



آموزش

JQUERY UI

www.tahlildadeh.com

نویسنده: مهندس افشین رفوآ

بسم الله الرحمن الرحيم

آموزشگاه تحلیلی داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

کتاب آموزشی JQueryUI

آموزشگاه تحلیلی داده
مؤلف مهندس افشین رفوآ



تقدیم به همه جویندگان علم که توان وامکان شرکت در کلاس های حضوری ما را ندارند.

فهرست

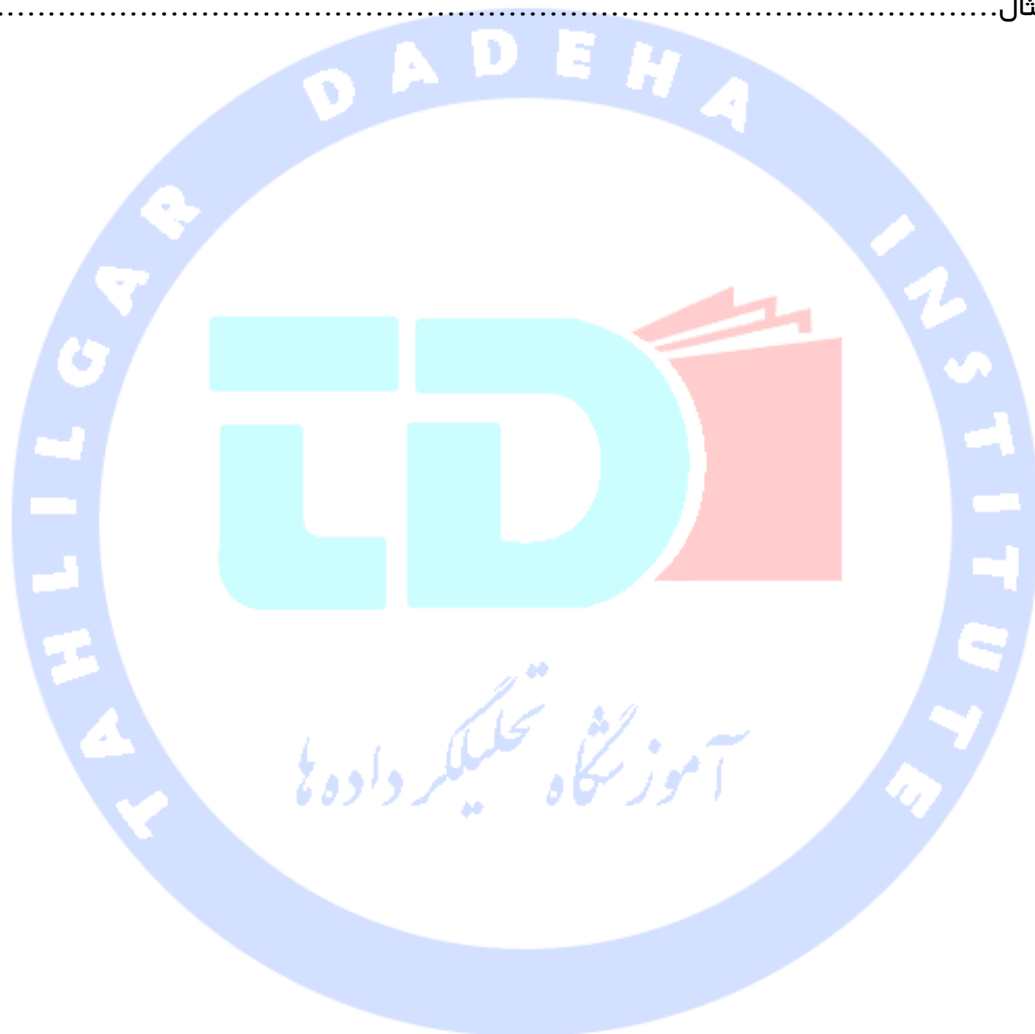
| | | |
|----|-------|--|
| 9 | | مقدمه |
| 9 | | توجه |
| 10 | | بررسی اجمالی JQueryUI |
| 10 | | ویژگی ها |
| 11 | | مزایای JQueryUI |
| 11 | | آموزش JQueryUI |
| 12 | | بینندگان |
| 12 | | پیش نیازها |
| 12 | | JqueryUI - Environment Setup |
| 13 | | دانلود خاص با Download Builder |
| 16 | | دانلود پایدار |
| 16 | | legacy |
| 17 | | دانلود UI Library از CDNs |
| 19 | | قابل جابجایی |
| 19 | | متد draggable() به دو فرم زیر مورد استفاده قرار می گیرد |
| 19 | | متد \$(selector, context).draggable (options) |
| 27 | | متد \$(selector, context).draggable ("action", [params]) |
| 30 | | JqueryUI - Droppable |
| 30 | | متد droppable() به دو روش مورد استفاده قرار می گیرد |
| 30 | | متد \$(selector, context).droppable (options) |
| 39 | | متد \$(selector, context).droppable ("action", params) |
| 44 | | JqueryUI - Resizable |
| 44 | | متد resizable() به دو شکل مورد استفاده قرار می گیرد |
| 45 | | متد \$(selector, context).resizable (options) |
| 55 | | متد \$(selector, context).resizable ("action", params) |
| 59 | | متد selectable در JQueryUI |
| 60 | | متد selectable() به دو شکل مورد استفاده قرار می گیرد |

| | |
|-----|---|
| 60 | متد \$(selector, context).selectable (options) |
| 66 | متد \$(selector, context).selectable ("action", params) |
| 71 | JqueryUI - Sortable |
| 71 | متد به دو فرم مورد استفاده قرار می گیرد |
| 71 | متد \$(selector, context).sortable (options) |
| 80 | متد \$(selector, context).sortable ("action", [params]) |
| 87 | Accordion Widget |
| 87 | متد accordion() به دو فرم استفاده می شود |
| 87 | متد \$(selector, context).accordion (options) |
| 96 | متد \$(selector, context).accordion ("action", params) |
| 106 | JqueryUI - Autocomplete |
| 107 | متد autocomplete() به دو فرم زیر استفاده می شود |
| 107 | متد \$(selector, context).autocomplete (options) |
| 113 | متد \$(selector, context).autocomplete ("action", params) |
| 119 | JqueryUI - Button |
| 119 | متد button() می تواند به دو شکل زیر استفاده می شود |
| 119 | متد \$(selector, context).button (options) |
| 122 | متد \$(selector, context).button ("action", params) |
| 126 | JqueryUI - DatePicker |
| 126 | متد datepicker() به دو شکل زیر مورد استفاده قرار می گیرد |
| 126 | متد \$(selector, context).datepicker (options) |
| 142 | متد \$(selector, context).datepicker ("action", [params]) |
| 144 | JqueryUI - Dialog |
| 145 | متد dialog() به دو فرم زیر استفاده می شود |
| 145 | متد \$(selector, context).dialog (options) |
| 152 | متد \$(selector, context).dialog ("action", [params]) |
| 158 | JqueryUI – Menu |
| 159 | متد menu() به دو شکل مورد استفاده قرار می گیرد |

| | |
|-----|--|
| 159 | متد \$(selector, context).menu (options) |
| 162 | متد \$(selector, context).menu ("action", params) |
| 169 | JqueryUI - Progressbar |
| 169 | متد progressbar() به دو روش زیر انجام می شود. |
| 170 | متد \$(selector, context).progressbar (options) |
| 172 | متد \$(selector, context).progressbar ("action", params) |
| 177 | JqueryUI - Slider |
| 177 | متد slider() به دو فرم قابل استفاده می باشد |
| 177 | متد \$(selector, context).slider (options) |
| 182 | متد \$(selector, context).slider ("action", params) |
| 187 | JqueryUI - Spinner |
| 187 | متد spinner() به دو فرم زیر قابل استفاده می باشد |
| 188 | متد \$(selector, context).spinner (options) |
| 193 | متد \$(selector, context).spinner ("action", params) |
| 198 | Extension Points |
| 199 | JqueryUI - Tooltip |
| 200 | متد به دو فرم زیر قابل استفاده می باشد |
| 200 | متد \$(selector, context).tooltip (options) |
| 204 | متد \$(selector, context).tooltip ("action", [params]) |
| 207 | JqueryUI - Position |
| 210 | JqueryUI - Widget Factory |
| 211 | ترکیب |
| 211 | Base Widget |
| 211 | گزینه ها |
| 212 | متدها |
| 215 | Events |
| 216 | افزودن گزینه ها به custom widget |
| 217 | افزودن متدها به custom widget |

| | |
|-----|--|
| 219 | افزودن رویدادها به Custom Widget |
| 220 | JqueryUI - Add Class |
| 220 | متد addClass() به شکل زیر می باشد |
| 224 | JqueryUI - Color Animation |
| 224 | ترکیب |
| 226 | JqueryUI - Effect |
| 226 | متد effect() دارای ترکیب زیر می باشد |
| 227 | افکت های JQueryUI |
| 228 | افکت - حرکت |
| 229 | افکت - انفجار |
| 230 | JqueryUI - Hide |
| 230 | متد hide() دارای ترکیب زیر است |
| 231 | افکت های JQueryUI |
| 231 | افکت - Blind |
| 232 | افکت - shake |
| 233 | JqueryUI - Remove Class |
| 234 | متد removeClass() با ترکیب اصلی خود به شکل زیر می باشد |
| 235 | مثال |
| 235 | Passing single class |
| 236 | JqueryUI - Show |
| 237 | متود show() دارای ترکیب زیر است |
| 237 | افکت های JQueryUI |
| 239 | مثال |
| 240 | نمایش با افکت Blind |
| 242 | JqueryUI - Switch Class |
| 242 | متد switchClass() دارای ترکیب اصلی زیر می باشد |
| 245 | JqueryUI- toggle |
| 245 | متد toggle() دارای ترکیب زیر می باشد |

| | |
|----------|---|
| 246..... | افکت های JQueryUI |
| 248..... | مثال |
| 249..... | JqueryUI - Toggle Class |
| 250..... | متد toggleClass() دارای ترکیب پایه ی خود به شکل زیر می باشد |
| 252..... | مثال |



مقدمه

زکات علم نشر آن است. حضرت علی (ع)

موسسه آموزشی تحلیل داده ، با حضور جمعی از متخصصین مجرب در زمینه برنامه نویسی در نظر دارد، مطالب آموزشی خود را در قالب کتاب های آموزشی و فیلم ، به صورت رایگان در دسترس عموم قرار دهد تا حتی آن دسته از عزیزانی که بنا به دلایل مالی، مسافت جغرافیایی و یا نداشتن وقت کافی ، امکان شرکت در دوره های حضوری برای آنها میسر نیست، از یادگیری بی بهره نمانند.

علاوه بر این علاقه مندان می توانند ، با ثبت نام در انجمن سایت تحلیل داده، سوالات خود را مطرح نموده و مدرسین آموزشگاه و اعضای انجمن در اسرع وقت، پاسخ های خود را، حتی الامکان به صورت فیلم، در دسترس عموم قرار دهند.

لذا از کلیه فعالان در این زمینه دعوت می شود، در این حرکت جمعی در کنار ما باشند و با حضور فعال خود در انجمن، گام موثری در بهبود سطح علمی جوانان کشور عزیزمان، ایران بردارند.

آموزشگاه حلکیر داده

توجه:

برای دانلود سورس کد مثال های کتاب ، به بخش مقالات Html در آدرس www.tahlildadeh.com مراجعه فرمایید.

بررسی اجمالی JQueryUI

JqueryUI یک کتابخانه ی قدرتمند **javascript** می باشد که در بالای کتابخانه ی **jQuery JavaScript** ساخته شده است. **UI** مخفف **User Interface** می باشد، که مجموعه ای **plug-in** ها برای **jQuery** می باشد که کاربردهای جدید به هسته ی کتابخانه ی **jQuery** اضافه می کند.

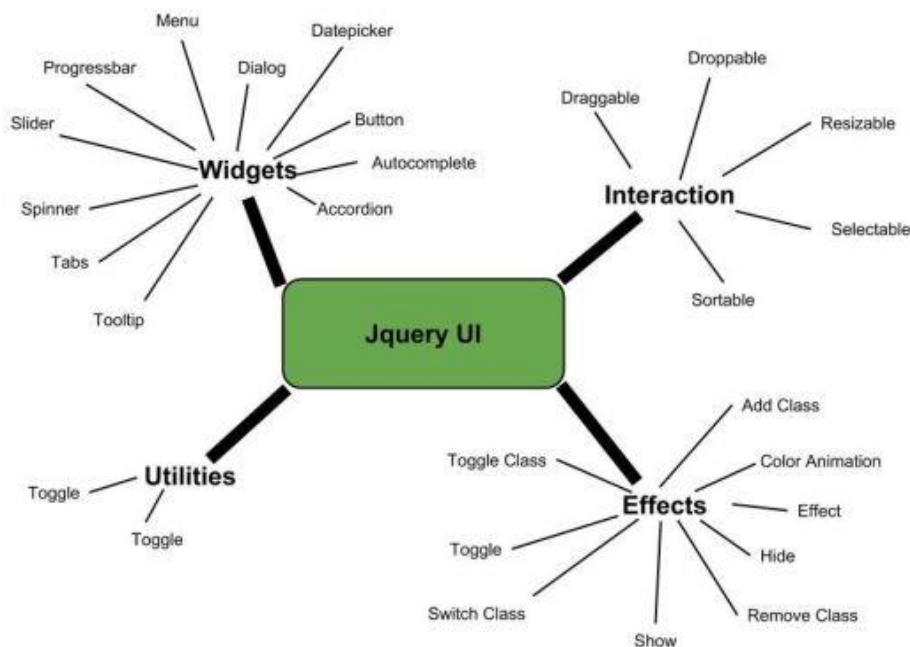
مجموعه ی **plug-in** ها در **JqueryUI** حاوی تعاملات اینترفیس، تاثیرات، انیمیشن ها، ویجت ها و زمینه هایی است که در بالای **jQuery JavaScript Library** ساخته شده اند.

در سپتامبر 2007، در یک پست وبلاگ جان رسیج (**John Resig**) روی **jquery.com** آن را منتشر کرد. آخرین انتشار **jQuery 1.6**، **1.10.4** یا ورژن های بعدی را ملزم کرد.

jQuery UI یک نرم افزار رایگان و با منبع باز می باشد که تحت لیسانس **MIT License** می باشد.

ویژگی ها

JqueryUI به چهار گروه تقسیم می شود، تعاملات، ویجت ها، افکت ها و بهربری ها. اینها در بخش های بعدی با جزئیات مورد بحث قرار خواهند گرفت. ساختار کتابخانه به شکل تصویر زیر می باشد:



تعاملات: اینها **plugin** های تعاملی مانند درگ، دراپ و تغییر اندازه و موارد دیگر هستند که به یوزر توانایی تعامل با عناصر **DOM** را می دهند.

ویجت ها: با استفاده از ویجت ها که پلاگین های **jQuery** هستند، می توانید عناصر اینترفیس یوزر، مانند **accordion,datepicker** و غیره را ایجاد کنید.

افکت ها: اینها روی افکت های داخلی **jQuery** ساخته شده اند. اینها حاوی یک سویت کامل از انیمیشن های سفارشی و انتقال هایی برای عناصر **DOM** هستند.

بهرمندی ها: اینها مجموعه ای از ابزار **modular** میباشند که کتابخانه ی **JqueryUI** به صورت داخلی از آنها استفاده می کند.

مزایای JqueryUI

API های منسجم و سازگار

پشتیبانی قابل درک مرورگر

دارای منبع آزاد و رایگان برای استفاده

پرونده سازی خوب

مکانیسم قالب دهی قدرتمند

پایداری و نگهداری این مطلب

آموزش JqueryUI

JqueryUI یکی از محبوب ترین چارچوب های **front end** حاضر می باشد. این برنامه اولین چارچوب **front-end** براق، بصری و قدرتمند همراه می باشد که برای گسترش سریعت و راحت تر وب استفاده می شود. این برنامه از **HTML**، **CSS** و **Javascript** استفاده می کند.

این آموزش به شما اصول **JqueryUI Framework** را آموزش خواهد داد که می توانید از آنها برای ایجاد برنامه های پیچیده ی وب **GUI** به آسانی استفاده کنید. این آموزش به بخش هایی مانند ساختار **JqueryUI** پایه، **JqueryUI** **CSS**، مولفه های **JqueryUI Layout** و **JqueryUI Plugins** می پردازد. هر کدام از این بخش ها حاوی موضوعات مرتبط با مثال های ساده و مفید می باشد.

بینندگان

این آموزش برای افرادی طراحی شده که دارای یک دانش مقدماتی از **HTML** و **CSS** می باشد و نیاز به توسعه ی وب سایت دارد. پس از تکمیل این آموزش و شما در یک سطح متوسط از تخصص در توسعه ی پروژه های وب با استفاده از **Twitter JqueryUI** قرار خواهید گرفت.

پیش نیازها

قبل از اینکه به پیشروی در این آموزش بپردازیم، فرض می کنم که شما تقریباً در مورد اصول **HTML** و **CSS** آگاهی دارید. اگر در این مورد این مفاهیم آگاهی ندارید، پیشنهاد می کنم به بررسی آموزش کوتاه ما در مورد آموزش **HTML** و **CSS** بپردازید.

JqueryUI - Environment Setup

در این بخش به بحث در مورد دانلود و برقراری کتابخانه ی **JqueryUI** خواهیم پرداخت. همچنین به طور مختصر مسیر این ساختار و محتوای آن را مطالعه خواهیم کرد. کتابخانه ی **JqueryUI** به دو روش در صفحه ی وب شما قابل استفاده می باشد:

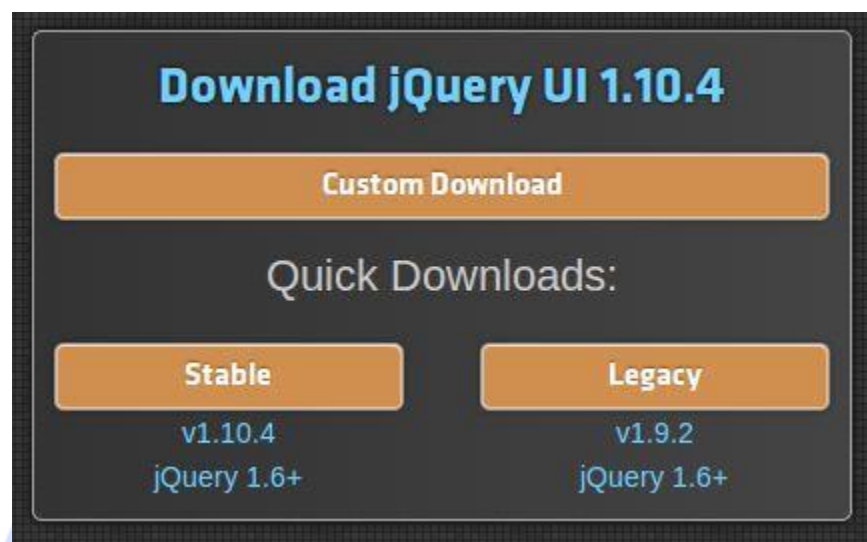
دانلود **UI Library** از وب سایت رسمی آن.

دانلود **UI Library** از **CDNs**

دانلود **UI Library** از وب سایت رسمی آن:

وقتی که لینک <http://jqueryui.com/> را باز می کنید، مشاهده خواهید کرد که سه گزینه برای دانلود

JqueryUI وجود دارد:



Custom Download: روی این دکمه کلیک کنید تا یک ورژن متعارف از کتابخانه را دانلود کنید.

Stable: روی این دکمه کلیک کنید تا آخرین ورژن و درواقع ورژن ثابت از **jQueryUI** را دریافت کنید.

Legacy: روی این دکمه کلیک کنید تا انتشار پیشین از کتابخانه ی **jQueryUI** را دریافت کنید.

دانلود خاص با Download Builder

با استفاده از **Download Builder** می توانید یک ساخت سفارشی (**custom build**) ایجاد کنید که حاوی تنها بخش هایی از کتابخانه ای است که مورد نیاز شما می باشد و این ورژن جدید از **jQueryUI** را براساس زمینه ی انتخاب شده، دانلود کنید. شما صفحه ای مانند زیر مشاهده خواهید کرد (همان صفحه در دو تصویر قرار می گیرد).:

| Interactions | | |
|---|--|--|
| <input checked="" type="checkbox"/> Toggle All | | |
| These add basic behaviors to any element and are used by many components below. | | |
| <input checked="" type="checkbox"/> Draggable | | Enables dragging functionality for any element. |
| <input checked="" type="checkbox"/> Droppable | | Enables drop targets for draggable elements. |
| <input checked="" type="checkbox"/> Resizable | | Enables resize functionality for any element. |
| <input checked="" type="checkbox"/> Selectable | | Allows groups of elements to be selected with the mouse. |
| <input checked="" type="checkbox"/> Sortable | | Enables items in a list to be sorted using the mouse. |

| Widgets | | |
|---|--|--|
| <input checked="" type="checkbox"/> Toggle All | | |
| Full-featured UI Controls - each has a range of options and is fully themeable. | | |
| <input checked="" type="checkbox"/> Accordion | | Displays collapsible content panels for presenting information in a limited amount of space. |
| <input checked="" type="checkbox"/> Autocomplete | | Lists suggested words as the user is typing. |
| <input checked="" type="checkbox"/> Button | | Enhances a form with themeable buttons. |
| <input checked="" type="checkbox"/> Datepicker | | Displays a calendar from an input or inline for selecting dates. |
| <input checked="" type="checkbox"/> Dialog | | Displays customizable dialog windows. |
| <input checked="" type="checkbox"/> Menu | | Creates nestable menus. |
| <input checked="" type="checkbox"/> Progressbar | | Displays a status indicator for loading state, standard percentage, and other progress indicators. |
| <input checked="" type="checkbox"/> Slider | | Displays a flexible slider with ranges and accessibility via keyboard. |
| <input checked="" type="checkbox"/> Spinner | | Displays buttons to easily input numbers via the keyboard or mouse. |
| <input checked="" type="checkbox"/> Tabs | | Transforms a set of container elements into a tab structure. |
| <input checked="" type="checkbox"/> Tooltip | | Shows additional information for any element on hover or focus. |

| Effects | | |
|--|--|--|
| <input checked="" type="checkbox"/> Fade Effect | | Fades an element. |
| <input checked="" type="checkbox"/> Fold Effect | | Folds an element first horizontally and then vertically. |
| <input checked="" type="checkbox"/> Highlight Effect | | Highlights the background of an element in a defined color for a custom duration. |
| <input checked="" type="checkbox"/> Pulse Effect | | Pulsates an element n times by changing the opacity to zero and back. |
| <input checked="" type="checkbox"/> Scale Effect | | Grows or shrinks an element and its content. Restores an element to its original size. |
| <input checked="" type="checkbox"/> Shake Effect | | Shakes an element horizontally or vertically n times. |
| <input checked="" type="checkbox"/> Slide Effect | | Slides an element in and out of the viewport. |
| <input checked="" type="checkbox"/> Transfer Effect | | Displays a transfer effect from one element to another. |

Theme

Select the theme you want to include or design a custom theme.

UI lightness

Theme Folder Name:

CSS Scope:

Download

این کار برای هنگامی که می خواهید فقط از یک **plugin** خاص یا یک ویژگی از کتابخانه ی **jQuery UI** استفاده کنید، مفید می باشد. ساختار مسیر این ورژن در تصویر زیر نمایش داده شده است:



فایل های فشرده نشده در مسیر **development-bundle** قرار می گیرند. بهترین استفاده از فایل فشرده نشده در هنگام توسعه و یا عیب زدایی می باشد؛ فایل ذخیره شده پهنای باند را ذخیره کرده و اجرا را در تولید بهبود می بخشد.

دانلود پایدار

روی دکمه ی **Stable** کلیک کنید که مستقیماً شما را به یک فایل **ZIP** هدایت می کند که حاوی منابع، مثال ها و مستنداتی برای آخرین ورژن از کتابخانه ی **JqueryUI** می باشد. محتویات فایل **ZIP** را روی یک مسیر **jqueryui** باز کنید.

این ورژن حاوی فایل هایی شامل همه ی وابستگی ها، مجموعه ی بزرگی از دموها و حتی مجموعه ی تست واحد کتابخانه می باشد. این ورژن برای شروع کار مفید می باشد.

دانلود legacy

روی دکمه ی **Legacy** کلیک کنید که مستقیماً به یک فایل **ZIP** از انتشار اصلی بخش قبل از کتابخانه ی **JqueryUI** هدایت می کند. این ورژن حاوی همه ی فایل ها، همه ی وابستگی ها، مجموعه ای بزرگ از دموها و حتی مجموعه ی تست واحد مربوط به کتابخانه می باشد. این ورژن برای شروع مفید می باشد.

دانلود UI Library از CDNs

یک **CDN** یا **Content Delivery Network** (شبکه ی ارائه ی محتوا) شبکه ای از سرورهای طراحی شده برای ارائه ی فایل ها به یوزرها می باشد. اگر از یک لینک **CDN** در صفحه ی وب خود استفاده کنید، این لینک مسئولیت فایل های میزبان را از سرورهای خود شما به یک مجموعه ی خارجی انتقال می دهد. این لینک همچنین یک مزیت دیگر نیز دارد که اگر بازدید کننده ای از صفحه ی وب شما، یک کپی از **JqueryUI** را از همان **CDN** دانلود کرده باشد، نیازی به دانلود مجدد آن نمی باشد.

jQuery Foundation, Google, Microsoft همه **CDN** هایی ارائه می دهند که هسته ی **jQuery** را به همراه **jQuery UI** میزبانی می کند.

از آنجایی که یک **CDN** شما را ملزم نمی کند تا میزبانی ورژن خود از **jQuery** و **jQuery UI** را انجام دهید، برای دموها و آزمایش عالی می باشد.

ما در حال استفاده از ورژن های **CDN** از کتابخانه در این آموزش می باشیم.

مثال:

اکنون اجازه دهید مثال ساده ای با استفاده از **JqueryUI** نگارش کنیم. اجازه بدهید یک فایل **HTML** ایجاد کنیم، محتوای زیر را به برچسب **<head>** کپی کنید:

```
<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>
```

جزئیات مربوط به کد بالا عبارتند از:

اولین خط زمینه ی **jQuery UI** (در مورد نمونه ی ما **ui-lightness**) را از طریق **CSS** اضافه می کند. این

CSS ظاهر **UI** ما را خواهد ساخت.

دومین خط کتابخانه ی **jQuery** را اضافه می کند، همانطور که **jQuery UI** در بالای کتابخانه ی **jQuery**

ساخته شده است.

سومین خط کتابخانه ی **jQuery UI** را اضافه می کند. این خط **jQuery UI** را در صفحه ی شما فعال می سازد.

اکنون اجازه بدهید محتوایی را به برچسب **<head>** اضافه کنیم:

```
<script type="text/javascript">
$(function () {
    $('#dialogMsg').dialog();
});
</script>
```

در **<body>** کد زیر را اضافه کنید:

```
<body>
<form id="form1" runat="server">
<div id="dialogMsg" title="First JQueryUI Example">
    Hello this is my first JQueryUI example.
</div>
</form>
</body>
```

کد کامل **HTML** مانند زیر می باشد. آن را با عنوان **myfirstexample.html** ذخیره کنید.

```
<!DOCTYPE html>
<head>
<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>
<script type="text/javascript">
$(function () {
    $('#dialogMsg').dialog();
});
</script>
</head>
<body>
<form id="form1" runat="server">
<div id="dialogMsg" title="First JQueryUI Example">
    Hello this is my first JQueryUI example.
</div>
</form>
</body>
</html>
```

صفحه ی بالا را در مرورگر خود باز کنید، صفحه ای مانند زیر را مشاهده خواهید کرد:



قابل جابجایی

jQueryUI متد **draggable()** را برای ساخت عنصر قابل جابجایی DOM ارائه می دهد. زمانیکه عنصر قابل جابجایی می باشد، می توانید با کلیک کردن روی بخش و انتقال آن در هر جایی در محدوده ی بخش ویو، آن را جابجا کنید.

ترکیب:

متد **draggable()** به دو فرم زیر مورد استفاده قرار می گیرد

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

متد **\$(selector, context).draggable (options)**

متد **draggable (options)** اعلام می کند که یک عنصر HTML می تواند در صفحه ی HTML جابجا شود. پارامتر options در واقع آبجکتی می باشد که رفتار عناصر را تعیین می کند.

ترکیب:

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

شما می توانید با استفاده از آبجکت **Javascript** یک یا چند گزینه را همزمان ارائه دهید. اگر بیشتر از یک گزینه برای ارائه وجود دارد، می توانید آنها را با استفاده از کاما (ویرگول) از یکدیگر مجزا کنید، مانند زیر:

\$(selector, context).draggable({option1: value1, option2: value2.....});

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

| Option | Description |
|-------------------|---|
| addClasses | اگر این گزینه روی false تنظیم شده است، مانع اضافه شدن گروه ui-draggable به لیست عناصر انتخاب شده ی DOM می شود. به طور پیش فرض این گزینه روی true تنظیم شده است. |
| appendTo | عنصری را مشخص می کند که در آن کمک کننده ی قابل جابجایی در هنگام درگ کردن، باید ضمیمه شود. مقدار این گزینه به طور پیش فرض "parent" می باشد. |
| axis | این گزینه درگ کردن به صورت افقی (X) و یا عمودی (Y) را تعیین می کند. مقادیر ممکن "x", "y" : می باشند. |
| cancel | شما می توانید از این گزینه برای جلوگیری از درگ کردن بر روی برخی عناصر استفاده کنید. به طور پیش فرض مقدار مربوطه آن گزینه ی "input,textarea, button,select,option" می باشد. |
| connectToSortable | شما می توانید از این گزینه برای تعیین لیستی استفاده کنید که عناصر آنها قابل تعویض می باشند. در انتهای جایگزینی، عنصر بخشی از لیست می باشد. مقدار آن به طور پیش فرض "false" می باشد. |
| containment | جابجایی را به داخل محدوده ی عنصر یا ناحیه ی تعیین شده محدود می کند. مقدار آن به طور پیش فرض "false" می باشد. |
| cursor | زمانی که عنصر جابجا می شود، پراپرتی CSS مکان نما را تعیین می کند. این گزینه در واقع شکل نشانگر ماوس را نمایش می دهد. مقدار آن به طور پیش فرض "auto" می باشد. |

| | |
|-----------|---|
| cursorAt | افست مربوط به درگ کمکی مربوط به نشانگر ماوس تنظیم می کند. با استفاده از ترکیبی از یک یا دو کلید، مختصات می تواند به عنوان یک hash ارائه شود. |
| delay | پس از اینکه اولین جابجایی ماوس انجام می شود، تاخیری در حدود هزارم ثانیه دارد. تغییر مکان ممکن است پس از این زمان آغاز شود. مقدار آن به طور پیش فرض "0" می باشد. |
| disabled | این گزینه وقتی روی true تنظیم باشد، توانایی جابجایی آیتم ها را غیرفعال می سازد. آیتم ها نمی توانند حرکت کنند تا زمانیکه عملکرد فعال شده باشد (با استفاده از عملکرد ("enable" draggable) مقدار آن به طور پیش فرض "false" می باشد. |
| distance | قبل از جابجایی، تعداد پیکسل هایی که ماوس باید جابجا شود، در نظر گرفته می شوند. مقدار آن به طور پیش فرض "1" می باشد. |
| grid | جابجایی کمکی را در هر پیکسل x و y به یک شبکه آغاز می کند. ردیف مربوطه باید به شکل [x, y] باشد. مقدار آن به طور پیش فرض "false" می باشد. |
| handle | اگر مشخص شده باشد، جابجایی از آغاز را محدود می کند، مگر اینکه mousedown روی عناصر خاص اتفاق بیفتد. مقدار آن به طور پیش فرض "false" می باشد. |
| helper | به یک عنصر کمکی اجازه می دهد تا برای درگ کردن صفحه نمایش استفاده شود. مقدار آن به طور پیش فرض "original" می باشد. |
| iframeFix | در طول یک درگ، مانع iframes ها در گرفتن رویدادهای mousemove می شود. مقدار آن به طور پیش فرض "false" می باشد. |
| opacity | در هنگام جابجایی حالت محوی از عنصر حرکت می کند. مقدار آن به طور پیش فرض "false" می باشد. |

| | |
|-------------------|--|
| refreshPositions | اگر روی true تنظیم شده باشد، همه ی موقعیت ها با هر حرکت ماوس محاسبه می شوند. به طور پیش فرض مقدار آن "false" می باشد. |
| revert | نشان می دهد که آیا در انتهای جابه جایی عنصر به عقب و موقعیت اصلی خود بازمی گردد. مقدار آن به طور پیش فرض "false" می باشد. |
| revertDuration | طول دوره ی جابه جایی (به هزارم ثانیه) پس از اینکه عنصر به موقعیت اصلی خود بازگشت. مقدار آن به طور پیش فرض "500" می باشد. |
| scope | علاوه بر پذیرش گزینه ی droppable، برای گروه بندی آیتم های draggable و droppable استفاده می شود. به طور پیش فرض مقدار آن "default" می باشد. |
| scroll | وقتی روی true تنظیم باشد، اگر آیتم از محدوده ی قابل مشاهده ی پنجره خارج شود، صفحه اسکرول خواهد شد. مقدار پیش فرض آن "true" می باشد. |
| scrollSensitivity | تعداد پیکسل هایی را نشان می دهد که ماوس باید از پنجره خارج شود تا باعث اسکرول شدن صفحه شود. مقدار آن به طور پیش فرض "20" می باشد. |
| scrollSpeed | Indicates the scrolling speed of the display once scrolling begins. By default its value is "20". سرعت اسکرول کردن صفحه نمایش را هنگام آغاز آن نشان می دهد. مقدار آن به طور پیش فرض "20" می باشد. |
| snap | نمایش آیتمی را که روی دیگر عناصر جابجا شده، تنظیم می کند. مقدار پیش فرض آن "false" می باشد. |
| snapMode | تعیین می کند که چگونه بین عنصر جابجا شده و موارد نمایش داده شده در <i>options.snap</i> تنظیم برقرار شود. مقدار پیش فرض آن "both" می باشد. |

| | |
|---------------|---|
| snapTolerance | تعداد پیکسل ها در موقعیت های مختلف را که برای انتشار تنظیمات لازم هستند، به حداکثر می رساند. مقدار پیش فرض آن "20" می باشد. |
| stack | z-index مربوط به مجموعه عناصر را کنترل می کند که با انتخابگر مطابقت دارد، همیشه آخرین آیتم درگ شده را جلو می آورد. در برنامه هایی مانند window managers بسیار موثر است. مقدار پیش فرض آن "false" می باشد. |
| zIndex | Z-index برای کمکی در هنگام درگ شدن. مقدار پیش فرض آن "false" می باشد. |

بخش زیر چند مثال کارآمد از عملکرد درگ نشان می دهد.

