概述  
Selectable插件允许用户对指定的元素进行选中的动作。此外还支持按住Ctrl键单击或拖拽选择多个元素。  
官方示例地址：http://jqueryui.com/demos/sortable/  
**所有的事件回调函数都有两个参数：event和ui，浏览器自有event对象，和经过封装的ui对象**ui.helper - 表示sortable元素的JQuery对象，通常是当前元素的克隆对象  
ui.position - 表示相对当前对象，鼠标的坐标值对象{top,left}  
ui.offset - 表示相对于当前页面，鼠标的坐标值对象{top,left}  
ui.item - 表示当前拖拽的元素  
ui.placeholder - 占位符（如果有定义的话）  
ui.sender - 当前拖拽元素的所属sortable对象（仅当元素是从另一个sortable对象传递过来时有用）

**·参数（参数名 : 参数类型 : 默认值）**  
**appendTo : String : 'parent'**Defines where the helper that moves with the mouse is being appended to during the drag (for example, to resolve overlap/zIndex issues).  
初始：$('.selector').sortable({ appendTo: 'body' });  
获取：var appendTo = $('.selector').sortable('option', 'appendTo');  
设置：$('.selector').sortable('option', 'appendTo', 'body');

**axis : String : false**  
如果有设置，则元素仅能横向或纵向拖动。可选值：'x', 'y'  
初始：$('.selector').sortable({ axis: 'x' });  
获取：var axis = $('.selector').sortable('option', 'axis');  
设置：$('.selector').sortable('option', 'axis', 'x');

**cancel : Selector : ':input,button'**  
阻止排序动作在匹配的元素上发生。  
初始：$('.selector').sortable({ cancel: 'button' });  
获取：var cancel = $('.selector').sortable('option', 'cancel');  
设置：$('.selector').sortable('option', 'cancel', 'button');

**containment : Element, String, Selector : false**  
约束排序动作只能在一个指定的范围内发生。可选值：DOM对象, 'parent', 'document', 'window', 或jQuery对象  
初始：$('.selector').sortable({ containment: 'parent' });  
获取：var containment = $('.selector').sortable('option', 'containment');  
设置：$('.selector').sortable('option', 'containment', 'parent');

**cursor : String : 'auto'**定义在开始排序动作时，如果的样式。  
初始：$('.selector').sortable({ cursor: 'crosshair' });  
获取：var cursor = $('.selector').sortable('option', 'cursor');  
设置：$('.selector').sortable('option', 'cursor', 'crosshair');

**cursorAt : Object : false**  
当开始移动时，鼠标定位在的某个位置上（最多两个方向）。可选值：{ top, left, right, bottom }.  
初始：$('.selector').sortable({ cursorAt: 'top' });  
获取：var cursorAt = $('.selector').sortable('option', 'cursorAt');  
设置：$('.selector').sortable('option', 'cursorAt', 'top');

**connectWith : Selector : false**  
允许sortable对象连接另一个sortable对象，可将item元素拖拽到另一个中。  
初始：$('.selector').sortable({ connectWith: '.otherlist' });  
获取：var connectWith = $('.selector').sortable('option', 'connectWith');  
设置：$('.selector').sortable('option', 'connectWith', '.otherlist');  
真实案例如下：（此处为两个sortable对象，其中一个移到另一个sortable对象后，新组成的仍然是sortable对象）  
JS代码：

HTML代码：

* Item 1
* Item 2
* Item 3
* Item 4
* Item 5
* Item 1
* Item 2
* Item 3
* Item 4
* Item 5

**delay : Integer : 0**  
以毫秒为单位，设置延迟多久才激活排序动作。此参数可防止误点击。  
初始：$('.selector').sortable({ delay: 500 });  
获取：var delay = $('.selector').sortable('option', 'delay');  
设置：$('.selector').sortable('option', 'delay', 500);

**distance : Integer : 1**  
决定至少要在元素上面拖动多少像素后，才正式触发排序动作。  
初始：$('.selector').sortable({ distance: 30 });  
获取：var distance = $('.selector').sortable('option', 'distance');  
设置：$('.selector').sortable('option', 'distance', 30);

