

jQuery UI 实例 - 拖动 (Draggable)

允许使用鼠标移动元素。

如需了解更多有关 draggable 交互的细节，请查看 API 文档 [可拖拽小部件 \(Draggable Widget \)](#)。

默认功能

在任意的 DOM 元素上启用 draggable 功能。通过鼠标点击并在视区中拖动来移动 draggable 对象。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 拖动 (Draggable) - 默认功能</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    #draggable { width: 150px; height: 150px; padding: 0.5em; }
  </style>
  <script>
    $(function() {
      $( "#draggable" ).draggable();
    });
  </script>
</head>
<body>

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

</body>
</html>
```

[查看演示](#)

自动滚动

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。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 拖动 (Draggable) - 自动滚动</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    #draggable, #draggable2, #draggable3 { width: 100px; height: 100px; padding: 0.5em; float: left; margin: 0 10px 10px 0; }
  </style>
  <script>
    $(function() {
      $( "#draggable" ).draggable({ scroll: true });
      $( "#draggable2" ).draggable({ scroll: true, scrollSensitivity: 100 });
      $( "#draggable3" ).draggable({ scroll: true, scrollSpeed: 100 });
    });
  </script>
</head>
<body>

  <div id="draggable" class="ui-widget-content">
    <p>Scroll 设置为 true, 默认设置</p>
  </div>

  <div id="draggable2" class="ui-widget-content">
    <p>scrollSensitivity 设置为 100</p>
  </div>

  <div id="draggable3" class="ui-widget-content">
    <p>scrollSpeed 设置为 100</p>
  </div>

  <div style="height: 5000px; width: 1px;"></div>

</body>
</html>
```

[查看演示](#)

约束运动

通过定义 draggable 区域的边界来约束每个 draggable 的运动。设置 `axis` 选项来限制 draggable 的路径为 x 轴或者 y 轴，或者使用 `containment` 选项来指定一个父级的 DOM 元素或者一个 jQuery 选择器，比如 'document.'。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 拖动 (Draggable) - 约束运动</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    .draggable { width: 90px; height: 90px; padding: 0.5em; float: left; margin: 0 10px 10px 0; }
    #draggable, #draggable2 { margin-bottom:20px; }
    #draggable { cursor: n-resize; }
    #draggable2 { cursor: e-resize; }
    #containment-wrapper { width: 95%; height:150px; border:2px solid #ccc; padding: 10px; }
    h3 { clear: left; }
  </style>
  <script>
    $(function() {
      $( "#draggable" ).draggable({ axis: "y" });
      $( "#draggable2" ).draggable({ axis: "x" });

      $( "#draggable3" ).draggable({ containment: "#containment-wrapper", scroll: false });
      $( "#draggable5" ).draggable({ containment: "parent" });
    });
  </script>
</head>
<body>

<h3>沿着轴约束运动：</h3>

<div id="draggable" class="draggable ui-widget-content">
  <p>只能垂直拖拽</p>
</div>

<div id="draggable2" class="draggable ui-widget-content">
  <p>只能水平拖拽</p>
</div>

<h3>或者在另一个 DOM 元素中约束运动：</h3>
<div id="containment-wrapper">
  <div id="draggable3" class="draggable ui-widget-content">
    <p>我被约束在盒子里</p>
  </div>
</div>
```

```
<div class="draggable ui-widget-content">
  <p id="draggable5" class="ui-widget-header">我被约束在父元素内</p>
</div>
</div>