عملکرد پیش فرض

در مثالی را مشاهده می کنید که یک عملکرد قابل جابجایی را توضیح می دهد که هیچ پارامتری به متد **draggable()** انتقال نمی دهد.

```

<!DOCTYPE html>
<head>
<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>
#draggable {
width: 150px;
height: 150px;
padding: 0.5em;
background: #eee;
}
</style>
<script>
$(function () {
$("#draggable").draggable();
});
</script>
</head>
<body>
<div id="draggable" class="ui-widget-content">
<p>Drag me !!!</p>
</div>
</body>

```

</html>

استفاده از `disable`، `distance` و `delay`

مثال زیر استفاده از سه گزینه ی مهم `delay` (b)، `disabled` (a) و `distance` (c) را در عملکرد درگ از **JqueryUI** نشان می دهد.

```
<!DOCTYPE html>
<head>
  <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>
</head>
<body>
  <div id="div1" style="border:solid 1px;background-color:gainsboro;">
    <span>You can't move me!</span><br /><br />
  </div>
  <div id="div2" style="border:solid 1px;background-color:grey;">
    <span>
      Dragging will start only after you drag me for 50px
    </span>
    <br /><br />
  </div>
  <div id="div3" style="border:solid 1px;background-color:gainsboro;">
    <span>
      You have to wait for 500ms for dragging to start!
    </span>
    <br /><br />
  </div>
  <script>
    $("#div1 span").draggable(
      { disabled: true }
    );
    $("#div2 span").draggable(
      { distance: 50 }
    );
    $("#div3 span").draggable(
      { delay: 500 }
    );
  </script>
</body>
</html>
```

اجزه بدهید کد بالا را در یک فایل **HTML** به نام **dragexample.htm** ذخیره کرده و آن را در یک مرورگر

استاندارد که **javascript** را پشتیبانی می کند، باز کنیم.

جنبش محدود

مثال های زیر چگونگی محدود کردن محیط عناصر را روی صفحه با استفاده از گزینه ی **containment** در عملکرد درگ کردن **jQueryUI** نشان می دهد.

```
<!DOCTYPE html>
<head>
  <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>
</head>
<body>
  <div id="div4" style="border:solid 1px;background-color:gainsboro;">
    <span>You can drag me only within this div.</span><br /><br />
  </div>
  <div id="div5" style="border:solid 1px;background-color:grey;">
    <span>You can drag me only along x axis.</span><br /><br />
  </div>
  <script>
    $("#div4 span").draggable({
      containment: "#div4"
    });
    $("#div5 span").draggable({
      axis: "x"
    });
  </script>
</body>
</html>
```

در اینجا از خارج شدن عناصر **** از یک **<div>** که ID آن **div4** می باشد، جلوگیری می شود. همچنین می توانید با استفاده از گزینه های محور، با ارزش **x** یا **y**، که توضیح داده شده اند، محدودیت هایی را اعمال کنید.

جابجایی محتوا با استفاده از تکثیر

مثال زیر چگونگی جابجایی یک آیتم را که کلون (**clone**) مربوط به آیتم انتخاب شده می باشد، نشان می دهد. این کار با استفاده از گزینه ی **helper** با مقدار **clone** انجام می شود.

```
<!DOCTYPE html>
<head>
  <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>
</head>
<body>
  <div id="div6" style="border:solid 1px;background:#eee;height:50px;">
    <span>You can duplicate me....</span>
  </div>
  <script>
    $("#div6 span").draggable({
      helper: "clone"
    });
  </script>
</body>
</html>
```

```

});
</script>
</body>
</html>

```

همانطور که مشاهده می کنید اولین عنصر که درگ می شود – تنها عنصر کلون منتقل شده است، در حالیکه آیتم اصلی در موقعیت اصلی خود باقی می ماند. اگر ماوس را رها کنید، عنصر کلون ناپدید شده و آیتم اصلی هنوز به جای خود باقیست.

دریافت مقدار گزینه ی فعلی

مثال زیر چگونگی دریافت مقدار مربوط به هر گزینه را در هر زمانی، در طول دوره ی اجرای نسخه ی شما توضیح می دهد. در اینجا مقدار گزینه های **cursor** و **cursorAt** را که در زمان اجرا تنظیم شده اند، خواهیم خواند. به روش مشابه می توانید مقدار گزینه های موجود دیگر را دریافت کنید.

```

<!DOCTYPE html>
<html>
<head>
<title>tahlildadeh</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>
</head>
<body>
<div id="divX" style="border:solid 1px;background:#eee;height:50px;">
<span>Click anywhere on me to see cursor type...</span>
</div>
<script>
/* First make the item draggable */
$("#divX span").draggable();

$("#divX span").bind('click', function (event) {
var cursor = $("#divX span").draggable("option", "cursor");
var cursorAt = $("#divX span").draggable("option", "cursorAt");
alert("Cursor type - " + cursor + ", cursorAt - " + cursorAt);
});
</script>
</body>
</html>

```

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

متد `draggable (action, params)` می تواند عملکردی را روی عناصر قابل حرکت اجرا کند، به عنوان مثال جلوگیری از جابجایی. **Action** به عنوان یک رشته در اولین استدلال تعیین شده و یک یا چند **params** می توانند بر اساس عملکرد ارائه شده، تعیین شوند.

اساسا در اینجا عملکردها چیزی به جز متدهای **jQuery** نیستند که می توانیم از آنها به فرم رشته استفاده کنیم.

ترکیب

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

جدول زیر عملکردهایی را برای این متد لیست می کند:

| Action | Description |
|---------------------------------|--|
| <code>destroy()</code> | قابلیت جابجایی را کاملاً حذف می کند. عناصر قابل جابه جایی نیستند. این امر عنصر را به وضعیت <code>pre-init</code> خود بازمی گرداند. |
| <code>disable()</code> | قابلیت جابجایی را غیرفعال می سازد. عناصر نمی توانند جابجا شوند تا زمانیکه فراخوانی <code>draggable("enable")</code> به متد انجام شود. |
| <code>enable()</code> | مدیریت جابه جایی را دوباره فعال می سازد. عناصر می توانند مجدداً جابجا شوند. |
| <code>option(optionName)</code> | مقدار کنونی مربوط به <code>optionName</code> تعیین شده را دریافت می کند. در اینجا <code>optionName</code> نام گزینه ای برای دریافت است و از نوع <code>String</code> می باشد. |
| <code>option()</code> | آبجکتی حاوی جفت های <code>key/value</code> را دریافت می کند که گزینه های قابل جابجایی کنونی را نشان می دهد. |

| | |
|---------------------------|---|
| option(optionName, value) | مقدار گزینه ی draggable را تنظیم می کند که با <i>optionName</i> تعیین شده در ارتباط می باشد. در اینجا <i>optionName</i> نام گزینه ای برای تنظیم می باشد و <i>value</i> نام مقداری است که برای این گزینه تنظیم می شود. |
| option(options) | یک یا چند گزینه را برای draggable تنظیم می کند. در اینجا options در واقع نقشه ای از جفت های option-value برای تنظیم می باشد. |
| widget() | یک آبجکت jQuery را که حاوی عنصر draggable می باشد، باز می گرداند. |

مثال ها

اکنون اجازه بدهید مثالی را مشاهده کنیم که از عملکردهای موجود در جدول بالا استفاده می کند. مثال زیر استفاده از عملکردهای **enable** و **disable** را توضیح می دهد:

```
<!DOCTYPE html>
<html>
<head>
<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>
</head>
<body>
<div id="div7" style="border:solid 1px;background-color:gainsboro;">
<span>You can't move me. Dragging is disabled.</span><br><br>
</div>
<div id="div8" style="border:solid 1px;background-color:grey;">
<span>You can move me. Dragging is enabled.</span><br><br>
</div>
<script>
$("#div7 span").draggable();
$("#div7 span").draggable('disable');
$("#div8 span").draggable();
$("#div8 span").draggable('enable');
</script>
</body>
</html>
```

همانطور که مشاهده می کنید، جابجایی اولین عنصر غیر فعال و جابجایی دومین عنصر فعال می باشد که می توانید جابجایی آن را امتحان کنید.

مدیریت رویداد روی عناصر جابجا شده

علاوه بر متد **draggable (options)**، در بخش های قبل مشاهده کردیم که **JqueryUI** متدهایی ارائه می دهد که برای یک رویداد خاص فعال می شدند. این متدها در جدول زیر لیست شده اند:

| Event Method | Description |
|-------------------|---|
| create(event, ui) | زمانی که draggable ایجاد می شود، آغاز به کار می کند. در اینجا رویداد از نوع Event بوده و آبجکت نیز از نوع ui می باشد. |
| drag(event, ui) | هنگام حرکت ماوس در طول جابه جایی، آغاز به کار می کند، در اینجا رویداد از نوع Event بوده و آبجکت نیز از نوع ui می باشد، مانند helper، position و object. |
| start(event, ui) | در هنگام شروع جابه جایی آغاز به کار می کند. در اینجا رویداد از نوع Event و ui از نوع object می باشد، مانند helper، position و object. |
| stop(event, ui) | در هنگام توقف جابه جایی آغاز به کار می کند. در اینجا رویداد از نوع Event و ui از نوع object می باشد، مانند helper، position و object. |

مثال ها

مثال زیر استفاده از متدی را در طول درگ کردن توضیح می دهد. این مثال استفاده از رویداد **drag** را بیان می کند.

```
<!DOCTYPE html>
<head>
  <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>
</head>
<body>
  <div id="div9" style="border:solid 1px;background-color:gainsboro;">
    <span>Drag me to check the event method firing</span><br />
  </div>
</body>
```

```

</div>
<script>
$( "#div9 span" ).draggable({
  cursor: "move",
  axis: "x",
  drag: function (event, ui) {
    alert("hi..");
  }
});
</script>
</body>
</html>

```

اکنون سعی کنید که محتوای نوشته شده را درگ کنید، مشاهده خواهید کرد که شروع یک رویداد درگ منجر به نمایش یک دیالوگ باکس می شود و مکان نما برای جابجایی آیکن، تغییر خواهد کرد و متن هم فقط در محور X حرکت خواهد کرد.

JqueryUI - Droppable

JQueryUI متد `droppable()` را برای ساخت هر گونه عنصر DOM از نوع `droppable` در یک هدف تعیین شده، ارائه می دهد.

ترکیب:

آموزشگاه تلکیر داده ها

متد `droppable()` به دو روش مورد استفاده قرار می گیرد

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

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

متد `$(selector, context).droppable (options)`

متد `droppable (options)` بیان می کند که یک عنصر HTML می تواند به عنوان عنصری مورد استفاده قرار بگیرد که می توانید در آن عناصر دیگری قرار دهید. پارامتر `options` آبجکتی است که رفتار عناصر شامل را تعیین می کند.

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

می توانید با استفاده از آبجکت **Javascript**، یک یا چند گزینه هم زمان ارائه دهید. اگر بیشتر از یک گزینه برای ارائه وجود داشته باشد، می توانید با استفاده از ویرگور آنها را از یکدیگر مجزا کنید، مانند ترکیب زیر:

`$(selector, context).droppable({option1: value1, option2: value2..... });`

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

| Option | Description |
|-------------|--|
| accept | این گزینه زمانی استفاده می شود که لازم است عنصرهای جابجایی را که قرار است برای دراپ کردن پذیرفته شوند، کنترل می کند. مقدار آن به طور پیش فرض * می باشد. |
| activeClass | این گزینه یک String می باشد که یک یا چند گروه CSS را نمایش می دهد که هنگام جابجایی یک عنصر پذیرفته شده، به عنصر droppable اضافه می شوند. مقدار پیش فرض آن false می باشد. |
| addClasses | وقتی این گزینه روی false تنظیم شده باشد، از اضافه شدن گروه <i>ui-droppable</i> به عناصر droppable جلوگیری می کند. مقدار پیش فرض آن true می باشد. |
| disabled | وقتی این گزینه روی true تنظیم شده باشد، قابلیت دراپ شدن (droppable) را غیرفعال خواهد کرد. مقدار پیش فرض آن false می باشد. |
| greedy | این گزینه زمانی استفاده می شود که لازم است عناصر قابل جابه جایی را کنترل کنید که قرار است برای قرار گرفتن در droppable های تو در تو پذیرفته شوند. مقدار پیش فرض |

| | |
|------------|---|
| | آن false می باشد. اگر این گزینه روی true تنظیم شده باشد، هر منبعی از droppable عنصر را دریافت نخواهد کرد. |
| hoverClass | این گزینه یک String می باشد که یک یا چند گروه CSS را نمایش می دهد که به عنصر droppable اضافه می شوند و یک عنصر پذیرفته شده وارد آن می شود. مقدار پیش فرض آن false می باشد. |
| scope | این گزینه برای محدود کردن عمل droppable از عناصر قابل جابه جایی فقط برای آیتم هایی که دارای <i>options.scope</i> مشابه هستند، استفاده می شود. مقدار پیش فرض آن "default" می باشد. |
| tolerance | این گزینه یک String می باشد که تعیین می کند کدام وضعیت برای امتحان کردن اینکه آیا یک draggable روی یک droppable معلق است، استفاده شود. مقدار پیش فرض آن "intersect" می باشد. |

بخش زیر چند مثال کاربردی از قابلیت دراپ کردن نشان خواهد داد.

قابلیت پیش فرض:

مثال زیر یک قابلیت ساده ی **droppable** را توضیح می دهد که هیچ پارامتری به **droppable()** انتقال نمی دهد.

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8" />
<title>jQuery UI Droppable - Default functionality</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>
#draggable-1 {
width: 100px;
height: 50px;
```



```
padding: 0.5em;
float: left;
margin: 22px 5px 10px 0;
}
#draggable-1 {
width: 120px;
height: 90px;
padding: 0.5em;
float: left;
margin: 10px;
}
</style>
<script>
$(function () {
    $("#draggable-1").draggable();
    $("#draggable-1").droppable();
});
</script>
</head>
<body>
<div id="draggable-1" class="ui-widget-content">
    <p>Drag me to my target</p>
</div>
<div id="draggable-1" class="ui-widget-header">
    <p>Drop here</p>
</div>
</body>
</html>
```

استفاده از `addClass`، `disabled` و `tolerance`

مثال زیر استفاده از سه گزینه ی `(a) disabled`، `(b) addClass` و `(c) tolerance` را عملکرد دراپ کردن JQueryUI نشان می دهد.

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Droppable - Default functionality</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>
#draggable-2 {
width: 100px;
height: 50px;
padding: 0.5em;
margin: 0px 5px 10px 0;
}
#draggable-2, #draggable-3, #draggable-4, #draggable-5 {
width: 120px;
height: 90px;
```

```

padding: 0.5em;
float: left;
margin: 10px;
}
</style>
<script>
$(function () {
  $("#draggable-2").draggable();
  $("#droppable-2").droppable({
    drop: function (event, ui) {
      $(this)
        .addClass("ui-state-highlight")
        .find("p")
        .html("Dropped!");
    }
  });
  $("#droppable-3").droppable({
    disabled: "true",
    drop: function (event, ui) {
      $(this)
        .addClass("ui-state-highlight")
        .find("p")
        .html("Dropped!");
    }
  });
  $("#droppable-4").droppable({
    tolerance: 'touch',
    drop: function (event, ui) {
      $(this)
        .addClass("ui-state-highlight")
        .find("p")
        .html("Dropped with a touch!");
    }
  });
  $("#droppable-5").droppable({
    tolerance: 'fit',
    drop: function (event, ui) {
      $(this)
        .addClass("ui-state-highlight")
        .find("p")
        .html("Dropped only when fully fit on the me!");
    }
  });
});
</script>
</head>
<body>
<div id="draggable-2" class="ui-widget-content">
  <p>Drag me to my target</p>
</div>
<div id="droppable-2" class="ui-widget-header">
  <p>Drop here</p>
</div>
<div id="droppable-3" class="ui-widget-header">
  <p>I'm disabled, you can't drop here!</p>
</div>

```

```

<div id="droppable-4" class="ui-widget-header">
  <p>Tolerance Touch!</p>
</div>
<div id="droppable-5" class="ui-widget-header">
  <p>Tolerance Fit!</p>
</div>
</body>
</html>

```

اکنون عنصر را روی باکس "Tolerance Touch" قرار دهید (فقط در کناره ی باکس حرکت کنید) و تغییرات عنصر هدف را مشاهده کنید. اکنون برای قرار دادن عنصر روی هدف "Tolerance Fit"، عنصر قابل جابجایی باید به طور کامل عنصر هدف را تحت پوشش قرار دهد، به عنوان مثال "Tolerance Fit".

عناصر را انتخاب کنید تا دراپ شوند:

مثال زیر استفاده از گزینه های **accept** و **scope** را در عملکرد درگ کردن JQueryUI نشان می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Droppable - Default functionality</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>
    .wrap {
      display: table-row-group;
    }
    #japanpeople, #indiapeople, #javatutorial, #springtutorial {
      width: 120px;
      height: 70px;
      padding: 0.5em;
      float: left;
      margin: 0px 5px 10px 0;
    }
    #japan, #india, #java, #spring {
      width: 140px;
      height: 100px;
      padding: 0.5em;
      float: left;
      margin: 10px;
    }
  </style>
  <script>
    $(function () {
      $("#japanpeople").draggable();
      $("#indiapeople").draggable();
    });
  </script>

```

```

$("#japan").droppable({
  accept: "#japanpeople",
  drop: function (event, ui) {
    $(this)
      .addClass("ui-state-highlight")
      .find("p")
      .html("Dropped!");
  }
});
$("#india").droppable({
  accept: "#indiapeople",
  drop: function (event, ui) {
    $(this)
      .addClass("ui-state-highlight")
      .find("p")
      .html("Dropped!");
  }
});
$("#javatutorial").draggable({ scope: "java" });
$("#springtutorial").draggable({ scope: "spring" });
$("#java").droppable({
  scope: "java",
  drop: function (event, ui) {
    $(this)
      .addClass("ui-state-highlight")
      .find("p")
      .html("Dropped!");
  }
});
$("#spring").droppable({
  scope: "spring",
  drop: function (event, ui) {
    $(this)
      .addClass("ui-state-highlight")
      .find("p")
      .html("Dropped!");
  }
});
});
</script>
</head>
<body>
<div class="wrap">
  <div id="japanpeople" class="ui-widget-content">
    <p>People to be dropped to Japan</p>
  </div>
  <div id="indiapeople" class="ui-widget-content">
    <p>People to be dropped to India</p>
  </div>
  <div id="japan" class="ui-widget-header">
    <p>Japan</p>
  </div>
  <div id="india" class="ui-widget-header">
    <p>India</p>
  </div>
</div>

```

```

<hr />
<div class="wrap">
  <div id="javatutorial" class="ui-widget-content">
    <p>People who want to learn Java</p>
  </div>
  <div id="springtutorial" class="ui-widget-content">
    <p>People who want to learn Spring</p>
  </div>
  <div id="java" class="ui-widget-header">
    <p>Java</p>
  </div>
  <div id="spring" class="ui-widget-header">
    <p>Spring</p>
  </div>
</div>
</body>
</html>

```

در اینجا مشاهده می کنید که می توانید عنصر "People from Japan" را فقط روی هدف "Japan" و عنصر "People from India" را روی هدف India قرار دهید. به طور مشابه محدوده ی مربوط به "People who want to learn Java" روی هدف "Java" و "People who want to learn Spring" روی هدف "Spring" target تنظیم می شود.

مدیریت ظاهر

مثال زیر استفاده از گزینه های activeClass و hoverClass از گروه JQueryUI را توضیح می دهد، که در تنظیم ظاهر به ما کمک می کند.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Droppable - Default functionality</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 type="text/css">
    #draggable-3 {
      width: 100px;
      height: 50px;
      padding: 0.5em;
      float: left;
      margin: 21px 5px 10px 0;
    }
    #droppable-6 {

```

```

width: 120px;
height: 90px;
padding: 0.5em;
float: left;
margin: 10px;
}
.active {
border-color: blue;
background: grey;
}
.hover {
border-color: red;
background: green;
}
</style>
<script>
$(function () {
$("#draggable-3").draggable();
$("#droppable-6").droppable({
activeClass: "active",
hoverClass: "hover",
drop: function (event, ui) {
$(this)
.addClass("ui-state-highlight")
.find("p")
.html("Dropped!");
}
});
});
</script>
</head>
<body>
<div id="draggable-3" class="ui-widget-content">
<p>Drag me to my target</p>
</div>
<div id="droppable-6" class="ui-widget-header">
<p>Drop here</p>
</div>
</body>
</html>

```

شما متوجه می شوید که در جابه جایی و یا تعلیق عنصر **"Drag me to my target"** رنگ عنصر **"Drop here"** را تغییر می دهد.

متد \$(selector, context).droppable ("action", params)

متد **droppable ("action", params)** می تواند عملی را روی عناصر **droppable** اجرا کند، مانند جلوگیری از قابلیت **droppable**. این عمل به عنوان یک رشته در اولین برهان تعیین شده است (به عنوان مثال "disable" مانع دراپ شدن می شود). در جدول زیر فعالیت هایی که منتقل می شوند را چک کنید.

ترکیب

`$(selector, context).droppable ("action", params);`

جدول زیر فعالیت های مختلفی را لیست می کند که در با این متد مورد استفاده قرار می گیرند:

| Action | Description |
|---------|---|
| destroy | این عملکرد قابلیت droppable مربوط به یک عنصر را کاملاً از بین می برد. این عناصر به وضعیت pre-init خود باز می گردند. |
| disable | این عملیات droppable را لغو می کند. عناصر دیگر عناصر droppable نیستند. این متد هیچ برهانی نمی پذیرد. |
| enable | این عملیات مجدداً عملکرد droppable را فعال سازی می کند و عناصر مجدداً می توانند عنصر droppable دریافت کنند. این متد هیچ برهانی را نمی پذیرد. |
| option | این فعالیت یک آبجکت حاوی جفت های key/valu را دریافت می کند که نشان دهنده ی گزینه های hash فعلی دارای قابلیت droppable می باشد. این متد هیچ برهانی را نمی پذیرد. |

| | |
|-----------------------------|--|
| option(optionName) | این فعالیت مقداری که معمولا عنصر droppable را با <i>optionName</i> تعیین شده مرتبط می سازد، دریافت می کند و مقدار String را به عنوان یک برهان می گیرد. |
| option(optionName, value) | این فعالیت مقدار گزینه ی droppable مرتبط با <i>optionName</i> تعیین شده را تنظیم می کند. |
| option(options) | این عمل یک یا چند گزینه برای droppable تنظیم می کند. گزینه های برهان نقشه ای از جفت های option-value می باشد که باید تنظیم شوند. |
| widget | این عمل یک آبجکت jQuery حاوی عنصر droppable یازمی گرداند. این متد هیچ برهانی را نمی پذیرد. |

مثال

اکنون اجازه بدهید با استفاده از عملکردهای ارائه شده در جدول بالا، مثالی را به شما توضیح دهیم. مثال زیر استفاده از متد **destroy()** را توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Droppable - Default functionality</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>
.droppable-4 {
  width: 90px;
  height: 50px;
  padding: 0.5em;
  float: left;
  margin: 0px 5px 10px 0;
  border: 1px solid red;
  background-color: #B9CD6D;
}
```



```

.droppable-7 {
  width: 100px;
  height: 90px;
  padding: 0.5em;
  float: left;
  margin: 10px;
  border: 1px solid black;
  background-color: #A39480;
}
.droppable.active {
  background-color: red;
}
</style>
<script>
$(function () {
  $('.draggable-4').draggable({ revert: true });
  $('.droppable-7').droppable({
    hoverClass: 'active',
    drop: function (e, ui) {
      $(this).html(ui.draggable.remove().html());
      $(this).droppable('destroy');
      $(this)
        .addClass("ui-state-highlight")
        .find("p")
        .html("i'm destroyed!");
    }
  });
});
</script>
</head>
<body>
<div class="draggable-4"><p>drag 1</p></div>
<div class="draggable-4"><p>drag 2</p></div>
<div class="draggable-4"><p>drag 3</p></div>
<div style="clear: both;padding:10px"></div>
<div class="droppable-7">drop here</div>
<div class="droppable-7">drop here</div>
<div class="droppable-7">drop here</div>
</body>
</html>

```

اگر "drag1" را روی رکدام از عناصر به نام "drop here" قرار دهید، متوجه خواهید شد که این عنصر دراپ شده و این عمل قابلیت یک عنصر را به طور کامل از بین می برد. شما می توانید "drag2" و "drag3" را مجدداً روی این عنصر قرار دهید.

علاوه بر متد **droppable (options)** که در بخش قبل مشاهده کردیم، **JqueryUI** متدهای رویدادی ارائه می دهد که برای یک رویداد خاص آغاز می شوند. این متدها در جدول زیر لیست شده اند:

| Event Method | Description |
|-----------------------|---|
| activate(event, ui) | این رویداد وقتی آغاز می شود که عنصر قابل جابه جایی پذیرفته شده جابه جایی را شروع کند. اگر بخواهید droppable را دارای ویژگی light up کنید، این متد می تواند مفید باشد. |
| create(event, ui) | این رویداد وقتی آغاز می شود که یک عنصر droppable ایجاد شده باشد، که رویداد از نوع Event است و ui از نوع object. |
| deactivate(event, ui) | این رویداد وقتی آغاز می شود که یک عنصر قابل جابه جایی، جابه جایی را متوقف کند که رویداد از نوع Event است و ui از نوع object. |
| drop(event, ui) | این عمل وقتی آغاز می شود که یک المنت روی droppable دراپ شود که براساس گزینه ی <i>tolerance</i> می باشد که رویداد از نوع Event است و ui از نوع object. |
| out(event, ui) | این رویداد وقتی آغاز می شود که یک المنت قابل جابه جایی پذیرفته شده از droppable خارج شود و براساس گزینه ی <i>tolerance</i> می باشد و رویداد از نوع Event است و ui از نوع object |

| | |
|-----------------|--|
| over(event, ui) | این رویداد وقتی آغاز می شود که یک عنصر قابل جابه جایی پذیرفته شده روی droppable درگ می شود. این متد براساس گزینه ی <i>tolerance</i> می باشد و رویداد از نوع Event است و ui از نوع object |
|-----------------|--|

مثال

مثال زیر استفاده از متد **event** را در هنگام عملکرد دراپ توضیح می دهد. این مثال استفاده از رویدادهای **drop**، **over** و **out** را توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Droppable - Default functionality</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>
    #draggable-5 {
      width: 100px;
      height: 50px;
      padding: 0.5em;
      float: left;
      margin: 22px 5px 10px 0;
    }
    #droppable-8 {
      width: 120px;
      height: 90px;
      padding: 0.5em;
      float: left;
      margin: 10px;
    }
  </style>
  <script>
    $(function () {
      $("#draggable-5").draggable();
      $("#droppable-8").droppable({
        drop: function (event, ui) {
          $(this)
            .addClass("ui-state-highlight")
            .find("p")
            .html("Dropped!");
        },
        over: function (event, ui) {
          $(this)
            .addClass("ui-state-highlight")
```

```

        .find("p")
        .html("moving in!");
    },
    out: function (event, ui) {
        $(this)
        .addClass("ui-state-highlight")
        .find("p")
        .html("moving out!");
    }
    });
});
</script>
</head>
<body>
<div id="draggable-5" class="ui-widget-content">
<p>Drag me to my target</p>
</div>
<div id="droppable-8" class="ui-widget-header">
<p>Drop here</p>
</div>
</body>
</html>

```

در اینجا متوجه خواهید شد که هنگامی که عنصر را درگ می کنید، چگونه پیغام در یک عنصر **droppable** تغییر می کند.

JqueryUI - Resizable

jQueryUI متد **resizable()** را برای تغییر اندازه ی هر عنصر در **DOM** می کند. این متد تغییر اندازه ی عنصری را که وقتگیر بود و کد **HTML** لازم داشت، بسیار ساده کرده است. متد **resizable()** یک آیکن در پایین صفحه و سمت راست نشان می دهد که برای تغییر اندازه می باشد.

ترکیب:

متد **resizable()** به دو شکل مورد استفاده قرار می گیرد

\$(selector, context).resizable (options) Method

\$(selector, context).resizable ("action", params) Method

متد \$(selector, context).resizable (options)

متد **resizable (options)** اعلام می کند که یک عنصر **HTML** می تواند تغییر اندازه دهد. پارامتر **option** آبجکتی است که رفتار عناصر دخیل در تغییر اندازه را تعیین می کند.

ترکیب

\$(selector, context).resizable (options);

در هنگام استفاده از آبجکت **javascript**، می توانید یک یا چند گزینه را ارائه دهید. اگر بیشتر از یک گزینه برای ارائه وجود داشته باشد، می توانید آنها را با استفاده از کاما از یکدیگر مجزا کنید، مانند ترکیب زیر:

\$(selector, context).resizable({option1: value1, option2: value2..... });

جدول زیر گزینه هایی را ارائه می دهد که با این متد استفاده می شوند:

| Option | Description |
|------------|--|
| alsoResize | این گزینه از نوع Selector، jQuery یا عنصر DOM می باشد. این گزینه عناصری را نمایش می دهد که در هنگام تغییر اندازه ی آبجکت اصلی، این عناصر نیز تغییر اندازه می دهند. مقدار پیش فرض آن false می باشد. |
| animate | وقتی این گزینه روی true تنظیم باشد، در طول دروره ی تغییر اندازه، وقتی دکمه ی ماوس رها می شود، برای فعالسازی افکت های بصری استفاده می شود. مقدار پیش فرض آن false می باشد (بدون افکت). |

| | |
|-----------------|---|
| animateDuration | این گزینه برای تنظیم مدت زمان افکت تغییر اندازه (به هزارم ثانیه) استفاده می شود. این گزینه تنها زمانی استفاده می شود که گزینه ی animate روی true تنظیم باشد. مقدار پیش فرض آن "slow" می باشد. |
| animateEasing | این گزینه برای تعیین اینکه در هنگام استفاده از گزینه ی animate کدام easing به کار گرفته شود، استفاده می شود. مقدار پیش فرض آن "swing" می باشد. |
| aspectRatio | این گزینه برای نشان دادن اینکه آیا نسبت ها را (طول و عرض) برای آیتم حفظ کنیم، به کار می رود. مقدار پیش فرض آن false می باشد. |
| autoHide | این گزینه برای پنهان کردن بزرگنمایی آیکن یا دسته ها استفاده می شود، به جز وقتی که ماوس روی آیتم قرار دارد. مقدار پیش فرض آن false می باشد. |
| cancel | این گزینه برای جلوگیری از تغییر اندازه روی عناصر تعیین شده استفاده می شود. مقدار پیش فرض آن input,textarea,button,select,option می باشد. |
| containment | این گزینه برای محدود کردن عناصر تغییر اندازه در یک محدوده ی مشخص، استفاده می شود. مقدار پیش فرض آن false می باشد. |
| delay | این گزینه برای تنظیم تحمل یا به تعویق انداختن آن در هزارم ثانیه استفاده می شود. تغییر اندازه یا جایگزینی پس از آن شروع خواهد شد. مقدار پیش فرض آن 0 است. |
| disabled | این گزینه وقتی روی true تنظیم شده باشد، مکانیزم تغییر اندازه را غیرفعال می سازد. ماوس دیگر عناصر را تغییر اندازه نخواهد داد تا اینکه با استفاده از resizable ("enable") مکانیزم فعالسازی شود. مقدار پیش فرض آن false می باشد. |

| | |
|-----------|---|
| distance | با این گزینه، تغییر اندازه تنها زمانی آغاز می شود که ماوس به اندازه ی یک فاصله (به پیکسل) حرکت می کند. به طور پیش فرض مقدار آن 1 pixel می باشد. این گزینه می تواند در هنگام کلیک کردن روی یک عنصر، مانع تغییر اندازه ی ناخواسته شود. |
| ghost | وقتی این گزینه روی true تنظیم شده باشد، یک کمک کننده ی نیمه شفاف برای تغییر اندازه نشان داده می شود. این آیتم هنگام رها کردن ماوس حذف خواهد شد. مقدار پیش فرض آن false است. |
| grid | این گزینه از نوع Array [x, y] می باشد، که تعداد پیکسل هایی را نشان می دهد که عنصر طی حرکت ماوس به طور افقی و عمودی گسترش می دهد. مقدار پیش فرض آن false می باشد. |
| handles | این گزینه رشته ای است که نشان می دهد کدام اضلاع یا زاویه های مورد نظر می توانند تغییر اندازه داشته باشند. مقادیر پیش فرض آن e, s, se می باشند. |
| helper | این گزینه برای افزودن یک گروه CSS به شیوه ی تغییر اندازه ی عنصر استفاده می شود. وقتی تغییر اندازه انجام شد، یک عنصر <div> ایجاد می شود که موردی است که مدرج شده است. زمانی که تغییر اندازه کامل شد، عنصر اصلی درست بوده و <div> حذف می شود. مقدار پیش فرض آن false می باشد. |
| maxHeight | این گزینه برای تنظیم حداکثر ارتفاع مجاز برای تغییر اندازه استفاده می شود. مقدار پیش فرض آن null است. |
| maxWidth | این گزینه برای تنظیم حداکثر عرض مجاز برای تغییر اندازه استفاده می شود. مقدار پیش فرض آن null است. |
| minHeight | این گزینه برای تنظیم حداقل ارتفاع مجاز برای تغییر اندازه استفاده می شود. مقدار پیش فرض آن 10 است. |

| | |
|----------|--|
| minWidth | این گزینه برای تنظیم حداقل عرض مجاز برای تغییر اندازه استفاده می شود. مقدار پیش فرض آن 10 است. |
|----------|--|

بخش بعدی چند مثال در مورد قابلیت تغییر اندازه به شما نشان خواهد داد.