**dropOnEmpty : Boolean : true**  
是否允許拖拽到一個空的sortable对象中。  
初始：$('.selector').sortable({ dropOnEmpty: false });  
获取：var dropOnEmpty = $('.selector').sortable('option', 'dropOnEmpty');  
设置：$('.selector').sortable('option', 'dropOnEmpty', false);

**forceHelperSize : Boolean : false**  
If true, forces the helper to have a size.  
初始：$('.selector').sortable({ forceHelperSize: true });  
获取：var forceHelperSize = $('.selector').sortable('option', 'forceHelperSize');  
设置：$('.selector').sortable('option', 'forceHelperSize', true);

**forcePlaceholderSize : Boolean : false**If true, forces the placeholder to have a size.  
初始：$('.selector').sortable({ forcePlaceholderSize: true });  
获取：var forcePlaceholderSize = $('.selector').sortable('option', 'forcePlaceholderSize');  
设置：$('.selector').sortable('option', 'forcePlaceholderSize', true);

**grid : Array : false**  
将排序对象的item元素视为一个格子处理，每次移动都按一个格子大小移动，数组值：[x,y]  
初始：$('.selector').sortable({ grid: [50, 20] });  
获取：var grid = $('.selector').sortable('option', 'grid');  
设置：$('.selector').sortable('option', 'grid', [50, 20]);

**handle : Selector, Element : false**  
限制排序的动作只能在item元素中的某个元素开始。  
初始：$('.selector').sortable({ handle: 'h2' });  
获取：var handle = $('.selector').sortable('option', 'handle');  
设置：$('.selector').sortable('option', 'handle', 'h2');

**helper : String, Function : 'original'**  
设置是否在拖拽元素时，显示一个辅助的元素。可选值：'original', 'clone'  
初始：$('.selector').sortable({ helper: 'clone' });  
获取：var helper = $('.selector').sortable('option', 'helper');  
设置：$('.selector').sortable('option', 'helper', 'clone');

**items : Selector : '> \*'**  
指定在排序对象中，哪些元素是可以进行拖拽排序的。  
初始：$('.selector').sortable({ items: 'li' });  
获取：var items = $('.selector').sortable('option', 'items');  
设置：$('.selector').sortable('option', 'items', 'li');

**opacity : Float : false**  
定义当排序时，辅助元素(helper)显示的透明度。  
初始：$('.selector').sortable({ opacity: 0.6 });  
获取：var opacity = $('.selector').sortable('option', 'opacity');  
设置：$('.selector').sortable('option', 'opacity', 0.6);

**placeholderType: StringDefault: false**  
设置当排序动作发生时，空白占位符的CSS样式。  
初始：$('.selector').sortable({ placeholder: 'ui-state-highlight' });  
获取：var placeholder = $('.selector').sortable('option', 'placeholder');  
设置：$('.selector').sortable('option', 'placeholder', 'ui-state-highlight');  
注意：此处的ui-state-highlight为类名，可在样式表里设置它的样式，也就是空白占位符的CSS样式。  
真实案例用法如下：

**revert : Boolean : false**  
如果设置成true，则被拖拽的元素在返回新位置时，会有一个动画效果。  
初始：$('.selector').sortable({ revert: true });  
获取：var revert = $('.selector').sortable('option', 'revert');  
设置：$('.selector').sortable('option', 'revert', true);

scroll : Boolean : true  
如果设置成true，则元素被拖动到页面边缘时，会自动滚动。  
初始：$('.selector').sortable({ scroll: false });  
获取：var scroll = $('.selector').sortable('option', 'scroll');  
设置：$('.selector').sortable('option', 'scroll', false);

scrollSensitivity : Integer : 20  
设置当元素移动至边缘多少像素时，便开始滚动页面。  
初始：$('.selector').sortable({ scrollSensitivity: 40 });  
获取：var scrollSensitivity = $('.selector').sortable('option', 'scrollSensitivity');  
设置：$('.selector').sortable('option', 'scrollSensitivity', 40);

scrollSpeed : Integer : 20  
设置页面滚动的速度。  
初始：$('.selector').sortable({ scrollSpeed: 40 });  
获取：var scrollSpeed = $('.selector').sortable('option', 'scrollSpeed');  
设置：$('.selector').sortable('option', 'scrollSpeed', 40);

