



# Web UI框架及应用

## 第4章 jQueryUI 交互组件



● 拖放组件

● 缩放组件

● 选择组件

● 排序组件



# 交互组件

交互组件不是存在于页面上的物理对象，而是赋予对象一组**内在的行为**。

使页面中的元素具有更富吸引力的外观和更好的交互性。

**拖放组件、缩放组件、选择组件、排序组件**



# 拖放组件

拖放（ drag-and-drop ）

是网页上常见的一种操作，用来移动某个元素（使用鼠标将其放置在另一个页面元素上）。

jQuery UI可以管理这些操作，并能对 “**拖拽**” （移动某元素）和 “**放置**” （搁放下刚刚移动的元素）加以区分。





# 拖动组件的基本结构

- 拖放组件draggable 功能可以应用在任意的 DOM 元素上 。

```
<div id="drag2" class="ui-widget-content">  
  <p>请拖动我！ </p>  
</div>
```

请拖动我！

```
$(function() {  
  $( "#draggable" ).draggable();  
});
```

draggable1.html



## 思考

为什么使用了 `draggable()` 方法会彻底改变已渲染页面中的HTML元素的外观

该方法遍历了HTML代码，在指定的可移动元素上自动增加了特定的CSS类，赋予了它们适当的样式。





# 主题化

自定义拖动组件——需要使用指定的CSS class 名称

ui-draggable : 拖动元素受影响的css。

ui-draggable-handle : 拖动手柄。

ui-draggable-dragging : 拖动过程中的css。



# draggable(options )方法

draggable(options )方法声明了一个在HTML页面中可以被移动的HTML元素。options参数是一个对象，用来指定相关元素的行为。

```
$(selector, context) . draggable(options )
```





# 配置拖动属性

## 管理元素的移动

属性	用途	默认
<code>cursor</code>	指定拖动光标的CSS样式，可用的值有： crosshair(十字形)、default(箭头形)、 pointer(手形)、move(十字交叉箭头形)...	auto
<code>delay</code>	指定开始拖动时延迟的毫秒数	0
<code>opacity</code>	元素移动时的透明度（介于0和1之间）。	1



# 配置拖动属性

可以在CSS中指定所要拖动的<div>元素的背景图片。

```
#drag {  
    background:url(img/draggable.png) no-repeat;  
    width:114px;  
    height:114px;  
}
```

```
$(function() {  
    var dragOpts = {  
        cursor: "pointer" };  
    $("#drag").draggable(dragOpts);  
});
```



draggable2.html



# 配置拖动属性

## 管理元素的移动限制

属性	用途
axis	限制拖动只能沿着某个方向轴进行，可设置x或y
containment	指定一个容器元素，拖动将限制在此元素范围内。此元素可以通过选择器、DOM元素或者字符串来指定。
scroll	使拖动元素的容器可自动卷动
scrollspeed	设置拖动元素的卷动速度
grid	使可拖动元素只能在页面中虚拟网格间移动
helper	定义一个虚构拖动元素，在拖动时替代实际的可拖动元素



# 1 拖放组件

当 draggable 移动到视区外时自动滚动文档。设置 scroll 选项为 true 来启用自动滚动，当滚动触发时进行微调，滚动速度是通过 scrollSensitivity 和 scrollSpeed 选项设置的。

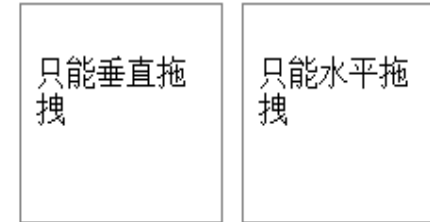
```
<script>
$(function() {
  $( "#draggable" ).draggable({ scroll: true });
  $( "#draggable2" ).draggable({ scroll: true, scrollSensitivity: 100 });
  $( "#draggable3" ).draggable({ scroll: true, scrollSpeed: 100 });
});
</script>
```