کارکرد پیش فرض:

مثال زیر قابلیت تغییر اندازه را نشان می دهد که هیچ پارامتری به متد **resizable()** انتقال نمی یابد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Resizable functionality</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>
  <!-- CSS -->
  <style>
    .ui-widget-header {
      background: #b9cd6d;
      border: 1px solid #b9cd6d;
      color: #FFFFFF;
      font-weight: bold;
    }
    .ui-widget-content {
      background: #cedc98;
      border: 1px solid #DDDDDD;
      color: #333333;
    }
    #resizable {
      width: 150px;
      height: 150px;
      padding: 0.5em;
      text-align: center;
      margin: 0;
    }
  </style>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#resizable").resizable();
    });
  </script>
</head>
<body>
  <!-- HTML -->
```



```

<div id="resizable" class="ui-widget-content">
  <h3 class="ui-widget-header">Pull my edges to resize me!!</h3>
</div>
</body>
</html>

```

مرزهای مربع را درگ کنید تا تغییر اندازه انجام شود.

استفاده از animate و ghost

مثال زیر استفاده از دو گزینه ی animate و ghost را در عملکرد تغییر اندازه ی JQueryUI توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Resizable functionality</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>
  <!-- CSS -->
  <style>
    .ui-widget-header {
      background: #b9cd6d;
      border: 1px solid #b9cd6d;
      color: #FFFFFF;
      font-weight: bold;
    }
    .ui-widget-content {
      background: #cedc98;
      border: 1px solid #DDDDDD;
      color: #333333;
    }
    #resizable-2, #resizable-3 {
      width: 150px;
      height: 150px;
      padding: 0.5em;
      text-align: center;
      margin: 0;
    }
  </style>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#resizable-2").resizable({
        animate: true
      });
      $("#resizable-3").resizable({
        ghost: true
      });
    });
  </script>

```

```

});
</script>
</head>
<body>
<!-- HTML -->
<div id="resizable-2" class="ui-widget-content">
  <h3 class="ui-widget-header">
    Pull my edges and Check the animation!!
  </h3>
</div><br>
<div id="resizable-3" class="ui-widget-content">
  <h3 class="ui-widget-header">I'm ghost!!</h3>
</div>
</body>
</html>

```

مرز مربع را برای تغییر اندازه درگ کنید تا تاثیر گزینه های **animate** و **ghost** را مشاهده کنید.

استفاده از **containment**، **minWidth** و **minHeight**

مثال زیر استفاده از سه گزینه ی **minWidth**، **containment**، **minHeight** را در عملکرد تغییر اندازه ی **JqueryUI** توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Resizable functionality</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>
    .ui-widget-header {
      background: #b9cd6d;
      border: 1px solid #b9cd6d;
      color: #FFFFFF;
      font-weight: bold;
    }

    .ui-widget-content {
      background: #cedc98;
      border: 1px solid #DDDDDD;
      color: #333333;
    }

    #container-1 {
      width: 300px;
      height: 300px;
    }

    #resizable-4 {

```

```

        background-position: top left;
        width: 150px;
        height: 150px;
    }
    #resizable-4, #container {
        padding: 0.5em;
    }
</style>
<script>
$(function () {
    $("#resizable-4").resizable({
        containment: "#container",
        minHeight: 70,
        minWidth: 100
    });
});
</script>
</head>
<body>
<div id="container" class="ui-widget-content">
    <div id="resizable-4" class="ui-state-active">
        <h3 class="ui-widget-header">
            Resize contained to this container
        </h3>
    </div>
</div>
</body>
</html>

```

مرزهای مربع را برای تغییر اندازه درگ کنید، بیشتر از حد مشخص شده نمی توانید تغییر اندازه دهید.

استفاده از `delay`، `distance` و `autoHide`

مثال زیر استفاده از سه گزینه ی `delay`، `distance` و `autoHide` را در عملکرد تغییر اندازه مربوط به

JqueryUI توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>jQuery UI Resizable functionality</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>
</head>
<style>
    .ui-widget-header {
        background: #b9cd6d;
        border: 1px solid #b9cd6d;
        color: #FFFFFF;
    }

```

```

    font-weight: bold;
}
.ui-widget-content {
    background: #cedc98;
    border: 1px solid #DDDDDD;
    color: #333333;
}
.square {
    width: 150px;
    height: 150px;
    border: 1px solid black;
    text-align: center;
    float: left;
    margin-left: 20px;
    -right: 20px;
}
</style>
<script>
$(function () {
    $("#resizable-5").resizable({
        delay: 1000
    });
    $("#resizable-6").resizable({
        distance: 40
    });
    $("#resizable-7").resizable({
        autoHide: true
    });
});
</script>
<head>
<body>
<div id="resizable-5" class="square ui-widget-content">
    <h3 class="ui-widget-header">
        Resize starts after delay of 1000ms
    </h3>
</div><br>
<div id="resizable-6" class="square ui-widget-content">
    <h3 class="ui-widget-header">
        Resize starts at distance of 40px
    </h3>
</div><br>
<div id="resizable-7" class="square ui-widget-content">
    <h3 class="ui-widget-header">
        Hover over me to see the magnification icon!
    </h3>
</div>
</body>
</html>

```

مرز مربع را برای تغییر اندازه درگ کنید، متوجه خواهید شد که:

اولین مربع پس از تاخیر **1000ms** تغییر اندازه می دهد.

دومین مربع پس از اینکه ماوس 40 px جابه جا شد، شروع به تغییر اندازه می کند.

اگر ماوس را روی سومین مربع حرکت دهید، آیکن بزرگنمایی ظاهر می شود.

استفاده از گزینه ی `alsoResize`

مثال زیر استفاده از گزینه ی `alsoResize` را در عملکرد تغییر اندازه ی `JqueryUI` توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Resizable functionality</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>
  <!-- CSS -->
  <style>
    .ui-widget-header {
      background: #b9cd6d;
      border: 1px solid #b9cd6d;
      color: #FFFFFF;
      font-weight: bold;
    }
    .ui-widget-content {
      background: #cedc98;
      border: 1px solid #DDDDDD;
      color: #333333;
    }
    #resizable-8, #resizable-9 {
      width: 150px;
      height: 150px;
      padding: 0.5em;
      text-align: center;
      margin: 0;
    }
  </style>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#resizable-8").resizable({
        alsoResize: "#resizable-9"
      });
      $("#resizable-9").resizable();
    });
  </script>
</head>
<body>
  <!-- HTML -->
  <div id="resizable-8" class="ui-widget-content">
    <h3 class="ui-widget-header">Resize!!</h3>
```

```

</div><br>
<div id="resizable-9" class="ui-widget-content">
  <h3 class="ui-widget-header">I also get resized!!</h3>
</div>
</body>
</html>

```

برای تغییر اندازه مرزهای مربع را درگ کنید، متوجه می شوید که دومین مربع نیز با اولین مربع تغییر اندازه خواهد داد.

استفاده از `aspectRatio`، `grid`

مثال زیر استفاده از گزینه های `aspectRatio` و `grid` را در تغییر اندازه ی `jQueryUI` توضیح می دهد:

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Resizable functionality</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>
    .ui-widget-header {
      background: #b9cd6d;
      border: 1px solid #b9cd6d;
      color: #FFFFFF;
      font-weight: bold;
    }
    .ui-widget-content {
      background: #cedc98;
      border: 1px solid #DDDDDD;
      color: #333333;
    }
    .square {
      width: 150px;
      height: 150px;
      border: 1px solid black;
      text-align: center;
      float: left;
      margin-left: 20px;
      margin-right: 20px;
    }
  </style>
  <script>
    $(function () {
      $("#resizable-10").resizable({
        aspectRatio: 10 / 3
      });
    });
  </script>

```

```

$( "#resizable-11" ).resizable({
  grid: [ 50, 20]
});
});
</script>
</head>
<body>
<div id="resizable-10" class="square ui-widget-content">
  <h3 class="ui-widget-header">
    Resize with aspectRatio of 10/3
  </h3>
</div>
<div id="resizable-11" class="square ui-widget-content">
  <h3 class="ui-widget-header">
    Snap me to the grid of [50,20]
  </h3>
</div>
</body>
</html>

```

مرز مربع را درگ کنید تا آن را تغییر اندازه دهید، اولین مربع با مربع دوم با مقیاس $10/3$ با نسبت هاب $[50,20]$ تغییر اندازه می دهد.

متد `$(selector, context).resizable ("action", params)`

متد `resizable ("action", params)` می تواند عملی را روی عناصر قابل تغییر اندازه اجرا کند، به عنوان مثال عمل فعال کردن و یا غیر فعال کردن قابلیت تغییر اندازه. این عملکرد به عنوان یک رشته در اولین برهان تعیین می شود (به عنوان مثال غیرفعال کردن و یا جلوگیری از تغییر اندازه). در جدول زیر عملکردهایی را که قابل انتقال هستند، بررسی کنید.

ترکیب:

`$(selector, context).resizable ("action", params);`

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

| Action | Description |
|--------|-------------|
|--------|-------------|

| | |
|-----------------------------------|--|
| <u>destroy</u> | این عملکرد قابلیت تغییر اندازه ی یک عنصر را کاملاً از بین می برد و آن را به وضعیت pre-init خود بازمی گرداند. |
| <u>disable</u> | این عملکرد قابلیت تغییر اندازه ی یک عنصر را غیرفعال می سازد. این متد هیچ برهانی را نمی پذیرد. |
| <u>enable</u> | این عملکرد قابلیت تغییر اندازه ی یک عنصر را فعال می سازد. این متد هیچ برهانی را نمی پذیرد. |
| <u>option(optionName)</u> | این عملکرد مقدار مشخص شده ی specifiedoptionName را بازمی گرداند. این گزینه با یکی از موارد استفاده شده با گزینه های قابل تغییر اندازه مطابقت دارد. |
| <u>option()</u> | یک آبجکت حاوی جفت های key/value می باشد که گزینه های قابل اندازه بندی مجدد را نمایش می دهد. این گزینه نیز برهانی نمی پذیرد. |
| <u>option(optionName, value)</u> | این عملکرد گزینه های اندازه بندی مجدد را با optionName مشخص شده، تنظیم می کند. این گزینه با یکی از گزینه های تغییر اندازه مطابقت دارد. |
| <u>option(options)</u> | این عملکرد یک یا چند گزینه را برای اندازه بندی مجدد تنظیم می کند. |
| <u>widget()</u> | این عملکرد یک آبجکت jQuery باز می گرداند که حاوی عنصر resizable می باشد. این متد برهانی را نمی پذیرد. |

مثال:

اکنون اجازه بدهید با استفاده از عملکردهای جدول بالا مثالی را مشاهده کنیم. مثال زیر استفاده از متدهای

destroy() و **disable()** را توضیح می دهد.

```
<!doctype html>
<html lang="en">
```



```

<head>
  <meta charset="utf-8">
  <title>jQuery UI Resizable functionality</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>
  <!-- CSS -->
  <style>
    .ui-widget-header {
      background: #b9cd6d;
      border: 1px solid #b9cd6d;
      color: #FFFFFF;
      font-weight: bold;
    }
    .ui-widget-content {
      background: #cedc98;
      border: 1px solid #DDDDDD;
      color: #333333;
    }
    #resizable-12, #resizable-13 {
      width: 150px;
      height: 150px;
      padding: 0.5em;
      text-align: center;
      margin: 0;
    }
  </style>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#resizable-12").resizable();
      $("#resizable-12").resizable('disable');
      $("#resizable-13").resizable();
      $("#resizable-13").resizable('destroy');
    });
  </script>
</head>
<body>
  <!-- HTML -->
  <div id="resizable-12" class="ui-widget-content">
    <h3 class="ui-widget-header">I'm disable!!</h3>
  </div><br>
  <div id="resizable-13" class="ui-widget-content">
    <h3 class="ui-widget-header">I'm Destroyed!!</h3>
  </div>
</body>
</html>

```

شما نمی توانید اولین مربع را تغییر اندازه دهید، چرا که اولین مربع غیرفعال و دومین مربع ناقص می باشد.

مدیریت رویداد روی عناصر resizable

علاوه بر متد **resizable (options)** که در بخش های قبل مشاهده کردیم، **JqueryUI** متدهای دیگری ارائه می دهد که برای یک رویداد خاص آغاز می شوند. این متدهای **event** در جدول زیر لیست شده اند:

| Event Method | Description |
|-------------------|--|
| create(event, ui) | این رویداد زمانی آغاز می شود که عنصر resizable ایجاد شده باشد. |
| resize(event, ui) | این رویداد زمانی آغاز می شود که دسته ی مربوط به تغییر اندازه ی عنصر درگ شود. |
| start(event, ui) | این رویداد با شروع یک عملیات تغییر اندازه آغاز می شود. |
| stop(event, ui) | این رویداد در پایان یک عملیات تغییر اندازه آغاز می شود. |

مثال:

مثال زیر استفاده از متد را در طول عملیات تغییر اندازه، توضیح می دهد. این مثال استفاده از رویدادهای

create و **resize** را توضیح می دهد:

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Resizable functionality</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>
  <!-- CSS -->
  <style>
    .ui-widget-header {
      background: #b9cd6d;
      border: 1px solid #b9cd6d;
      color: #FFFFFF;
      font-weight: bold;
    }
  </style>
```

```

.ui-widget-content {
    background: #cedc98;
    border: 1px solid #DDDDDD;
    color: #333333;
}
#resizable-14 {
    width: 150px;
    height: 150px;
    padding: 0.5em;
    text-align: center;
    margin: 0;
}
</style>
<!-- Javascript -->
<script>
$(function () {
    $("#resizable-14").resizable({
        create: function (event, ui) {
            $("#resizable-15").text("I'm Created!!");
        },
        resize: function (event, ui) {
            $("#resizable-16").text("top = " + ui.position.top +
                ", left = " + ui.position.left +
                ", width = " + ui.size.width +
                ", height = " + ui.size.height);
        }
    });
});
</script>
</head>
<body>
<!-- HTML -->
<div id="resizable-14" class="ui-widget-content">
    <h3 class="ui-widget-header">Resize !!</h3>
</div><br>
<span id="resizable-15"></span><br>
<span id="resizable-16"></span>
</body>
</html>

```

متد selectable در JQuery UI

jQueryUI متد **selectable()** را برای انتخاب عنصر **DOM** به طور مجزا یا در گره، ارائه می دهد. با این متد عناصر با درگ کردن یک باکس با ماوس بر روی آنها انتخاب می شوند. همچنین می توان با کلیک کردن و با درگ کردن عناصر در حالیکه دکمه **Ctrl** را نگاه داشته اید، انتخاب کنید، که به این وسیله انتخاب چند مورد ممکن می شود.

ترکیب:

متد () selectable به دو شکل مورد استفاده قرار می گیرد

\$(selector, context).selectable (options) Method

\$(selector, context).selectable ("action", params) Method

متد \$(selector, context).selectable (options)

متد **selectable (options)** بیان می کند که یک عنصر **HTML** حاوی آیتم های قابل انتخاب می باشد. پارامتر **options** آبجکتی است که رفتار عناصر انتخابی را تعیین می کند.

ترکیب:

\$(selector, context).selectable (options);

می توانید با استفاده از آبجکت **Javascript** یک یا چند گزینه را ارائه دهید. اگر بیشتر از یک گزینه برای ارائه وجود دارد، با استفاده از کاما می توانید آنها را از یکدیگر مجزا کنید، مانند زیر:

\$(selector, context).selectable({option1: value1, option2: value2..... });

جدول زیر گزینه هایی را ارائه می دهد که با این متد استفاده می شوند:

| Option | Description |
|--------|-------------|
|--------|-------------|

| | |
|-------------|---|
| appendTo | این گزینه بیان می کند که کدام عنصر مجموعه (lasso) باید ضمیمه شود. مقدار پیش فرض آن body می باشد. |
| autoRefresh | اگر این گزینه روی true تنظیم شده باشد، موقعیت و اندازه ی هر آیتم قابل انتخاب، در ابتدای یک عملکرد انتخاب محاسبه می شود. مقدار پیش فرض آن true می باشد. |
| cancel | این گزینه انتخاب را ممنوع می کند. به طور پیش فرض مقدار آن input,textarea,button,select,option می باشد. |
| delay | این گزینه برای تنظیم زمان به هزارم ثانیه در زمان شروع انتخاب، استفاده می شود. این گزینه می تواند از انتخاب های ناخواسته جلوگیری کند. مقدار پیش فرض آن 0 است. |
| disabled | وقتی این گزینه روی true تنظیم شده باشد، مکانیزم انتخاب غیرفعال می شود. کاربران نمی توانند موردی را انتخاب کنند تا اینکه مکانیزم با استفاده از دستورالعمل selectable ("enable") به حالت قبل باز گردد. مقدار پیش فرض آن false می باشد. |
| distance | این گزینه فاصله ای است که ماوس (به پیکسل) باید جابه جا شود تا مشخص شود که انتخاب در حال پیشروی است. این مسئله به عنوان مثال برای جلوگیری از کلیک های ساده از اینکه به عنوان یک گروه ساده در نظر گرفته شود، جلوگیری می کند. مقدار پیش فرض آن 0 است. |
| filter | این گزینه انتخابگری است که نشان می دهد کدام عناصر می توانند بخشی از مجموعه باشند. مقدار پیش فرض آن * می باشد. |
| tolerance | این گزینه تعیین می کند که از کدام حالت برای امتحان کردن این مورد که مجموعه ی کمک کننده (lasso) باید یک آیتم را انتخاب کند، استفاده می شود. مقدار پیش فرض آن touch است. |

بخش بعد به شما چند مثال کاربردی از قابلیت انتخاب را نشان خواهد داد.

قابلیت پیش فرض

مثال زیر، نمونه‌ی ساده‌ای از قابلیت انتخاب می‌باشد که پارامتری به متد **selectable()** انتقال نمی‌دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI selectable-1</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>
    #selectable-1 .ui-selecting {
      background: #707070;
    }
    #selectable-1 .ui-selected {
      background: #EEEEEE;
      color: #000000;
    }
    #selectable-1 {
      list-style-type: none;
      margin: 0;
      padding: 0;
      width: 20%;
    }
    #selectable-1 li {
      margin: 3px;
      padding: 0.4em;
      font-size: 16px;
      height: 18px;
    }
    .ui-widget-content {
      background: #cedc98;
      border: 1px solid #DDDDDD;
      color: #333333;
    }
  </style>
  <script>
    $(function () {
      $("#selectable-1").selectable();
    });
  </script>
</head>
<body>
  <ol id="selectable-1">
    <li class="ui-widget-content">Product 1</li>
    <li class="ui-widget-content">Product 2</li>
    <li class="ui-widget-content">Product 3</li>
    <li class="ui-widget-content">Product 4</li>
  </ol>
```

```

<li class="ui-widget-content">Product 5</li>
<li class="ui-widget-content">Product 6</li>
<li class="ui-widget-content">Product 7</li>
</ol>
</body>
</html>

```

سعی کنید روی تولیدات کلیک کنید، از کلید **CTRL** استفاده کنید تا چند مورد را انتخاب کنید.

استفاده از **delay** و **distance**

مثال زیر استفاده از دو گزینه **delay** و **distance** را در عملکرد **selectable** مربوط به **JqueryUI** را توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Selectable</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>
#selectable-2 .ui-selecting, #selectable-3 .ui-selecting {
background: #707070;
}
#selectable-2 .ui-selected, #selectable-3 .ui-selected {
background: #EEEEEE;
color: #000000;
}
#selectable-2, #selectable-3 {
list-style-type: none;
margin: 0;
padding: 0;
width: 20%;
}
#selectable-2 li, #selectable-3 li {
margin: 3px;
padding: 0.4em;
font-size: 16px;
height: 18px;
}
.ui-widget-content {
background: #cedc98;
border: 1px solid #DDDDDD;
color: #333333;
}
</style>
<script>

```

```

$(function () {
    $("#selectable-2").selectable({
        delay: 1000
    });
    $("#selectable-3").selectable({
        distance: 100
    });
});
</script>
</head>
<body>
<h3>Starts after delay of 1000ms</h3>
<ol id="selectable-2">
<li class="ui-widget-content">Product 1</li>
<li class="ui-widget-content">Product 2</li>
<li class="ui-widget-content">Product 3</li>
</ol>
<h3>Starts after mouse moves distance of 100px</h3>
<ol id="selectable-3">
<li class="ui-widget-content">Product 4</li>
<li class="ui-widget-content">Product 5</li>
<li class="ui-widget-content">Product 6</li>
<li class="ui-widget-content">Product 7</li>
</ol>
</body>
</html>

```

سعی کنید روی تولیدات کلیک کنید، با استفاده از دکمه ی **CTRL** چند مورد را با هم انتخاب کنید. متوجه خواهید شد که انتخاب محصول **1** و **2** و **3** پس از یک وقفه ی **1000ms** انجام می شود. انتخاب محصولات **4**، **5**، **6** و **7** به طور مجزا انجام پذیر نمی باشند. انتخاب تنها پس از اینکه ماوس یک فاصله ی **100 px** را جابه جا می شود، آغاز می شود.

استفاده از فیلتر

مثال زیر استفاده از دو گزینه ی **delay** و **distance** را در قابلیت انتخاب از **JqueryUI** نشان می دهد.

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI selectable-4</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>
#selectable-4 .ui-selecting {

```



```

        background: #707070;
    }
    #selectable-4 .ui-selected {
        background: #EEEEEE;
        color: #000000;
    }
    #selectable-4 {
        list-style-type: none;
        margin: 0;
        padding: 0;
        width: 20%;
    }
    #selectable-4 li {
        margin: 3px;
        padding: 0.4em;
        font-size: 16px;
        height: 18px;
    }
    .ui-widget-content {
        background: #cedc98;
        border: 1px solid #DDDDDD;
        color: #333333;
    }
</style>
<script>
$(function () {
    $("#selectable-4").selectable({
        filter: "li:first-child"
    });
});
</script>
</head>
<body>
<ol id="selectable-4">
<li class="ui-widget-content">Product 1</li>
<li class="ui-widget-content">Product 2</li>
<li class="ui-widget-content">Product 3</li>
<li class="ui-widget-content">Product 4</li>
<li class="ui-widget-content">Product 5</li>
<li class="ui-widget-content">Product 6</li>
<li class="ui-widget-content">Product 7</li>
</ol>
</body>
</html>

```

سعی کنید روی محصولات کلیک کنید. متوجه خواهید شد که تنها اولین محصول قابلیت انتخاب شدن را دارد.

متد `$(selector, context).selectable ("action", params)`

متد `selectable ("action", params)` می تواند یک عملکرد را روی عناصر دارای قابلیت انتخاب اجرا کند، از جمله جلوگیری از قابلیت انتخاب. این عملکرد به عنوان یک رشته در اولین برهان تعیین می شود (به عنوان مثال `disable` برای متوقف کردن انتخاب). عملکردهایی را که منتقل می شوند در جدول زیر بررسی کنید.

ترکیب

`$(selector, context).selectable ("action", params);`

جدول زیر عملکردهای مختلفی را نشان می دهد که با این متد قابل استفاده می باشند:

| Action | Description |
|-----------------------------|--|
| destroy | این عملکرد قابلیت انتخاب را از یک عنصر کاملاً حذف می کند. عناصر به وضعیت pre-init خود بازمی گردند. |
| disable | این عملکرد قابلیت انتخاب را غیرفعال می سازد. این متد هیچ برهانی را نمی پذیرد. |
| enable | این عملکرد قابلیت انتخاب را فعال می سازد. این متد هیچ برهانی را نمی پذیرد. |
| option(optionName) | این عملکرد مقدار مربوط به optionName تعیین شده را به دست می آورد. |
| option() | این عملکرد آبجکتی حاوی جفت های key/value را می گیرد که نشان دهنده ی گزینه های انتخابی hash می باشند. |
| option(optionName, value) | این فعالیت مقدار گزینه ی selectale مربوط به optionName تعیین شده را تنظیم می کند. optionName نام گزینه ای است که قرار است تنظیم شود و value مقداری است که قرار است برای این گزینه تنظیم شود. |

| | |
|-------------------|---|
| option(options) | این عملکرد یک یا چند گزینه را برای انتخاب تنظیم می کند. Options نقشه ای است از جفت های option-value که قرار است تنظیم شوند. |
| refresh | این عملکرد باعث می شود که اندازه و موقعیت عناصر انتخابی دوباره تنظیم شوند. بیشتر زمانی استفاده می شود که گزینه ی autoRefresh غیر فعال باشد (در حالیکه روی false تنظیم باشد). این متد هیچ برهانی را قبول نمی کند. |
| widget | این عملکرد یک آبجکت jQuery می باشد که حاوی عنصر انتخابی می باشد. این متد هیچ برهانی را نمی پذیرد. |

مثال

اکنون اجازه بدهید که با استفاده از عملکردهای بالا، مثالی را بررسی کنیم. مثال زیر استفاده از متدهای **disable()** و **option(optionName, value)** را توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Selectable</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>
#selectable-5 .ui-selecting, #selectable-6 .ui-selecting {
background: #707070;
}
#selectable-5 .ui-selected, #selectable-6 .ui-selected {
background: #EEEEEE;
color: #000000;
}
#selectable-5, #selectable-6 {
list-style-type: none;
margin: 0;
padding: 0;
width: 20%;
}
#selectable-5 li, #selectable-6 li {
margin: 3px;
padding: 0.4em;
}
```

```

        font-size: 16px;
        height: 18px;
    }
    .ui-widget-content {
        background: #cedc98;
        border: 1px solid #DDDDDD;
        color: #333333;
    }
</style>
<script>
$(function () {
    $("#selectable-5").selectable();
    $("#selectable-5").selectable('disable');
    $("#selectable-6").selectable();
    $("#selectable-6").selectable("option", "distance", 1);
});
</script>
</head>
<body>
<h3>Disabled using disable() method</h3>
<ol id="selectable-5">
    <li class="ui-widget-content">Product 1</li>
    <li class="ui-widget-content">Product 2</li>
    <li class="ui-widget-content">Product 3</li>
</ol>
<h3>Select using method option( optionName, value )</h3>
<ol id="selectable-6">
    <li class="ui-widget-content">Product 4</li>
    <li class="ui-widget-content">Product 5</li>
    <li class="ui-widget-content">Product 6</li>
    <li class="ui-widget-content">Product 7</li>
</ol>
</body>
</html>

```

سعی کنید روی محصولات کلیک کنید، از دکمه ی **CTRL** برای انتخاب چند مورد استفاده کنید. متوجه خواهید شد که محصول 1، 2 و 3 غیرفعال می باشند. انتخاب محصول 4، 5، 6 و 7 پس از حرکت ماوس به اندازه ی 1 px اتفاق می افتد.

مدیریت رویداد روی عناصر انتخابی

علاو بر متد **selectable (options)** که در بخش قبل مشاهده کردیم، **JqueryUI** متدهای دیگری را ارائه می دهد که برای یک رویداد خاص آغاز می شوند. این متدها در جدول زیر لیست شده اند:

| Event Method | Description |
|------------------------|--|
| create(event, ui) | این رویداد زمانی آغاز میشود که عنصری دارای قابلیت بختخاب (selectable) ایجاد شود. رویداد از نوع Event و ui از نوع Object می باشد. |
| selected(event, ui) | این رویداد برای هر عنصری که انتخاب می شود، آغاز می شود. رویداد از نوع Event و ui از نوع Object می باشد. |
| selecting(event, ui) | این رویداد برای هر عنصری با قابلیت انتخاب که قرار است انتخاب شود، آغاز می شود. در حالیکه رویداد از نوع Event است، ui از نوع Object است. |
| start(event, ui) | این رویداد در ابتدای عملکرد انتخاب آغاز می شود. در حالیکه رویداد از نوع Event است، ui از نوع Object است. |
| stop(event, ui) | این رویداد در انتهای عملکرد انتخاب آغاز می شود. در حالیکه رویداد از نوع Event است، ui از نوع Object است. |
| unselected(event, ui) | این رویداد در انتهای عملکرد انتخاب برای هر عنصری که از حالت انتخاب خارج شده باشد، آغاز می شود. در حالیکه رویداد از نوع Event است، ui از نوع Object است. |
| unselecting(event, ui) | این رویداد در زمان عملکرد انتخاب برای عنصر انتخاب شده که قرار است از حالت انتخاب خارج شود، آغاز می شود. در حالیکه رویداد از نوع Event است، ui از نوع Object است. |

مثال زیر استفاده از متدی را در زمان قابلیت انتخاب، توضیح می دهد. این مثال استفاده از رویداد **selected** را توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
```

```

<meta charset="utf-8">
<title>jQuery UI selectable-7</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>
    #selectable-7 .ui-selecting {
        background: #707070;
    }
    #selectable-7 .ui-selected {
        background: #EEEEEE;
        color: #000000;
    }
    #selectable-7 {
        list-style-type: none;
        margin: 0;
        padding: 0;
        width: 20%;
    }
    #selectable-7 li {
        margin: 3px;
        padding: 0.4em;
        font-size: 16px;
        height: 18px;
    }
    .ui-widget-content {
        background: #cedc98;
        border: 1px solid #DDDDDD;
        color: #333333;
    }
    .resultarea {
        background: #cedc98;
        border-top: 1px solid #000000;
        border-bottom: 1px solid #000000;
        color: #333333;
        font-size: 14px;
    }
</style>
<script>
$(function () {
    $("#selectable-7").selectable({
        selected: function () {
            var result = $("#result").empty();
            $(".ui-selected", this).each(function () {
                var index = $("#selectable-7 li").index(this);
                result.append(" #" + (index + 1));
            });
        }
    });
});
</script>
</head>
<body>
<h3>Events</h3>
<ol id="selectable-7">
    <li class="ui-widget-content">Product 1</li>

```

```

<li class="ui-widget-content">Product 2</li>
<li class="ui-widget-content">Product 3</li>
<li class="ui-widget-content">Product 4</li>
<li class="ui-widget-content">Product 5</li>
<li class="ui-widget-content">Product 6</li>
<li class="ui-widget-content">Product 7</li>
</ol>
<span class="resultarea">Selected Product</span><span></span>
<span id=result class="resultarea"></span>
</body>
</html>

```

سعی کنید که روی محصولات کلیک کنید، با استفاده از دکمه ی **CTRL** چند مورد را انتخاب کنید. شما تعداد موارد انتخاب شده را در پایین مشاهده خواهید کرد.

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:

| 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. ایک گزینه به آیتم ها اجازه می دهد تا به لیست های دیگر جابه جاشوند 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 ارکیب 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 زمانی که 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.