tolerance : String : 'intersect'  
设置当拖动元素越过其它元素多少时便对元素进行重新排序。可选值：'intersect', 'pointer'  
intersect：至少重叠50%  
pointer：鼠标指针重叠元素  
初始：$('.selector').sortable({ tolerance: 'pointer' });  
获取：var tolerance = $('.selector').sortable('option', 'tolerance');  
设置：$('.selector').sortable('option', 'tolerance', 'pointer');

zIndex : Integer : 1000  
设置在排序动作发生时，元素的z-index值。  
初始：$('.selector').sortable({ zIndex: 5 });  
获取：var zIndex = $('.selector').sortable('option', 'zIndex');  
设置：$('.selector').sortable('option', 'zIndex', 5);

**·事件**  
start  
当排序动作开始时触发此事件。  
定义：$('.selector').sortable({ start: function(event, ui) { ... } });  
绑定：$('.selector').bind('sortstart', function(event, ui) { ... });

sort  
当元素发生排序时触发此事件。  
定义：$('.selector').sortable({ sort: function(event, ui) { ... } });  
绑定：$('.selector').bind('sort', function(event, ui) { ... });

change  
当元素发生排序且坐标已发生改变时触发此事件。  
定义：$('.selector').sortable({ change: function(event, ui) { ... } });  
绑定：$('.selector').bind('sortchange', function(event, ui) { ... });

beforeStop  
当排序动作结束之前触发此事件。此时占位符元素和辅助元素仍有效。  
定义：$('.selector').sortable({ beforeStop: function(event, ui) { ... } });  
绑定：$('.selector').bind('sortbeforeStop', function(event, ui) { ... });

stop  
当排序动作结束时触发此事件。  
定义：$('.selector').sortable({ stop: function(event, ui) { ... } });  
绑定：$('.selector').bind('sortstop', function(event, ui) { ... });

update  
当排序动作结束时且元素坐标已经发生改变时触发此事件。  
定义：$('.selector').sortable({ update: function(event, ui) { ... } });  
绑定：$('.selector').bind('sortupdate', function(event, ui) { ... });

receive  
当一个已连接的sortable对象接收到另一个sortable对象的元素后触发此事件。  
定义：$('.selector').sortable({ receive: function(event, ui) { ... } });  
绑定：$('.selector').bind('sortreceive', function(event, ui) { ... });

over  
当一个元素拖拽移入另一个sortable对象后触发此事件。  
定义：$('.selector').sortable({ over: function(event, ui) { ... } });  
绑定：$('.selector').bind('sortover', function(event, ui) { ... });

out  
当一个元素拖拽移出sortable对象移出并进入另一个sortable对象后触发此事件。  
定义：$('.selector').sortable({ out: function(event, ui) { ... } });  
绑定：$('.selector').bind('sortout', function(event, ui) { ... });

activate  
当一个有使用连接的sortable对象开始排序动作时，所有允许的sortable触发此事件。  
定义：$('.selector').sortable({ activate: function(event, ui) { ... } });  
绑定：$('.selector').bind('sortactivate', function(event, ui) { ... });

deactivate  
当一个有使用连接的sortable对象结束排序动作时，所有允许的sortable触发此事件。  
定义：$('.selector').sortable({ deactivate: function(event, ui) { ... } });  
绑定：$('.selector').bind('sortdeactivate', function(event, ui) { ... });

**·方法**  
destory  
从元素中移除拖拽功能。  
用法：.sortable( 'destroy' )

disable  
禁用元素的拖拽功能。  
用法：.sortable( 'disable' )

enable  
启用元素的拖拽功能。  
用法：.sortable( 'enable' )

option  
获取或设置元素的参数。  
用法：.sortable( 'option' , optionName , [value] )

serialize  
获取或设置序列化后的每个item元素的id属性。  
用法：.sortable( 'serialize' , [options] )

toArray  
获取序列化后的每个item元素的id属性的数组。  
用法：.sortable( 'toArray' ）

refresh  
手动重新刷新当前sortable对象的item元素的排序。  
用法：.sortable( 'refresh' )

refreshPositions  
手动重新刷新当前sortable对象的item元素的坐标，此方法可能会降低性能。  
用法：.sortable( 'refreshPositions' )

cancel  
取消当前sortable对象中item元素的排序改变。  
用法：.sortable( 'cancel' )