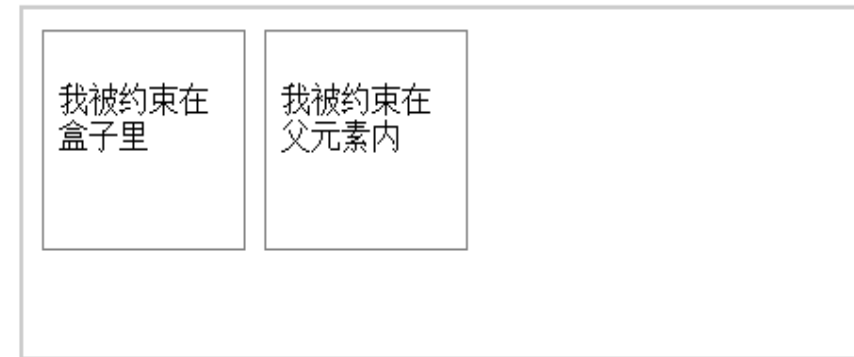
# 通过设置边界来约束运动

```
$(function() {  
    $( "#draggable" ).draggable({ axis: "y" });  
    $( "#draggable2" ).draggable({ axis: "x" });  
    $( "#draggable3" ).draggable({ containment:  
"#containment-wrapper", scroll: false });  
    $( "#draggable5" ).draggable({ containment:  
"parent" });  
});
```

沿着轴约束运动:



在另一个 DOM 元素中约束运动:



draggable3.html



# helper元素

helper元素是用于在拖动过程中显示的虚拟对象，而不是移动实际的可拖动对象。

```
var dragOpts = {  
    helper: helperMaker  
};  
  
function helperMaker() {  
    return $("

").css({  
        border: "4px solid #cccccc",  
        opacity: "0.5",  
        height: "110px",  
        width: "120px"  
    });  
}


```

draggable4.html



# 事件

draggable 上的 start、drag 和 stop 事件。

属性	用途
start	拖拽开始时触发。
drag	拖拽期间触发。
stop	拖拽停止时触发。

draggable5.html



# 放置组件的基本结构

- 放置组件（ Droppable ）是为拖动对象定义有效的**放置目标**。即为拖动元素提供可投放的地点，并且在将拖动元素投放到该区域时，触发某操作。

```
<div id="drag"></div>  
<div id="target"></div>
```

```
$(function() {  
    $("#drag").draggable();  
    $("#target").droppable();  
});
```

droppable1.html





# 配置放置属性

## 管理放置类

属性	用途
<code>accept</code>	设置放置对象所能接受的元素
<code>acctiveClass</code>	设置在可接受的拖动元素处于拖动状态时，放置对象的样式
<code>greedy</code>	当拖动对象被拖入到嵌套的放置对象中时，用于阻止放置事件被连环调用
<code>hoverClass</code>	指定放置对象在拖动对象被移动到其中时的样式
<code>tolerance</code>	设置所接受的拖动对象被认为完成投放的触发模式



# Tolerance属性

tolerance属性指示了放置对象探测拖动对象是否已完成放置

属性	用途
fit	拖动对象完全处于放置对象的边界之内才会认为完成放置
intersect	至少有25%的拖动对象进入放置对象边界之内才会认为完成放置
pointer	鼠标指针重叠在 droppable 上
touch	只要拖动对象的边缘与放置对象的边缘相接触就会认为完成放置