آموزشگاه حلکیر داده

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;
    }
  </style>
```

```

padding-left: 1.5em;
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>

```

تولیدات بالا را دوباره تنظیم کنید، از ماوس برای درگ کردن آیتم ها استفاده کنید.

استفاده از گزینه های delay و distance

مثال زیر استفاده از دو گزینه ی (a) delay و (b) distance را در قابلیت 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-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 مرتب سازی تصادفی چه با زمان و چه با فاصله، ما چند هزارم ثانیه برای عنصری تنظیم کرده ایم که قبل از شروع شدن مرتب سازی نیاز به درگ شدن دارد. ما این کار را با استفاده از گزینه ی **distance** نیز انجام داده ایم.

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 `dropOnEmpty` و `connectWith`

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 : **#sortable-5, #sortable-6** : **connectWith**).

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 کاملاً 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 مجدداً 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 ی 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

`$(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>
</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.

Accordion Widget

Accordion Widget در **jQueryUI** یک نگهدارنده ی محتوا قابل ارتقا و پیش ساخته بر اساس **jQuery** است که به بخش های مختلفی تقسیم می شود که احتمالاً شبیه به تب هستند. **jQuery** متد **accordion()** را برای دستیابی به این امر ارائه می دهد.

ترکیب:

متد **accordion()** به دو فرم استفاده می شود

\$(selector, context).accordion (options) Method

\$(selector, context).accordion ("action", params) Method

متد **\$(selector, context).accordion (options)**

متد **accordion (options)** اعلام می کند که یک عنصر **HTML** و محتوای آن باید مانند یک منوهای **accordion** تنظیم و مورد بررسی قرار بگیرند. پارامتر **options** آبجکتی است که ظاهر و رفتار منوی مورد نظر را تعیین می کند.

\$(selector, context).accordion (options);

شما می توانید یک یا چند گزینه را با استفاده از آبجکت **javascript** همزمان ارائه دهید. اگر بیشتر از یک گزینه برای ارائه وجود دارد، آنها را می توانید با استفاده از کاما مانند زیر، جدا کنید:

```
$(selector, context).accordion({option1: value1, option2: value2.....});
```

جدول زیر گزینه های مختلفی را که با این متد استفاده می شوند، ارائه می کند:

| Option | Description |
|---------|--|
| active | <p>ایندکس مربوط به منویی را که باز است، وقتی که صفحه برای اولین بار دیده می شود، نشان می دهد. مقدار پیش فرض آن 0 است، به عنوان مثال اولین منو.</p> <p>Option - active</p> <p>Indicates the index of the menu that is open when the page is first accessed. By default its value is 0, i.e the first menu.</p> <p>ایندکس منویی که باز است را نشان می دهد، وقتی صفحه برای اولین بار دیده می شود. مقدار پیش فرض آن 0 است، به عنوان مثال اولین منو.</p> <p>می تواند از انواع زیر باشد:</p> <ul style="list-style-type: none"> Boolean: اگر روی false تنظیم شده باشد، همه ی پنل ها را خراب می کند. لازم است که گزینه ی collapsible روی true تنظیم شده باشد. Integer: یک شاخص 0 محور از پنل که فعال است. یک مقدار منفی پنل را از آخرین پنل، به عقب می راند. <pre>\$(".selector").accordion({ active: 2 });</pre> |
| animate | <p>این گزینه برای تنظیم چگونگی محرک سازی پنل های در حال تغییر استفاده می شود. مقدار پیش فرض آن {} است.</p> |

| | |
|-------------|--|
| | <p>Option - animate</p> <p>این گزینه برای امتحان کردن چگونگی حرکت پنل های در حال تغییر استفاده می شود. مقدار پیش فرض آن {} است.</p> <p>این گزینه می تواند از انواع زیر باشد:</p> <ul style="list-style-type: none"> Boolean: مقدار false انیمیشن ها را غیرفعال خواهد کرد. Number: طول دوره به هزارم ثانیه می باشد. String: نام easing برای استفاده با مدت زمان به طور پیش فرض. Object: تنظیمات انیمیشن با پراپرتی های easing و duration. <pre>\$(".selector").accordion({ animate: "bounceslide" });</pre> |
| collapsible | <p>وقتی این گزینه روی true تنظیم شده باشد، به یوزرها اجازه می دهد تا با کلیک کردن روی یک منو آن را ببندند. به طور پیش فرض کلیک کردن روی تیتل پنل باز، هیچ تاثیری ندارد. مقدار پیش فرض آن false است.</p> <p>Option - collapsible</p> <p>وقتی این گزینه روی true تنظیم شده باشد، به یوزرها اجازه می دهد تا با کلیک کردن روی یک منو آن را ببندند. به طور پیش فرض کلیک کردن روی تیتل پنل باز، هیچ تاثیری ندارد. مقدار پیش فرض آن false است.</p> <pre>\$(".selector").accordion({ collapsible: true });</pre> |
| disabled | <p>وقتی این گزینه روی true تنظیم شده باشد، accordion را غیرفعال می کند. مقدار پیش فرض آن false است.</p> |

| | |
|-------------|--|
| | <p>Option - disabled</p> <p>وقتی این گزینه روی true تنظیم شده باشد، accordion را غیرفعال می کند. مقدار پیش فرض آن false است.</p> <p><code>\$(".selector").accordion({ disabled: true });</code></p> |
| event | <p>این گزینه رویداد استفاده شده برای انتخاب تیتراک accordion تعیین می کند. مقدار پیش فرض آن click است.</p> <p>Option - event</p> <p>این گزینه رویداد استفاده شده برای انتخاب تیتراک accordion تعیین می کند. مقدار پیش فرض آن click است.</p> <p><code>\$(".selector").accordion({ event: "mouseover" });</code></p> |
| header | <p>این گزینه یک انتخابگر یا یک عنصر را برای رد شدن از الگوی پیش فرض برای تشخیص عناصر تیتراک، تعیین می کند. مقدار پیش فرض آن <code>:first-child,>:not(li):even</code> است.</p> <p>Option - header</p> <p><code>\$(".selector").accordion({ header: "h3" });</code></p> |
| heightStyle | <p>این گزینه برای کنترل طول accordion و پنل ها استفاده می شود. مقدار پیش فرض آن auto است.</p> <p>Option - heightStyle</p> <p>این گزینه برای کنترل طول accordion و پنل ها استفاده می شود. مقدار پیش فرض آن auto است.</p> <p>مقادیر ممکن عبارتند از:</p> |

| | |
|-------|--|
| | <ul style="list-style-type: none"> • auto: همه ی پنل ها براساس طول بلندترین پنل تنظیم خواهند شد. • fill: براساس طول 'accordion' اصلی به میزان طول در دسترس گسترده می شود. • content: هر پنل دارای طولی به اندازه ی محتوای آن است. <pre>\$(".selector").accordion({ heightStyle: "fill" });</pre> |
| icons | <p>این گزینه آبجکتی است که آیکن هایی را تعریف می کند که برای سمت چپ تیتر متن، برای پنل های باز و بسته، تعریف می کند. آیکن مورد استفاده برای پنل های بسته به عنوان نام header تعیین می شود، در حالیکه آیکن مورد استفاده برای پنل های باز headerSelected تعیین می شود. مقدار پیش فرض آن { "header": "ui-icon-triangle-1-e", "activeHeader": "ui-icon-triangle-1-s" } است.</p> <p>Option – icons</p> <p>این گزینه آبجکتی است که آیکن هایی را تعریف می کند که برای سمت چپ تیتر متن، برای پنل های باز و بسته، تعریف می کند. آیکن مورد استفاده برای پنل های بسته به عنوان نام header تعیین می شود، در حالیکه آیکن مورد استفاده برای پنل های باز headerSelected تعیین می شود. مقدار پیش فرض آن { "header": "ui-icon-triangle-1-e", "activeHeader": "ui-icon-triangle-1-s" } است.</p> <pre>\$(".selector").accordion({ icons: { "header": "ui-icon-plus", "activeHeader": "ui-icon-minus" } });</pre> |

بخش بعدی چند مثال کاربردی در رابطه با قابلیت **accordion widget** ارائه می شود.

قابلیت پیش فرض

مثال زیر قابلیت ساده ی **accordion widget** را بدون انتقال پارامتر در متد **accordion()** توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Accordion 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>
<script>
$(function () {
    $("#accordion-1").accordion();
});
</script>
<style>
#accordion-1 {
    font-size: 14px;
}
</style>
</head>
<body>
<div id="accordion-1">
<h3>Tab 1</h3>
<div>
<p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit,
        sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
        Ut enim ad minim veniam, quis nostrud exercitation ullamco
        laboris nisi ut aliquip ex ea commodo consequat.
    </p>
</div>
<h3>Tab 2</h3>
<div>
<p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit,
        sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
        Ut enim ad minim veniam, quis nostrud exercitation ullamco
        laboris nisi ut aliquip ex ea commodo consequat.
    </p>
</div>
<h3>Tab 3</h3>
<div>
<p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit,
        sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
        Ut enim ad minim veniam, quis nostrud exercitation ullamco
        laboris nisi ut aliquip ex ea commodo consequat.
    </p>
</div>
</div>
</body>
</html>

```

روی عنوان ها (Tab 1, Tab 2, Tab 3) کلیک کنید تا محتوای شکسته شده به بخش های منطقی را گسترش داده / خراب کنید (expand/collapse).

مثال زیر استفاده از گزینه ی collapsible را در accordion widget از JQueryUI توضیح می دهد:

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Accordion 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>
  <script>
    $(function () {
      $("#accordion-2").accordion({
        collapsible: true
      });
    });
  </script>
  <style>
    #accordion-2 {
      font-size: 14px;
    }
  </style>
</head>
<body>
  <div id="accordion-2">
    <h3>Tab 1</h3>
    <div>
      <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit,
        sed do eiusmod tempor incididunt ut labore et dolore magna
        aliqua. Ut enim ad minim veniam, quis nostrud exercitation
        ullamco laboris nisi ut aliquip ex ea commodo consequat.
      </p>
    </div>
    <h3>Tab 2</h3>
    <div>
      <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit,
        sed do eiusmod tempor incididunt ut labore et dolore magna
        aliqua. Ut enim ad minim veniam, quis nostrud exercitation
        ullamco laboris nisi ut aliquip ex ea commodo consequat.
      </p>
    </div>
    <h3>Tab 3</h3>
    <div>
      <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit,
        sed do eiusmod tempor incididunt ut labore et dolore magna
        aliqua. Ut enim ad minim veniam, quis nostrud exercitation
        ullamco laboris nisi ut aliquip ex ea commodo consequat.
      </p>
      <ul>
```

```

</li>List item one</li>
</li>List item two</li>
</li>List item three</li>
</ul>
</div>
</div>
</body>
</html>

```

در اینجا **collapsible** را به **true** تنظیم کرده ایم. روی تیترا **accordion** کلیک کنید، این باعث سقوط بخش فعال می شود.

استفاده از **heightStyle**

مثال زیر استفاده از گزینه های **heightStyle** را در **accordion widget** از **jQueryUI** توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Accordion 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>
<script>
$(function () {
  $("#accordion-3").accordion({
    heightStyle: "content"
  });
  $("#accordion-4").accordion({
    heightStyle: "fill"
  });
});
</script>
<style>
#accordion-3, #accordion-4 {
  font-size: 14px;
}
</style>
</head>
<body>
<h3>Height style-content</h3>
<div style="height:250px">
<div id="accordion-3">
<h3>Tab 1</h3>
<div>
<p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore

```

```

magna aliqua.
</p>
<ul>
  <li>List item one</li>
  <li>List item two</li>
  <li>List item three</li>
  <li>List item four</li>
  <li>List item five</li>
</ul>
</div>
<h3>Tab 2</h3>
<div>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore
    magna aliqua.
  </p>
</div>
<h3>Tab 3</h3>
<div>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore
    magna aliqua.
  </p>
</div>
</div>
<div><br><br></div>
<h3>Height style-Fill</h3>
<div style="height:250px">
  <div id="accordion-4">
    <h3>Tab 1</h3>
    <div>
      <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing
        elit, sed do eiusmod tempor incididunt ut labore
        et dolore magna aliqua.
      </p>
      <ul>
        <li>List item one</li>
        <li>List item two</li>
        <li>List item three</li>
        <li>List item four</li>
        <li>List item five</li>
      </ul>
    </div>
    <h3>Tab 2</h3>
    <div>
      <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing
        elit, sed do eiusmod tempor incididunt ut labore
        et dolore magna aliqua.
      </p>
    </div>
    <h3>Tab 3</h3>
    <div>

```

```

<p>
  Lorem ipsum dolor sit amet, consectetur adipisicing
  elit, sed do eiusmod tempor incididunt ut labore
  et dolore magna aliqua.
</p>
</div>
</div>
</div>
</body>
</html>

```

در اینجا دو **accordion** وجود دارد که اولی دارای گزینه ی **heightStyle** است که بر روی **content** تنظیم شده که به پنل های **accordion** اجازه می دهد تا طول اصلی خود را حفظ کنند. دومین **accordion** دارای گزینه ی **heightStyle** است که روی **fill** تنظیم شده است، متن به طور خودکار ابعاد **accordion** را به طول ظرف اصلی تنظیم می کند.

متد `$(selector, context).accordion("action", params)`

متد `$(selector, context).accordion("action", params)` می تواند یک اکشن را روی عناصر **accordion** اجرا می کند، مانند انتخاب کردن/ از حالت انتخاب در آوردن منوی **accordion**. این اکشن به عنوان یک رشته در اولین برهان تعیین می شود (برای مثال **"disable"** همه ی منوها را غیرفعال می سازد.) اکشن هایی را که می توانند منتقل شوند، در جدول زیر بررسی کنید.

`$(selector, context).accordion("action", params);`

جدول زیر اکشن های مختلفی را لیست می کند که با این متد استفاده می شوند:

| Action | Description |
|--------|-------------|
|--------|-------------|

| | |
|----------------|---|
| <p>destroy</p> | <p>این اکشن قابلیت accordion را از یک عنصر به طور کامل از بین می برد. عناصر به وضعیت pre-init خود بازمی گردند.</p> <p>Action - destroy</p> <p>این اکشن قابلیت accordion را از یک عنصر به طور کامل از بین می برد. عناصر به وضعیت pre-init خود بازمی گردند.</p> <p><code>\$(".selector").accordion("destroy");</code></p> |
| <p>disable</p> | <p>این اکشن منوها را غیرفعال می کند. هیچ کلیک کردنی به حساب نمی آید. این متد هیچ برهانی را نمی پذیرد.</p> <p>Action - disable</p> <p>این اکشن منوها را غیرفعال می کند. هیچ کلیک کردنی به حساب نمی آید. این متد هیچ برهانی را نمی پذیرد.</p> <p><code>\$(".selector").accordion("disable");</code></p> |
| <p>enable</p> | <p>این اکشن همه ی منوها را مجددا فعال می سازد. کلیک کردن ها دوباره در نظر گرفته می شوند. این متد هیچ برهانی را نمی پذیرد.</p> <p>Action - enable</p> <p>این اکشن همه ی منوها را مجددا فعال می سازد. کلیک کردن ها دوباره در نظر گرفته می شوند. این متد هیچ برهانی را نمی پذیرد.</p> <p><code>\$(".selector").accordion("enable");</code></p> |

| | |
|------------------------------------|---|
| <p>option(optionName)</p> | <p>این اکشن آخرین مقدار مرتبط به عنصر accordion را با optionName تعیین شده، دریافت می کند و مقدار یک String را به عنوان برهان در نظر می گیرد.</p> <p>Action - option(optionName)</p> <p>این اکشن آخرین مقدار مرتبط به عنصر accordion را با optionName تعیین شده، دریافت می کند و مقدار یک String را به عنوان برهان در نظر می گیرد.</p> <pre>var isDisabled = \$(".selector").accordion("option", "disabled");</pre> |
| <p>option</p> | <p>این اکشن یک آبجکت حاوی جفت های key/value را دریافت می کند که گزینه های فعلی hash مربوط به accordion را نمایش می دهد.</p> <p>Action - option</p> <p>این اکشن یک آبجکت حاوی جفت های key/value را دریافت می کند که گزینه های فعلی hash مربوط به accordion را نمایش می دهد.</p> <pre>var options = \$(".selector").accordion("option");</pre> |
| <p>option(optionName, value)</p> | <p>این اکشن مقدار گزینه ی accordion مربوط به optionName تعیین شده را تنظیم می کند.</p> <p>Action - option(optionName, value)</p> <p>این اکشن مقدار گزینه ی accordion مربوط به optionName تعیین شده را تنظیم می کند.</p> <pre>\$(".selector").accordion("option", "disabled", true);</pre> |

| | |
|--------------------------|--|
| <p>option(options)</p> | <p>این اکشن یک یا چند گزینه را برای accordion تنظیم می کند.</p> <p>Action - option(options)</p> <p>این اکشن یک یا چند گزینه را برای accordion تنظیم می کند. Options در واقع نقشه ای است از جفت های option-value برای تنظیم.</p> <pre>\$(".selector").accordion("option", { disabled: true });</pre> |
| <p>refresh</p> | <p>این اکشن هر تیترو پنلی را که به طور مستقیم به DOM اضافه شده یا از آن حذف شدند را پردازش می کند. سپس طول پنل های accordion را مجددا محاسبه می کند. نتایج بستگی به محتوا و گزینه ی heightStyle دارد. این متد هیچ برهانی را نمی پذیرد.</p> <p>Action - refresh</p> <p>این اکشن هر تیترو پنلی را که به طور مستقیم به DOM اضافه شده یا از آن حذف شدند را پردازش می کند. سپس طول پنل های accordion را مجددا محاسبه می کند. نتایج بستگی به محتوا و گزینه ی heightStyle دارد. این متد هیچ برهانی را نمی پذیرد.</p> <pre>\$(".selector").accordion("refresh");</pre> |
| <p>widjet</p> | <p>این اکشن عنصر accordion widjet را بازمی گرداند، عنصری که با نام گروه -ui accordion تفسیر می شود.</p> |

| | |
|--|---|
| | <p>Action - widget</p> <p>این اکشن عنصر accordion widget را بازمی گرداند، عنصری که با نام گروه ui-accordion تفسیر می شود.</p> <p><code>var widget = \$(".selector").accordion("widget");</code></p> |
|--|---|

مثال:

اکنون اجازه بدهید مثالی را در مورد اکشن های جدول بالا بررسی کنیم. مثال زیر استفاده از متد `option()` را توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Accordion 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>
  <script>
    $(function () {
      $("#accordion-5").accordion({
        disabled: false
      });
      $("input").each(function () {
        $(this).change(function () {
          if ($(this).attr("id") == "disableaccordion") {
            $("#accordion-5").accordion("option", "disabled", true);
          } else {
            $("#accordion-5").accordion("option", "disabled", false);
          }
        });
      });
    });
  </script>
  <style>
    #accordion-5 {
      font-size: 14px;
    }
  </style>
</head>
<body>
  <div id="accordion-5">

```

```

<h3>Tab 1</h3>
<div>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna
    aliqua. Ut enim ad minim veniam, quis nostrud
    exercitation ullamco laboris nisi ut aliquip ex ea
    commodo consequat.
  </p>
</div>
<h3>Tab 2</h3>
<div>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna
    aliqua. Ut enim ad minim veniam, quis nostrud
    exercitation ullamco laboris nisi ut aliquip ex ea
    commodo consequat.
  </p>
</div>
<h3>Tab 3</h3>
<div>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna
    aliqua. Ut enim ad minim veniam, quis nostrud
    exercitation ullamco laboris nisi ut aliquip ex ea
    commodo consequat.
  </p>
  <ul>
    <li>List item one</li>
    <li>List item two</li>
    <li>List item three</li>
  </ul>
</div>
</div>
<div style="margin-top:30px">
  <input type="radio" name="disable" id="disableaccordion"
    value="disable">Disable accordion
  <input type="radio" name="disable" id="enableaccordion" checked
    value="enable">Enable accordion
</div>
</body>
</html>

```

در اینجا فعالسازی و غیرفعالسازی **accordion** ها را توضیح دادیم. برای بررسی هر اکشن، دکمه های مربوط را انتخاب کنید.

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



| Event Method | Description |
|----------------------------------|---|
| <p>activate(event, ui)</p> | <p>این رویداد وقتی آغاز می شود که یک منو فعال می شود. این رویداد تنها برای فعالسازی پنل کاربرد دارد و برای پنل اصلی، در هنگام ایجاد accordion widget، استفاده نمی شود.</p> <p>Event - activate(event, ui)</p> <p>این رویداد وقتی آغاز می شود که یک منو فعال می شود. این رویداد تنها برای فعالسازی پنل کاربرد دارد و برای پنل اصلی، در هنگام ایجاد accordion widget، استفاده نمی شود. در حالیکه رویداد از نوع Event و ui از نوع Object است. مقادیر ممکن برای ui عبارتند از:</p> <p>newHeader: یک آبجکت jQuery که تیتري را نمایش می دهد که اخیرا فعال بوده است.</p> <p>oldHeader: یک آبجکت jQuery را نشان می دهد که اخیرا غیرفعال شده است.</p> <p>newPanel: یک آبجکت jQuery است که پنلی را نشان می دهد که اخیرا فعال بود.</p> <p>oldPanel: یک آبجکت jQuery است که پنلی را نشان می دهد که اخیرا غیر فعال بود.</p> <p><code>\$(".selector").accordion({ activate: function (event, ui) { } });</code></p> |
| <p>beforeActivate(event, ui)</p> | <p>این رویداد قبل از فعالسازی پنل، آغاز می شود. برای جلوگیری از فعالسازی پنل، این رویداد می تواند کنسل شود.</p> |

| | |
|--------------------------|--|
| | <p>Event - beforeActivate(event, ui)</p> <p>این رویداد قبل از فعالسازی پنل، آغاز می شود. برای جلوگیری از فعالسازی پنل، این رویداد می تواند کنسل شود. رویداد از نوع Event و ui از نوع object است. مقادیر ممکن برای ui عبارتند از:</p> <p>newHeader: یک آبجکت jQuery است که تیتري را نشان می دهد که قرار است فعالسازی شود.</p> <p>oldHeader: یک آبجکت jQuery است که تیتري را نشان می دهد که قرار است غیر فعالسازی شود.</p> <p>newPanel: یک آبجکت jQuery است که پنلی را نشان می دهد که قرار است فعالسازی شود.</p> <p>oldPanel: یک آبجکت jQuery است که پنلی را نشان می دهد که قرار است غیر فعالسازی شود.</p> <p>آموزشگاه تحلیکیر داده ها</p> <pre>\$(".selector").accordion({beforeActivate: function (event, ui) { }});</pre> |
| <p>create(event, ui)</p> | <p>این رویداد با ایجاد accordion ایجاد می شود.</p> <p>Event - create(event, ui)</p> <p>این رویداد با ایجاد accordion ایجاد می شود. در آن رویداد از نوع Event و ui از نوع object است. مقادیر ممکن برای ui عبارتند از:</p> <ul style="list-style-type: none"> header: یک آبجکت jQuery است که تیتري فعال را نمایش می دهد. |

- panel: یک آبجکت jQuery است که پنل فعال را نمایش می دهد.

```
$(".selector").accordion({create: function (event, ui) { }});
```

مثال:

مثال زیر استفاده از متد رویداد را در **accordion widgets** توضیح می دهد. این مثال استفاده از رویدادهای **create**, **beforeActive** و **active** را توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Accordion 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>
  <script>
    $(function () {
      $("#accordion-6").accordion({
        create: function (event, ui) {
          $("#span#result").html($("#span#result").html() + "<b>Created</b><br>");
        },
        beforeActivate: function (event, ui) {
          $("#span#result").html($("#span#result").html() + ", <b>beforeActivate</b><br>");
        },
        activate: function (event, ui) {
          $("#span#result").html($("#span#result").html() + "<b>activate</b><br>");
        }
      });
    });
  </script>
  <style>
    #accordion-6 {
      font-size: 14px;
    }
  </style>
</head>
<body>
  <div id="accordion-6">
    <h3>Tab 1</h3>
    <div>
      <p>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit,
        sed do eiusmod tempor incididunt ut labore et dolore
        magna aliqua. Ut enim ad minim veniam, quis nostrud
        exercitation ullamco laboris nisi ut aliquip ex ea
        commodo consequat.
      </p>
    </div>
    <h3>Tab 2</h3>
  </div>
</body>
</html>
```

```

<div>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit,
    sed do eiusmod tempor incididunt ut labore et dolore
    magna aliqua. Ut enim ad minim veniam, quis nostrud
    exercitation ullamco laboris nisi ut aliquip ex ea
    commodo consequat.
  </p>
</div>
<h3>Tab 3</h3>
<div>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit,
    sed do eiusmod tempor incididunt ut labore et dolore
    magna aliqua. Ut enim ad minim veniam, quis nostrud
    exercitation ullamco laboris nisi ut aliquip ex ea
    commodo consequat.
  </p>
  <ul>
    <li>List item one</li>
    <li>List item two</li>
    <li>List item three</li>
  </ul>
</div>
</div>
<hr />
<span id=result></span>
</body>
</html>

```

در اینجا متنی را براساس رویدادها در پایین نمایش می دهیم.

JqueryUI - Autocomplete

Autocompletion مکانیزمی است که به طور مداوم در وب سایت های مدرن برای ارائه ی لیستی از پیشنهادات به کاربرها برای اولین حرفی که در تکست باکس تایپ کرده، استفاده می شود. سپس کاربر می تواند یک آیتم را از لیست انتخاب کند که در فیلد ورودی نمایش داده خواهد شد. این ویژگی باعث می شود کاربر مجبور به وارد کردن کامل یک لغت یا مجموعه ای لغات نباشد.

jQuery یک ویجت **autocomplete** را ارائه می دهد – کنترلی که بسیار مشابه یک **<select>** **dropdown** کار می کند، اما انتخاب ها را فیلتر می کند تا تنها مواردی را نمایش دهد که کاربر در حال تایپ در یک کنترل می باشد. **jQuery** متد **autocomplete()** برای ایجاد لیست پیشنهادات در زیر فیلد ورودی، ارائه می دهد و گروه های جدید **CSS** را به عناصر اضافه می کند تا به آنها استایل جدید را بدهد.

هر فیلدی که می تواند یک ورودی دریافت کند، می تواند به یک **Autocomplete** تبدیل شود، یعنی عناصر **<input>** ، عناصر **<textarea>** ، و عناصری با ویژگی **contenteditable**.

متد () autocomplete به دو فرم زیر استفاده می شود

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

`$(selector, context).autocomplete ("action", params) Method`

متد `$(selector, context).autocomplete (options)`

متد **autocomplete (options)** اعلام می کند که یک عنصر **<input> HTML** باید به عنوان یک فیلد ورودی تنظیم شود که در بالای لیستی از پیشنهادات نمایش داده خواهد شد. پارامتر **options** آبجکتی است که رفتار لیست پیشنهاد ها را در هنگام تایپ کردن کاربر در فیلد ورودی، تعیین می کند.

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

با استفاده از آبجکت **Javascript** می توانید یک یا چند گزینه را همزمان ارائه داده و سپس آنها را به وسیله ی کاما از یکدیگر مجزا کنید:

`$(selector, context).autocomplete({option1: value1, option2: value2..... });`

جدول زیر گزینه های مختلفی را ارائه می دهد که با این متد استفاده می شوند.

| Option | Description |
|------------------|---|
| appendTo | این گزینه برای ضمیمه کردن یک عنصر به منو استفاده می شود. مقدار پیش فرض آن null است. |
| autoFocus | اگر این گزینه روی true تنظیم شده باشد، وقتی منو نمایش داده شود، به طور خودکار اولین آیتم منو متمرکز می شود. مقدار پیش فرض آن false است. |
| delay | این گزینه یک عدد صحیح می باشد که زمان انتظار را قبل از هماهنگ کردن مقادیر، به هزارم ثانیه نشان می دهد (همانطور که توسط گزینه ی source تعیین شده است). وقتی یک داده ی |

| | |
|-----------|--|
| | غیرداخلی با دادن وقت به کاربر برای وارد کردن کاراکترهای بیشتر قبل از شروع جستجو، مورد استفاده قرار می گیرد، این امر trashing را کاهش می دهد. مقدار پیش فرض آن 300 است. |
| disabled | اگر این گزینه روی true تنظیم شده باشد، autocomplete widget از ابتدا غیرفعال خواهد بود. مقدار پیش فرض آن false است. |
| minLength | تعداد کاراکترهایی که باید قبل از هماهنگ سازی مقادیر (همانطور که توسط گزینه ی source تعیین شد)، وارد شوند. می تواند از تعیین مقادیر بیش از حد بزرگ جلوگیری کند وقتی که تعداد کاراکترها برای تنظیم عدد کافی نیستند. مقدار پیش فرض آن 1 است. |
| position | این گزینه موقعیت پیشنهادهای منو را در ارتباط با عنصر ورودی مرتبط تشخیص می دهد. گزینه ی Of پیش فرض عنصر ورودی است، اما می توانید عنصر دیگری را در مقابل این موقعیت تعیین کنید. مقدار پیش فرض آن { my: "left top", at: "left bottom", collision: "none" } است. |
| source | این گزینه روشی را تعیین می کند که در آن داده ی هماهنگ با داده ی ورودی، تنظیم می شود. یک مقدار باید ارائه شود، در غیر این صورت autocomplete widget ایجاد نخواهد شد. مقدار پیش فرض آن none; must be specified is است. |

بخش بعدی مثال های کاربردی در مورد قابلیت ویجت **autocomplete** را ارائه می دهد.

قابلیت پیش فرض

مثال زیر به طور ساده قابلیت ویجت **autocomplete** را توضیح میدهد که هیچ پارامتری را به متد

autocomplete() انتقال نمی دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Autocomplete functionality</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>
<!-- Javascript -->
<script>
$(function () {
    var availableTutorials = [
        "ActionScript",
        "Bootstrap",
        "C",
        "C++",
    ];
    $("#autocomplete-1").autocomplete({
        source: availableTutorials
    });
});
</script>
</head>
<body>
<!-- HTML -->
<div class="ui-widget">
<p>Type "a" or "s"</p>
<label for="autocomplete-1">Tags: </label>
<input id="autocomplete-1">
</div>
</body>
</html>

```

استفاده از autofocus

مثال زیر استفاده از گزینه ی **autoFocus** را در ویجت **autocomplete** از **jQuery** توضیح می دهد:

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Autocomplete functionality</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>
<!-- Javascript -->
<script>
$(function () {
    var availableTutorials = [
        "ActionScript",
        "Bootstrap",
        "C",
        "C++",
    ];
    $("#autocomplete-2").autocomplete({
        source: availableTutorials,
        autoFocus: true
    });
});

```

```

</script>
</head>
<body>
  <!-- HTML -->
  <div class="ui-widget">
    <p>Type "a" or "s"</p>
    <label for="autocomplete-2">Tags: </label>
    <input id="autocomplete-2">
  </div>
</body>
</html>

```

استفاده از minLength و delay

مثال زیر استفاده از گزینه های **minLength** و **delay** را در ویجت **autocomplete** از **jQuery** توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Autocomplete functionality</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>
  <!-- Javascript -->
  <script>
    $(function () {
      var availableTutorials = [
        "ActionScript",
        "Bootstrap",
        "C",
        "C++",
        "Ecommerce",
        "Jquery",
        "Groovy",
        "Java",
        "JavaScript",
        "Lua",
        "Perl",
        "Ruby",
        "Scala",
        "Swing",
        "XHTML"
      ];
      $("#autocomplete-3").autocomplete({
        minLength: 2,
        delay: 500,
        source: availableTutorials
      });
    });
  </script>

```

```

</head>
<body>
  <!-- HTML -->
  <div class="ui-widget">
    <p>Type two letter for e.g:ja,sc etc</p>
    <label for="autocomplete-3">Tags: </label>
    <input id="autocomplete-3">
  </div>
</body>
</html>

```

استفاده از label

مثال زیر استفاده از گزینه ی label را در autocomplete widget از JQueryUI توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Autocomplete functionality</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>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#autocomplete-4").autocomplete({
        source: [
          { label: "India", value: "IND" },
          { label: "Australia", value: "AUS" }
        ]
      });
    });
  </script>
</head>
<body>
  <!-- HTML -->
  <div class="ui-widget">
    <p>Type I OR A</p>
    <input id="autocomplete-4">
  </div>
</body>
</html>

```

استفاده از source خارجی

مثال زیر استفاده از فایل خارجی را برای گزینه ی source در autocomplete widget از JQueryUI

توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Autocomplete functionality</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>
  <!-- Javascript -->
  <script>
    $( "#autocomplete-5" ).autocomplete({
      source: "jqueryui/search.php",
      minLength: 2
    });
  </script>
</head>
<body>
  <input id="autocomplete-5">
</body>
</html>

```

فایل **search.php** مانند فایل بالا، در همان موقعیت جایگزین می شود (**autocompleteexample.html**).
محتوای **search.php** در زیر نمایش داده می شود.

```

<?
$term = $_GET[ "term" ];
$companies = array(
    array( "label" => "JAVA", "value" => "1" ),
    array( "label" => "DATA IMAGE PROCESSING", "value" => "2" ),
    array( "label" => "JAVASCRIPT", "value" => "3" ),
    array( "label" => "DATA MANAGEMENT SYSTEM", "value" => "4" ),
    array( "label" => "COMPUTER PROGRAMMING", "value" => "5" ),
    array( "label" => "SOFTWARE DEVELOPMENT LIFE CYCLE", "value" => "6" ),
    array( "label" => "LEARN COMPUTER FUNDAMENTALS", "value" => "7" ),
    array( "label" => "IMAGE PROCESSING USING JAVA", "value" => "8" ),
    array( "label" => "CLOUD COMPUTING", "value" => "9" ),
    array( "label" => "DATA MINING", "value" => "10" ),
    array( "label" => "DATA WAREHOUSE", "value" => "11" ),
    array( "label" => "E-COMMERCE", "value" => "12" ),
    array( "label" => "DBMS", "value" => "13" ),
    array( "label" => "HTTP", "value" => "14" )
);
$result = array();
foreach ( $companies as $company ) {
    $companyLabel = $company[ "label" ];
    if ( strpos( strtoupper( $companyLabel ), strtoupper( $term ) )
        !== false ) {
        array_push( $result, $company );
    }
}
echo json_encode( $result );
?>

```


متد \$(selector, context).autocomplete ("action", params)

متد **autocomplete ("action", params)** می تواند فعالیتی را روی لیست پیشنهادات انجام دهد، به عنوان مثال پنهان کردن و یا نمایش دادن. فعالیت به عنوان یک **String** در اولین برهان نمایش داده می شود (به عنوان مثال close برای پنهان کردن لیست). اکشن هایی را که قابل انتقال هستند، در جدول زیر مشاهده می کنید.

`$(selector, context).autocomplete ("action", params);`

جدول زیر اکشن هایی را ارائه می دهد که با این متد استفاده می شوند:

| Action | Description |
|-------------------------|--|
| close | این اکشن لیست پیشنهادات را در Autocomplete پنهان می کند. این متد هیچ آرگیومنتی نمی پذیرد. |
| destroy | این اکشن قابلیت Autocomplete را حذف می کند. لیست پیشنهادات پاک می شود و این متد هیچ آرگیومنتی نمی پذیرد. |
| disable | این اکشن مکانیزم autocomplete را غیرفعال می سازد. لیست پیشنهادات دیگر ظاهر نمی شود. این متد هیچ آرگیومنتی نمی پذیرد. |
| enable | این اکشن مکانیزم autocomplete را مجددا فعالسازی می کند. لیست پیشنهادات نیز دوباره نمایش داده خواهند شد. این متد هیچ آرگیومنتی نمی پذیرد. |
| option(optionName) | این اکشن مقدار تعیین شده برای پارامتر optionName را بازیابی می کند. این گزینه با یکی از گزینه های به کار برده شده در autocomplete مطابقت دارد. |

| | |
|-----------------------------------|---|
| option | این اکشن یک آبجکت حاوی جفت های key/value را دریافت می کند که گزینه های hash مربوط به autocomplete را نشان می دهد. |
| option(optionName, value) | این اکشن مقدار گزینه ی autocomplete را مربوط به optionName تعیین شده، تنظیم می کند. مبحث optionName نام گزینه ای است که باید تنظیم شود و value مقداری است که باید برای گزینه تنظیم شود. |
| option(options) | این اکشن یک یا چند گزینه را برای autocomplete تنظیم می کند. مبحث options نقشه ای است از جفت های option-value که قرار است تنظیم شوند. |
| search([value]) | این اکشن به دنبال نوعی سازگاری بین مقدار رشته و منبع داده است (در options.source تعیین شده است). حداقل تعداد کارکترها باید به مقدار برسند، در غیر این صورت جستجو اجرا نمی شود. |
| widget | عنصر DOM را بازمی گرداند که با لیست پیشنهادها مطابقت دارد. این اکشن، آبجکتی است از گروه jQuery که امکان دسترسی آسان به لیست را بدون استفاده از انتخابگرهای jQuery فراهم می کند. |

مثال:

اکنون اجازه بدهید با استفاده از اکشن های بالا، مثالی را بررسی کنیم. مثال زیر استفاده از متد **option(optionName, value)** را توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Autocomplete functionality</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>
  <!-- Javascript -->
  <script>
    $(function () {
      var availableTutorials = [
        "ActionScript",

```

```

"Bootstrap",
"C",
"C++",
"Ecommerce",
"jQuery",
"Groovy",
"Java",
"JavaScript",
"Lua",
"Perl",
"Ruby",
"Scala",
"Swing",
"XHTML"
];
$("#autocomplete-6").autocomplete({
  source: availableTutorials
});
$("#autocomplete-6").autocomplete("option", "position", { my: "right-10 top+10", at: "right top" });
});
</script>
</head>
<body>
<!-- HTML -->
<div class="ui-widget">
  <p>Type "a" or "s"</p>
  <label for="autocomplete-6">Tags: </label>
  <input id="autocomplete-6">
</div>
</body>
</html>

```

نکات گسترش

autocomplete widget می تواند همانطور که با ساخته شد، گسترش یابد. وقتی ویجت ها را گسترش می دهید، قادرید که رفتارهایی را به متدهای موجود اضافه کرده و یا حذف کنید. جدول زیر متدهایی را ارائه می دهد که در نکات گسترش با همان ثبات **API** رفتار می کنند که متدهای پلاگین در بالا داشتند.

| Method | Description |
|--------------------------------------|---|
| <code>_renderItem(ul, item)</code> | این متد ایجاد هر گزینه را در منوی ویجت کنترل می کند. این متد یک عنصر <code></code> جدید ایجاد می کند، آن را به منو ضمیمه کرده و بازمی گرداند. |

| | |
|-------------------------------------|--|
| <code>_renderMenu(ul, items)</code> | این متد ساخت منوی ویجت را کنترل می کند. |
| <code>_resizeMenu()</code> | این متد اندازه ی منو را قبل از نمایش داده شدن آن کنترل می کند. عنصر منو در <code>menu.element</code> موجود می باشد. این متد هیچ برهانی را نمی پذیرد. |