</body>
</html>
```

[查看演示](#)

光标样式

当拖拽对象时定位光标。默认情况下，光标是出现在被拖拽对象的中间。使用 `cursorAt` 选项来指定相对于 `draggable` 的另一个位置（指定一个相对于 `top`、`right`、`bottom`、`left` 的像素值）。通过提供一个带有有效的 CSS 光标值的 `cursor` 选项，来自定义光标的外观。有效的 CSS 光标值包括：`default`、`move`、`pointer`、`crosshair`，等等。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 拖动 (Draggable) - 光标样式</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    #draggable, #draggable2, #draggable3 { width: 100px; height: 100px; padding: 0.5em; float: left; margin: 0 10px 10px 0; }
  </style>
  <script>
    $(function() {
      $( "#draggable" ).draggable({ cursor: "move", cursorAt: { top: 56, left: 56 } });
      $( "#draggable2" ).draggable({ cursor: "crosshair", cursorAt: { top: -5, left: -5 } });
      $( "#draggable3" ).draggable({ cursorAt: { bottom: 0 } });
    });
  </script>
</head>
<body>

  <div id="draggable" class="ui-widget-content">
    <p>我总是在中间（相对于鼠标）</p>
  </div>

  <div id="draggable2" class="ui-widget-content">
    <p>我的光标是在 left -5 和 top -5</p>
  </div>
```

```
<div id="draggable3" class="ui-widget-content">
  <p>我的光标位置只控制了 'bottom' 值</p>
</div>

</body>
</html>
```

[查看演示](#)

延迟开始

通过 `delay` 选项设置延迟开始拖拽的毫秒数。通过 `distance` 选项设置光标被按下且拖拽指定像素后才允许拖拽。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 拖动 (Draggable) - 延迟开始</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    #draggable, #draggable2 { width: 120px; height: 120px; padding: 0.5em; float: left; margin: 0 10px 10px 0; }
  </style>
  <script>
    $(function() {
      $( "#draggable" ).draggable({ distance: 20 });
      $( "#draggable2" ).draggable({ delay: 1000 });
      $( ".ui-draggable" ).disableSelection();
    });
  </script>
</head>
<body>

  <div id="draggable" class="ui-widget-content">
    <p>只有把我拖拽了 20 像素后，拖拽才开始</p>
  </div>

  <div id="draggable2" class="ui-widget-content">
    <p>不管 distance 是多少，您都必须拖拽并等待 1000ms 后拖拽才开始</p>
  </div>
```

```
</body>
</html>
```

[查看演示](#)

事件

draggable 上的 start、drag 和 stop 事件。拖拽开始时触发 start 事件，拖拽期间触发 drag 事件，拖拽停止时触发 stop 事件。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 拖动 (Draggable) - 事件</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    #draggable { width: 16em; padding: 0 1em; }
    #draggable ul li { margin: 1em 0; padding: 0.5em 0; } * html #draggable ul li { height: 1%; }
    #draggable ul li span.ui-icon { float: left; }
    #draggable ul li span.count { font-weight: bold; }
  </style>
  <script>
    $(function() {
      var $start_counter = $( "#event-start" ),
          $drag_counter = $( "#event-drag" ),
          $stop_counter = $( "#event-stop" ),
          counts = [ 0, 0, 0 ];

      $( "#draggable" ).draggable({
        start: function() {
          counts[ 0 ]++;
          updateCounterStatus( $start_counter, counts[ 0 ] );
        },
        drag: function() {
          counts[ 1 ]++;
          updateCounterStatus( $drag_counter, counts[ 1 ] );
        },
        stop: function() {
          counts[ 2 ]++;
          updateCounterStatus( $stop_counter, counts[ 2 ] );
        }
      });
    });
```

```

function updateCounterStatus( $event_counter, new_count ) {
    // 首先更新视觉状态...
    if ( !$event_counter.hasClass( "ui-state-hover" ) ) {
        $event_counter.addClass( "ui-state-hover" )
        .siblings().removeClass( "ui-state-hover" );
    }
    // ...然后更新数字
    $( "span.count", $event_counter ).text( new_count );
}
});
</script>
</head>
<body>

<div id="draggable" class="ui-widget ui-widget-content">

    <p>请拖拽我，触发一连串的事件。</p>

    <ul class="ui-helper-reset">
        <li id="event-start" class="ui-state-default ui-corner-all"><span class="ui-icon ui-icon-play"></span>"start" 被调用 <span class="count">0</span>x</li>
        <li id="event-drag" class="ui-state-default ui-corner-all"><span class="ui-icon ui-icon-arrow-4"></span>"drag" 被调用 <span class="count">0</span>x</li>
        <li id="event-stop" class="ui-state-default ui-corner-all"><span class="ui-icon ui-icon-stop"></span>"stop" 被调用 <span class="count">0</span>x</li>
    </ul>
</div>

</body>
</html>

```

[查看演示](#)

Handles

只有当光标在 draggable 上指定部分时才允许拖拽。使用 `handle` 选项来指定用于拖拽对象的元素（或元素组）的 jQuery 选择器。

Or prevent dragging when the cursor is over a specific element (or group of elements) within the draggable. Use the `cancel` option to specify a jQuery selector over which to "cancel" draggable functionality.