# 放置事件回调函数

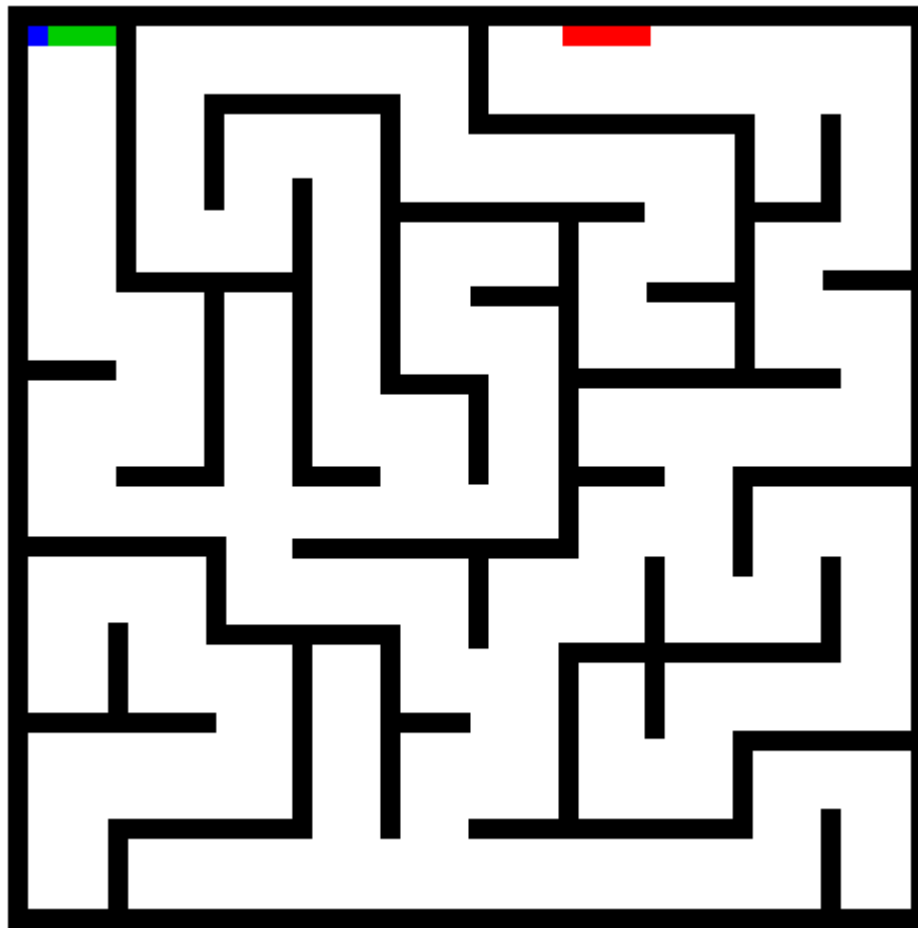
## 管理放置事件

属性	用途
activate	当所接受的拖动对象开始拖动
deactivate	当所接受的拖动对象停止拖动
drop	当所接受的拖动对象被放入至放置对象中
out	当所接受的拖动对象从放置对象内部移出其边界
over	当所接受的拖动对象被移动到放置对象的边界之内



# JavaScript迷宫小游戏

## 拖放组件实例

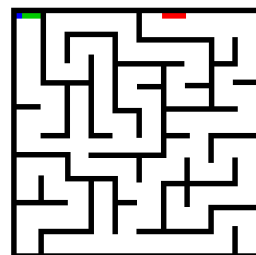


[dragMaze.html](#)



# JavaScript迷宫小游戏

1. 使用for循环为迷宫添加墙壁
2. 为拖动元素定义一个简单的配置对象（containment属性）限制拖动元素只能在迷宫内部移动。创建拖动组件。
3. 为墙壁定义配置对象。每个墙壁看作放置组件并接受拖动元素。指定tolerance属性值为touch，并为over属性增加一个回调函数。因为只要拖动对象一接触墙壁，该函数就会被执行。
4. 增加另外一个放置对象以配置迷宫的末端，为此对象设置所接受的元素，即拖动元素。指定在拖动组件结束放置时执行的函数。
5. 把墙壁和末端元素设成放置组件。
6. 设置迷宫中的墙壁样式。





