

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

JqueryUI - Sortable

مدرس : مهندس افشین رفوآ

JqueryUI - Sortable

**jQueryUI** متد **sortable()** را ارائه می دهد تا دوباره عناصر را در یک لیست یا گرید با استفاده از ماوس، مرتب سازی کنیم. این متد عملکرد مرتب سازی را براساس یک رشته که به عنوان اولین پارامتر منتقل می شود، اجرا می کند.

ترکیب:

متد به دو فرم مورد استفاده قرار می گیرد:

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

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

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

متد **sortable (options)** اعلام می کند که یک عنصر **HTML** حاوی عناصر قابل تعویض است. پارامتر **options** آبجکتی است که رفتار عناصر را در زمان مرتب سازی مجدد، تعیین می کند.

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

جدول زیر گزینه ای مختلفی را ارائه می دهد که با این متد مورد استفاده قرار می گیرند:

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 561 - واحد 7

88146323 - 88446780 - 88146330

Option	Description
appendTo	این گزینه عنصری را تعیین می کند که در آن عنصر جدید با options ایجاد شده و helper در زمان جابه جایی / درگ وارد خواهد شد. مقدار پیش فرض آن parent است.
axis	این گزینه مختصات جابجایی را نشان می دهد ( X افقی و Y عمودی می باشد. ). مقدار پیش فرض آن false است.
cancel	این گزینه برای جلوگیری از مرتب سازی عناصر با کلیک کردن بر روی هرکدام از این انتخابگرها، استفاده می شود. مقدار پیش فرض آن "input,textarea,button,select,option" می باشد.
connectWith	این گزینه یک selector است که یک عنصر قابل مرتب سازی دیگر را تشخیص می دهد که می تواند آیتم هایی را بپذیرد. یک گزینه به آیتم ها اجازه می دهد تا به لیست های دیگر جابه جا شوند که یک تعامل مفید و متداول با یوزر است. اگر حذف شود، هیچ کدام از عناصر با یکدیگر در ارتباط نیستند. مقدار پیش فرض آن false است.
containment	این گزینه عنصری را نشان می دهد که داخل آن جایگزینی اتفاق می افتد. عنصر به وسیله ی یک انتخابگر ( تنها اولین آیتم در لیست مورد توجه قرار خواهد گرفت)، یک DOM، یا یک رشته ی parent (parent element) یا "window" (صفحه ی HTML) نمایش داده می شود.
cursor	پراپرتی مکان نمای CSS را در هنگام جابه جایی، تعیین می کند. این گزینه شکل مکان نمای ماوس را نشان می دهد. مقدار پیش فرض آن auto است.

cursorAt	افست کمک کننده ی درگ را در ارتباط با مکان نمای ماوس، تنظیم می کند. مختصات به عنوان یک رشته ی hash با استفاده از اרכیب یک یا دو کلید، ارائه می شود: { top, left, right, bottom }. مقدار پیش فرض آن false است.
delay	تاخیری به اندازه ی هزارم ثانیه، پس از اینکه اولین جابه جایی ماوس اتفاق می افتد. جایگزینی ممکن است پس از این زمان اتفاق بیفتد. مقدار پیش فرض آن 0 است.
disabled	این گزینه اگر روی true تنظیم شده باشد، قابلیت مرتب سازی را غیرفعال می سازد. مقدار پیش فرض آن false است.
distance	تعداد پیکسل هایی که قبل از شروع مرتب سازی ماوس باید جابه جا شود. اگر تعیین شده باشد، تا زمانیکه ماوس بیشتر از این میزان جابه جا شود، مرتب سازی شروع نخواهد شد. مقدار پیش فرض آن 1 است.
dropOnEmpty	اگر این گزینه روی false تنظیم شده باشد، آیتم ها نمی توانند در اتصال یک مرتب سازی خالی دراپ شوند. مقدار پیش فرض آن true است.
forceHelperSize	اگر این گزینه روی true تنظیم شده باشد، کمک کننده را وادار به داشتن یک سایز می کند. مقدار پیش فرض آن false است.
forcePlaceholderSize	وقتی این گزینه روی true تنظیم شده باشد، وقتی آیتم جابه جا می شود، اندازه ی place holder را محاسبه می کند. این گزینه تنها در صورتی مفید است که options.placeholder آغاز شود. مقدار پیش فرض آن false است.