或者当光标在 draggable 内指定元素（或元素组）上时不允许拖拽。使用 `handle` 选项来指定取消拖拽功能的 jQuery 选择器。

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">

```

```
<title>jQuery UI 拖动 (Draggable) - Handles</title>
<link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
<script src="//code.jquery.com/jquery-1.9.1.js"></script>
<script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
<link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
<style>
#draggable, #draggable2 { width: 100px; height: 100px; padding: 0.5em; float: left; margin: 0 10px 10p
x 0; }
#draggable p { cursor: move; }
</style>
<script>
$(function() {
    $( "#draggable" ).draggable({ handle: "p" });
    $( "#draggable2" ).draggable({ cancel: "p.ui-widget-header" });
    $( "div, p" ).disableSelection();
});
</script>
</head>
<body>

<div id="draggable" class="ui-widget-content">
    <p class="ui-widget-header">您只可以在这个范围内拖拽我</p>
</div>

<div id="draggable2" class="ui-widget-content">
    <p>您可以把我向四周拖拽&hellip;</p>
    <p class="ui-widget-header">&hellip;但是您不可以在这个范围内拖拽我</p>
</div>

</body>
</html>
```

[查看演示](#)

还原位置

当带有布尔值 `revert` 选项的 `draggable` 停止拖拽时，返回 `draggable`（或它的助手）到原始位置。

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>jQuery UI 拖动 (Draggable) - 还原位置</title>
    <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
    <script src="//code.jquery.com/jquery-1.9.1.js"></script>
    <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
    <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
```



```
<style>
  #draggable, #draggable2 { width: 100px; height: 100px; padding: 0.5em; float: left; margin: 0 10px 10px 0; }
</style>
<script>
$(function() {
  $( "#draggable" ).draggable({ revert: true });
  $( "#draggable2" ).draggable({ revert: true, helper: "clone" });
});
</script>
</head>
<body>

<div id="draggable" class="ui-widget-content">
  <p>还原</p>
</div>

<div id="draggable2" class="ui-widget-content">
  <p>还原助手</p>
</div>

</body>
</html>
```

[查看演示](#)

对齐到元素或网格

对齐 draggable 到 DOM 元素的内部或外部边界。使用 `snap`、`snapMode` (`inner`, `outer`, `both`) 和 `snapTolerance` (当调用对齐时, draggable 与元素之间的距离, 以像素为单位) 选项。

或者对齐 draggable 到网格。通过 `grid` 选项设置网格单元的尺寸 (以像素为单位的高度或宽度)。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 拖动 (Draggable) - 对齐到元素或网格</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    .draggable { width: 90px; height: 80px; padding: 5px; float: left; margin: 0 10px 10px 0; font-size: .9em; }
    .ui-widget-header p, .ui-widget-content p { margin: 0; }
    #snaptarget { height: 140px; }
```

```
</style>
<script>
$(function() {
    $( "#draggable" ).draggable({ snap: true });
    $( "#draggable2" ).draggable({ snap: ".ui-widget-header" });
    $( "#draggable3" ).draggable({ snap: ".ui-widget-header", snapMode: "outer" });
    $( "#draggable4" ).draggable({ grid: [ 20, 20 ] });
    $( "#draggable5" ).draggable({ grid: [ 80, 80 ] });
});
</script>
</head>
<body>

<div id="snaptarget" class="ui-widget-header">
    <p>我是对齐目标</p>
</div>

<br style="clear:both">

<div id="draggable" class="draggable ui-widget-content">
    <p>默认 (snap: true) , 对齐到所有其他的 draggable 元素</p>
</div>

<div id="draggable2" class="draggable ui-widget-content">
    <p>我只对齐到大盒子</p>
</div>