مدیریت رویداد روی عنصر `autocomplete`

علاوه بر متد `autocomplete (options)` که در بخشهای قبل مشاهده کردیم، `JqueryUI` متدهای دیگری ارائه می دهد که برای یک رویداد خاص آغاز می شوند. این متدها در زیر ارائه شده اند:

| Event Method | Description |
|--------------------------------|---|
| <code>change(event, ui)</code> | این رویداد زمانی آغاز می شود که مقدار عنصر <code><input></code> براساس یک انتخاب تغییر می کند. در زمان آغاز، این رویداد همیشه پس از آغاز رویداد <code>close</code> اتفاق می افتد. |
| <code>close(event, ui)</code> | هر زمانی که منوی <code>autocomplete</code> بسته شود، این رویداد آغاز می شود. |
| <code>create(event, ui)</code> | وقتی <code>autocomplete</code> ایجاد می شود، این رویداد آغاز می شود. |
| <code>focus(event, ui)</code> | هرزمان که تمرکز بر روی یکی از انتخاب های منو باشد، این رویداد آغاز می شود. مقدار تمرکز بر روی <code><input></code> تنظیم می شود، مگر اینکه کنسل شود (به عنوان مثال با گزارش دادن <code>false</code>). |
| <code>open(event, ui)</code> | پس از اینکه داده آماده شده است و منو در آستانه ی باز شدن می باشد، این رویداد آغاز می شود. |

| | |
|----------------------------------|---|
| <code>response(event, ui)</code> | پس از اینکه جستجو کامل شده و منو نمایش داده شود، این رویداد آغاز می شود. این رویداد همیشه با تکمیل شدن جستجو آغاز می شود، حتی اگر منو نمایش داده نشود، زیرا هیچ نتیجه ای وجود ندارد، یا Autocomplete غیرفعال شده است. |
| <code>search(event, ui)</code> | این رویداد پس از اینکه ضوابط <code>delay</code> و <code>minLength</code> اجرا شده اند، آغاز می شود، درست قبل از اینکه مکانیزم تعیین شده توسط منبع، فعال شود. اگر کنسل شود، عملکرد جستجو هم متوقف می شود. |
| <code>select(event, ui)</code> | این رویداد زمانی آغاز می شود که یک مقدار از منوی autocomplete انتخاب شده باشد. کنسل کردن این رویداد از تنظیم مقدار برای عنصر <code><input></code> جلوگیری می کند (اما از بسته شدن منو جلوگیری نمی کند.) |

مثال:

مثال زیر استفاده از متد رویداد را در **autocomplete widgets** توضیح می دهد. این مثال استفاده از رویدادهای **focus** و **select** را توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Autocomplete functionality</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>
    #project-label {
      display: block;
      font-weight: bold;
      margin-bottom: 1em;
    }
    #project-icon {
      float: left;
      height: 32px;
      width: 32px;
    }
    #project-description {
      margin: 0;
  
```

```

padding: 0;
}
</style>
<!-- Javascript -->
<script>
$(function () {
    var projects = [
        {
            value: "java",
            label: "Java",
            desc: "write once run anywhere",
        },
        {
            value: "jquery-ui",
            label: "jQuery UI",
            desc: "the official user interface library for jQuery",
        },
        {
            value: "Bootstrap",
            label: "Twitter Bootstrap",
            desc: "popular front end frameworks ",
        }
    ];
    $("#project").autocomplete({
        minLength: 0,
        source: projects,
        focus: function (event, ui) {
            $("#project").val(ui.item.label);
            return false;
        },
        select: function (event, ui) {
            $("#project").val(ui.item.label);
            $("#project-id").val(ui.item.value);
            $("#project-description").html(ui.item.desc);
            return false;
        }
    })
    .data("ui-autocomplete")._renderItem = function (ul, item) {
        return $("- 
")
            .append("<a>" + item.label + "<br>" + item.desc + "</a>")
            .appendTo(ul);
    };
});
</script>
</head>
<body>
<div id="project-label">Select a project (type "a" for a start):</div>
<input id="project">
<input type="hidden" id="project-id">
<p id="project-description"></p>
</body>
</html>

```

JqueryUI - Button

jQueryUI متد **button()** را ارائه می دهد برای تبدیل عناصر **HTML** (مانند دکمه ها، ورودی ها و **anchor** ها) به دکمه های دارای موضوع (**themeable**)، با تنظیم حرکات ماوس روی آنها، ارائه می دهد، که همه به طور واضحی توسط **jQuery UI** تنظیم شده اند.

برای گروه بندی دکمه های رادیو، **Button** همچنین یک ویجت اضافه به نام **Buttonset** ارائه می دهد. **Buttonset** با انتخاب یک حامل عنصر (که حاوی دکمه های رادیو است) و فراخوانی متد **buttonset()** استفاده می شود.

ترکیب:

متد **button()** می تواند به دو شکل زیر استفاده می شود

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

`$(selector, context).button ("action", params) Method`

متد **\$(selector, context).button (options)**

متد **button (options)** اعلام می کند که یک عنصر **HTML** باید مانند یک دکمه در نظر گرفته شود. پارامتر **options** آبجکتی است که رفتار و ظاهر دکمه را تعیین می کند.

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

شما می توانید با استفاده از آبجکت **Javascript** یک یا چند گزینه را همزمان ارائه دهید. اگر بیشتر از یک گزینه وجود داشته باشد، می توانید آنها را با استفاده از کاما از یکدیگر جدا کنید، مانند ترکیب زیر:

`$(selector, context).button({option1: value1, option2: value2.....});`

جدول زیر لیست های **option** های مختلف را ارائه می دهد که با این متد مورد استفاده قرار می گیرند:

| Option | Description |
|----------|---|
| disabled | این گزینه دکمه ی تنظیم شده روی true را غیرفعال می سازد. مقدار پیش فرض آن false است. |
| icons | این گزینه تعیین می کند که یک یا دو آیکن در button نمایش داده شوند: آیکن های اولیه در سمت چپ و آیکن های ثانویه در سمت راست. آیکن اولیه با پراپرتی primary و آیکن ثانویه با پراپرتی secondary تشخیص داده می شود. مقدار پیش فرض آن primary: null, secondary: null می باشد. |
| label | این گزینه متنی را برای نمایش روی دکمه تعیین می کند که برچسب معمول را رد می کند. اگر حذف شود، برچسب معمول نمایش داده می شود. در مورد دکمه های radio و چک باکس ها، برچسب معمول <>label است که با کنترل در ارتباط است. مقدار پیش فرض آن null است. |
| text | این گزینه تعیین می کند که آیا متن روی دکمه نمایش داده شود. اگر به عنوان false تعیین شده باشد، متن حذف می شود، اگر (و فقط اگر) گزینه ی icons حداقل یک آیکن را تعیین کنند. مقدار پیش فرض آن true است. |

قابلیت پیش فرض

مثال زیر قابلیت دکمه ی ویجت را توضیح می دهد، که هیچ پارامتری به متد **button()** منتقل نمی شود.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Buttons functionality</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>
  <script>
    $(function () {
      $("#button-1, #button-2, #button-3, #button-4").button();
      $("#button-5").buttonset();
    });
  </script>
</head>

```



```

<body>
  <button id="button-1">A button element</button>
  <input id="button-2" type="submit" value="A submit button">
  <a id="button-3" href="">An anchor</a>
  <input type="checkbox" id="button-4">
  <label for="button-4">Toggle</label>
  <br><br>
  <div id="button-5">
    <input type="checkbox" id="check1">
    <label for="check1">Left</label>
    <input type="checkbox" id="check2">
    <label for="check2">Middle</label>
    <input type="checkbox" id="check3">
    <label for="check3">Right</label>
  </div>
</body>
</html>

```

این مثال **markup** را که می توان برای دکمه ها استفاده کرد، توضیح می دهد: یک دکمه، یک ورودی **submit** و یک **anchor**.

استفاده از **text**، **icons** و **disabled**

مثال زیر استفاده از سه گزینه ی **text**، **icons** و **disabled** را در دکمه ی عملکرد **jQueryUI** توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Buttons functionality</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>
  <script>
    $(function () {
      $("#button-6").button({
        icons: {
          primary: "ui-icon-locked"
        },
        text: false
      });
      $("#button-7").button({
        disabled: true
      });
      $("#button-8").button({
        icons: {

```

```

        primary: "ui-icon-gear",
        secondary: "ui-icon-triangle-1-s"
    }
    });
    });
</script>
<head>
<body>
<button id="button-6">
    Button with icon only
</button>
<button id="button-7">
    Button Disabled
</button>
<button id="button-8">
    Button with two icons
</button>
</body>
</html>

```

متد `$(selector, context).button ("action", params)`

متد `button ("action", params)` می تواند عملی را روی دکمه اجرا کند، عملی مانند غیر فعال کردن دکمه. این عمل به عنوان یک رشته در اولین آرگیومنت تعیین می شود (به عنوان مثال `disable` برای غیرفعال کردن دکمه). اکشن هایی را که می توانند منتقل شوند، در جدول زیر چک کنید.

`$(selector, context).button ("action", params);`

جدول زیر اکشن های مختلفی را که توسط این متد استفاده می شوند، نشان می دهد:

| Action | Description |
|---------|--|
| destroy | این اکشن قابلیت یک دکمه را از یک عنصر کاملاً حذف می کند. عناصر به وضعیت pre-init خود بازمی گردند. این متد هیچ آرگیومنتی نمی پذیرد. |
| disable | این اکشن قابلیت دکمه ی مربوط به یک عنصر را کاملاً غیرفعال می سازد. این متد هیچ آرگیومنتی نمی پذیرد. |

| | |
|-----------------------------|--|
| enable | این اکشن قابلیت دکمه ی مربوط به یک عنصر را کاملا فعال می سازد. این متد هیچ آرگیومنتی نمی پذیرد. |
| option(optionName) | این اکشن مقدار گزینه ی تعیین شده در optionName را بازمی گرداند. optionName درواقع یک string است. |
| option | این اکشن یک آبجکت حاوی جفت های key/value را بازمی گرداند که گزینه های hash مربوط به دکمه ی موجود را نمایش می دهد. |
| option(optionName, value) | این اکشن مقدار گزینه ی دکمه، مربوط به optionName تعیین شده را تنظیم می کند. |
| option(options) | این اکشن یک یا چند گزینه را برای دکمه تعیین می کند. در اینجا options نقشه ای است از جفت های option-value که تنظیم می شوند. |
| refresh | این اکشن نمایش دکمه ها را تازه سازی می کند. این اکشن زمانی مفید است که دکمه ها به وسیله ی برنامه مدیریت می شوند و نمایش لزوما با وضعیت داخلی تطابق ندارد. این متد هیچ آرگیومنتی نمی پذیرد. |
| widget | این متد یک آبجکت jQuery را باز می گرداند که حاوی یک دکمه می باشد. این متد هیچ آرگیومنتی نمی پذیرد. |

مثال:

اکنون اجازه بدهید با استفاده از اکشن های جدول بالا، مثالی را بررسی کنیم. مثال زیر استفاده از متدهای

destroy() و **disable()** را توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
```

```

<meta charset="utf-8">
<title>jQuery UI Buttons functionality</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>
</script>
$(function () {
    $("#button-9").button({
        text: false,
        icons: {
            primary: "ui-icon-seek-start"
        }
    });
    $("#button-9").button('destroy');
    $("#button-10").button({
        icons: {
            primary: "ui-icon-seek-prev"
        }
    });
    $("#button-10").button('disable');
    $("#button-11").button({
        text: false,
        icons: {
            primary: "ui-icon-play"
        }
    });
});
</script>
<head>
<body>
<button id="button-9">
    I'm destroyed
</button>
<button id="button-10">
    I'm disabled
</button>
<button id="button-11">
    play
</button>
</body>
</html>

```

مدیریت رویداد روی دکمه ها

علاوه بر متد **button (options)** که در بخش های قبل مشاهده کردیم، **jQueryUI** متدهای رویداد ارائه می دهد که برای یک رویداد خاص آغاز می شوند. این متدهای رویداد در جدول زیر ارائه شده اند:

| Event Method | Description |
|-------------------|--|
| create(event, ui) | وقتی دکمه ایجاد شده باشد، این رویداد آغاز می شود. در اینجا رویداد از نوع Event است و ui از نوع Object. |

مثال:

مثال زیر استفاده از متد رویداد را برای قابلیت دکمه ی ویجت توضیح می دهد. این استفاده از رویداد **create** را توضیح می دهد:

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Buttons functionality</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>
.resultarea {
background: #cedc98;
border-top: 1px solid #000000;
border-bottom: 1px solid #000000;
color: #333333;
font-size: 14px;
}
</style>
<script>
$(function () {
$("#button-12").button({
create: function () {
$("#p#result").html($("#p#result")
.html() + "<b>created<b><br>");
}
});
});
</script>
<head>
<body>
<button id="button-12">
A button element
</button>
<p class="resultarea" id=result>
</p>
</body>
</html>

```

JqueryUI - Datepicker

Datepickers در **jQueryUI** به یوزرها اجازه می دهد تا تاریخ ها را به آسانی و به صورت بصری وارد کنند. شما می توانید فرمت تاریخ و زبان را تنظیم کنید، محدوده ی تاریخ انتخابی را تعیین کنید و آن را به آسانی به دکمه ها و گزینه های مسیریابی اضافه کنید.

jQueryUI متد **datepicker()** را ارائه می دهد که یک **datepicker** ایجاد می کند و ظاهر عناصر **HTML** را روی صفحه با افزودن گروه های جدید **CSS**، تغییر می دهد. عناصر **<div>**، **<input>** و **** را در یک مجموعه به یک کنترل **datepicker** تغییر می دهد.

به طور پیش فرض، برای عناصر **<input>**، وقتی فیلدهای متن مرتبط تمرکز را به دست می آورند، تقویم **datepicker** در یک پوشش کوچک باز می شود. برای یک تقویم داخلی **datepicker** را به یک عنصر **<div>** یا **** ضمیمه کنید.

ترکیب:

متد **datepicker()** به دو شکل زیر مورد استفاده قرار می گیرد

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

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

متد **\$(selector, context).datepicker (options)**

متد **datepicker (options)** اعلام می کند که یک عنصر **<nput>** (یا **<div>** یا ****)، بسته به اینکه شما نمایش تقویم خود را چگونه انتخاب کرده باشید) باید به عنوان یک **datepicker** تنظیم شود. پارامتر **options** آبجکتی است که رفتار و ظاهر عناصر **datepicker** را تنظیم می کند.

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

شما می توانید با استفاده از آجکت **javascript** یک یا چند گزینه را ارائه دهید. اگر بیشتر از یک گزینه برای ارائه وجود داشته باشد، آنها را با استفاده از کاما از یکدیگر مجزا می کنید، مانند ترکیب زیر:

```
$(selector, context).datepicker({option1: value1, option2: value2..... });
```

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

| Option | Description |
|------------|--|
| altField | این گزینه یک انتخابگر jQuery برای فیلدی تعیین می کند که با هر مجموعه تاریخی آپدیت می شود. گزینه ی altFormat می تواند برای تنظیم فرمت این مقدار استفاده شود. این گزینه برای تنظیم مقادیر تاریخ به یک عنصر ورودی مخفی شده، بسیار مفید است تا به سرور ارائه شود، در حالیکه یک فرمت user-friendly به یوزر نمایش می دهد. مقدار پیش فرض آن "" است. |
| altFormat | این گزینه زمانی استفاده می شود که یک گزینه ی altField تعیین شده باشد. در واقع فرمتی را برای نوشتن مقدار برای عنصر جایگزین تعیین می کند. مقدار پیش فرض آن "" است. |
| appendText | این گزینه یک مقدار String است که پس از عنصر <input> قرار می گیرد، که برای نمایش دستورالعمل ها به یوزر در نظر گرفته شده است. مقدار پیش فرض آن "" است. |
| autoSize | این گزینه وقتی روی true تنظیم شده باشد، عنصر را تغییر اندازه می دهد تا فرمت datepicker را طوری که با گزینه ی dateFormat تنظیم شده، در خود جای دهد. مقدار پیش فرض آن false است. |
| beforeShow | این گزینه یک تابع بازگشتی است که درست قبل از نمایش یک datepicker تحریک می شود، با عنصر <input> و نمونه ی datepicker که به عنوان یک پارامتر منتقل می |

| | |
|-----------------|---|
| | <p>شود. این عملکرد می تواند گزینه ای hash را بازگرداند که برای اصلاح datePicker استفاده می شوند. مقدار پیش فرض آن "" است.</p> |
| beforeShowDay | <p>این گزینه یک تابع بازگشتی است که تاریخ را به عنوان پارامتر می گیرد که برای هر روز، درست قبل از نمایش آن، فراخوانده می شود، که تاریخ به عنوان تنها پارامتر منتقل می شود. این عمل برای رد کردن بعضی رفتارهای پیش فرض از عناصر روز، استفاده می شود. این تابع باید یک ردیف دارای سه عنصر بازگرداند. مقدار پیش فرض آن null است.</p> |
| buttonImage | <p>این گزینه مسیر را به یک تصویر تعیین می کند تا روی دکمه ی نمایش داده شود که با تنظیم گزینه ی showOn برای یک یا هر دو دکمه، فعال می شود. اگر buttonText نیز ارائه شود، دکمه ی متن، موجودیت alt می شود. مقدار پیش فرض آن "" می باشد.</p> |
| buttonImageOnly | <p>این گزینه اگر روی true تنظیم شده باشد، تعیین می کند که تصویر تعیین شده توسط buttonImage باید به تنهایی ظاهر شود (نه روی یک دکمه). گزینه ی showOn باید روی یکی از دکمه ها یا هر دوی آنها برای ظاهر شدن تصویر، تنظیم شود. مقدار پیش فرض آن false است.</p> |
| buttonText | <p>این گزینه عنوانی برای دکمه تعیین می کند که به وسیله ی تنظیم گزینه ی showOn برای یک یا هر دو دکمه، فعال می شود. مقدار پیش فرض آن "... " است.</p> |
| calculateWeek | <p>این گزینه یک تابع سفارشی برای محاسبه و بازگردانی عدد هفته است برای تاریخی که به عنوان یک پارامتر مجزا منتقل می شود. اجرای پیش فرض، اجرایی است که به وسیله ی تابع \$.datepicker.iso8601Week() ارائه می شود.</p> |

| | |
|----------------|---|
| changeMonth | این گزینه اگر روی true تنظیم شده باشد، یک منو برای ماه (month dropdown) نمایش داده می شود که به یوزر اجازه می دهد تا مستقیماً و بدون استفاده از دکمه های فلش، ماه را تغییر دهد. مقدار پیش فرض آن false است. |
| changeYear | این گزینه اگر روی true تنظیم شده باشد، یک منوی رو به پایین برای سال نمایش داده می شود، می توان بدون استفاده از دکمه های فلش آن را تغییر داد. گزینه ی yearRange می تواند برای بررسی این مسئله که آیا سال ها در دسترس مجموعه هستند، استفاده می شود. مقدار پیش فرض آن false است. |
| closeText | این گزینه متن را برای جایگزینی عنوان پیش فرض Done برای بستن دکمه، تعیین می کند. این گزینه زمانی استفاده می شود که پدل دکمه از طریق گزینه ی showButtonPanel نمایش داده می شود. مقدار پیش فرض آن done است. |
| constrainInput | این گزینه اگر روی true تنظیم شده باشد، متن ورودی به عنصر <input> محدود به کاراکترهایی می شود که به وسیله ی گزینه ی dateFormat مجاز هستند. مقدار پیش فرض آن true است. |
| currentText | این گزینه تعیین می کند که متن جایگزین عنوان پیش فرض Today برای دکمه ی فعلی شود. اگر پدل دکمه از طریق گزینه ی showButtonPanel نمایش داده شود، این گزینه استفاده می شود. مقدار پیش فرض آن Today است. |
| dateFormat | این گزینه فرمت تاریخ مورد استفاده را تعیین می کند. مقدار پیش فرض آن mm/dd/yy است. |
| dayNames | عنصر عنصر این گزینه یک ردیف دارای 7 عنصر است که نام کامل روزها را ارائه می دهد که عنصر 0 نشان دهنده ی یکشنبه است. این گزینه برای متمرکز کردن کنترل استفاده |

| | |
|---------------|---|
| | می شود. مقدار پیش فرض آن ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"] است. |
| dayNamesMin | این گزینه یک ردیف دارای 7 عنصر است که نام کوتاه شده ی روزها را نمایش می دهد که عنصر 0 نشان دهنده ی یکشنبه است و به عنوان تیترهای ستون استفاده می شود. برای متمرکز کردن کنترل استفاده می شود. مقدار پیش فرض آن ["Su", "Mo", "Tu", "We", "Th", "Fr", "Sa"] است. |
| dayNamesShort | این گزینه یک ردیف دارای 7 عنصر ارائه می دهد که نام کوتاه شده ی روزها را نمایش می دهد با عنصر 0 که نشان دهنده ی روز یکشنبه است. ان گزینه برای متمرکز کردن کنترل استفاده می شود. مقدار پیش فرض آن ["Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"] است. |
| defaultDate | این گزینه تاریخ اولیه را برای کنترل تنظیم می کند، اگر عنصر <input type="text"/> هیچ مقداری نداشته باشد، مقدار پیش فرض را برای today نادیده می گیرد. این گزینه می تواند یک نمونه ی Date باشد، تعداد روزها از امروز یا یک رشته که یک تاریخ کامل یا نسبی را تعیین می کند. مقدار پیش فرض آن null است. |
| duration | این گزینه سرعت انیمیشنی را تعیین می کند که datepicker را ظاهر می سازد. می تواند یکی از گزینه های slow, normal, یا fast باشد، یا مدت زمانی که برای انیمیشن طول می کشد (به هزارم ثانیه). مقدار پیش فرض آن normal است. |
| firstDay | این گزینه تعیین می کند که کدام روز به عنوان اولین روز هفته نشان داده می شود و در سمت چپ جدول نمایش داده خواهد شد. مقدار پیش فرض آن 0 است. |

| | |
|------------------|--|
| gotoCurrent | وقتی این گزینه روی true تنظیم شده باشد، روز موجود در لینک بر روی تاریخ انتخاب شده تنظیم می شود، پیش فرض مربوط به امروز را نادیده می گیرد. مقدار پیش فرض آن false است. |
| hideIfNoPrevNext | این گزینه اگر روی true تنظیم شده باشد، لینک های next و previous را پنهان می کند، وقتی مناسب نیستند، همانطور که با تنظیمات گزینه های minDate و maxDate تعیین شده است. مقدار پیش فرض آن false است. |
| isRTL | وقتی این گزینه روی true تنظیم شده باشد، متمرکز سازی ها را به عنوان زبان از راست به چپ مشخص می کند. مقدار پیش فرض آن false است. |
| maxDate | این گزینه حداکثر تاریخ قابل انتخابی را برای کنترل تعیین می کند. این می تواند یک نمونه ی Date، تعداد روزها از امروز یا یک رشته که یک تاریخ کامل یا نسبی را تعیین می کند، باشد. مقدار پیش فرض آن null است. |
| minDate | این گزینه حداقل تاریخ قابل انتخابی را برای کنترل تعیین می کند. این می تواند یک نمونه ی Date، تعداد روزها از امروز یا یک رشته که یک تاریخ کامل یا نسبی را تعیین می کند، باشد. مقدار پیش فرض آن null است. |
| monthNames | این گزینه یک ردیف 12 عنصری است که نام کامل ماه ها را ارائه می دهد که عنصر 0 نشان دهنده ی ماه ژانویه می باشد. برای متمرکز کردن کنترل استفاده می شود. مقدار پیش فرض آن ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"] است. |
| monthNamesShort | این گزینه یک ردیف 12 عنصری است که نام کوتاه ماه ها را ارائه می دهد که عنصر 0 نشان دهنده ی ماه ژانویه می باشد. برای متمرکز کردن کنترل استفاده می شود. مقدار |

| | |
|------------------------|---|
| | پیش فرض آن ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"] است. |
| navigationAsDateFormat | این گزینه اگر روی true تنظیم شده باشد، مسیریابی لینک ها برای nextText, prevText و currentText از طریق تابع \$.datepicker.formatDate() قبل از نمایش، منتقل می شوند. این گزینه به فرمت های تاریخ اجازه می دهد تا برای گزینه هایی ارائه شوند که با مقادیر مربوط جایگزین می شوند. مقدار پیش فرض آن false است. |
| nextText | این گزینه متنی را تعیین می کند که برای مسیریابی لینک ماه بعدی، عنوان پیش فرض را جایگزین Next می کند. ThemeRoller این متن را جایگزین یک آیکن می کند. مقدار پیش فرض آن Next است. |
| numberOfMonths | این گزینه تعداد ماه هایی را تعیین می کند که در datepicker نمایش داده می شوند. مقدار پیش فرض آن 1 است. |
| onChangeMonthYear | این گزینه یک callback است که وقتی datepicker وارد یک ماه یا سال جدید می شود، با سال انتخاب شده، فراخوانده می شود و ماه و نمونه ی datepicker به عنوان پارامترها منتقل می شوند، و کانتکست تابع به عنصر فیلد ورودی تنظیم می شود. مقدار پیش فرض آن null است. |
| onClose | این گزینه یک callback است که وقتی datepicker بسته می شود، فراخوانده می شود، تاریخ انتخاب شده را به عنوان متن انتقال می دهد و نمونه ی datepicker و کانتکست تابع بر روی عنصر فیلد ورودی تنظیم شده اند. مقدار پیش فرض آن null است. |

| | |
|-------------------|--|
| onSelect | این گزینه یک callback است که وقتی یک تاریخ انتخاب می شود، فراخوانده می شود و ماه و نمونه ی datePicker به عنوان پارامترها منتقل می شوند، و کانتکست تابع به عنصر فیلد ورودی تنظیم می شود. مقدار پیش فرض آن null است. |
| prevText | این گزینه متن را برای جایگزینی عنوان پیش فرض Prev برای مسیریابی لینک ماه قبل، تعیین می کند. (دقت داشته باشید که ThemeRoller این متن را با یک آیکن جایگزین می کند.). مقدار پیش فرض آن PrevdefaultDatedayNamesMin است. |
| selectOtherMonths | این گزینه اگر روی true تنظیم شده باشد، روزهای نمایش داده شده ی قبل یا بعد از ماه ها، قابل انتخاب هستند. چنین روزهایی نمایش داده نمی شوند، مگر اینکه گزینه ی showOtherMonths روی true باشد. مقدار پیش فرض آن false است. |
| shortYearCutoff | اگر عدد مربوط به این گزینه مقداری بین سال های 0 و 99 باشد، قبل از آنها هر عدد 2 رقمی مربوط به قرن قبل در نظر گرفته می شود. اگر این گزینه یک string باشد، مقدار تحت تبدیل عددی قرار می گیرد و به سال جاری اضافه می شود. مقدار پیش فرض 10+ است که 10 سال از سال جاری را نشان می دهد. |
| showAnim | این گزینه مجموعه ای از نام انیمیشنی را تعیین می کند که برای نمایش دادن و یا پنهان کردن datePicker استفاده می شوند. اگر تعیین شده باشد، باید یکی از انیمیشن های show (the default), fadeIn, slideDown یا هر انیمیشن show/hide مربوط به jQuery UI باشد. مقدار پیش فرض آن show است. |
| showButtonPanel | اگر این گزینه روی true تنظیم شده باشد، یک پنل دکمه در پایین datePicker نمایش داده می شود که حاوی دکمه های فعلی و دکمه های بسته شده می باشد. عنوان این دکمه ها از طریق گزینه های currentText و closeText نمایش داده می شوند. مقدار پیش فرض آن false است. |

| | |
|--------------------|--|
| showCurrentAtPos | این گزینه ایندکس 0 محور را مشخص می کند، که از بالا سمت چپ، آغاز می شود، جایی که ماه جاری باید در بین چند ماه نمایش داده شود. مقدار پیش فرض آن 0 است. |
| showMonthAfterYear | این گزینه اگر روی true تنظیم شده باشد، موقعیت های ماه و سال را که در عنوان datepicker معکوس شده اند، مشخص می کند. مقدار پیش فرض آن false است. |
| showOn | این گزینه زمان ظاهر شدن datepicker را تعیین می کند. مقادیر ممکن عبارتند از focus, button یا هر دو. مقدار پیش فرض آن focus است. |
| showOptions | این گزینه یک hash را ارائه می دهد که وقتی یک انیمیشن jQuery UI برای گزینه ی showAnim تعیین شده باشد، به انیمیشن منتقل می شود. مقدار پیش فرض آن {} است. |
| showOtherMonths | این گزینه اگر روی true تنظیم شده باشد، تاریخ های قبل یا بعد از اولین و آخرین روزهای ماه جاری نمایش داده می شوند. این تاریخ ها قابل انتخاب نیستند، مگر اینکه گزینه ی selectOtherMonths روی true تنظیم شده باشد. مقدار پیش فرض آن false است. |
| showWeek | این گزینه اگر روی true تنظیم شده باشد، عدد هفته در یک ستون در سمت چپ ماه نمایش داده می شود. گزینه ی calculateWeek می تواند برای تغییر روشی استفاده شود که در آن، این مقدار تعیین شده است. مقدار پیش فرض آن false است. |
| stepMonths | این گزینه تعداد ماه هایی را تعیین می کند که وقتی یکی از کنترل های مسیریابی ماه کلیک می شود، جابه جا می شوند. مقدار پیش فرض آن 1 است. |
| weekHeader | این گزینه متنی را برای نمایش برای عدد ستون هفته تعیین می کند، مقدار پیش فرض WK را، وقتی showWeek روی true تنظیم است، نادیده می گیرد. مقدار پیش فرض آن WK است. |

| | |
|------------|--|
| yearRange | این گزینه محدودیت هایی را تعیین می کند که در آنها سال ها در منوهای رو به پایین، به شکل from:to ، وقتی changeYear روی true تنظیم شده باشد، نمایش داده می شوند. مقادیر می توانند کامل یا نسبی باشند (به عنوان مثال 2005:2005+2، برای 2005 تا 2 ال از هم اکنون.). پیشوند c می تواند برای ساخت مقادیر نسبی از سال انتخاب شده، استفاده شود، به جای سال جاری (مثال 3c-2:c) مقدار پیش فرض آن c-10:c+10 است. |
| yearSuffix | این گزینه متن های اضافه را بعد از سال در تیتیر datepicker نمایش می دهد. مقدار پیش فرض آن "" است. |

بخش بعدی قسمت هایی از عملکرد و قابلیت **datepicker** را نشان می دهد.

قابلیت پیش فرض

مثال زیر قابلیت **datepicker** را توضیح می دهد که هیچ پارامتری به متد **datepicker()** انتقال نمی دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Datepicker functionality</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>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#datepicker-1").datepicker();
    });
  </script>
</head>
<body>
  <!-- HTML -->
  <p>Enter Date: <input type="text" id="datepicker-1"></p>
</body>
</html>
```

مثال زیر قابلیت یک **datepicker** داخلی را توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Datepicker functionality</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>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#datepicker-2").datepicker();
    });
  </script>
</head>
<body>
  <!-- HTML -->
  Enter Date: <div id="datepicker-2"></div>
</body>
</html>
```

در مثال بالا از عنصر **<div>** برای دریافت **datepicker** داخلی استفاده کردیم.

استفاده از **appendText**، **dateFormat**، **altField** و **altFormat**

مثال زیر استفاده از گزینه های مهم **altField** (c) **dateFormat** (b) **appendText** (a) و **altFormat** (d)

را در عملکرد **datepicker** از **jQueryUI** نشان می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Datepicker functionality</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>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#datepicker-3").datepicker({
        appendText: "(yy-mm-dd)",
        dateFormat: "yy-mm-dd",
        altField: "#datepicker-4",

```



```

        altFormat: "DD, d MM, yy"
    });
});
</script>
</head>
<body>
    <!-- HTML -->
    <p>Enter Date: <input type="text" id="datepicker-3"></p>
    <p>Alternate Date: <input type="text" id="datepicker-4"></p>
</body>
</html>

```

در مثال بالا فرمت تاریخ برای اولین ورودی به شکل **yy-mm-dd** تنظیم شده است. اگر شما از **datepicker** تاریخی را انتخاب کنید، همان تاریخ در دومین فیلد ورودی که فرمت تاریخ آن به شکل است **DD, d MM, yy**، منعکس می شود.

استفاده از **beforeShowDay**

مثال زیر استفاده از گزینه **beforeShowDay** را در عملکرد **datepicker** از **JqueryUI** نشان می دهد.

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>jQuery UI Datepicker functionality</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>
    <!-- Javascript -->
    <script>
        $(function () {
            $("#datepicker-5").datepicker({
                beforeShowDay: function (date) {
                    var dayOfWeek = date.getDay();
                    // 0 : Sunday, 1 : Monday, ...
                    if (dayOfWeek == 0 || dayOfWeek == 6) return [false];
                    else return [true];
                }
            });
        });
    </script>
</head>
<body>
    <!-- HTML -->
    <p>Enter Date: <input type="text" id="datepicker-5"></p>
</body>
</html>

```

در مثال بالا شنبه و یک شنبه غیرفعال هستند.

استفاده از `buttonImage`، `showOn` و `buttonImageOnly`

مثال زیر استفاده از سه گزینه ی `buttonImage` (b)، `showOn` (a) و `buttonImageOnly` (c) را در عملکرد `datepicker` از `JqueryUI` نشان می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Datepicker functionality</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>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#datepicker-6").datepicker({
        showOn: "button",
        buttonImage: "/jqueryui/images/calendar-icon.png",
        buttonImageOnly: true
      });
    });
  </script>
</head>
<body>
  <!-- HTML -->
  <p>Enter Date: <input type="text" id="datepicker-6"></p>
</body>
</html>
```

در مثال بالا آیکنی نمایش داده می شود که با کلیک کردن روی آن `datepicker` باز می شود.

