* Overview
* Options
* Events
* Methods
* Theming

jQuery UI Selectable

## Overview

The jQuery UI Selectable plugin allows for elements to be selected by dragging a box (sometimes called a lasso) with the mouse over the elements. Also, elements can be selected by click or drag while holding the Ctrl/Meta key, allowing for multiple (non-contiguous) selections.

### Dependencies

* UI Core
* UI Widget
* UI Mouse

### Example

* Demo
* View Source

A simple jQuery UI Selectable.

$("#selectable").selectable();

<!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">  
#selectable .ui-selecting {  
 background: silver;  
}  
#selectable .ui-selected {  
 background: gray;  
}  
</style>  
  
 <script>  
 $(document).ready(function() {  
 $("#selectable").selectable();  
 });  
 </script>  
</head>  
<body style="font-size:62.5%;">  
   
<ul id="selectable">  
<li>Item 1</li>  
<li>Item 2</li>  
<li>Item 3</li>  
<li>Item 4</li>  
<li>Item 5</li>  
</ul>  
  
</body>  
</html>

## Options

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

#### **autoRefresh**Type: Boolean Default: true This determines whether to refresh (recalculate) the position and size of each selectee at the beginning of each select operation. If you have many many items, you may want to set this to false and call the refresh method manually. Code examplesInitialize a selectable with the autoRefresh option specified. $( ".selector" ).selectable({ autoRefresh: false });Get or set the autoRefresh option, after init. //getter var autoRefresh = $( ".selector" ).selectable( "option", "autoRefresh" ); //setter $( ".selector" ).selectable( "option", "autoRefresh", false );

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

#### **delay**Type: Integer Default: 0 Time in milliseconds to define when the selecting should start. It helps preventing unwanted selections when clicking on an element. Code examplesInitialize a selectable with the delay option specified. $( ".selector" ).selectable({ delay: 20 });Get or set the delay option, after init. //getter var delay = $( ".selector" ).selectable( "option", "delay" ); //setter $( ".selector" ).selectable( "option", "delay", 20 );

#### **distance**Type: Integer Default: 0 Tolerance, in pixels, for when selecting should start. If specified, selecting will not start until after mouse is dragged beyond distance. Code examplesInitialize a selectable with the distance option specified. $( ".selector" ).selectable({ distance: 20 });Get or set the distance option, after init. //getter var distance = $( ".selector" ).selectable( "option", "distance" ); //setter $( ".selector" ).selectable( "option", "distance", 20 );

#### **filter**Type: Selector Default: '\*' The matching child elements will be made selectees (able to be selected). Code examplesInitialize a selectable with the filter option specified. $( ".selector" ).selectable({ filter: 'li' });Get or set the filter option, after init. //getter var filter = $( ".selector" ).selectable( "option", "filter" ); //setter $( ".selector" ).selectable( "option", "filter", 'li' );

### **tolerance**Type: String Default: 'touch'

* Possible values: 'touch', 'fit'.
  + **fit**: draggable overlaps the droppable entirely
  + **touch**: draggable overlaps the droppable any amount

## Code examplesInitialize a selectable with the tolerance option specified. $( ".selector" ).selectable({ tolerance: 'fit' });Get or set the tolerance option, after init. //getter var tolerance = $( ".selector" ).selectable( "option", "tolerance" ); //setter $( ".selector" ).selectable( "option", "tolerance", 'fit' );

Events

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

#### **selected**Type: selectableselected This event is triggered at the end of the select operation, on each element added to the selection. Code examplesSupply a callback function to handle the selected event as an init option. $( ".selector" ).selectable({ selected: function(event, ui) { ... } });Bind to the selected event by type: selectableselected. $( ".selector" ).bind( "selectableselected", function(event, ui) { ... });

#### **selecting**Type: selectableselecting This event is triggered during the select operation, on each element added to the selection. Code examplesSupply a callback function to handle the selecting event as an init option. $( ".selector" ).selectable({ selecting: function(event, ui) { ... } });Bind to the selecting event by type: selectableselecting. $( ".selector" ).bind( "selectableselecting", function(event, ui) { ... });

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

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

#### **unselected**Type: selectableunselected This event is triggered at the end of the select operation, on each element removed from the selection. Code examplesSupply a callback function to handle the unselected event as an init option. $( ".selector" ).selectable({ unselected: function(event, ui) { ... } });Bind to the unselected event by type: selectableunselected. $( ".selector" ).bind( "selectableunselected", function(event, ui) { ... });

#### **unselecting**Type: selectableunselecting This event is triggered during the select operation, on each element removed from the selection. Code examplesSupply a callback function to handle the unselecting event as an init option. $( ".selector" ).selectable({ unselecting: function(event, ui) { ... } });Bind to the unselecting event by type: selectableunselecting. $( ".selector" ).bind( "selectableunselecting", function(event, ui) { ... });

## Methods

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

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

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

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

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

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

### **refresh**Signature: .selectable( "refresh" ) Refresh the position and size of each selectee element. This method can be used to manually recalculate the position and size of each selectee element. Very useful if autoRefresh is set to false.

## Theming

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

### Sample markup with jQuery UI CSS Framework classes

<ul class="**ui-selectable**">

   <li class="**ui-selectee**"></li>

   <li class="**ui-selectee**"></li>

   <li class="**ui-selectee**"></li>

</ul>

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

**<ul>**

**<li></li>**

**<li></li>**

**<li></li>**

**</ul>.**