● 拖放组件

● 缩放组件

● 选择组件

● 排序组件

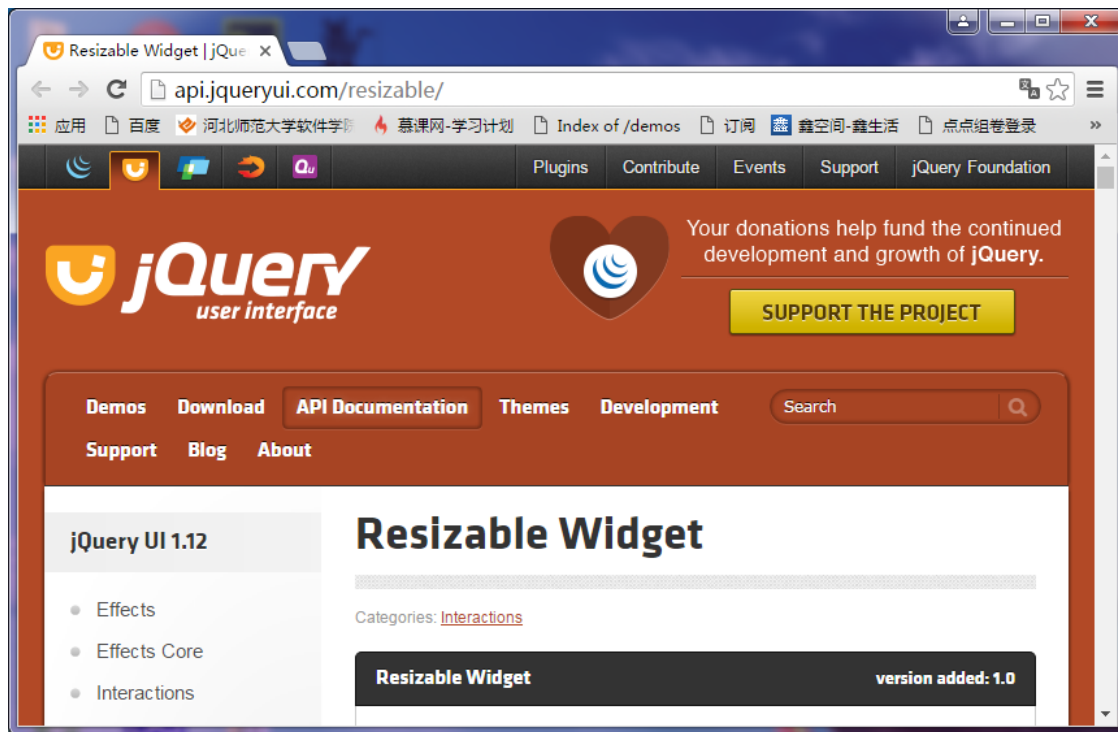


## 缩放组件

缩放 ( resizable )

是网页上常见的一种操作，使元素可以通过鼠标改变其尺寸。

jQuery UI允许缩放页面上的每一个元素。





## 缩放组件的基本结构

- 缩放组件resizable 功能可以应用在任意的 DOM 元素上。

```
<div class="resize">  
  <p>这是一个可以进行缩放元素  
  </p>  
</div>
```

```
$(function() {  
  $(".resize").resizable();  
});
```