grid	این گزینه یک ردیف [x, y] است که تعداد پیکسل هایی را نشان می دهد که عنصر مرتب سازی، به صورت افقی یا عمودی در زمان جایگزینی ماوس، جابه جا می کند. مقدار پیش فرض آن false است.
handle	اگر مشخص شده باشد، مرتب سازی را از ابتدا محدود می سازد، مگر اینکه mousedown اتفاق بیفتد. مقدار پیش فرض آن false است.
helper	به یک عنصر کمک کننده اجازه می دهد تا مورد استفاده قرار بگیرد. مقدار پیش فرض آن original است.
items	این گزینه تعیین می کند که کدام آیتم ها در داخل عنصر DOM مرتب سازی شوند. مقدار پیش فرض آن * > است.
opacity	این گزینه میزان محو بودن کمک کننده را در هنگام مرتب سازی مشخص می کند. مقدار پیش فرض آن false است.
placeholder	این گزینه برای نام گروه یا در غیر اینصورت برای فضای سفید، استفاده می شود. مقدار پیش فرض آن false است.
revert	این گزینه تعیین می کند که آیا آیتم های قابل مرتب سازی باید با استفاده از یک انیمیشن ساده به موقعیت های جدید خود بازگردند. مقدار پیش فرض آنها false است.
scroll	این گزینه برای غیرفعال ساختن اسکرول استفاده می شود. اگر روی true تنظیم شده باشد، در هنگام رسیدن به لبه، اسکرول در صفحه اتفاق می افتد. مقدار پیش فرض آن true است.

scrollSensitivity	این گزینه تعداد پیکسل هایی را نشان می دهد که ماوس باید از منطقه دید خارج شود تا اسکرول اتفاق بیفتد. مقدار پیش فرض آن 20 است. این گزینه تنها با گزینه های اسکرول تنظیم شده به true استفاده می شود.
scrollSpeed	این گزینه سرعت اسکرول شدن را هنگامی که اسکرول شدن آغاز می شود، نشان می دهد. مقدار پیش فرض آن 20 است.
tolerance	این گزینه یک String است که تعیین می کند که کدام وضعیت استفاده شود برای امتحان کردن این مطلب که آیا آیتم جابه جا شده در حال حرکت روی یک آیتم دیگر است. مقدار پیش فرض آن "intersect" است.
zIndex	این گزینه z-index را برای element/helper نمایش می دهد در حالیکه ذخیره شده است. مقدار پیش فرض آن 1000 است.

بخش بعد چند مثال کاربردی در مورد قابلیت درگ به شما نشان می دهد.

### قابلیت پیش فرض:

مثال زیر قابلیت مرتب سازی را توضیح می دهد و هیچ پارامتری را به متد sortable() انتقال نمی دهد.

```
<!DOCTYPE html>
<head>
  <title>jQuery UI Sortable - Example</title>
  <link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css" rel="stylesheet">
  <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
  <script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <style>
    #sortable-1 {
      list-style-type: none;
      margin: 0;
      padding: 0;
      width: 25%;
    }
    #sortable-1 li {
      margin: 0 3px 3px 3px;
      padding: 0.4em;
      padding-left: 1.5em;
    }
  </style>
</head>
<body>
  <ul style="list-style-type: none;" id="sortable-1">
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
    <li>Item 4</li>
    <li>Item 5</li>
  </ul>
</body>
</html>
```

```

        font-size: 17px;
        height: 16px;
    }
    .default {
        background: #cedc98;
        border: 1px solid #DDDDDD;
        color: #333333;
    }
</style>
<script>
$(function () {
    $("#sortable-1").sortable();
});
</script>
</head>
<body>
<ul id="sortable-1">
<li class="default">Product 1</li>
<li class="default">Product 2</li>
<li class="default">Product 3</li>
<li class="default">Product 4</li>
<li class="default">Product 5</li>
<li class="default">Product 6</li>
<li class="default">Product 7</li>
</ul>
</body>
</html>

```

توليدات بالا را دوباره تنظيم كنيد، از ماوس براي درگ كردن آيتم ها استفاده كنيد.

