging helper relative to the mouse cursor. Coordinates can be given as a hash using a combination of one or two keys: { top, left, right, bottom }. Code examplesInitialize a draggable with the cursorAt option specified. $( ".selector" ).draggable({ cursorAt: { left: 5 } });Get or set the cursorAt option, after init. //getter var cursorAt = $( ".selector" ).draggable( "option", "cursorAt" ); //setter $( ".selector" ).draggable( "option", "cursorAt", { left: 5 } );

#### **delay**Type: Integer Default: 0 Time in milliseconds after mousedown until dragging should start. This option can be used to prevent unwanted drags when clicking on an element. Code examplesInitialize a draggable with the delay option specified. $( ".selector" ).draggable({ delay: 500 });Get or set the delay option, after init. //getter var delay = $( ".selector" ).draggable( "option", "delay" ); //setter $( ".selector" ).draggable( "option", "delay", 500 );

#### **distance**Type: Integer Default: 1 Distance in pixels after mousedown the mouse must move before dragging should start. This option can be used to prevent unwanted drags when clicking on an element. Code examplesInitialize a draggable with the distance option specified. $( ".selector" ).draggable({ distance: 30 });Get or set the distance option, after init. //getter var distance = $( ".selector" ).draggable( "option", "distance" ); //setter $( ".selector" ).draggable( "option", "distance", 30 );

#### **grid**Type: Array Default: false Snaps the dragging helper to a grid, every x and y pixels. Array values: [x, y] Code examplesInitialize a draggable with the grid option specified. $( ".selector" ).draggable({ grid: [50, 20] });Get or set the grid option, after init. //getter var grid = $( ".selector" ).draggable( "option", "grid" ); //setter $( ".selector" ).draggable( "option", "grid", [50, 20] );

#### **handle**Type: Element, Selector Default: false If specified, restricts drag start click to the specified element(s). Code examplesInitialize a draggable with the handle option specified. $( ".selector" ).draggable({ handle: 'h2' });Get or set the handle option, after init. //getter var handle = $( ".selector" ).draggable( "option", "handle" ); //setter $( ".selector" ).draggable( "option", "handle", 'h2' );

#### **helper**Type: String, Function Default: 'original' Allows for a helper element to be used for dragging display. Possible values: 'original', 'clone', Function. If a function is specified, it must return a DOMElement. Code examplesInitialize a draggable with the helper option specified. $( ".selector" ).draggable({ helper: 'clone' });Get or set the helper option, after init. //getter var helper = $( ".selector" ).draggable( "option", "helper" ); //setter $( ".selector" ).draggable( "option", "helper", 'clone' );

#### **iframeFix**Type: Boolean, Selector Default: false Prevent iframes from capturing the mousemove events during a drag. Useful in combination with cursorAt, or in any case, if the mouse cursor is not over the helper. If set to true, transparent overlays will be placed over all iframes on the page. If a selector is supplied, the matched iframes will have an overlay placed over them. Code examplesInitialize a draggable with the iframeFix option specified. $( ".selector" ).draggable({ iframeFix: true });Get or set the iframeFix option, after init. //getter var iframeFix = $( ".selector" ).draggable( "option", "iframeFix" ); //setter $( ".selector" ).draggable( "option", "iframeFix", true );

#### **opacity**Type: Float Default: false Opacity for the helper while being dragged. Code examplesInitialize a draggable with the opacity option specified. $( ".selector" ).draggable({ opacity: 0.35 });Get or set the opacity option, after init. //getter var opacity = $( ".selector" ).draggable( "option", "opacity" ); //setter $( ".selector" ).draggable( "option", "opacity", 0.35 );

#### **refreshPositions**Type: Boolean Default: false If set to true, all droppable positions are calculated on every mousemove. Caution: This solves issues on highly dynamic pages, but dramatically decreases performance. Code examplesInitialize a draggable with the refreshPositions option specified. $( ".selector" ).draggable({ refreshPositions: true });Get or set the refreshPositions option, after init. //getter var refreshPositions = $( ".selector" ).draggable( "option", "refreshPositions" ); //setter $( ".selector" ).draggable( "option", "refreshPositions", true );

#### **revert**Type: Boolean, String Default: false If set to true, the element will return to its start position when dragging stops. Possible string values: 'valid', 'invalid'. If set to invalid, revert will only occur if the draggable has not been dropped on a droppable. For valid, it's the other way around. Code examplesInitialize a draggable with the revert option specified. $( ".selector" ).draggable({ revert: true });Get or set the revert option, after init. //getter var revert = $( ".selector" ).draggable( "option", "revert" ); //setter $( ".selector" ).draggable( "option", "revert", true );