这是一个可以进行缩  
放的元素

resizable1.html





为什么使用了 `resizable()` 方法会彻底改变已渲染页面中的HTML元素的外观

该方法遍历了HTML代码，在指定的可缩放元素上自动增加了特定的CSS类，赋予了它们适当的样式。





自定义缩放组件——需要使用指定的CSS class 名称

ui-resizable : 缩放元素的css。

ui-resizable-handle : 缩放手柄。

ui-resizable-e : 右侧拖动插件。

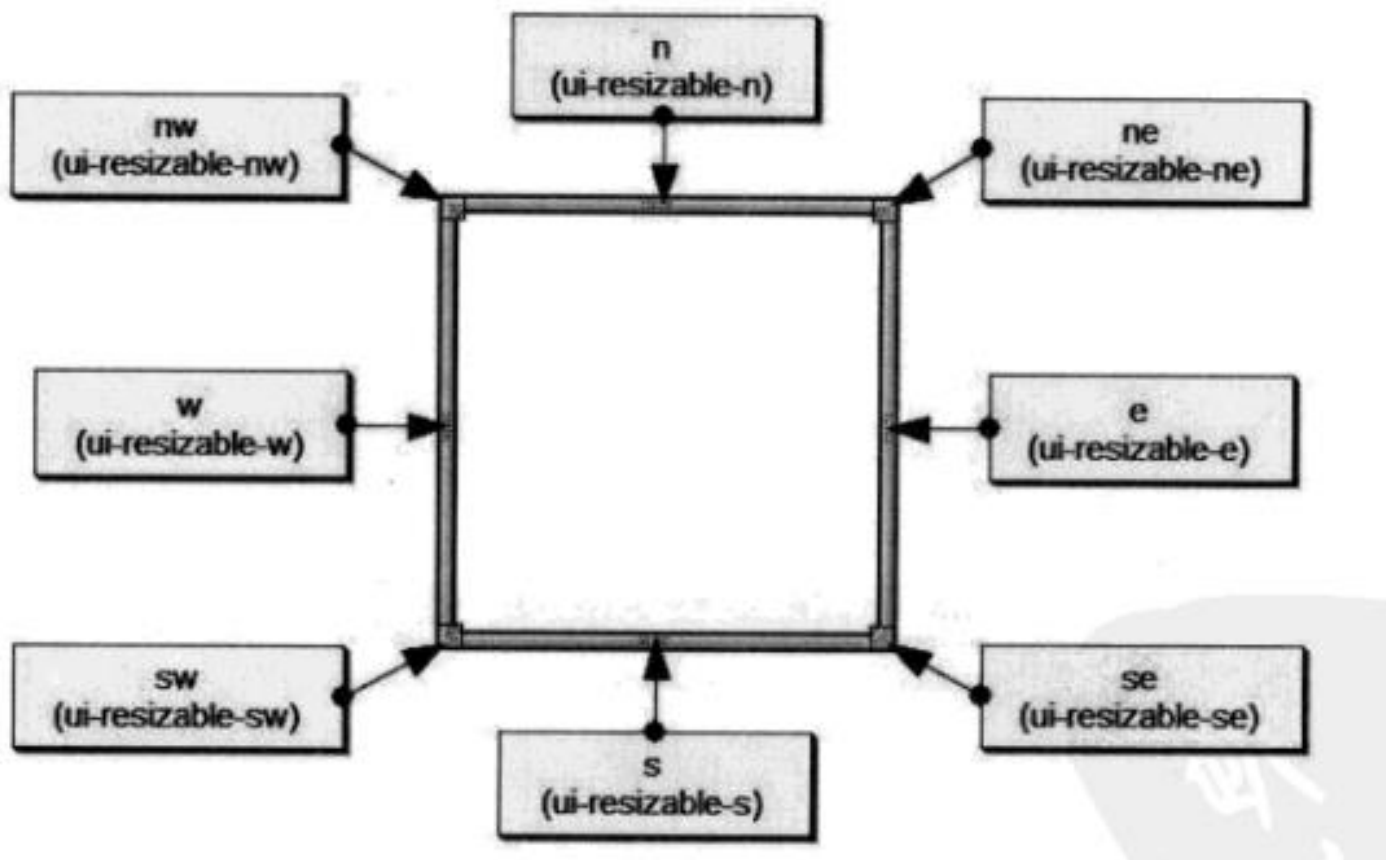
ui-resizable-s : 底部拖动插件。

ui-resizable-se : 右下角拖动插件。



## 缩放组件的样式

设置resizable插件允许生成在元素的哪个边上，可选值：n, s, w, e, ne, se, sw, nw, all





## 缩放组件的样式

```
.ui-resizable-e {  
    background:url(..../img/resizable-e.gif) repeat right center;  
}  
.ui-resizable-s {  
    background:url(..../img/resizable-s.gif) repeat center top;  
}  
.ui-resizable-se {  
    background:url(..../img/resizable-se.gif) repeat;  
}
```



resizable2.html



# resizable(options )方法

resizable(options )方法声明了一个在HTML页面中可以被缩放的HTML元素。options参数是一个对象，用来指定缩放时相关元素的行为。

```
$(selector, context) . draggable(options )
```



## 缩放组件的可配置属性

### 管理缩放元素及特效

属性	用途	默认值
<code>animate</code>	缩放是否带动画效果	false
<code>animateDuration</code>	动画的速度，可以使用毫秒数作为他的值，也可使用字符串，slow、normal、fast	slow
<code>alsoResize</code>	缩放时使用选择器同时改变另外一个元素	false
<code>autoHide</code>	隐藏尺寸改变手柄，直到鼠标指针指向该位置	false
<code>cancel</code>	其值为一个选择器，指定不可被缩放的元素	input
<code>delay</code>	缩放延迟	0
<code>helper</code>	激活helper元素以显示尺寸改变的过程	null