استفاده از گزينه هاي `distance` و `delay`

مثال زير استفاده از دو گزينه `delay` (a) و `distance` (b) را در قابليت مرتب سازي JQueryUI توضيح

مي دهد.

```

<!DOCTYPE html>
<head>
<title>jQuery UI Sortable - Example</title>
<link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css" rel="stylesheet">
<script src="http://code.jquery.com/jquery-1.10.2.js"></script>
<script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
<style>
#sortable-2, #sortable-3 {
    list-style-type: none;
    margin: 0;
    padding: 0;
    width: 25%;
}
#sortable-2 li, #sortable-3 li {

```

```

margin: 0 3px 3px 3px;
padding: 0.4em;
padding-left: 1.5em;
font-size: 17px;
height: 16px;
}
.default {
background: #cedc98;
border: 1px solid #DDDDDD;
color: #333333;
}
}
</style>
<script>
$(function () {
  $("#sortable-2").sortable({
    delay: 500
  });
  $("#sortable-3").sortable({
    distance: 30
  });
});
</script>
</head>
<body>
<h3>Delay by 500ms</h3>
<ul id="sortable-2">
<li class="default">Product 1</li>
<li class="default">Product 2</li>
<li class="default">Product 3</li>
<li class="default">Product 4</li>
</ul>
<h3>Distance Delay by 30px</h3>
<ul id="sortable-3">
<li class="default">Product 1</li>
<li class="default">Product 2</li>
<li class="default">Product 3</li>
<li class="default">Product 4</li>
</ul>
</body>
</html>

```

تولیدات بالا را مجدداً تنظیم کنید و از ماوس برای درگ کردن آیتم‌ها استفاده کنید. به خاطر جلوگیری از مرتب‌سازی تصادفی چه با زمان و چه با فاصله، ما چند هزارم ثانیه برای عنصری تنظیم کرده ایم که قبل از شروع شدن مرتب‌سازی نیاز به درگ شدن دارد. ما این کار را با استفاده از گزینه **distance** نیز انجام داده ایم.

[استفاده از placeholder](#)

مثال زیر استفاده از گزینه **placeholder** را در عملکرد مرتب سازی **JqueryUI** نشان می دهد.

```
<!DOCTYPE html>
<head>
  <title>jQuery UI Sortable - Example</title>
  <link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css" rel="stylesheet">
  <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
  <script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <style>
    #sortable-4 {
      list-style-type: none;
      margin: 0;
      padding: 0;
      width: 25%;
    }
    #sortable-4 li {
      margin: 0 3px 3px 3px;
      padding: 0.4em;
      padding-left: 1.5em;
      font-size: 17px;
      height: 16px;
    }
    .highlight {
      border: 1px solid red;
      font-weight: bold;
      font-size: 45px;
      background-color: #333333;
    }
    .default {
      background: #cedc98;
      border: 1px solid #DDDDDD;
      color: #333333;
    }
  </style>
  <script>
    $(function () {
      $("#sortable-4").sortable({
        placeholder: "highlight"
      });
    });
  </script>
</head>
<body>
  <ul id="sortable-4">
    <li class="default">Product 1</li>
    <li class="default">Product 2</li>
    <li class="default">Product 3</li>
    <li class="default">Product 4</li>
    <li class="default">Product 5</li>
    <li class="default">Product 6</li>
    <li class="default">Product 7</li>
  </ul>
</body>
</html>
```



## استفاده از گزینه های connectWith و dropOnEmpty

مثال زیر استفاده از گزینه های connectWith (a) و dropOnEmpty (b) را در عملکرد مرتب