استفاده از `defaultDate`، `dayNamesMin` و `duration`

مثال زیر استفاده از سه گزینه ی مهم `dayNamesMin` (b)، `dayNamesMin` (a) و `duration` (c) را در عملکرد `datepicker` از `JqueryUI` نشان می دهد.

```
<!doctype html>
```

```

<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Datepicker functionality</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>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#datepicker-7").datepicker({
        defaultDate: +9,
        dayNamesMin: ["So", "Mo", "Di", "Mi", "Do", "Fr", "Sa"],
        duration: "slow"
      });
    });
  </script>
</head>
<body>
  <!-- HTML -->
  <p>Enter Date: <input type="text" id="datepicker-7"></p>
</body>
</html>

```

در مثال بالا نام های روزها با استفاده از **dayNamesMin** تغییر کرده اند. همچنین مشاهده می کنید که تاریخ پیش فرض تنظیم شده است.

استفاده از **prevText**، **nextText**، **showOtherMonths** و **selectOtherMonths**

مثال زیر استفاده از گزینه های مهم **prevText (b)**، **nextText (c)**، **showOtherMonths (d)** و **selectOtherMonths** را در **datepicker** از **jQueryUI** نشان می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Datepicker functionality</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>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#datepicker-8").datepicker({
        prevText: "click for previous months",
        nextText: "click for next months",
        showOtherMonths: true,

```

```

        selectOtherMonths: false
    });
    $("#datepicker-9").datepicker({
        prevText: "click for previous months",
        nextText: "click for next months",
        showOtherMonths: true,
        selectOtherMonths: true
    });
});
</script>
</head>
<body>
<!-- HTML -->
<p>Enter Start Date: <input type="text" id="datepicker-8"></p>
<p>Enter End Date: <input type="text" id="datepicker-9"></p>
</body>
</html>

```

در مثال بالا لینک های **prev** و **next** دارای عنوان می باشند. اگر روی عنصر کلیک کنید، **datepicker** باز می شود. اکنون در اولین **datepicker**، تاریخ های ماه های دیگر غیرفعال هستند، چرا که **selectOtherMonths** روی **false** تنظیم شده است. در دومین **datepicker** برای دومین ورودی، **selectOtherMonths** روی **true** تنظیم شده است.

استفاده از **changeYear**، **changeMonth**، **numberOfMonth**

مثال زیر استفاده از گزینه های مهم **changeYear** (b)، **changeMonth** (a) و **numberOfMonths** (c) را در عملکرد **datepicker** از **JqueryUI** نشان می دهد.

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Datepicker functionality</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>
<!-- Javascript -->
<script>
$(function () {
    $("#datepicker-10").datepicker({
        changeMonth: true,
        changeYear: true,
        numberOfMonths: [2, 2]
    });
});
</script>

```

```

</head>
<body>
  <!-- HTML -->
  <p>Enter Date: <input type="text" id="datepicker-10"></p>
</body>
</html>

```

در مثال بالا منوهای رو به پایین را برای فیلدهای **Month** و **Year** مشاهده کردید. در اینجا ما در حال نمایش ماه در ساختار یک ردیف از [2,2] هستیم.

استفاده از **showWeek**، **yearSuffix** و **showAnim**

مثال زیر استفاده از گزینه های مهم **yearSuffix** (b)، **showWeek** (a) و **showAnim** (c) را عملکرد **datepicker** از **JqueryUI** نشان می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Datepicker functionality</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>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#datepicker-11").datepicker({
        showWeek: true,
        yearSuffix: "-CE",
        showAnim: "slide"
      });
    });
  </script>
</head>
<body>
  <!-- HTML -->
  <p>Enter Date: <input type="text" id="datepicker-11"></p>
</body>
</html>

```

در مثال بالا مشاهده می کنید که اعداد مربوط به هفته در سمت چپ نمایش داده می شوند، چرا که **showWeek** روی **true** تنظیم شده است. سال داری پسوند **"-CE"** می باشد.

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

متد `datepicker (action, params)` می تواند اکشنی را روی تقویم اجرا کند، از جمله انتخاب یک تاریخ جید. `action` به عنوان یک رشته در اولین آرگيومنت تعیین شده و به طور انتخابی یک یا چند `param` می توانند براساس اکشن داده شده، ارائه شوند.

اساسا اکشن ها متدهای **jQuery** هستند که می توانیم به فرم رشته از آنها استفاده کنیم.

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

جدول زیر اکشن هایی را برای این متد لیست می کند:

| Action | Description |
|--|--|
| <code>destroy()</code> | این گزینه قابلیت <code>datepicker</code> را کاملاً حذف می کند. این اکشن عنصر را به حالت <code>pre-init</code> خود بازمی گرداند. این متد هیچ آرگيومنتی نمی پذیرد. |
| <code>dialog(date [, onSelect] [, settings] [, pos])</code> | این اکشن <code>datepicker</code> را در یک دیاوگ باکش <code>jQuery UI</code> نمایش می دهد. |
| <code>getDate()</code> | این اکشن <code>Date</code> را مطابق تاریخ انتخاب شده بازمی گرداند. این متد هیچ آرگيومنتی نمی پذیرد. |
| <code>hide()</code> | این اکشن <code>date picker</code> از قبل باز شده را می بندد. این متد هیچ آرگيومنتی نمی پذیرد. |
| <code>isDisabled()</code> | این اکشن قابلیت <code>date picker</code> را بررسی می کند، اگر غیرفعال شده باشد. این متد هیچ آرگيومنتی نمی پذیرد. |
| <code>option(optionName)</code> | این اکشن مقدار مربوط به <code>optionName</code> تعیین شده را بازمی گرداند. |

| | |
|-----------------------------|---|
| option() | این اکشن یک آبجکت حاوی جفت های key/value را دریافت می کند که گزینه های hash مربوط به datepicker فعلی را نشان می دهند. این متد هیچ آرگیومنتی نمی پذیرد. |
| option(optionName, value) | این گزینه مقدار مربوط به گزینه ی datepicker را که با optionName تعیین شده در ارتباط است، تنظیم می کند. |
| option(options) | این اکشن یک یا چند گزینه را برای datepicker تنظیم می کند. |
| refresh() | این اکشن، پس از ساختن چند تعدیل خارجی، date picker را مجددا طراحی می کند. این متد هیچ برهانی را نمی پذیرد. |
| setDate(date) | این اکشن تاریخ تعیین شده را به عنوان تاریخ کنونی از datepicker تنظیم می کند. |
| show() | این اکشن datepicker را باز می کند. اگر datepicker به یک ورودی ضمیمه شده باشد، ورودی باید برای datepicker قابل مشاهده باشد. این متد هیچ برهانی را نمی پذیرد. |
| widget() | این اکشن یک آبجکت jQuery که حاوی datepicker می باشد، بازمی گرداند. |

مثال زیر استفاده از برخی اکشن های لیست شده در جدول بالا را نشان می دهد.

استفاده از اکشن setDate()

اکنون اجازه بدهید با استفاده از اکشن های جدول بالا، مثالی را مشاهده کنیم. مثال زیر استفاده از اکشن های **setDate** را توضیح می دهد.

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

```

<title>jQuery UI Datepicker functionality</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>
<!-- Javascript -->
<script>
$(function () {
    $("#datepicker-12").datepicker();
    $("#datepicker-12").datepicker("setDate", "10w+1");
});
</script>
</head>
<body>
<!-- HTML -->
<p>Enter Date: <input type="text" id="datepicker-12"></p>
</body>
</html>

```

استفاده از اکشن show()

مثال زیر استفاده از اکشن show را توضیح می دهد:

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Datepicker functionality</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>
<!-- Javascript -->
<script>
$(function () {
    $("#datepicker-13").datepicker();
    $("#datepicker-13").datepicker("show");
});
</script>
</head>
<body>
<!-- HTML -->
<p>Enter Date: <input type="text" id="datepicker-13"></p>
</body>
</html>

```

JqueryUI - Dialog

دیالوگ باکس ها یکی از روش های خوب برای نمایش اطلاعات روی یک صفحه ی HTML هستند. یک دیالوگ باکس پنجره را با یک تیترو یک محدوده ی محتوا، معلق می کند. این پنجره را می توان جابه جا کرد، تغییر اندازه داد، و البته با استفاده از آیکون X پیش فرض آن را بست.

jQueryUI متد dialog() را ارائه می دهد که کد HTML نوشته شده روی صفحه را به کد HTML برای نمایش در یک دیالوگ باکس تبدیل می کند.

متد dialog() به دو فرم زیر استفاده می شود

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

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

متد `$(selector, context).dialog (options)`

متد `dialog (options)` اعلام می کند که یک عنصر HTML می تواند به شکل یک دیالوگ باکس نمایش داده شود. پارامتر `options` آبجکتی است که ظاهر و رفتار آن پنجره را تعیین می کند.

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

شما می توانید با استفاده از آبجکت Javascript یک یا چند گزینه را همزمان نمایش دهید. اگر بیشتر از یک گزینه برای نمایش وجود داشته باشد، آنها را با استفاده از کاما مجزا می کنید، مانند ترکیب زیر:

`$(selector, context).dialog({option1: value1, option2: value2..... });`

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

| Option | Description |
|----------|---|
| appendTo | اگر این گزینه روی false تنظیم شده باشد، از افزوده شدن گروه ui-draggable به لیست عناصر انتخاب شده ی DOM جلوگیری می کند. مقدار پیش فرض آن true است. |

| | |
|---------------|---|
| autoOpen | این گزینه مگر اینکه بر روی false تنظیم شده باشد، دیالوگ باکس بر ایجاد آن باز می شود. وقتی روی false تنظیم شده باشد، دیالوگ باکس با فراخوانی dialog('open') باز می شود. مقدار پیش فرض آن true است. |
| buttons | این گزینه دکمه ها را به دیالوگ باکس اضافه می کند. اینها به عنوان آبجکت هایی لیست شده اند و هر پراپرتی متن روی دکمه است. مقدار آن یک تابع پاسخ است که وقتی یوزر روی دکمه کلیک می کند، فراخوانده می شود. مقدار پیش فرض آن {} است. |
| closeOnEscape | مگر اینکه گزینه روی false تنظیم شده باشد، وقتی یوزر روی دکمه ی Escape کلیک کند دیالوگ باکس بسته خواهد شد. مقدار پیش فرض آن true است. |
| closeText | این گزینه حاوی متنی است برای جایگزینی Close برای دکمه ی بستن. مقدار پیش فرض آن "close" است. |
| dialogClass | اگر این گزینه روی false تنظیم شده باشد، از افزوده شدن گروه -ui draggable به لیست عناصر انتخاب شده ی DOM جلوگیری می کند. مقدار پیش فرض آن "" است. |
| draggable | مگر اینکه این گزینه روی false تنظیم شده باشد، دیالوگ باکس با کلیک کزدن و درگ کردن نوار عنوان، draggable خواهد بود. مقدار پیش فرض آن true است. |
| height | این گزینه طول دیالوگ باکس را تنظیم می کند. مقدار پیش فرض آن auto است. |

| | |
|-----------|---|
| hide | این گزینه برای تنظیم اثر بسته شدن دیاپالوگ باکس استفاده می شود. مقدار پیش فرض آن null است. |
| maxHeight | این گزینه حداکثر طول گزینه را به پیکسل، تنظیم می کند که در آن دیاپالوگ باکس قابل تغییر اندازه می باشد. مقدار پیش فرض آن false است. |
| maxWidth | این گزینه بیشترین عرض را که در آن دیاپالوگ باکس می تواند تغییر اندازه پیدا کند، به پیکسل تنظیم می کند. مقدار پیش فرض آن false است. |
| minHeight | این گزینه حداقل طول را به پیکسل، که طبق آن دیاپالوگ باکس می تواند تغییر اندازه پیدا کند، تنظیم می کند. مقدار پیش فرض آن 150 است. |
| minWidth | این گزینه حداقل عرض را به پیکسل تنظیم می کند که طبق آن دیاپالوگ باکس می تواند تغییر اندازه پیدا کند. مقدار پیش فرض آن 150 است. |
| modal | اگر این گزینه روی true تنظیم شده باشد، دیاپالوگ باکس دارای رفتار کیفیتی خواهد بود؛ آیتم های دیگر روی صفحه غیرفعال خواهند شد، به عنوان مثال با هم تعامل نخواهند داشت. دیاپالوگ های مودال ایجاد یک همپوشانی در زیر دیاپالوگ اما بالای عناصر پیچ دیگر، دارند. مقدار پیش فرض آن false است. |
| position | این گزینه موقعیت اولیه ی دیاپالوگ باکس را تعیین می کند. می تواند یکی از موقعیت های از پیش تعریف شده باشد: center (the default), left, right, top, bottom. همچنین می تواند یک ردیف دارای دو عنصر با مقادیر left و top یا موقعیت های متن مانند ['right', 'top'] باشد. مقدار پیش فرض آن { my: "center", at: "center", of: window } is است. |

| | |
|-----------|---|
| resizable | اگر اینکه روی false تنظیم شده باشد، دیالوگ باکس در همه ی جهت ها قابل تغییر اندازه خواهد بود. مقدار پیش فرض آن true است. |
| show | این گزینه تاثیری است که هنگامی استفاده می شود که دیالوگ باکس باز شده است. مقدار پیش فرض آن null است. |
| title | این گزینه متن مربوط به نوار عنوان را در دیالوگ باکس تعیین می کند. مقدار پیش فرض آن null است. |
| width | این گزینه عرض دیالوگ باکس را به پیکسل تعیین می کند. مقدار پیش فرض آن 300 است. |

بخش بعد به شما چند مثال از قابلیت دیالوگ ارائه خواهد کرد.

قابلیت پیش فرض

مثال زیر استفاده از قابلیت دیالوگ را، بدون انتقال هیچ پارامتری به متد **dialog()**، توضیح می دهد:

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Dialog functionality</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>
<!-- CSS -->
<style>
.ui-widget-header, .ui-state-default, ui-button {
background: red;
border: 1px solid #b9cd6d;
color: #FFFFFF;
font-weight: bold;
}
</style>
<!-- Javascript -->
<script>
$(function () {
$("#dialog-1").dialog({
```

```

        autoOpen: false,
    });
    $("#opener").click(function () {
        $("#dialog-1").dialog("open");
    });
});
</script>
</head>
<body>
<!-- HTML -->
<div id="dialog-1" title="Dialog Title goes here...">This my first jQuery UI Dialog!</div>
<button id="opener">Open Dialog</button>
</body>
</html>

```

استفاده از buttons، title و position

مثال زیر استفاده از گزینه های مهم **position** و **buttons, title** را در دیاالوگ ویجت از **jQueryUI** توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Dialog functionality</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>
<!-- CSS -->
<style>
.ui-widget-header, .ui-state-default, ui-button {
background: #b9cd6d;
border: 1px solid #b9cd6d;
color: #FFFFFF;
font-weight: bold;
}
</style>
<!-- Javascript -->
<script>
$(function () {
    $("#dialog-2").dialog({
        autoOpen: false,
        buttons: {
            OK: function () { $(this).dialog("close"); }
        },
        title: "Success",
        position: {
            my: "left center",
            at: "left center"
        }
    });
    $("#opener-2").click(function () {
        $("#dialog-2").dialog("open");
    });
});

```

```

});
});
</script>
</head>
<body>
<!-- HTML -->
<div id="dialog-2" title="Dialog Title goes here...">This my first jQuery UI Dialog!</div>
<button id="opener-2">Open Dialog</button>
</body>
</html>

```

استفاده از show، hide و height

مثال زیر استفاده از گزینه های مهم **hide**، **show** و **height** را در دیالوگ ویجت **jQueryUI** توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Dialog functionality</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>
<!-- CSS -->
<style>
.ui-widget-header, .ui-state-default, ui-button {
background: #b9cd6d;
border: 1px solid #b9cd6d;
color: #FFFFFF;
font-weight: bold;
}
</style>
<!-- Javascript -->
<script>
$(function () {
$("#dialog-3").dialog({
autoOpen: false,
hide: "puff",
show: "slide",
height: 200
});
$("#opener-3").click(function () {
$("#dialog-3").dialog("open");
});
});
</script>
</head>
<body>
<!-- HTML -->

```

```

<div id="dialog-3" title="Dialog Title goes here...">This my first jQuery UI Dialog!</div>
<button id="opener-3">Open Dialog</button>
</body>
</html>

```

استفاده از گزینه ی کیفیتی (modal)

مثال زیر استفاده از گزینه های **position** و **buttons, title** را در دیالوگ ویجت **jQueryUI** توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Dialog functionality</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>
  <!-- CSS -->
  <style>
    .ui-widget-header, .ui-state-default, ui-button {
      background: #b9cd6d;
      border: 1px solid #b9cd6d;
      color: #FFFFFF;
      font-weight: bold;
    }
  </style>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#dialog-4").dialog({
        autoOpen: false,
        modal: true,
        buttons: {
          OK: function () { $(this).dialog("close"); }
        },
      });
      $("#opener-4").click(function () {
        $("#dialog-4").dialog("open");
      });
    });
  </script>
</head>
<body>
  <!-- HTML -->
  <div id="dialog-4" title="Dialog Title goes here...">This my first jQuery UI Dialog!</div>
  <button id="opener-4">Open Dialog</button>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
    ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat.
  </p>

```

```

<label for="textbox">Enter Name: </label>
<input type="text">
</body>
</html>

```

متد \$(selector, context).dialog ("action", [params])

متد **dialog (action, params)** می تواند اکشنی را روی دیالوگ باکس اجرا کند، مانند بستن دیالوگ باکس. **action** به عنوان یک رشته در اولین آرگیومننت تعیین می شود و به طور انتخابی، یک یا چند **params**، بر اساس اکشن داده شده، ارائه می شوند.

اساسا در اینجا اکشن ها متدهای **jQuery** هستند که می توانیم به فرم رشته مورد استفاده قرار دهیم.

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

جدول زیر اکشن هایی را ارائه می دهد که با این متد مورد استفاده قرار می گیرند:

| Action | Description |
|-------------|--|
| close() | این اکشن دیالوگ باکس را می بندد. این متد هیچ برهانی را نمی پذیرد. |
| destroy() | این اکشن دیالوگ باکس را کاملا حذف می کند و عنصر را به حالت pre-init خود بازمی گرداند. این متد هیچ برهانی را نمی پذیرد. |
| isOpen() | اگر دیالوگ باکس باز باشد، این اکشن چک می شود. این متد هیچ برهانی را نمی پذیرد. |
| moveToTop() | این اکشن دیالوگ باکس های هماهنگ را پیش زمینه قرار می دهد. این متد هیچ برهانی را نمی پذیرد. |
| open() | این اکشن دیالوگ باکس را باز می کند. این متد هیچ برهانی را نمی پذیرد. |

| | |
|--|--|
| option(optionName) | این اکشن مقدار مرتبط با optionName تعیین شده را دریافت می کند. در اینجا optionName نام گزینه ای برای دریافت می باشد. |
| option() | این اکشن یک آبجکت حاوی جفت های key/value را دریافت می کند که گزینه های hash مربوط به دیالوگ را نمایش می دهد. این متد هیچ برهانی نمی پذیرد. |
| option(optionName, value) | این گزینه مقدار گزینه ی دیالوگ را در ارتباط با optionName تنظیم شده، تعیین شده، تنظیم می کند. |
| option(options) | این اکشن یک یا چند گزینه را برای دیالوگ تنظیم می کند. |
| widget() | This action returns the dialog box's widget element; the element annotated with the ui-dialog class name. This method does not accept any arguments. |

مثال:

اکنون اجازه بدهید مثالی را در رابطه با اکشن های جدول بالا بررسی کنیم. مثال زیر استفاده از متدهای **close()** و **isOpen()** را توضیح می دهد:

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Dialog functionality</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>
<!-- CSS -->
<style>
.ui-widget-header, .ui-state-default, ui-button {
background: #b9cd6d;
border: 1px solid #b9cd6d;
color: #FFFFFF;
font-weight: bold;

```

```

}
</style>
<!-- Javascript -->
<script type="text/javascript">
$(function () {
    $("#toggle").click(function () {

        ($("#dialog-5").dialog("isOpen") == false) ? ($("#dialog-5").dialog("open") : ($("#dialog-5").dialog("close"));

    });
    $("#dialog-5").dialog({ autoOpen: false });
});
</script>
</head>
<body>
<button id="toggle">Toggle dialog!</button>
<div id="dialog-5" title="Dialog Title!">
    Click on the Toggle button to open and cose this dialog box.
</div>
</body>
</html>

```

روی دکمه ی **Toggle** کلیک کرده و این دیالوگ باکس را ببندید.

مدیریت رویداد روی دیالوگ باکس

علاوه بر متد dialog (options) که در بخش های قبل مشاهده کردیم، JQueryUI متدهای رویداد ارائه می دهد که برای یک رویداد خاص آغاز می شوند. این متدها در جدول زیر ارائه شده اند:

| Event Method | Description |
|-------------------------------|---|
| beforeClose(event, ui) | این رویداد زمانی آغاز می شود که دیالوگ باکس در آستانه ی بسته شدن است. بازگرداندن false از بسته شدن دیالوگ باکس جلوگیری می کند. این رویداد در دسترس دیالوگ باکس هایی است که در ارزیابی موفق نیستند. در اینجا رویداد از نوع Event و ui از نوع Object است. |
| close(event, ui) | این رویداد زمانی آغاز می شود که دیالوگ باکس بسته شده است. در اینجا رویداد از نوع Event و ui از نوع Object است. |

| | |
|-------------------------------------|---|
| <code>create(event, ui)</code> | این رویداد زمانی آغاز می شود که دیاپوگ باکس ایجاد می شود. در اینجا رویداد از نوع Event و ui از نوع Object است. |
| <code>drag(event, ui)</code> | این رویداد به طور مکرر، در هنگام جابه جایی دیاپوگ باکس در هنگام درگ کردن، آغاز می شود. در اینجا رویداد از نوع Event و ui از نوع Object است. |
| <code>dragStart(event, ui)</code> | این رویداد زمانی آغاز می شود که یک پاسخ از دیاپوگ باکس به وسیله ی درگ کردن عنوان آن، ارائه می شود. commences |
| <code>dragStop(event, ui)</code> | این رویداد زمانی آغاز می شود که یک عملیات درگ به پایان می رسد. در اینجا رویداد از نوع Event و ui از نوع Object است. |
| <code>focus(event, ui)</code> | این رویداد زمانی آغاز می شود که دیاپوگ تمرکز را به دست می آورد. در اینجا رویداد از نوع Event و ui از نوع Object است. |
| <code>open(event, ui)</code> | این رویداد زمانی آغاز می شود که دیاپوگ باکس باز شده است. در اینجا رویداد از نوع Event و ui از نوع Object است. |
| <code>resize(event, ui)</code> | این رویداد در هنگام تغییر اندازه ی یک دیاپوگ باکس، به طور مکرر آغاز می شود. در اینجا رویداد از نوع Event و ui از نوع Object است. |
| <code>resizeStart(event, ui)</code> | این رویداد زمانی آغاز می شود که تغییر اندازه ی دیاپوگ باکس شروع شود. در اینجا رویداد از نوع Event و ui از نوع Object است. |
| <code>resizeStop(event, ui)</code> | این رویداد زمانی آغاز می شود که تغییر اندازه ی دیاپوگ باکس به پایان برسد. در اینجا رویداد از نوع Event و ui از نوع Object است. |

مثال زیر استفاده از رویدادهای لیست شده در جدول بالا را توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Dialog functionality</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>
  <!-- CSS -->
  <style>
    .ui-widget-header, .ui-state-default, ui-button {
      background: #b9cd6d;
      border: 1px solid #b9cd6d;
      color: #FFFFFF;
      font-weight: bold;
    }
    .invalid {
      color: red;
    }
    textarea {
      display: inline-block;
      width: 100%;
      margin-bottom: 10px;
    }
  </style>
  <!-- Javascript -->
  <script type="text/javascript">
    $(function () {
      $("#dialog-6").dialog({
        autoOpen: false,
        buttons: {
          OK: function () {
            $(this).dialog("close");
          }
        },
        beforeClose: function (event, ui) {
          if ($("#terms").prop("checked")) {
            event.preventDefault();
            $("#[for=terms]").addClass("invalid");
          }
        },
        width: 600
      });
      $("#opener-5").click(function () {
        $("#dialog-6").dialog("open");
      });
    });
  </script>
</head>
<body>
  <div id="dialog-6">
    <p>You must accept these terms before continuing.</p>
    <textarea>
```

This Agreement and the Request constitute the entire agreement of the parties with respect to the subject matter of the Request. This Agreement shall be governed by and construed in accordance with the laws of the State, without giving effect to principles of conflicts of law.

```
</textarea>
<div>
  <label for="terms">I accept the terms</label>
  <input type="checkbox" id="terms">
</div>
</div>
<button id="opener-5">Open Dialog</button>
</body>
</html>
```

استفاده از متد **resize**

مثال زیر استفاده از متد **resize** را توضیح می دهد:

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Dialog functionality</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>
  <!-- CSS -->
  <style>
    .ui-widget-header, .ui-state-default, ui-button {
      background: #b9cd6d;
      border: 1px solid #b9cd6d;
      color: #FFFFFF;
      font-weight: bold;
    }
  </style>
  <!-- Javascript -->
  <script type="text/javascript">
    $(function () {
      $("#dialog-7").dialog({
        autoOpen: false,
        resize: function (event, ui) {
          $(this).dialog("option", "title",
            ui.size.height + " x " + ui.size.width);
        }
      });
      $("#opener-6").click(function () {
        $("#dialog-7").dialog("open");
      });
    });
  </script>
</head>
<body>
  <div id="dialog-7" title="Dialog Title goes here...">Resize this dialog to see the dialog co-ordinates in the title!</div>
```

```
<button id="opener-6">Open Dialog</button>
</body>
</html>
```

این دیالوگ را تغییر اندازه دهید تا هماهنگی را در عنوان مشاهده کنید.

نکات گسترش

دیالوگ ویجت با **widget factory** ساخته شده و می تواند گسترده شود. برای گسترش ویجت ها، می توانیم متدهای موجود را نادیده گرفته یا آنها را اضافه کنیم. جدول زیر یک نکته ی گسترش (**extension point**) با همان **API** ارائه می دهد، همانطور که متدهای دیالوگ در جدول بالا ارائه شدند.

| Method | Description |
|---------------------------------|--|
| _allowInteraction(event) | این متد به یوزر اجازه می دهد تا با whitelisting عناصر که زیرمجموعه ی دیالوگ نیستند، اما یوزر می تواند از آنها استفاده کند، با یک عنصر هدف تعامل داشته باشد. در اینجا رویداد Event می باشد. |

jQuery UI Widget Factory یک مورد توسعه پذیر است که بر اساس آن همه ی ویجت های **jQuery UI** ساخته می شوند. استفاده از **widget factory** برای ساخت یک پلاگین تسهیلاتی را ارائه می دهد و همچنین کنوانسیون هایی برای کارهای معمول مانند آشکارسازی روش پلاگین و تغییر گزینه های بعد از نمونه.

JqueryUI – Menu

یک منو ویجت معمولاً حاوی یک نوار منوی اصلی با منوهای پیش رو می باشد. آیتم های این منوها اغلب دارای منوهای فرعی نیز می باشند. یک منو با استفاده از عناصر مارک آپ به همراه برقراری ارتباط منبع – جزء (**parent-child**) ایجاد می شود (با استفاده از **** یا ****). هر آیتم منو دارای یک عنصر لنگر می باشد.

منو ویجت در **jQueryUI** می تواند برای منوهای درون خطی و پاپ آپ استفاده شود، یا به عنوان یک اساس برای ساخت سیستم های منوی پیچیده تر استفاده می شود. به عنوان مثال می توانید با استفاده از موقعیت های سفارشی، منوهای تو دو تو ایجاد کنید.

jQueryUI متدهای **menu()** را برای ایجاد یک منو ارائه می دهد.

ترکیب:

متد **menu()** به دو شکل مورد استفاده قرار می گیرد

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

`$(selector, context).menu ("action", params) Method`

متد `$(selector, context).menu (options)`

متد **menu (options)** اعلام می کند که یک عنصر **HTML** و محتوای آن باید به عنوان منو تنظیم و رفتار شود. پارامتر **options** آبجکتی است که ظاهر و رفتار آیتم های منو را تعیین می کند.

ترکیب

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

در هنگام استفاده از آبجکت **Javascript**، می توانید یک یا چند گزینه ارائه دهید. اگر بیشتر از یک گزینه برای ارائه وجود دارد، آنها را مانند زیر با استفاده از کاما (ویرگول) از یکدیگر جدا می کنید:

`$(selector, context).menu({option1: value1, option2: value2..... });`

جدول زیر گزینه هایی مختلفی را لیست می کند که می توان با این متد استفاده کرد:

| Option | Description |
|--------|-------------|
|--------|-------------|

| | |
|----------|--|
| disabled | این گزینه اگر روی true تنظیم شده باشد، منو را غیرفعال می کند. به طور پیش فرض مقدار آن false می باشد. |
| icons | این گزینه آیکن ها را برای منوهای فرعی تنظیم می کند. به طور پیش فرض مقدار آن می باشد { submenu: "ui-icon-carat-1-e" }. |
| menus | این گزینه یک انتخاب گر برای عناصری است که به عنوان ظرف منو در نظر گرفته می شوند، شامل منوهای فرعی. مقدار آن به طور پیش فرض ui می باشد. |
| position | این گزینه موقعیت منوهای فرعی را در ارتباط با آیتم منوی اصلی تنظیم می کند. به طور پیش فرض مقدار آن { my: "left top", at: "right top" } می باشد. |
| role | این گزینه برای شخصی سازی نقش های ARIA برای منو و آیتم های منو استفاده می شود. مقدار آن به طور پیش فرض menu می باشد. |

کارکرد پیش فرض

در زیر مثال ساده ای از کارکرد منو ویجت را مشاهده می کنید که هیچ پارامتری را به متد **menu()** انتقال نمی دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Menu functionality</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>
  <!-- CSS -->
  <style>
    .ui-menu {
      width: 200px;
    }
  </style>
  <!-- Javascript -->
  <script>
```



```

$(function () {
    $("#menu-1").menu();
});
</script>
</head>
<body>
<!-- HTML -->
<ul id="menu-1">
<li><a href="#">Spring</a></li>
<li><a href="#">Hibernate</a></li>
<li>
<a href="#">Java</a>
<ul>
<li><a href="#">Java IO</a></li>
<li><a href="#">Swing</a></li>
<li><a href="#">Jaspr Reports</a></li>
</ul>
</li>
<li><a href="#">JSF</a></li>
<li><a href="#">HTML5</a></li>
</ul>
</body>
</html>

```

در مثال بالا مشاهده می کنید که منوی **themeable** با عملکردهای ماوس و صفحه کلید برای مسیریابی طراحی شده است.

آموزشگاه تحلیکیر داده ها

استفاده از آیکن ها و موقعیت ها

مثال زیر استفاده از دو گزینه ی **icons** و **position** را در عملکرد منوی از **jQueryUI** نشان می دهد.