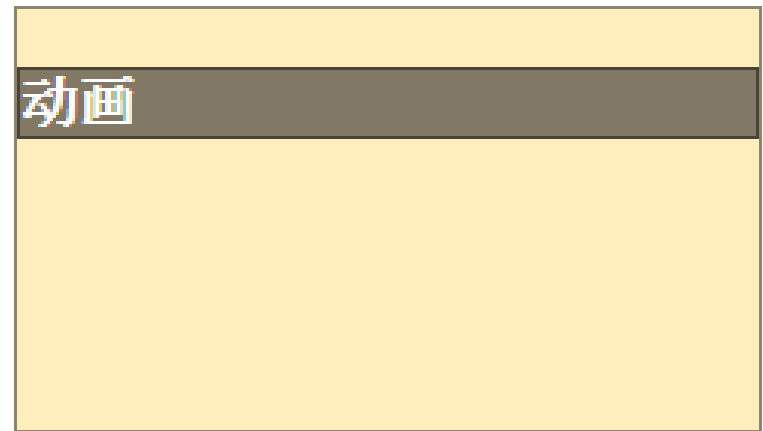


## 尺寸改变动画

使用 animate 选项（布尔值）使缩放行为动画化。

当该选项设置为 true 时，拖拽轮廓到所需的位置，元素会在拖拽停止时以动画形式调整到该尺寸。

```
$(function() {  
    $( "#resizable" ).resizable({  
        animate: true  
    });  
});
```



resizable3.html



## 缩放组件的可配置属性

### 对缩放做的限制

属性	用途	默认值
comtainment	限制缩放适用于哪个元素范围。此元素可以通过选择器、DOM元素或父元素来指定。	false
maxHeight	元素允许被放大到的最大高度	null
maxWidth	元素允许被放大到的最大宽度	null
minHeight	元素允许被缩小到的最小高度	10
minWidth	元素允许被缩小到的最小宽度	10





## 限制缩放区域

定义缩放区域的边界。使用 `containment` 选项来指定一个父级的 DOM 元素或一个 jQuery 选择器

```
$(function() {  
    $( "#resizable" ).resizable({  
        containment: "#container"  
    });  
});
```

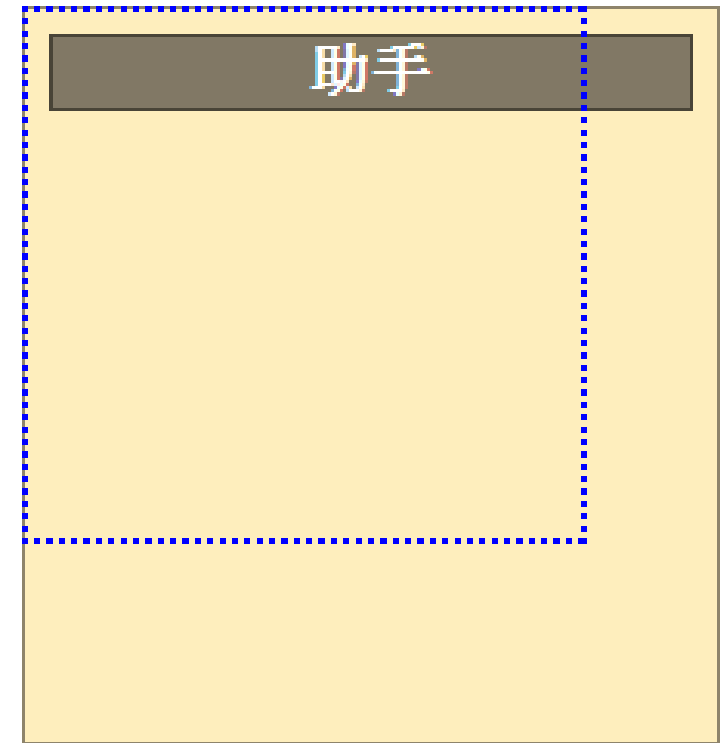


resizable4.html



通过设置 helper 选项为一个 CSS class，当缩放时只显示元素的轮廓。

```
$(function() {  
    $( "#resizable" ).resizable({  
        helper: "ui-resizable-helper"  
    });  
});
```



resizable5.html



● 拖放组件

● 缩放组件

● 选择组件

● 排序组件



## 选择组件

### 缩放组件 ( Selectable )

用来确定一系列可选的元素。

在某个 DOM 元素上或者一组元素上启用 selectable 功能。通过鼠标拖拽或点击的方式选择条目。按住 Ctrl 键，选择多个不相邻的条目。

```
$(function() {  
    $( "#selectable" ).selectable();  
});
```