سازی JQueryUI توضیح می دهد.

```
<!DOCTYPE html>
<head>
  <title>jQuery UI Sortable - Example</title>
  <link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css" rel="stylesheet">
  <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
  <script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <style>
    #sortable-5, #sortable-6, #sortable-7 {
      list-style-type: none;
      margin: 0;
      padding: 0;
      width: 20%;
      float: left;
    }
    #sortable-5 li, #sortable-6 li, #sortable-7 li {
      margin: 0 3px 3px 3px;
      padding: 0.4em;
      padding-left: 1.5em;
      font-size: 17px;
      height: 16px;
    }
    .default {
      background: #cedc98;
      border: 1px solid #DDDDDD;
      color: #333333;
    }
  </style>
  <script>
    $(function () {
      $("#sortable-5, #sortable-6").sortable({
        connectWith: "#sortable-5, #sortable-6"
      });
      $("#sortable-7").sortable({
        connectWith: "#sortable-5",
        dropOnEmpty: false
      });
    });
  </script>
</head>
<body>
  <ul id="sortable-5">
    <h3>List 1</h3>
    <li class="default">A</li>
    <li class="default">B</li>
    <li class="default">C</li>
    <li class="default">D</li>
  </ul>
  <ul id="sortable-6">
```

```

<h3>List 2</h3>
<li class="default">a</li>
<li class="default">b</li>
<li class="default">c</li>
<li class="default">d</li>
</ul>
<ul id="sortable-7">
<h3>List 3</h3>
<li class="default">e</li>
<li class="default">f</li>
<li class="default">g</li>
<li class="default">h</li>
</ul>
</body>
</html>

```

آیتم ها را از یک لیست **List1** به لیست دیگر **List 2** و بالعکس، با انتقال یک انتخابگر به **connectWith**، مرتب سازی کنید. این کار به وسیله ی گروه بندی همه ی لیست های مرتبط با یک گروه **CSS** و سپس انتقال آن گروه به عملکرد **sortable**، انجام می شود. ( برای مثال : **connectWith: '#sortable-5, #sortable-6**).

سعی کنید آیتم ها را به تحت عنوان **List 3** به **List2** یا **List 1** درگ کنید. همانطور که گزینه ی **dropOnEmpty** را به **false** تنظیم کرده ایم، دراپ کردن این آیتم ها امکان پذیر نیست.

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

متد **sortable (action, params)** می تواند اکشنی را روی عناصر قابل مرتب شدن انجام دهد، از جمله جلوگیری از جایگزینی. این اکشن به عنوان یک رشته در اولین برهان تعیین شده و به طور انتخابی، یک یا چند **param** براساس اکشن ارائه شده، ایجاد می شوند.

اساسا اکشن ها در اینجا چیزی به جز متدهای **jQuery** نیستند، که می توانیم به شکل رشته از آنها استفاده کنیم.

ترکیب:

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

جدول زیر اکشن های مربوط به این متد را ارائه می دهد:

Action	Description
cancel()	این اکشن عملکرد مرتب سازی فعلی را کنسل می کند. این متد در بین گرداننده ها برای دریافت مرتب سازی و متوقف کردن آن، بسیار مفید است. این متد هیچ برهانی نمی پذیرد.
destroy()	این اکشن عملکرد قابلیت مرتب سازی را کاملا حذف می کند و عناصر را به حالت pre-init خود بازمی گرداند. این متد هیچ برهانی نمی پذیرد.
disable()	این اکشن هرگونه قابلیت مرتب سازی را در مجموعه غیرفعال می سازد. به هر حال قابلیت مرتب سازی عناصر حذف نمی شود و با فراخوانی متغیر فعال این متد، قابل بازگشت است. این متد هیچ برهانی نمی پذیرد.
enable()	قابلیت مرتب سازی را مجددا روی هر کدام از عناصر sortable در مجموعه، فعال می سازد. دقت داشته باشید که این متد sortability را به هیچ کدام از عناصر اضافه نمی کند. این متد هیچ برهانی نمی پذیرد.
option( optionName )	این اکشن مقدار مربوط به optionName تعیین شده را دریافت می کند، که optionName نام گزینه ای است که دریافت می کنیم.
option()	آبجکتی حاوی جفت های key/value را دریافت می کند که گزینه های قابل مرتب سازی hash را نمایش می دهند. این متد هیچ برهانی نمی پذیرد.
option( optionName, value )	این اکشن مقدار گزینه ی قابل مرتب سازی را با optionName تعیین شده، تنظیم می کند، که optionName نام گزینه ای است برای تنظیم و value مقداری است که برای گزینه تنظیم می شود.