#### **revertDuration**Type: Integer Default: 500 The duration of the revert animation, in milliseconds. Ignored if revert is false. Code examplesInitialize a draggable with the revertDuration option specified. $( ".selector" ).draggable({ revertDuration: 1000 });Get or set the revertDuration option, after init. //getter var revertDuration = $( ".selector" ).draggable( "option", "revertDuration" ); //setter $( ".selector" ).draggable( "option", "revertDuration", 1000 );

#### **scope**Type: String Default: 'default' Used to group sets of draggable and droppable items, in addition to droppable's accept option. A draggable with the same scope value as a droppable will be accepted by the droppable. Code examplesInitialize a draggable with the scope option specified. $( ".selector" ).draggable({ scope: 'tasks' });Get or set the scope option, after init. //getter var scope = $( ".selector" ).draggable( "option", "scope" ); //setter $( ".selector" ).draggable( "option", "scope", 'tasks' );

#### **scroll**Type: Boolean Default: true If set to true, container auto-scrolls while dragging. Code examplesInitialize a draggable with the scroll option specified. $( ".selector" ).draggable({ scroll: false });Get or set the scroll option, after init. //getter var scroll = $( ".selector" ).draggable( "option", "scroll" ); //setter $( ".selector" ).draggable( "option", "scroll", false );

#### **scrollSensitivity**Type: Integer Default: 20 Distance in pixels from the edge of the viewport after which the viewport should scroll. Distance is relative to pointer, not the draggable. Code examplesInitialize a draggable with the scrollSensitivity option specified. $( ".selector" ).draggable({ scrollSensitivity: 40 });Get or set the scrollSensitivity option, after init. //getter var scrollSensitivity = $( ".selector" ).draggable( "option", "scrollSensitivity" ); //setter $( ".selector" ).draggable( "option", "scrollSensitivity", 40 );

#### **scrollSpeed**Type: Integer Default: 20 The speed at which the window should scroll once the mouse pointer gets within the scrollSensitivity distance. Code examplesInitialize a draggable with the scrollSpeed option specified. $( ".selector" ).draggable({ scrollSpeed: 40 });Get or set the scrollSpeed option, after init. //getter var scrollSpeed = $( ".selector" ).draggable( "option", "scrollSpeed" ); //setter $( ".selector" ).draggable( "option", "scrollSpeed", 40 );

#### **snap**Type: Boolean, Selector Default: false If set to a selector or to true (equivalent to '.ui-draggable'), the draggable will snap to the edges of the selected elements when near an edge of the element. Code examplesInitialize a draggable with the snap option specified. $( ".selector" ).draggable({ snap: true });Get or set the snap option, after init. //getter var snap = $( ".selector" ).draggable( "option", "snap" ); //setter $( ".selector" ).draggable( "option", "snap", true );

#### **snapMode**Type: String Default: 'both' Determines which edges of snap elements the draggable will snap to. Ignored if snap is false. Possible values: 'inner', 'outer', 'both' Code examplesInitialize a draggable with the snapMode option specified. $( ".selector" ).draggable({ snapMode: 'outer' });Get or set the snapMode option, after init. //getter var snapMode = $( ".selector" ).draggable( "option", "snapMode" ); //setter $( ".selector" ).draggable( "option", "snapMode", 'outer' );

#### **snapTolerance**Type: Integer Default: 20 The distance in pixels from the snap element edges at which snapping should occur. Ignored if snap is false. Code examplesInitialize a draggable with the snapTolerance option specified. $( ".selector" ).draggable({ snapTolerance: 40 });Get or set the snapTolerance option, after init. //getter var snapTolerance = $( ".selector" ).draggable( "option", "snapTolerance" ); //setter $( ".selector" ).draggable( "option", "snapTolerance", 40 );

#### **stack**Type: Selector Default: false Controls the z-Index of the set of elements that match the selector, always brings to front the dragged item. Very useful in things like window managers. Code examplesInitialize a draggable with the stack option specified. $( ".selector" ).draggable({ stack: ".products" });Get or set the stack option, after init. //getter var stack = $( ".selector" ).draggable( "option", "stack" ); //setter $( ".selector" ).draggable( "option", "stack", ".products" );

#### **zIndex**Type: Integer Default: false z-index for the helper while being dragged. Code examplesInitialize a draggable with the zIndex option specified. $( ".selector" ).draggable({ zIndex: 2700 });Get or set the zIndex option, after init. //getter var zIndex = $( ".selector" ).draggable( "option", "zIndex" ); //setter $( ".selector" ).draggable( "option", "zIndex", 2700 );

## Events

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

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

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

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

## Methods

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

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

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

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

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

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

## Theming

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

### Sample markup with jQuery UI CSS Framework classes

<div class="**ui-draggable**"></div>

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