This can be selected

This can be selected

This can be selected

This can be selected

selectable1.html



## Selectee样式类名

缩放组件 ( Selectable )

所有被设置为可选择的元素初始时都被设置了一个样式类 `ui-selectable`。

`ui-selecting` 选择方框包含了可选元素时

`ui-selected` 选择交互结束时，被选择的元素

`ui-unselecting` 先前被选过但现在不在所选范围



## 配置属性

属性	默认值	用途
autoRefresh	true	在开始选择操作前，自动刷新每个选项的位置和尺寸
filter	*	指定设为可选项的子元素

```
$(function() {  
    var selectableObj = {  
        filter: ".selectable"  
    }  
    $("#selectables").selectable(selectableObj);  
});
```

selectable2.html



## 自定义网格

让 selectable 条目显示为网格，使用 CSS 使得它们带有相同的尺寸且浮动显示。

1	2	3	4
5	6	7	8
9	10	11	12

selectable3.html



● 拖放组件

● 缩放组件

● 选择组件

● 排序组件





## 排序组件

### 排序组件 ( Sortable )

在任意的 DOM 元素上启用 sortable 功能。

通过鼠标点击并拖拽元素到列表中的一个新的位置，其它条目会自动调整。默认情况下，sortable 各个条目共享 draggable 属性。

```
$(function() {  
    $("#sortables").sortable();  
});
```



## 排序组件

sortable()方法遍历了HTML代码并给元素自动增加了特定的CSS类，赋予了它们适当的样式。

ui-sortable：整个课排序组件

ui-sortable-handle：可排序元素

ui-sortable-helper：正在被移动的元素

ui-sortable-helper：占位元素（div）



## 4 sortable(options)方法

sortable(options)方法声明了一个含有可调序元素的HTML元素。options参数是一个对象，用来指定与调序相关的行为。

```
$(selector, context) . sortable( options)
```

```
$(selector, context) . sortable( "action" , params)
```



## 4 sortable(options )方法

管理可移动元素。

选项	功能	默认值
disabled	如果设置为 true , 则禁用该 sortable。	false
cursor	定义当排序时被显示的光标。 crosshair(十字形)    default(箭头形) pointer(手形)        move(十字交叉箭头型) wait(系统繁忙)       help(帮助)	auto
delay	指定延迟时间, 只有在鼠标持续拖动超过延迟时间后, 才开始移动元素。	0毫秒
distance	指定一个距离, 以像素为单位, 只有在鼠标持续拖动超过此距离后, 才开始移动元素。	1像素



## 4 sortable(options )方法

管理可移动元素。

选项	功能	默认值
axis	限制排序组件只能沿一个方向轴拖动，x或y	none
containment	限制排序组件的条目在拖动时不能越过容器。 parent、window、document、jQuery选择器	parent
opacity	元素移动时的透明度设置	1(不透明 )
placeholder	指定用于放置排序元素的空白位置CSS样式类	none



## 占位符

占位符定义了预留给待排序条目的空白区域，作为它排序后的新位置。

占位符的位置并不固定，它随着待排序条目的移动而移动到将要替换的目标位置。

```
var sortOpts = {  
    placeholder: "empty"  
};
```

```
<style>  
    .empty { background-color:  
#cdfdcd; }  
</style>
```



## sortable(options )方法

管理可调序元素的事件。

选项	功能
start	当排序开始时触发该事件
stop	当排序停止时触发该事件
beforeStop	在stop之前被调用，此时占位元素仍然在列表中
sort	在排序期间触发该事件
change	在元素和一个被拖动元素调换位置之后被调用
update	当用户停止排序且 DOM 位置改变时触发该事件



谢谢