option( options )	یک یا چند گزینه را برای sortable تنظیم می کند. در اینجا options نقشه ای است از جفت های option-value برای تنظیم.
refresh()	این اکشن در صورت لزوم لیست آیتم ها را تازه سازی می کند. این متد هیچ برهانی را نمی پذیرد. فراخوانی این متد باعث می شود آیتم های جدید برای تشخیص به sortable اضافه شوند.
toArray( options )	این متد یک ردیف از مقادیر id از عناصر قابل مرتب سازی را در ترتیب ذخیره شده، بازمی گرداند. این متد options را به عنوان پارامتر خود دارد تا ترتیب ذخیره شده را به حالت شخصی در آورد.
serialize( options )	این متد یک رشته ی query سریالی شده را بازمی گرداند که از sortable شکل گرفته است.
refreshPositions()	این متد بیشتر به صورت داخلی برای تازه سازی اطلاعات حافظه داخلی استفاده می شود. این متد هیچ برهانی (argument) را نمی پذیرد.
widget()	این متد یک آبجکت jQuery حاوی عنصر قابل مرتب سازی را باز می گرداند. این متد هیچ برهانی (argument) را نمی پذیرد.

مثال:

اکنون اجازه بدهید مثالی را بررسی کنیم که از اکشن های جدول بالا استفاده می کند. مثال زیر استفاده از متد **toArray( options )** را توضیح می دهد.

```
<!DOCTYPE html>
<head>
  <title>jQuery UI Sortable - Example</title>
  <link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css" rel="stylesheet">
  <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
  <script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <style>
```

```

#sortable-8 {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 25%;
  float: left;
}
#sortable-8 li {
  margin: 0 3px 3px 3px;
  padding: 0.4em;
  padding-left: 1.5em;
  font-size: 17px;
  height: 16px;
}
.default {
  background: #cedc98;
  border: 1px solid #DDDDDD;
  color: #333333;
}
</style>
<script>
$(function () {
  $('#sortable-8').sortable({
    update: function (event, ui) {
      var productOrder = $(this).sortable('toArray').toString();
      $('#sortable-9').text(productOrder);
    }
  });
});
</script>
</head>
<body>
<ul id="sortable-8">
  <li id="1" class="default">Product 1</li>
  <li id="2" class="default">Product 2</li>
  <li id="3" class="default">Product 3</li>
  <li id="4" class="default">Product 4</li>
</ul>
<br>
<h3><span id="sortable-9"></span></h3>
</body>
</html>

```

مرتب سازی آیتم ها را امتحان کنید، ترتیب آیتم ها در پایین نمایش داده شده است. در اینجا ما

`$(this).sortable('toArray').toString()` را فرا می خوانیم که لیستی از همه ی آیتم های `id` ارائه می دهد

و ممکن است به صورت `1,2,3,4` به نظر برسد.

مدیریت رویداد روی عناصر قابل مرتب سازی:

علاوه بر متد **sortable (options)** که در بخش های قبلی مشاهده کردیم، **JqueryUI** متدهایی را ارائه می دهد که برای یک رویداد خاص آغاز می شوند. این متدها در جدول زیر ارائه شده اند:

Event Method	Description
activate(event, ui)	هنگامی که یک عملکرد مرتب سازی روی یک sortable متصل آغاز می شود، این رویداد شروع به کار می کند.
beforeStop(event, ui)	زمانی که یک عملکرد مرتب سازی رو به پایان است، این رویداد با کمک کننده و placeholder که هنوز معتبر هستند، شروع می شود.
change(event, ui)	زمانی که عنصر مرتب شده موقعیت خود را در داخل DOM تغییر دهد، این رویداد آغاز می شود.
create(event, ui)	هنگامی که sortable ایجاد می شود، این رویداد آغاز می شود.
deactivate(event, ui)	وقتی که یک مرتب سازی متصل متوقف می شود و در حال نشر روی sortable متصل است، این رویداد آغاز می شود.
out(event, ui)	این رویداد زمانی آغاز می شود که آیتم مرتب سازی از لیست متصل دور می شود.
over(event, ui)	این رویداد وقتی آغاز می شود که یک آیتم مرتب سازی (sort) وارد لیست اتصال می شود.
receive(event, ui)	این رویداد زمانی آغاز می شود که یک لیست اتصال یک آیتم مرتب سازی از یک لیست دیگر دریافت کرده است.

remove(event, ui)	این رویداد زمانی آغاز می شود که آیتم sort از لیست اتصال حذف شده و به لیست دیگری درگ می شود.
sort(event, ui)	این رویداد مکررا با حرکت ماوس در زمان یک عملکرد مرتب سازی آغاز می شود.
start(event, ui)	این رویداد زمانی آغاز می شود که یک عملکرد مرتب سازی آغاز می شود.
stop(event, ui)	این رویداد زمانی آغاز می شود که یک عملکرد مرتب سازی به پایان رسیده است.
update(event, ui)	این رویداد زمانی آغاز می شود که یک عملکرد مرتب سازی متوقف شده و موقعیت آیتم نیز تغییر کرده است.

#### مثال:

مثال زیر استفاده از متدهای رویداد را در زمان عملکرد **drop** نشان می دهد. این مثال در واقع استفاده از متدهای **start**, **receive** و **stop** را توضیح می دهد.

```
<!DOCTYPE html>
<head>
<title>jQuery UI Sortable - Example</title>
<link href="http://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css" rel="stylesheet">
<script src="http://code.jquery.com/jquery-1.10.2.js"></script>
<script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
<style>
#sortable-10, #sortable-11 {
list-style-type: none;
margin: 0;
padding: 0;
width: 80%;
}
#sortable-10 li, #sortable-11 li {
margin: 0 3px 3px 3px;
padding: 0.4em;
padding-left: 1.5em;
font-size: 17px;
height: 16px;
}
.highlight {
border: 1px solid #000000;
font-weight: bold;
}
```

```

font-size: 45px;
background-color: #cedc98;
}
.default {
background: #cedc98;
border: 1px solid #DDDDDD;
color: #333333;
}
.wrap {
display: table-row-group;
}
.wrap1 {
float: left;
width: 100px;
}
</style>
<script>
$(function () {
$("#sortable-10").sortable({
start: function (event, ui) {
$("#span#result").html($("#span#result").html()
+ "<b>start</b><br>");
},
receive: function (event, ui) {
$("#span#result").html($("#span#result").html()
+ ", receive");
},
stop: function (event, ui) {
$("#span#result").html($("#span#result").html()
+ "<b>stop</b><br>");
}
});
$("#sortable-11").sortable({
connectWith: "#sortable-10, #sortable-11"
});
});
</script>
</head>
<body>
<div class="wrap">
<div class="wrap1">
<h3>List 1</h3>
<ul id="sortable-10">
<li class="default">A</li>
<li class="default">B</li>
<li class="default">C</li>
<li class="default">D</li>
</ul>
</div>
<div class="wrap1">
<h3>List 2</h3>
<ul id="sortable-11">
<li class="default">a</li>
<li class="default">b</li>
<li class="default">c</li>
<li class="default">d</li>
</ul>
</div>

```



```
</ul>
</div>
</div>
<hr />
<span id=result></span>
</body>
</html>
```

سعی کنید آیتم ها را در **List 1** مرتب سازی کنید، مشاهده خواهید کرد که پیغام در رویداد **stop** و **start** نمایش داده می شود. اکنون آیتم ها را از **List 2** به **List 1** قرار دهید، مجدداً یک پیغام روی رویداد **receive** نمایش داده می شود.