<div id="draggable3" class="draggable ui-widget-content">
    <p>我只对齐到大盒子的外边缘</p>
</div>

<div id="draggable4" class="draggable ui-widget-content">
    <p>我对齐到一个 20 x 20 网格</p>
</div>

<div id="draggable5" class="draggable ui-widget-content">
    <p>我对齐到一个 80 x 80 网格</p>
</div>

</body>
</html>
```

[查看演示](#)

视觉反馈

给用户提供反馈，就像以助手方式拖拽对象一样。 `helper` 选项接受值 'original' (用光标移动 draggable 对象)，'clone' (用光标移动 draggable 的副本)，或者一个返回 DOM 元素的函数 (该元素在拖拽期间显示在光标附近)。通过 `opacity` 选项控制助手的透明度。

为了区别哪一个 draggable 正在被拖拽，让在运动中的 draggable 保持最前。如果正在拖拽，使用 `zIndex` 选项来设置助手的高度 z-index，或者使用 `stack` 选项来确保当停止拖拽时，最后一个被拖拽的项目总是出现在同组其他项目的上面。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 拖动 (Draggable) - 视觉反馈</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    #draggable, #draggable2, #draggable3, #set div { width: 90px; height: 90px; padding: 0.5em; float: left; margin: 0 10px 10px 0; }
    #draggable, #draggable2, #draggable3 { margin-bottom: 20px; }
    #set { clear: both; float: left; width: 368px; height: 120px; }
    p { clear: both; margin: 0; padding: 1em 0; }
  </style>
  <script>
    $(function() {
      $( "#draggable" ).draggable({ helper: "original" });
      $( "#draggable2" ).draggable({ opacity: 0.7, helper: "clone" });
      $( "#draggable3" ).draggable({
        cursor: "move",
        cursorAt: { top: -12, left: -20 },
        helper: function( event ) {
          return $( "<div class='ui-widget-header'>I'm a custom helper</div>" );
        }
      });
      $( "#set div" ).draggable({ stack: "#set div" });
    });
  </script>
</head>
<body>

<h3 class="docs">助手: </h3>

<div id="draggable" class="ui-widget-content">
  <p>原始的</p>
</div>

<div id="draggable2" class="ui-widget-content">
  <p>半透明的克隆</p>
```

```
</div>

<div id="draggable3" class="ui-widget-content">
  <p>自定义助手 (与 cursorAt 结合) </p>
</div>

<h3 class="docs">堆叠: </h3>
<div id="set">
  <div class="ui-widget-content">
    <p>我们是 draggables..</p>
  </div>

  <div class="ui-widget-content">
    <p>..它的 z-index 是自动控制的..</p>
  </div>

  <div class="ui-widget-content">
    <p>..带有 stack 选项。</p>
  </div>
</div>

</body>
</html>
```

[查看演示](#)

jQuery UI Draggable + Sortable

Draggable 与 Sortable 的无缝交互。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 拖动 (Draggable) + 排序 (Sortable) </title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    ul { list-style-type: none; margin: 0; padding: 0; margin-bottom: 10px; }
    li { margin: 5px; padding: 5px; width: 150px; }
  </style>
  <script>
    $(function() {
      $( "#sortable" ).sortable({
        revert: true
```

```
});  
$( "#draggable" ).draggable({  
    connectToSortable: "#sortable",  
    helper: "clone",  
    revert: "invalid"  
});  
$( "ul, li" ).disableSelection();  
});  
</script>  
</head>  
<body>  
  
<ul>  
    <li id="draggable" class="ui-state-highlight">请拖拽我</li>  
</ul>  
  
<ul id="sortable">  
    <li class="ui-state-default">Item 1</li>  
    <li class="ui-state-default">Item 2</li>  
    <li class="ui-state-default">Item 3</li>  
    <li class="ui-state-default">Item 4</li>  
    <li class="ui-state-default">Item 5</li>  
</ul>  
  
  
</body>  
</html>
```

[查看演示](#)[← jQuery UI API – 工具提示框部件 \(Tooltip Widget \)](#)[jQuery UI 实例 – 放置 \(Droppable \) →](#)[✎ 点我分享笔记](#)