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Menu functionality</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>
<!-- CSS -->
<style>
.ui-menu {
width: 200px;
}
</style>
<!-- Javascript -->
<script>
$(function () {
    $("#menu-2").menu({

```

```

        icons: { submenu: "ui-icon-circle-triangle-e" },
        position: { my: "right top", at: "right-10 top+5" }
    });
});
</script>
</head>
<body>
<!-- HTML -->
<ul id="menu-2">
<li><a href="#">Spring</a></li>
<li><a href="#">Hibernate</a></li>
<li>
<a href="#">Java</a>
<ul>
<li><a href="#">Java IO</a></li>
<li><a href="#">Swing</a></li>
<li><a href="#">Jaspr Reports</a></li>
</ul>
</li>
<li><a href="#">JSF</a></li>
<li><a href="#">HTML5</a></li>
</ul>
</body>
</html>

```

در مثال بالا مشاهده می کنید که آیکن **image** را برای لیست منوی فرعی به کار برده ایم و همچنین موقعیت منوی فرعی را تغییر داده ایم.

متد `$(selector, context).menu ("action", params)`

متد `menu ("action", params)`، می تواند یک فعالیت را روی عناصر منو اجرا کند، مانند فعال / غیرفعال کردن منو. این فعالیت به عنوان یک رشته در اولین مبحث تعیین می شود (به عنوان مثال **"disable"** منو را غیر فعال می سازد). فعالیت هایی را که قابل انتقال هستند در جدول زیر بررسی کنید.

ترکیب

`$(selector, context).menu ("action", params);`

جدول زیر فعالیت های مختلفی را لیست می کند که با این متد مورد استفاده قرار می گیرند:

| Action | Description |
|--------|-------------|
|--------|-------------|

| | |
|--|--|
| <code>blur([event])</code> | این فعالیت تمرکز را از یک منو حذف می کند و با تنظیم مجدد هر کدام از عناصرهای فعال استایل، رویداد <code>blur</code> مربوط به منو را آغاز می کند. در حالیکه رویداد از نوع <code>Event</code> می باشد و نشان می دهد که چه چیزی باعث تیرگی در منو شده است. |
| <code>collapse([event])</code> | این عملکرد منوی فرعی فعال را می بندد. در حالیکه رویداد از نوع <code>Event</code> می باشد و نشان می دهد که چه چیزی باعث از هم پاشیدن منو شده است. |
| <code>collapseAll([event], [all])</code> | این فعالیت تمام منوهای فرعی را می بندد. |
| <code>destroy()</code> | این عملکرد کارکرد منو را کاملاً حذف می کند. این عمل باعث بازگشت عنصر به وضعیت <code>pre init</code> – خود خواهد شد. این متد هیچ برهانی (<code>argument</code>) را نمی پذیرد. |
| <code>disable()</code> | این عملکرد منو را غیرفعال می کند. این متد نیز هیچ برهانی را نمی پذیرد. |
| <code>enable()</code> | این عملکرد منو را فعال می سازد. این متد نیز هیچ برهانی نمی پذیرد. |
| <code>expand([event])</code> | این عمل منوی فرعی زیر آیتم فعال را، اگر وجود داشته باشد، باز می کند. در حالیکه رویداد از نوع <code>Event</code> می باشد و نشان می دهد که چه چیزی باعث باز شدن منو شده است. |
| <code>focus([event], item)</code> | این عملکرد یک آیتم خاص از منو را فعال می سازد، به این صورت که شروع به باز کردن منوهای فرعی کرده و تمرکز رویداد منو را آغاز می کند. در حالیکه رویداد از نوع <code>Event</code> می باشد و نشان می دهد که چه چیزی باعث منو تمرکز را به دست آورد و آیتم یک آبجکت <code>jQuery</code> می باشد که آیتم منو به <code>focus/activate</code> را نمایش می دهد. |
| <code>isFirstItem()</code> | این فعالیت یک مقدار <code>boolean</code> را باز می گرداند که اگر آیتم فعال حال حاضر منو، اولین آیتم منو باشد، بیان می دارد. این متد هیچ برهانی را نمی پذیرد. |

| | |
|-----------------------------|---|
| isLastItem() | این فعالیت یک مقدار Boolean بازمی گرداند که اگر آیتم فعال حال حاضر منو، آخرین آیتم منو باشد، بیان می دارد. این متد هیچ برهانی را نمی پذیرد. |
| next([event]) | این فعالیت وضعیت فعال را به آیتم بعدی منو محول می کند. از آنجایی که رویداد از نوع Event می باشد و نشان می دهد چه چیزی باعث آغاز جابجایی تمرکز شده است. |
| nextPage([event]) | این عملکرد وضعیت فعال را به اولین آیتم منو در زیر یک منوی رونده scrollable menu انتقال می دهد، یا اگر منو حرکت کننده نباشد آخرین آیتم آن قرار می گیرد. از آنجایی که رویداد از نوع Event می باشد و نشان می دهد که چه چیزی باعث شروع جابجایی تمرکز شده است. |
| option(optionName) | این فعالیت مقدار فعلی مربوط به optionName را دریافت می کند. از آنجایی که optionName از نوع String می باشد و نام گزینه ی به دست آمده را نشان می دهد. |
| option() | این فعالیت جفت های حاوی key/value را دریافت می کند که نشان دهنده ی گزینه های حال حاضر منو می باشد. |
| option(optionName, value) | این فعالیت مقدار گزینه ی منو را که در ارتباط با optionName تعیین شده می باشد، تنظیم می کند. در حالیکه optionName از نوع String می باشد و نام گزینه را برای تنظیم نشان می دهد و مقدار از نوع Object می باشد که مقدار مربوط به تنظیم گزینه را نشان می دهد. |
| option(options) | این عملکرد یک یا چند گزینه را برای منو تنظیم می کند. در حالیکه گزینه ها از نوع Object می باشند و نقشه ای از جفت های option-value برای تنظیم نشان می دهند. |
| previous([event]) | این عملکرد وضعیت فعال را به آیتم منوی پیشین بازمی گرداند. در اینجا رویدادها از نوع Events می باشند و نشان می دهند که چه چیزی باعث حرکت تمرکز را آغاز کرد. |

| | |
|----------------------------|--|
| previousPage([event]) | این فعالیت وضعیت فعال را به اولین آیتم منو در بالای یک scrollable یا در غیر این صورت به اولین آیتم منو حرکت می دهد. در اینجا رویداد از نوع Event می باشد و نشان می دهد که چه چیزی باعث آغاز حرکت تمرکز شد. |
| refresh() | این فعالیت آیتم های منوهای زیرمجموعه و منوهای اصلی را که هنوز مقدار دهی نشده اند، مقدار دهی می کند. این متد هیچ برهانی نمی پذیرد. |
| select([event]) | این فعالیت آیتم منو فعال را انتخاب می کند، همه ی منوهای زیرمجموعه را می شکنند و منو انتخاب رویداد را آغاز می کند. در اینجا رویداد از نوع Event می باشد و مشخص می کند که چه چیزی مجموعه را آغاز کرد. |
| widget() | این فعالیت یک آبجکت jQuery باز می گرداند که حاوی منو می باشد. این متد نیز هیچ برهانی نمی پذیرد. |

مثال زیر استفاده از متد **disable()** را توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Menu functionality</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>
  <!-- CSS -->
  <style>
    .ui-menu {
      width: 200px;
    }
  </style>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#menu-3").menu();
      $("#menu-3").menu("disable");
    });
  </script>

```

```

</script>
</head>
<body>
<!-- HTML -->
<ul id="menu-3">
<li><a href="#">Spring</a></li>
<li><a href="#">Hibernate</a></li>
<li>
<a href="#">Java</a>
<ul>
<li><a href="#">Java IO</a></li>
<li><a href="#">Swing</a></li>
<li><a href="#">Jaspr Reports</a></li>
</ul>
</li>
<li><a href="#">JSF</a></li>
<li><a href="#">HTML5</a></li>
</ul>
</body>
</html>

```

در مثال بالا مشاهده می کنید که منو غیرفعال می باشد.

استفاده از متدهای focus و collapseAll

مثال زیر استفاده از متدهای focus() و collapseAll را نشان می دهد:

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Menu functionality</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>
<!-- CSS -->
<style>
.ui-menu {
width: 200px;
}
</style>
<!-- Javascript -->
<script>
$(function () {
var menu = $("#menu-4").menu();
$("#menu-4").menu("focus", null, $("#menu-4").menu().find(".ui-menu-item:last"));
$(menu).mouseleave(function () {
menu.menu('collapseAll');
});
});
</script>
</head>
<body>

```

```

<!-- HTML -->
<ul id="menu-4">
  <li><a href="#">Spring</a></li>
  <li><a href="#">Hibernate</a></li>
  <li><a href="#">JSF</a></li>
  <li><a href="#">HTML5</a></li>
  <li>
    <a href="#">Java</a>
    <ul>
      <li><a href="#">Java IO</a></li>
      <li><a href="#">Swing</a></li>
      <li><a href="#">Jaspr Reports</a></li>
    </ul>
  </li>
</ul>
</body>
</html>

```

در مثال بالا مشاهده می کنید که تمرکز روی آخرین آیتم منو می باشد. اکنون منوی زیرمجموعه را باز کرده و وقتی ماوس از روی منوی زیرمجموعه به بخش دیگری می رود، منوی زیرمجموعه بسته می شود.

مدیریت رویداد روی عناصر منو

علاوه بر متد **menu (options)** ، در بخش های قبلی مشاهده کردیم که **JqueryUI** متدهای رویداد را ارائه می دهد که برای یک رویداد خاص آغاز شدند. این متدهای رویداد در زیر ارائه شده اند:

| Event Method | Description |
|--------------------------|--|
| blur(event, ui) | این رویداد وقتی آغاز می شود که یک منو تمرکز را از دست می دهد. |
| create(event, ui) | این رویداد زمانی آغاز می شود که منو ایجاد شده است. |
| focus(event, ui) | این رویداد زمانی آغاز می شود که منو تمرکز را به دست می آورد و یا وقتی آیتم های منو فعال می شوند. |
| select(event, ui) | این رویداد زمانی آغاز می شود که یک آیتم منو انتخاب شود. |

مثال:

مثال زیر استفاده از متد رویداد را برای کاربرد منو ویجت توضیح می دهد. این مثال استفاده از رویداد

create، blur و focus را توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Menu functionality</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>
  <!-- CSS -->
  <style>
    .ui-menu {
      width: 200px;
    }
  </style>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#menu-5").menu({
        create: function (event, ui) {
          var result = $("#result");
          result.append("Create event<br>");
        },
        blur: function (event, ui) {
          var result = $("#result");
          result.append("Blur event<br>");
        },
        focus: function (event, ui) {
          var result = $("#result");
          result.append("focus event<br>");
        }
      });
    });
  </script>
</head>
<body>
  <!-- HTML -->
  <ul id="menu-5">
    <li><a href="#">Spring</a></li>
    <li><a href="#">Hibernate</a></li>
    <li><a href="#">JSF</a></li>
    <li><a href="#">HTML5</a></li>
    <li>
      <a href="#">Java</a>
      <ul>
        <li><a href="#">Java IO</a></li>
        <li><a href="#">Swing</a></li>
        <li><a href="#">Jaspr Reports</a></li>
      </ul>
    </li>
  </ul>
```



```

</li>
</ul>
<span id="result"></span>
</body>
</html>

```

در مثال بالا ما در حال چاپ پیغام ها براساس رویدادهای آغاز شده هستیم.

JqueryUI - Progressbar

نوارهای پیشروی (progress bar) میزان پیشروی یک عملکرد یا یک پروسه را نشان می دهد. می توانیم نوتر پیشروی را به دو گروه **determinate progress bar** و **indeterminate progress bar** تقسیم بندی کنیم.

Determinate progress bar (نوار پیشروی مشخص) تنها در موقعیت هایی استفاده می شود که سیستم می تواند به درستی وضعیت فعلی را آپدیت کند. نوار پیشروی **determinate** هرگز نباید از چپ به راست پر شود و سپس برای یک فرایند مجزا به عقب بازگردد.

اگر وضعیت حقیقی نمی تواند محاسبه شود، **indeterminate progress bar** باید برای ارائه ی بازخورد یوزر به کار گرفته شود.

jQueryUI یک نوار پیشروی با استفاده ی آسان ارائه می دهد که می توانیم از این طریق یوزرها را مطلع کنیم که اجرای درخواست انجام شده برای برنامه ی ما سخت می باشد. **jQueryUI** متد **progressbar()** را برای ایجاد نوارهای پیشروی ارائه می دهد.

ترکیب:

متد **progressbar()** به دو روش زیر انجام می شود

\$(selector, context).progressbar (options) Method

\$(selector, context).progressbar ("action", params) Method

متد \$(selector, context).progressbar (options)

متد **progressbar (options)** اعلام می کند که یک عنصر **HTML** می تواند به فرم یک نوار پیشروی مدیریت شود. پارامتر **options** آبجکتی است که ظاهر و رفتار نوار پیشروی را تعیین می کند.

ترکیب

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

شما می توانید با استفاده از آبجکت **Javascript** یک یا چند گزینه را همزمان ارائه دهید. اگر بیشتر از یک گزینه برای ارائه وجود دارد، می توانید آنها را با استفاده از کاما از یکدیگر مجزا کنید، مانند ترکیب زیر:

`$(selector, context).progressbar({option1: value1, option2: value2..... });`

جدول زیر گزینه های مختلفی را ارائه می دهد که می توانید با این متد استفاده کنید:

| Option | Description |
|----------|---|
| disabled | وقتی این گزینه روی true تنظیم شده باشد، نوار پیشروی را غیرفعال می سازد. مقدار پیش فرض آن false می باشد. |
| max | این گزینه حداکثر مقدار برای یک نوار پیشروی را تعیین می کند. مقدار پیش فرض آن 100 می باشد. |
| value | این گزینه مقدار اولیه ی نوار پیشروی را تعیین می کند. مقدار پیش فرض آن 0 است. |

در بخش بعد مثال هایی از قابلیت نوار پیشروی مشاهده خواهید کرد.

مثال زیر توضیح ساده ای از قابلیت نوار پیشروی را ارائه می دهد که هیچ پارامتری به متد `progressbar()` انتقال نمی دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI ProgressBar functionality</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>
    .ui-widget-header {
      background: #cedc98;
      border: 1px solid #DDDDDD;
      color: #333333;
      font-weight: bold;
    }
  </style>
  <script>
    $(function () {
      $("#progressbar-1").progressbar({
        value: 30
      });
    });
  </script>
</head>
<body>
  <div id="progressbar-1"></div>
</body>
</html>
```

اجازه بدهید کد بالا را در فایل **HTML** به نام **progressbarexample.htm** ذخیره کنیم و آن را در یک مرورگر استاندارد که **javascript** را پشتیبانی می کند، باز کنیم. پس از آن باید نتیجه ی زیر را مشاهده. اکنون می توانید با نتیجه بازی کنید.

این مثال ایجاد یک نوار پیشروی را با استفاده از متد **progressbar()** نشان می دهد. این یک نوار پیشروی پیش فرض می باشد.

استفاده از max و value

مثال زیر استفاده از گزینه های **values** و **max** را در قابلیت نوار پیشروی **jQueryUI** نشان می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI ProgressBar functionality</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>
    .ui-widget-header {
      background: #cedc98;
      border: 1px solid #DDDDDD;
      color: #333333;
      font-weight: bold;
    }
  </style>
  <script>
    $(function () {
      var progressbar = $("#progressbar-2");
      $("#progressbar-2").progressbar({
        value: 30,
        max: 300
      });
      function progress() {
        var val = progressbar.progressbar("value") || 0;
        progressbar.progressbar("value", val + 1);
        if (val < 99) {
          setTimeout(progress, 100);
        }
      }
      setTimeout(progress, 3000);
    });
  </script>
</head>
<body>
  <div id="progressbar-2"></div>
</body>
</html>

```

در اینجا مشاهده می کنید که نوار پیشروی از راست به چپ پر شده و وقتی مقدار به 300 می رسد، متوقف می شود.

متد `$(selector, context).progressbar ("action", params)`

متد `progressbar ("action", params)` می تواند عملکردهایی را روی نوار پیشروی اجرا کند، از جمله تغییر درصد پیشروی. عملکرد به عنوان یک رشته در اولین مبحث مطرح می شود (به عنوان مثال مقداری برای تغییر درصد پیشروی). در جدول زیر عملکردهایی را بررسی خواهیم کرد که می توانند انتقال یابند.

ترکیب:

`$(selector, context).progressbar ("action", params);`

جدول زیر **action** های مختلفی را لیست می کند که با این متد استفاده می شوند:

| Action | Description |
|-----------------------------|--|
| destroy | این عملکرد قابلیت نوار پیشروی را از یک عنصر به طور کامل حذف می کند. عناصر به وضعیت pre-init خود بازمی گردند. این متد هیچ برهانی (argument) را نمی پذیرد. |
| disable | این عمل قابلیت نوار پیشروی را روی یک عنصر غیرفعال می سازد. این متد هیچ برهانی را نمی پذیرد. |
| enable | این عمل قابلیت نوار پیشروی را روی یک عنصر فعال می سازد. این متد هیچ برهانی را نمی پذیرد. |
| option(optionName) | این عمل مقدار کنونی را، که با optionName تعیین شده در ارتباط است، بازمی گرداند. در اینجا optionName یک رشته می باشد. |
| option | این عمل، یک آبجکت حاوی جفت های key/value را دریافت می کند که گزینه های نوار پیشروی hash را نشان می دهند. این متد هیچ برهانی را نمی پذیرد. |
| option(optionName, value) | این عمل مقدار گزینه ی نوار پیشروی را که با optionName تعیین شده در ارتباط است، تنظیم می کند. |
| option(options) | این عمل یک یا چند گزینه را برای نوارهای پیشروی تعیین می کند. برهان options نقشه ای است که دارای جفت های option-value است که قرار است تنظیم شوند. |

| | |
|----------------|--|
| value | این عمل مقدار کنونی options.value را بازمی گرداند که درصد پیشروی نوار پیشروی را نشان می دهد. |
| value(value) | این عمل یک مقدار جدید برای درصد پر شده در نوار پیشروی را نشان می دهد. مقدار argument میتواند یک عدد یا یک Boolean باشد. |
| widget | این عمل یک آبجکت jQuery را که حاوی نوار پیشروی می باشد، بازمی گرداند. این متد هیچ برهانی را نمی پذیرد. |

مثال:

اکنون اجازه بدهید با استفاده از عملگردهای جدول بالا، مثالی را بررسی کنیم. مثال زیر استفاده از متدهای **disable()** و **option(optionName, value)** را توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI ProgressBar functionality</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>
    .ui-widget-header {
      background: #cedc98;
      border: 1px solid #DDDDDD;
      color: #333333;
      font-weight: bold;
    }
  </style>
</script>
$(function () {
  $("#progressbar-3").progressbar({
    value: 30
  });
  $("#progressbar-3").progressbar('disable');
  $("#progressbar-4").progressbar({
    value: 30
  });
  var progressbar = $("#progressbar-4");
  $("#progressbar-4").progressbar("option", "max", 1024);
  function progress() {
    var val = progressbar.progressbar("value") || 0;
    progressbar.progressbar("value", val + 1);
    if (val < 99) {

```

```

        setTimeout(progress, 100);
    }
}
setTimeout(progress, 3000);
});
</script>
</head>
</body>
<h3>Disabled Progressbar</h3>
<div id="progressbar-3"><div><br>
<h3>Progressbar with max value set</h3>
<div id="progressbar-4"><div>
</body>
</html>

```

مدیریت رویداد روی بخش های نوار پیشروی

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

در اینجا مشاهده می کنید که نوار پیشروی مقدار تغییر یافته ی خود را تغییر می دهد و در هنگام کامل شدن رویداد پیغام **"Loading Completed"** نمایش داده می شود.

| Event Method | Description |
|---------------------|--|
| change(event, ui) | هروقت که مقدار نوار پیشروی تغییر کند، این رویداد آغاز می شود. رویداد از نوع Event بوده و ui از نوع Object می باشد. |
| complete(event, ui) | وقتی که نوار پیشروی به حداکثر مقدار خود برسد، این رویداد آغاز می شود. رویداد از نوع Event بوده و ui از نوع Object می باشد. |

| | |
|----------------------|---|
| create(event, ui) | وقتی نوار پیشروی ایجاد می شود، این رویداد آغاز می شود. رویداد از نوع Event بوده و ui از نوع Object می باشد. |
|----------------------|---|

مثال:

مثال زیر متدی را توضیح می دهد که در حین قابلیت نوار پیشروی استفاده می شود. این مثال استفاده از رویدادهای **change** و **complete** را توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI ProgressBar functionality</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>
    .ui-widget-header {
      background: #cedc98;
      border: 1px solid #DDDDDD;
      color: #333333;
      font-weight: bold;
    }
    .progress-label {
      position: absolute;
      left: 50%;
      top: 13px;
      font-weight: bold;
      text-shadow: 1px 1px 0 #fff;
    }
  </style>
  <script>
    $(function () {
      var progressbar = $("#progressbar-5");
      progressLabel = $(".progress-label");
      $("#progressbar-5").progressbar({
        value: false,
        change: function () {
          progressLabel.text(
            progressbar.progressbar("value") + "%");
        },
        complete: function () {
          progressLabel.text("Loading Completed!");
        }
      });
      function progress() {
        var val = progressbar.progressbar("value") || 0;
```



```

        progressbar.progressbar("value", val + 1);
        if (val < 99) {
            setTimeout(progress, 100);
        }
    }
    setTimeout(progress, 3000);
});
</script>
<head>
<body>
<div id="progressbar-5">
<div class="progress-label">
    Loading...
</div>
</div>
</body>
</html>

```

JqueryUI - Slider

Slider زمانی استفاده می شود که یک مقدار عددی در داخل یک محدوده ی خاص به دست می آید. مزیت یک **slider** برای ورودی یک متن این است که برای کاربر غیر ممکن می شود که یک مقدار بد وارد کند. هر مقداری که می توانند با **slider** انتخاب کنند، معتبر خواهد بود.

jQueryUI از طریق **slider widget** به ما یک کنترل **slider** ارائه می دهد. **jQueryUI** متد **slider()** را ارائه می دهد که ظاهر عناصر **HTML** را در صفحه تغییر می دهد و گروه های جدید **CSS** اضافه می کند که سبک مناسبی به آنها می دهد.

ترکیب:

متد **slider ()** به دو فرم قابل استفاده می باشد

\$(selector, context).slider (options) Method

\$(selector, context).slider ("action", params) Method

متد **\$(selector, context).slider (options)**

متد **slider (options)** اعلام می کند که یک عنصر **HTML** باید به عنوان یک **slider** مدیریت شود. پارامتر **options** آبجکتی است که رفتار و ظاهر یک **slider** را تعیین می کند.

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

شما می توانید یک یا چند گزینه را همزمان با استفاده از آبجکت **javascript** ارائه دهید. اگر بیشتر از یک گزینه برای ارائه وجود داشته باشد، آنها را با استفاده از کاما از یکدیگر مجزا می کنید، مانند ارکیب زیر:

`$(selector, context).slider({option1: value1, option2: value2..... });`

جدول زیر گزینه های مختلفی را که با این متد به کار گرفته می شود، نشان می دهد:

| Option | Description |
|-------------|--|
| animate | این گزینه اگر روی true تنظیم شده باشد، وقتی که کاربر مستقیماً روی مختصات کلیک می کند، یک اثر انیمیشنی ایجاد می شود. مقدار پیش فرض آن false می باشد. |
| disabled | این گزینه اگر روی true تنظیم شده باشد، slider را غیرفعال می سازد. مقدار آن به طور پیش فرض false می باشد. |
| max | این گزینه بالاترین مقداری را که slider می تواند در محدوده داشته باشد، تعیین می کند - وقتی دسته به سمت راست (برای slider های افقی) یا به سمت بالا (برای slider های عمودی) حرکت داده می شود، مقدار نمایش داده می شود. مقدار آن به طور پیش فرض 100 می باشد. |
| min | این گزینه بالاترین مقداری را که slider می تواند در محدوده داشته باشد، تعیین می کند - وقتی دسته به سمت چپ (برای slider های افقی) یا به سمت پایین (برای slider های عمودی) حرکت داده می شود، مقدار نمایش داده می شود. مقدار آن به طور پیش فرض 0 می باشد. |
| orientation | این گزینه گرایش افقی یا عمودی slider را نشان می دهد. مقدار پیش فرض آن horizontal می باشد. |

| | |
|--------|---|
| range | این گزینه تعیین می کند که آیا یک محدوده یک slider را مشخص می کند یا نه. مقدار آن به طور پیش فرض false است. |
| step | این گزینه فواصل گسسته بین مقادیر حداکثر و حداقل را تعیین می کند که slider اجازه ی نمایش آن را دارد. مقدار پیش فرض آن 1 می باشد. |
| value | این گزینه مقدار اولیه ی یک slider دارای یک هندل را تعیین می کند. اگر چند هندل وجود داشته باشد، مقدار مربوط به اولین هندل را تعیین می کند. مقدار پیش فرض آن 1 است. |
| values | این گزینه از نوع array می باشد و باعث ایجاد چند هندل می شود و مقدار اولیه را برای آن هندل ها تعیین می کند. این گزینه ردیفی از مقادیر ممکن، هر کدام برای یک هندل، باشد. مقدار پیش فرض آن null است. |

بخش بعد چند مثال در رابطه با قابلیت slider به شما ارائه می دهد.

قابلیت پیش فرض

مثال زیر قابلیت slider را توضیح می دهد که هیچ پارامتری به متد slider() انتقال نمی دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Slider functionality</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>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#slider-1").slider();
    });
  </script>
</head>
<body>
  <!-- HTML -->
  <div id="slider-1"></div>

```

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

در مثال بالا یک slider افقی پایه وجود دارد که دارای یک هندل مجزا می باشد که به وسیله ی ماوس یا با استفاده از کلیدهای جهت روی صفحه کلید، جابه جا می شود.

استفاده از value، animate و orientation

مثال زیر استفاده از سه گزینه ی **animate (b) value (a)** و **orientation (c)** را در عملکرد slider از JQueryUI توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Slider functionality</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>
<!-- Javascript -->
<script>
$(function () {
  $("#slider-2").slider({
    value: 60,
    animate: "slow",
    orientation: "horizontal"
  });
});
</script>
</head>
<body>
<!-- HTML -->
<div id="slider-2"></div>
</body>
</html>
```

در مثال بالا مقدار slider، به عنوان مثال مقدار اولیه روی 60 تنظیم شده است، و هندل را در مقدار اولیه ی 60 مشاهده می کنید. اکنون مستقیما روی محور کلیک کرده و افکت انیمیشن را مشاهده کنید.

مثال زیر استفاده از سه گزینه ی (c) max، (b) min، (a) range و (d) values را در عملکرد slider از JQueryUI توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Slider functionality</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>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#slider-3").slider({
        range: true,
        min: 0,
        max: 500,
        values: [35, 200],
        slide: function (event, ui) {
          $("#price").val("$" + ui.values[0] + " - $" + ui.values[1]);
        }
      });
      $("#price").val("$" + $("#slider-3").slider("values", 0) +
        " - $" + $("#slider-3").slider("values", 1));
    });
  </script>
</head>
<body>
  <!-- HTML -->
  <p>
    <label for="price">Price range:</label>
    <input type="text" id="price"
      style="border:0; color:#b9cd6d; font-weight:bold;">
  </p>
  <div id="slider-3"></div>
</body>
</html>
```

در مثال بالا دامنه ی گزینه را برای گرفتن دامنه ای از مقادیر با دو هندل، روی true تنظیم کردیم. فضای بین هندل ها با رنگ زمینه ی متفاوت پر می شود تا مقادیر انتخاب شده را نشان دهد.

متد `$(selector, context).slider ("action", params)`

متد `("action", params)` یک عمل روی **slider** را مجاز می کند، به عنوان مثال حرکت مکان نما به یک موقعیت جدید. این عمل به عنوان یک رشته در اولین مبحث مطرح می شود (برای مثال **value** برای نشان دادن یک مقدار جدید از مکان نما). فعالیت هایی را که قابل انتقال هستند، در جدول زیر بررسی کنید.

ترکیب:

`$(selector, context).slider ("action", params);`

جدول زیر فعالیت هایی را لیست می کند که با این متد مورد استفاده قرار می گیرند:

| Action | Description |
|----------------------|--|
| destroy | این فعالیت قابلیت slider را از روی یک گزینه به طور کامل حذف می کند. عناصر به وضعیت pre-init خود بازمی گردند. این متد هیچ برهانی (argument) قبول نمی کند. |
| disable | این عمل قابلیت slider را غیرفعال می سازد. این متد هیچ برهانی (argument) قبول نمی کند. |
| enable | این فعالیت قابلیت slider را فعال می سازد. این متد هیچ برهانی (argument) قبول نمی کند. |
| option(optionName) | این فعالیت مقدار معین گزینه ی param را بازمی گرداند. این گزینه با یکی از گزینه های استفاده شده با slider (options) هماهنگ است. optionName نام گزینه ی دریافتی است. |

| | |
|-----------------------------|--|
| option() | این فعالیت یک آبجکت حاوی جفت های key/value را دریافت می کند که نمایشگر گزینه های slider فعلی hash می باشد. |
| option(optionName, value) | این فعالیت مقدار گزینه ی slider را در ارتباط با optionName تعیین شده، تنظیم می کند. optionName نام گزینه ای است که باید تنظیم شود و value مقداری است که برای این گزینه تنظیم می شود. |
| option(options) | این عمل یک یا چند گزینه را برای slider تنظیم می کند. گزینه های argument نقشه ای است از جفت های option-value که قرار است تنظیم شوند. |
| value | این عملکرد مقدار مربوط به options.value (the slider) of را بازمی گرداند. فقط اگر slider منحصر به فرد می باشد، از آن استفاده کنید، در غیر اینصورت از slider ("values") استفاده کنید. این عمل نیز هیچ برهانی را نمی پذیرد. |
| value(value) | این عمل مقدار مربوط به slider را تنظیم می کند. |
| values | این عمل مقدار کنونی مربوط به of options.values را بازمی گرداند (مقدار slider ها یک ردیف می باشد). این عمل نیز هیچ برهانی را نمی پذیرد. |
| values(index) | این عمل مقداری برای هندل تعیین شده دریافت می کند. در اینجا index از نوع عدد صحیح می باشد و یک شاخص صفر محور از هندل است. |
| values(index, value) | این عمل برای هندل تعیین شده، مقدار تنظیم می کند. در اینجا index از نوع شاخص صفر محور است و value مقداری است که باید تنظیم شود. |
| values(values) | این عمل برای تمام هندل ها مقدار تنظیم می کند. |
| widget | این عمل یک آبجکت jQuery حاوی slider را بازمی گرداند. این متد هیچ برهانی را نمی پذیرد. |

مثال:

اکنون اجازه بدهید مثالی را با استفاده از عملکردهای جدول بالا بررسی کنیم. مثال زیر استفاده از متدهای **value()** و **disable()** را توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Slider functionality</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>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#slider-4").slider({
        orientation: "vertical"
      });
      $("#slider-4").slider('disable');
      $("#slider-5").slider({
        orientation: "vertical",
        value: 50,
        slide: function (event, ui) {
          $("#minval").val(ui.value);
        }
      });
      $("#minval").val($("#slider-5").slider("value"));
    });
  </script>
</head>
<body>
  <!-- HTML -->
  <div id="slider-4"></div>
  <p>
    <label for="minval">Minumum value:</label>
    <input type="text" id="minval"
      style="border:0;color:#b9cd6d;font-weight:bold;">
  </p>
  <div id="slider-5"></div>
</body>
</html>
```

در مثال بالا اولین slider غیرفعال بوده و دومین slider دارای مقدار تنظیم شده ی 50 می باشد.

علاوه بر متد **slider (options)** ، در بخش قبل مشاهده کردیم که **JqueryUI** متدهایی را ارائه می دهد که برای یک رویداد خاص آغاز می شوند. این متدهای رویداد در زیر ارائه شده اند:

| Event Method | Description |
|-------------------|--|
| change(event, ui) | این رویداد زمانی آغاز می شود که مقدار هندل تغییر کند، چه از طریق عمل یا از طریق برنامه نویسی. |
| create(event, ui) | این رویداد با ایجاد slider آغاز می شود. |
| slide(event, ui) | این رویداد با جابه جایی ماوس در هنگام درگ شدن هندل از طریق slider، آغاز می شود. گزارش مقدار false اسلاید را کنسل می کند. |
| start(event, ui) | وقتی کاربر شروع به اسلاید کردن کند، این رویداد آغاز می شود. |
| stop(event, ui) | این رویداد در هنگام توقف یک اسلاید، آغاز می شود. |

مثال:

مثال زیر استفاده از متد رویداد را طی عملکرد **slider** توضیح می دهد. این مثال استفاده از رویدادهای **start**، **stop**، **change** و **slide** را توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Slider functionality</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>
<!-- Javascript -->
<script>
$(function () {
    $("#slider-6").slider({
        range: true,
        min: 0,
        max: 500,
        values: [35, 200],
        start: function (event, ui) {
            $("#startvalue")
                .val("$" + ui.values[0] + " - $" + ui.values[1]);
        },
        stop: function (event, ui) {
            $("#stopvalue")
                .val("$" + ui.values[0] + " - $" + ui.values[1]);
        },
        change: function (event, ui) {
            $("#changevalue")
                .val("$" + ui.values[0] + " - $" + ui.values[1]);
        },
        slide: function (event, ui) {
            $("#slidevalue")
                .val("$" + ui.values[0] + " - $" + ui.values[1]);
        }
    });
});
</script>
</head>
<body>
<!-- HTML -->
<div id="slider-6"></div>
<p>
    <label for="startvalue">Start:</label>
    <input type="text" id="startvalue"
        style="border:0; color:#b9cd6d; font-weight:bold;">
</p>
<p>
    <label for="stopvalue">Stop:</label>
    <input type="text" id="stopvalue"
        style="border:0; color:#b9cd6d; font-weight:bold;">
</p>
<p>
    <label for="changevalue">Change:</label>
    <input type="text" id="changevalue"
        style="border:0; color:#b9cd6d; font-weight:bold;">
</p>
<p>
    <label for="slidevalue">Slide:</label>
    <input type="text" id="slidevalue"
        style="border:0; color:#b9cd6d; font-weight:bold;">
</p>
</body>
</html>

```

JqueryUI - Spinner

Spinner widget فلش های بالا / پایین را به سمت چپ یک باکس ورودی اضافه می کند، بنابراین به یوزر اجازه می دهد تا یک مقدار را در باکس ورودی کم و زیاد کند. علاوه بر این این امکان را برای یوزر فراهم می کند تا یک مقدار را به طور مستقیم تایپ کند، یا اینکه یک مقدار موجود را با استفاده از ماوس، صفحه کلید و یا اسکرول تغییر دهد. این برنامه همچنین دارای مرحله ی برای رد کردن برخی مقادیر است. علاوه بر ویژگی های عددی پایه، جهانی کردن طراحی گزینه ها را نیز امکان پذیر می سازد (برای مثال ارز، مجزاکننده های مختلف، اعشار و غیره) بنابراین یک باکس مناسب بین المللی شده ارائه می دهد.

مثال زیر به **Globalize** بستگی دارد. شما می توانید فایل های جهانی شده را از <https://github.com/jquery/globalize> دریافت کنید. روی لینک های منتشر شده کلیک کنید، ورژن مورد نظر خود را انتخاب کنید و فایل **.zip** یا **tar.gz** را دانلود کنید. این فایل ها را باز کرده و فایل های زیر را روی فولدر مربوط به مثال، کپی کنید.

lib/globalize.js: این فایل حاوی کد **javascript** برای مواجه شدن با محلی کردن ها وجود دارد.

lib/globalize.culture.js: این فایل حاوی مجموعه ی کاملی از داخلی هایی است که کتابخانه ی **Globalize** با آن روبرو می شود.

این فایل ها در فولدر **external** از کتابخانه ی **jquery-ui** نیز وجود دارند.

jQueryUI یک متد **spinner()** ارائه می دهد که یک **spinner** ایجاد می کند.

ترکیب:

متد **spinner()** به دو فرم زیر قابل استفاده می باشد

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

`$(selector, context).spinner ("action", params) Method`

متد \$(selector, context).spinner (options)

متد **spinner (options)** اعلام می کند که یک عنصر **HTML** و محتوای آن باید مانند یک **spinner** مدیریت و برخورد شود. پارامتر **options** آبجکتی است که ظاهر و رفتار عناصر موجود در **spinner** را مشخص می کند.

ترکیب

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

شما می توانید با استفاده از آبجکت **Javascript**، یک یا چند گزینه را همزمان ارائه دهید. اگر بیشتر از یک گزینه وجود دارد، می توانید آنها را با استفاده از کاما از یکدیگر مجزا کنید، همانطور که در زیر مشاهده می کنید:

`$(selector, context).spinner({option1: value1, option2: value2.....});`

جدول زیر گزینه های مختلفی را لیست می کند که با این متد مورد استفاده قرار می گیرند:

| Option | Description |
|----------|---|
| culture | این گزینه culture را برای تجزیه و قالب بندی مقدار تنظیم می کند. مقدار پیش فرض آن null می باشد که به این معناست که culture تنظیم شده در Globalize استفاده می شود. |
| disabled | اگر این گزینه روی true تنظیم شده باشد، spinner را غیرفعال می کند. به طور پیش فرض مقدار آن false می باشد. |
| icons | این گزینه آیکن ها را برای استفاده ی دکمه ها تنظیم می کند، یک آیکن را که به وسیله ی jQuery UI CSS Framework ارائه شده، هماهنگ می کند. مقدار پیش فرض آن { down: "ui-icon-triangle-1-s", up: "ui-icon-triangle-1-n" } می باشد. |

| | |
|--------------|--|
| incremental | این گزینه تعداد مراحل را کنترل می کند که هنگام پایین نگاه داشتن دکمه ی spin مورد نیاز است. مقدار پیش فرض آن true می باشد. |
| max | این گزینه حداکثر مقدار مجاز را نشان می دهد. مقدار پیش فرض آن null است که به این معناست که هیچ حداکثر اجرایی وجود ندارد. |
| min | این گزینه حداقل مقدار مجاز را نشان می دهد. مقدار پیش فرض آن null است که به این معناست که هیچ حداقل اجرایی وجود ندارد. |
| numberFormat | این گزینه فرمت اعدادی را نشان می دهد که برای Globalize شدن منتقل شدند. متداول ترین ها عبارتند از n برای اعداد اعشاری و c برای ارزش پول. مقدار آن به طور پیش فرض null می باشد. |
| page | این گزینه تعداد مراحل مورد نیاز برای صفحه بندی از طریق متدهای pageUp/pageDown را نشان می دهد. مقدار آن به طور پیش فرض 10 است. |
| step | این گزینه اندازه ی مراحل مورد نیاز برای spinning از طریق دکمه ها یا از طریق متدهای thestepUp()/stepDown() را نشان می دهد. در صورت وجود، ویژگی عنصر step استفاده می شود و گزینه به طور مشخص تنظیم نشده است. |

در بخش بعدی چند مثال در مورد قابلیت **spinner widget** ارائه خواهد شد.

قابلیت پیش فرض

مثال زیر یک قابلیت **spinner widget** را توضیح می دهد که پارامتری به متد **spinner()** انتقال نمی دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Spinner functionality</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>
<!-- CSS -->
<style type="text/css">
  #spinner-1 input {
    width: 100px;
  }
</style>
<!-- Javascript -->
<script>
  $(function () {
    $("#spinner-1").spinner();
  });
</script>
</head>
<body>
  <!-- HTML -->
  <div id="example">
    <input type="text" id="spinner-1" value="0" />
  </div>
</body>
</html>

```

استفاده از گزینه های min، max و step

مثال زیر استفاده از سه گزینه ی min، max و step را در spinner widget از JQueryUI توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Spinner functionality</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>
  <!-- CSS -->
  <style type="text/css">
    #spinner-2, #spinner-3 input {
      width: 100px;
    }
  </style>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#spinner-2").spinner({
        min: -10,
        max: 10
      });
      $('#spinner-3').spinner({
        step: 100,
        min: -1000000,
        max: 1000000
      });
    });
  </script>

```

```

    });
  });
</script>
</head>
<body>
  <!-- HTML -->
  <div id="example">
    Spinner Min, Max value set:
    <input type="text" id="spinner-2" value="0" /><br><br>
    Spinner increments/decrements in step of 100:
    <input type="text" id="spinner-3" value="0" />
  </div>
</body>
</html>

```

در مثال بالا مشاهده می کنید که در اولین **spinner** مقادیر **max** و **min** به ترتیب بر روی 10 و 10 - تنظیم شده اند. از این رو با انتقال این مقادیر، **spinner** افزایش / کاهش را متوقف خواهد کرد. در **spinner** دوم، مقدار را در مرحله ی 100 افزایش / کاهش می دهد.

استفاده از گزینه ی **icons**

مثال زیر استفاده از گزینه ی **icons** را در **spinner widget** از **jQueryUI** توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Spinner functionality</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>
  <!-- CSS -->
  <style type="text/css">
    #spinner-5 input {
      width: 100px;
    }
  </style>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#spinner-5").spinner({
        icons: {
          down: "custom-down-icon", up: "custom-up-icon"
        }
      });
    });
  </script>

```

```

</head>
<body>
  <!-- HTML -->
  <div id="example">
    <input type="text" id="spinner-5" value="0" />
  </div>
</body>
</html>

```

در مثال بالا متوجه می شوید که **spinner** تصاویر تغییر یافته است.

استفاده از گزینه های **culture**، **numberFormat** و **page**

مثال زیر استفاده از سه گزینه ی **culture**، **numberFormat** و **page** را در **spinner widget** از **JqueryUI** توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Spinner functionality</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>
  <script src="/jqueryui/jquery-ui-1.10.4/external/jquery.mousewheel.js"></script>
  <script src="/jqueryui/jquery-ui-1.10.4/external/globalize.js"></script>
  <script src="/jqueryui/jquery-ui-1.10.4/external/globalize.culture.de-DE.js"></script>
  <script>
    $(function () {
      $("#spinner-4").spinner({
        culture: "de-DE",
        numberFormat: "C",
        step: 2,
        page: 10
      });
    });
  </script>
</head>
<body>
  <p>
    <label for="spinner-4">Amount to donate:</label>
    <input id="spinner-4" name="spinner" value="5">
  </p>
</body>
</html>

```


در مثال بالا مشاهده می کنید که **spinner** تعداد را در فرمت حاضر با عنوان **numberFormat** نمایش می دهد که روی **C** تنظیم شده و **culture** روی **deDE** تنظیم شده است. در اینجا از فایل های **Globalize** از کتابخانه **jquery-ui** استفاده کرده ایم.

متد `$(selector, context).spinner ("action", params)`

متد `spinner ("action", params)` می تواند فعالیتی را روی عناصر **spinner** اجرا کند، مانند فعال / غیرفعال کردن **spinner**. فعالیت به عنوان یک رشته در اولین **argument** تعیین می شود (به عنوان مثال گزینه ی **disable** در واقع **spinner** را غیر فعال می سازد). فعالیت هایی را که قابل انتقال هستند در جدول زیر بررسی کنید.

ترکیب

`$(selector, context).spinner ("action", params);`

جدول زیر فعالیت های مختلفی را نشان می دهد که با این متد استفاده می شود:

| Action | Description |
|----------------------|--|
| destroy | این فعالیت قابلیت spinner را از روی یک عنصر از بین می برد. عناصر به وضعیت pre-init خود بازمی گردند. این متد هیچ برهانی (argument) را نمی پذیرد. |
| disable | این عمل قابلیت spinner را غیرفعال می سازد. این متد هیچ برهانی را نمی پذیرد. |
| enable | این عمل قابلیت spinner را فعال می سازد. این متد هیچ برهانی را نمی پذیرد. |
| option(optionName) | این متد مقدار مربوط به optionName را دریافت می کند. در واقع optionName نام گزینه ی دریافتی می باشد. |

| | |
|-----------------------------|---|
| option | این عمل یک آبجکت حاوی جفت های key/value را دریافت می کند که گزینه های مربوط به spinner کنونی را نمایش می دهد. این متد هیچ برهانی را نمی پذیرد. |
| option(optionName, value) | این عمل مقدار گزینه ی spinner مربوط به optionName تعیین شده را تنظیم می کند. |
| option(options) | این عمل یک یا چند گزینه را برای spinner تنظیم می کند. |
| pageDown([pages]) | این عمل مقدار را با تعداد مشخص شده ی صفحات، کاهش می دهد، همانطور که در گزینه ی page تعریف شد. |
| pageUp([pages]) | این عمل مقدار را با تعداد مشخص شده ی صفحات، افزایش می دهد، همانطور که در گزینه ی page تعریف شد. |
| stepDown([steps]) | این عمل مقدار را به وسیله ی تعداد تعیین شده ی مراحل کاهش می دهد. |
| stepUp([steps]) | این عمل مقدار را به وسیله ی تعداد تعیین شده ی مراحل کاهش می دهد. |
| value | این عمل مقدار حاضر را به عنوان یک عدد دریافت می کند. این مقدار براساس گزینه های numberFormat و culture تجزیه می شود. این متد هیچ برهانی را نمی پذیرد. |
| value(value) | این عمل مقدار را تنظیم می کند، اگر مقدار انتقال یابد، براساس گزینه های numberFormat و culture تجزیه می شود. |
| widget | این عمل عنصر spinner widget را بازمی گرداند، عنصری که با نام گروه ui-spinner توضیح داده می شود. |

مثال زیر استفاده از فعالیت های جدول بالا را توضیح می دهد.

استفاده از فعالیت stepUp، stepDown، pageUp و pageDown

مثال زیر استفاده از متدهای **stepUp**، **stepDown**، **pageUp** و **pageDown** را توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Spinner functionality</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>
  <!-- CSS -->
  <style type="text/css">
    #spinner-6 input {
      width: 100px;
    }
  </style>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#spinner-6").spinner();
      $('#button').button();
      $('#stepUp-2').click(function () {
        $("#spinner-6").spinner("stepUp");
      });
      $('#stepDown-2').click(function () {
        $("#spinner-6").spinner("stepDown");
      });
      $('#pageUp-2').click(function () {
        $("#spinner-6").spinner("pageUp");
      });
      $('#pageDown-2').click(function () {
        $("#spinner-6").spinner("pageDown");
      });
    });
  </script>
</head>
<body>
  <!-- HTML -->
  <input id="spinner-6" />
  <br />
  <button id="stepUp-2">Increment</button>
  <button id="stepDown-2">Decrement</button>
  <button id="pageUp-2">Increment Page</button>
  <button id="pageDown-2">Decrement Page</button>
</body>
</html>
```

در مثال بالا از دکمه های مرتبط با افزایش / کاهش **spinner** استفاده شد.

استفاده از فعالیت های enable و disable

مثال زیر استفاده از متدهای **enable** و **disable** را توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Spinner functionality</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>
  <!-- CSS -->
  <style type="text/css">
    #spinner-7 input {
      width: 100px;
    }
  </style>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#spinner-7").spinner();
      $('#button').button();
      $('#stepUp-3').click(function () {
        $("#spinner-7").spinner("enable");
      });
      $('#stepDown-3').click(function () {
        $("#spinner-7").spinner("disable");
      });
    });
  </script>
</head>
<body>
  <!-- HTML -->
  <input id="spinner-7" />
  <br />
  <button id="stepUp-3">Enable</button>
  <button id="stepDown-3">Disable</button>
</body>
</html>
```

در مثال بالا از دکمه های **Enable/Disable** برای فعالسازی یا غیرفعال کردن **spinner** استفاده می کنیم.

مدیریت رویداد روی عناصر spinner

علاوه برمتد **spinner (options)** که در بخش های قبل 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 |
|-------------------|--|
| change(event, ui) | این رویداد وقتی آغاز می شود که مقدار spinner تغییر کرده و ورودی دیگر مد نظر نیست. |
| create(event, ui) | این رویداد وقتی آغاز می شود که spinner ایجاد شود. |
| spin(event, ui) | این رویداد در زمان افزایش / کاهش آغاز می شود. |
| start(event, ui) | این رویداد قبل از یک spin آغاز می شود. می تواند کنسل شود و از پیشروی spin جلوگیری کند. |
| stop(event, ui) | این رویداد پس از یک spin آغاز می شود. |

مثال:

مثال زیر متد استفاده شده در **spinner widget** را توضیح می دهد. این مثال استفاده از رویدادهای

change, **spin** و **stop** را توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Spinner functionality</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>
  <!-- CSS -->
  <style type="text/css">
    #spinner-8 input {
      width: 100px;
    }
  </style>
  <!-- Javascript -->
  <script>

```

```

$(function () {
  $("#spinner-8").spinner({
    spin: function (event, ui) {
      var result = $("#result-2");
      result.append("Spin Value: " + $("#spinner-8").spinner("value"));
    },
    change: function (event, ui) {
      var result = $("#result-2");
      result.append("Change value: " + $("#spinner-8").spinner("value"));
    },
    stop: function (event, ui) {
      var result = $("#result-2");
      result.append("Stop value: " + $("#spinner-8").spinner("value"));
    }
  });
});
</script>
</head>
<body>
<!-- HTML -->
<div id="example">
  <input type="text" id="spinner-8" value="0" />
</div>
<span id="result-2"></span>
</body>
</html>

```

در مثال بالا مقدار **spinner** را تغییر دهید و پیغامی که در زیر برای چرخش و توقف رویدادها نمایش داده می شود را مشاهده کنید. اکنون تمرکز **spinner** را تغییر داده و سپس پیغامی را روی رویداد تغییر مشاهده خواهید کرد.

Extension Points

Spinner widget با **widget factory** ساخته می شود و می تواند گسترده شود. برای گسترش ویجت ها می توانیم یا برخی رفتارهای متدهای موجود را نادیده بگیریم و یا اینکه به آنها مواردی اضافه کنیم. جدول زیر **extension point** را با ثبات **API** و با همان روش **spinner** ذکر شده در جدول فوق، ارائه می دهد.

| Method | Description |
|--------|-------------|
|--------|-------------|

| | |
|------------------------------------|--|
| <code>_buttonHtml(event)</code> | این متد یک String را بازمی گرداند که یک HTML می باشد. این HTML می تواند برای دکمه های افزایش و کاهش spinner استفاده شود. هر دکمه باید دارای یک گروه ui-spinner-button باشد که برای رویدادهای مرتبط با کار نامگذاری می شود. این متد هر برهانی را نمی پذیرد. |
| <code>_uiSpinnerHtml(event)</code> | این متد HTML را برای استفاده در پوشاندن عنصر <code><input></code> مربوط به spinner، تعیین می کند. این متد هر برهانی (argument) را نمی پذیرد. |

jQuery UI Widget Factory یک اساس توسعه پذیر می باشد که روی آن همه ی ویجت های **jQuery UI** ساخته می شوند. استفاده از **widget factory** برای ساختن یک پلاگین، سهولت هایی برای مدیریت وضعیت و همچنین کنوانسیون هایی برای کارهای معمولی سازد، از جمله نمایش دادن متدهای پلاگین و تغییر گزینه ها پس از نمونه.

JqueryUI - Tooltip

ویجت **tooltip** در **jQueryUI** ابزار داخلی را جایگزین می کند. این ویجت زمینه های جدید اضافه کرده و اجازه ی تنظیم آن را می دهد. ابتدا اجازه بدهید **tooltip** ها را تعریف کرده و درک کنیم. **Tooltip** ها به هر عنصری ضمیمه می شوند. برای نمایش **tooltip** ها یک تیتلر مرتبط به عناصر ورودی و یک تیتلر در ارتباط با مقداری که قرار است به عنوان **tooltip** استفاده شود، اضافه کنید. وقتی که با ماوس خود روی **tooltip** حرکت می کنید، تیتلر در یک باکس کوچک در کنار عنصر نمایش داده می شود.

jQueryUI متد **tooltip()** را برای افزودن **tooltip** به هر عنصر ارائه می دهد. این متد به طور پیش فرض یک انیمیشن محو ارائه می دهد تا **tooltip** را نمایش داده و پنهان کند.

ترکیب:

متد به دو فرم زیر قابل استفاده می باشد

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

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

متد `$(selector, context).tooltip (options)`

متد `tooltip (options)` اعلام می کند که یک `tooltip` می تواند به یک عنصر `HTML` اضافه شود. پارامتر `options` آبجکتی است که رفتار و ظاهر `tooltip` را تعیین می کند.

ترکیب:

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

با استفاده از آبجکت `javascript` می توانید یک یا چند گزینه را همزمان ارائه دهید. اگر بیشتر از یک گزینه وجود داشته باشد، می توانید آنها را با استفاده از کاما از یکدیگر جدا کنید، مانند ترکیب زیر:

`$(selector, context).tooltip({option1: value1, option2: value2..... });`

جدول زیر گزینه های مختلفی را ارائه می دهد که با این متد استفاده می شوند:

| Option | Description |
|---------|---|
| content | این گزینه محتوای یک <code>tooltip</code> را نمایش می دهد. مقدار پیش فرض آن <code>function returning the title attribute</code> می باشد. |

| | |
|--------------|---|
| disabled | این گزینه وقتی روی true تنظیم شده باشد، tooltip را غیرفعال می سازد. مقدار پیش فرض آن false است. |
| hide | این گزینه افکت انیمیشن را در زمان پنهان کردن tooltip نمایش می دهد. مقدار پیش فرض آن true است. |
| items | این گزینه نشان می دهد که کدام آیتم ها می توانند tooltip ها را نمایش دهند. مقدار پیش فرض آن [title] است. |
| position | این گزینه موقعیت tooltip w.r.t عنصر مرتبط به هدف، را نشان می دهد. مقدار پیش فرض آن function returning the title attribute از my, at, of, مقادیر ممکن عبارتند از collision, using, within. |
| show | این گزینه نشان می دهد که نمایش tooltip را انجام دهیم. مقدار پیش فرض آن true است. |
| tooltipClass | این گزینه گروهی است که به tooltip widget اضافه می شود، برای ابزاری مانند هشدارها یا خطاها. مقدار پیش فرض آن null است. |
| track | این گزینه وقتی روی true تنظیم شده باشد، tooltip ماوس را حرکت می دهد / دنبال می کند. مقدار پیش فرض آن false است. |

بخش بعدی چند مثال کاربردی برای کارایی tooltip ارائه خواهد داد.

قابلیت پیش فرض

مثال زیر قابلیت tooltip را توضیح می دهد که هیچ پارامتری به متد tooltip() انتقال نمی دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Tooltip functionality</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>
  <!-- Javascript -->
  <script>
    $(function () {
  
```

```

    $("#tooltip-1").tooltip();
    $("#tooltip-2").tooltip();
  });
</script>
</head>
<body>
  <!-- HTML -->
  <label for="name">Name:</label>
  <input id="tooltip-1" title="Enter You name">
  <p>
    <a id="tooltip-2" href="#" title="Nice tooltip">
      I also have a tooltip
    </a>
  </p>
</body>
</html>

```

در مقال بالا روی لینک های بالا حرکت کنید یا از کلید تب استفاده کنید تا تمرکز را روی هر کدام از عناصر قرار دهید.

استفاده از `content`، `track` و `disabled`

مثال زیر استفاده از سه گزینه ی مهم `(a) content (b) track` و `(c) disabled` را در قابلیت `tooltip` از **jQueryUI** توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Tooltip functionality</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>

  <!-- Javascript -->
  <script>
    $(function () {
      $("#tooltip-3").tooltip({
        content: "<strong>Hi!</strong>",
        track: true
      }),
      $("#tooltip-4").tooltip({
        disabled: true
      });
    });
  </script>
</head>
<body>
  <!-- HTML -->
  <label for="name">Message:</label>
  <input id="tooltip-3" title="tooltip"><br><br><br>

```

```

<label for="name">Tooltip disabled for me:</label>
<input id="tooltip-4" title="tooltip">
</body>
</html>

```

در مثال بالا محتوای **tooltip** از اولین باکس بر روی استفاده از گزینه ی **content** تنظیم شده است. متوجه می شوید که **tooltip** ماوس را دنبال می کند. **Tooltip** مربوط به باکس دوم غیرفعال می باشد.

استفاده از position

مثال زیر استفاده از گزینه ی **position** را در قابلیت **tooltip** از **JqueryUI** نشان می دهد.

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Tooltip functionality</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>
<!-- CSS -->
<style>
body {
margin-top: 100px;
}
.ui-tooltip-content::after, .ui-tooltip-content::before {
content: "";
position: absolute;
border-style: solid;
display: block;
left: 90px;
}
.ui-tooltip-content::before {
bottom: -10px;
border-color: #AAA transparent;
border-width: 10px 10px 0;
}
.ui-tooltip-content::after {
bottom: -7px;
border-color: white transparent;
border-width: 10px 10px 0;
}
</style>
<!-- Javascript -->
<script>
$(function () {
$("#tooltip-7").tooltip({
position: {
my: "center bottom",
at: "center top-10",
collision: "none"
}
}
}
)

```

```

});
});
</script>
</head>
<body>
<!-- HTML -->
<label for="name">Enter Date of Birth:</label>
<input id="tooltip-7" title="Please use MM.DD.YY format.">
</body>
</html>

```

در مثال بالا موقعیت **tooltip** بر روی بالای باکس ورودی تنظیم شده است.

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

متد `tooltip (action, params)` می تواند عملی را روی عناصر **tooltip** انجام دهد، مانند غیرفعال کردن **Action.tooltip** به عنوان یک رشته در اولین برهان تعیین شده و یک یا چند **params** براساس عمل داده شده، ارائه می شوند.

اساسا در اینجا این اکشن ها متدهای **jQuery** می باشند که به فرم **string** استفاده می شوند.

ترکیب:

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

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

| Action | Description |
|-----------|--|
| close() | این اکشن tooltip را می بندد. این متد هیچ برهانی (argument) قبول نمی کند. |
| destroy() | این اکشن قابلیت tooltip را به طور کل حذف می کند و عنصر را به وضعیت pre-init خود بازخواهد گرداند. این اکشن هیچ برهانی را نمی پذیرد. |
| disable() | این اکشن tooltip را غیرفعال می سازد. این متد هیچ برهانی را نمی پذیرد. |
| enable() | این اکشن tooltip را فعالسازی می کند. این متد هیچ برهانی را نمی پذیرد. |

| | |
|-----------------------------|---|
| open() | این اکشن به طور برنامه ریزی شده ای tooltip را باز می کند. این متد هیچ برهانی را نمی پذیرد. |
| option(optionName) | این اکشن مقدار مربوط به optionName را برای tooltip دریافت می کند. این متد هیچ برهانی را نمی پذیرد. |
| option() | این اکشن یک آبجکت حاوی جفت های key/value را دریافت می کند، که گزینه های فعلی tooltip را hash نمایش می دهد. این متد هیچ برهانی را نمی پذیرد. |
| option(optionName, value) | این اکشن مقدار گزینه ی tooltip مرتبط با optionName تعیین شده را تنظیم می کند. |
| option(options) | این اکشن یک یا چند گزینه برای tooltip تنظیم می کند. |
| widget() | این اکشن یک آبجکت jQuery حاوی عنصر اصلی را بازمی گرداند. این متد هیچ برهانی نمی پذیرد. |

مثال:

اکنون اجازه بدهید از اکشن های جدول بالا استفاده کنیم. مثال زیر استفاده از اکشن های **enable** و **disable** را توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Tooltip functionality</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>
  <!-- Javascript -->
  <script>
    $(function () {
      $("#tooltip-8").tooltip({
        //use 'of' to link the tooltip to your specified input
        position: { of: '#myInput', my: 'left center', at: 'left center' },
      }),
      $('#myBtn').click(function () {
        $('#tooltip-8').tooltip("open");
      });
    });
  </script>
</head>
<body style="padding:100px;">

```

```

<!-- HTML -->
<a id="tooltip-8" title="Message" href="#"></a>
<input id="myInput" type="text" name="myInput" value="0" size="7" />
<input id="myBtn" type="submit" name="myBtn" value="myBtn" class="myBtn" />
</body>
</html>

```

در مثال بالا روی دکمه ی **myBtn** کلیک کنید که یک **tooltip** برای شما نمایش داده خواهد شد.

مدیریت رویداد روی عناصر tooltip

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

| Event Method | Description |
|-------------------|--|
| create(event, ui) | زمانیکه یک tooltip ایجاد می شود، این متد آغاز به کار می کند. درحالیکه رویداد از نوع Event می باشد، ui از نوع object است. |
| close(event, ui) | با بسته شدن tooltip فعال می شود. معمولا روی focusout یا mouseleave آغاز به کار می کند. درحالیکه رویداد از نوع Event می باشد، ui از نوع object است. |
| open(event, ui) | با نمایش یا اجرای tooltip آغاز به کار می کند. معمولا روی focusin یا mouseover آغاز می شود. درحالیکه رویداد از نوع Event می باشد، ui از نوع object است. |

مثال:

مثال زیر استفاده از یک متد را در طول دوره ی قابلیت **tooltip** توضیح می دهد. این مثال در واقع استفاده از رویدادهای **open** و **close** را توضیح می دهد.

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI Tooltip functionality</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>
<!-- Javascript -->

```

```

<script>
$(function () {
  $('.tooltip-9').tooltip({
    items: 'a.tooltip-9',
    content: 'Hello welcome...',
    show: "slideDown", // show immediately
    open: function (event, ui) {
      ui.tooltip.hover(
        function () {
          $(this).fadeTo("slow", 0.5);
        }
      );
    }
  });
});
$(function () {
  $('.tooltip-10').tooltip({
    items: 'a.tooltip-10',
    content: 'Welcome to Tutorialspoint...',
    show: "fold",
    close: function (event, ui) {
      ui.tooltip.hover(function () {
        $(this).stop(true).fadeTo(500, 1);
      },
        function () {
          $(this).fadeOut('500', function () {
            $(this).remove();
          });
        }
      );
    }
  });
});
});
</script>
</head>
<body style="padding:100px;">
<!-- HTML -->
<div id="target">
  <a href="#" class="tooltip-9">Hover over me!</a>
  <a href="#" class="tooltip-10">Hover over me too!</a>
</div>
</body>
</html>

```

در مثال بالا tooltip برای **Hover over me!** فوراً ناپدید می‌شود، در حالیکه tooltip برای **Hover over me** **too!** پس از مدت زمان 1000ms محو می‌شود.

JqueryUI - Position

در این فصل به بررسی یکی از روش ابزارهای **jqueryUi** به نام متد **position()** خواهیم پرداخت. یک متد **position()** به شما اجازه می‌دهد تا موقعیت یک عنصر را با توجه به یک عنصر دیگر یا رویداد ماوس، تعیین کنید.

jQuery UI متد **position()** را از مرکز **jQuery** به نحوی گسترش می دهد که به شما اجازه می دهد که تا چگونگی موقعیت گذاری یک عنصر را به نحو دلخواه خود، توصیف کنید، درست همانطور که یک شخص را توصیف می کنید. به جای کار با اعداد، با لغات معنادار (از قبیل چپ و راست) و روابط کار می کنید.

در زیر دستور مربوط به متد **position()** را مشاهده می کنید:

position(options)

در اینجا **options** از نوع **Object** می باشد و اطلاعاتی را ارائه می دهد که تعیین می کنند چگونه عناصر قرار گرفته در یک مجموعه در موقعیت خود قرار بگیرند. جدول زیر گزینه های مختلفی را ارائه می دهد که با این متد مورد استفاده قرار می گیرند:

| Option | Description |
|-----------|---|
| my | این گزینه موقعیت عناصر در هم پیچیده را تعیین می کند (آنهایی که مجددا موقعیت گذاری شده اند) تا با عنصر مورد نظر یا موقعیت همتراز شود. مقدار پیش فرض آن center است. |
| at | این گزینه از نوع String است و موقعیت عنصر هدف را در مقابل عناصری که مجددا موقعیت گذاری می شوند، تعیین می کند. مقادیر گزینه ی my را می گیرد. مقدار پیش فرض آن center است. |
| of | این پارامتر از نوع Selector یا Element یا jQuery یا Event است. موقعیت عنصر هدف را در مقابل عناصری که مجددا موقعیت گذاری می شوند، و یا یک نمونه Event که حاوی مختصات ماوس برای استفاده به عنوان موقعیت هدف است را تشخیص می دهد. مقدار پیش فرض آن null است. |
| collision | این گزینه از نوع String است و قوانین مربوط به گسترش عنصر ورای پنجره در هر راستایی، تعیین می کند. مقدار پیش فرض آن flip است. |
| using | این گزینه ساختاری است که عملکرد داخلی را، که موقعیت عنصر را تغییر می دهد، جایگزین می نماید. برای هر عنصر در هم پیچیده ای با یک آبجکت hash و پراپرتی های left و top فرا خوانده می شود، که این پراپرتی |

| | |
|--------|---|
| | ها بر روی موقعیت محاسبه ی شده ی هدف تنظیم شده اند و عنصر به عنوان کانتکست عملکرد تنظیم شده است. مقدار پیش فرض آن null است. |
| within | این گزینه یک عنصر Selector یا Element یا jQuery می باشد و به شما اجازه می دهد تا تعیین کنید کدام عنصر به عنوان bounding box برای تشخیص برخورد، استفاده شود. این امر در صورتی که نیاز باشد عنصر در یک بخش خاص از صفحه ی شما قرار بگیرد، مفید می باشد. مقدار پیش فرض آن window است. |

مثال:

مثال زیر استفاده از متد position را توضیح می دهد:

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI position method 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>
<!-- CSS -->
<style>
.positionDiv {
    position: absolute;
    width: 75px;
    height: 75px;
    background: #b9cd6d;
}
#targetElement {
    width: 300px;
    height: 500px;
    padding-top: 50px;
}
</style>
<script>
$(function () {
    // Position the dialog offscreen to the left, but centered vertically
    $("#position1").position({
        my: "center",
        at: "center",
        of: "#targetElement"
    });
    $("#position2").position({
        my: "left top",
        at: "left top",
```

```

of: "#targetElement"
});
$("#position3").position({
  my: "right-10 top+10",
  at: "right top",
  of: "#targetElement"
});
$(document).mousemove(function (event) {
  $("#position4").position({
    my: "left+3 bottom-3",
    of: event,
    collision: "fit"
  });
});
});
</script>
</head>
<body>
<div id="targetElement">
  <div class="positionDiv" id="position1">Box 1</div>
  <div class="positionDiv" id="position2">Box 2</div>
  <div class="positionDiv" id="position3">Box 3</div>
  <div class="positionDiv" id="position4">Box 4</div>
</div>
</body>
</html>

```

در این مثال مشاهده می کنیم که:

Box 1 در مرکز عنصر **div** ردیف شده (به صورت افقی و عمودی).

Box 2 در بالای سمت چپ عنصر **div** ردیف شده است (به صورت افقی و عمودی).

Box 3 در گوشه ی سمت راست و در بالای پنجره نمایش داده می شود، اما برخی لایه ها را رها می کند، طوری که پیغام بیشتر به بیرون رانده می شود. این کار با استفاده از مقادیر افقی و عمودی **my** یا **at** انجام می شود.

برای **Box 4** مقدار **of** به عنوان یک آبجکت رویداد تنظیم شده است. این یک رویداد است که با نشانگر در ارتباط است و با رویداد ماوس در ارتباط می باشد.

JqueryUI - Widget Factory

قبلاً تنها راه برای نوشتن کنترل های سفارشی (**custom controls**) در **jQuery** گسترش فضا نام **\$.fn** بود. این کار برای ویجت های ساده موثر بود. تصور کنید که ویجت های دارای وضعیت بیشتری ساخته اید که خیلی

سریع سنگین می شود. برای کمک در فرایند ساخت ویجت ها، **Widget Factory** در **jQuery UI** معرفی شد که بسیاری از **boilerplate** ها را حذف می کند که به ویژه با تنظیم یک ویجت در ارتباط هستند.

یک **jQueryUI Widget Factory** تابعی است که نام یک رشته و یک آجکت را به عنوان برهان ها، خود گرفته و یک **jQuery plugin** و یک **Class** ایجاد می کند تا قابلیت آن را در هم بپیچد.

ترکیب

در زیر ترکیب مربوط به متد **jQueryUI Widget Factory** را مشاهده می کنید:

```
jQuery.widget( name [, base ], prototype )
```

name: رشته ای است که حاوی یک **namespace** و **widget name** از ویجت برای ایجاد می باشد.

base: ویجت پایه برای دریافت فرم و باید یک سازنده باشد که می تواند با لغت کلیدی **new** نمونه گذاری شود. پیش فرض برای **jQuery.Widget**.

prototype: یک آجکت برای اینکه به عنوان یک نمونه ی اولیه برای ویجت برای دریافت فرم، استفاده می شود. برای نمونه **jQuery UI** دارای یک پلاگین **"mouse"** است که روی آن بقیه ی تعاملات پلاگین ها بر اساس این مورد می باشند، که برای به دست آوردن **draggable**, **droppable** و غیره می باشد. تمام مواردی که از پلاگین های ماوس دریافت می کنند: **(jQuery.widget("ui.draggable", \$.ui.mouse, {...})**؛ اگر شما این آرگومان را تامین نکنید، ویجت مستقیماً از **jQuery.Widget**، **"base widget"** دریافت می کند. (به تفاوت بین **widget** با **w** کوچک و **Widget** با **W** دقت داشته باشید.)

Base Widget

Base widget ویجتی است که به وسیله ی **widget factory** استفاده می شود.

گزینه ها

جدول زیر گزینه های مختلفی را ارائه می دهد که با **Base widget** استفاده می شوند.

| Option | Description |
|--------------|---|
| disabledhide | این گزینه اگر روی true تنظیم شده باشد، ویجت را غیرفعال می سازد. مقدار پیش فرض آن false است. |
| hide | این گزینه چگونگی محرک سازی پنهان کردن عنصر را تعیین می کند. مقدار پیش فرض آن null است. |
| show | این گزینه چگونگی محرک سازی نمایش عنصر را تعیین می کند. مقدار پیش فرض آن null است. |

متدها

جدول زیر متدهای مختلفی را ارائه می دهد که با **base widget** مورد استفاده قرار می گیرند:

| Action | Description |
|--------------------------------------|--|
| <code>_create()</code> | این متد سازنده ی widget است. هیچ پارامتری وجود ندارد، اما <code>this.element</code> and <code>this.options</code> تقریباً تنظیم شده اند. |
| <code>_delay(fn [, delay])</code> | این متد تابع ارائه شده را پس از تاخیر تعیین شده، تحریک می کند. <code>timeout ID</code> را برای استفاده با <code>clearTimeout()</code> باز می گرداند. |
| <code>_destroy()</code> | متد عمومی <code>destroy()</code> که همه ی داده های مشترک، رویدادها و غیره را پاک می کند و به این واگذار می کند – متد <code>destroy()</code> برای <code>custom, widget-specific</code> <code>cleanup</code> . |

| | |
|--|---|
| <code>_focusable(element)</code> | این متد عنصر را برای به کار گرفتن گروه <code>theui-state-focus</code> تنظیم می کند. گرداننده های رویداد به طور خودکار پاک می شوند. |
| <code>_getCreateEventData()</code> | همه ی ویجت ها رویداد <code>creat</code> را آغاز می کنند. به طور پیش فرض هیچ داده ای در رویداد ارائه نمی شود. اما این متد می تواند یک آبجکتی را بازگرداند که به عنوان داده ی رویداد <code>creat</code> منتقل خواهد شد. |
| <code>_getCreateOptions()</code> | این متد به ویجت اجازه می دهد تا یک متد <code>custom</code> برای گزینه های تعریف شده در دوره ی نمونه گذاری، تعیین کند. گزینه های ارائه شده توسط یوزر، گزینه های مربوط به این متد را رد می کنند که گزینه های پیش فرض را نیز رد می کنند. |
| <code>_hide(element, option [, callback])</code> | این متد یا با استفاده از متدهای داخلی انیمیشن یا با استفاده از افکت های <code>custom</code> ، یک عنصر را فوراً پنهان می کند. گزینه ی <code>hide</code> را برای مقادیر ممکن برای گزینه بررسی کنید. |
| <code>_hoverable(element)</code> | این متد عنصری را تنظیم می کند برای به کار گرفتن گروه <code>ui-state-hover</code> روی <code>hover</code> . این گرداننده های رویداد به طور خودکار تا از بین رفتن تمیز می شوند. |
| <code>_init()</code> | هر زمانی که <code>plugin</code> بدون هیچ آرگیومنتی یا تنها با یک گزینه ی <code>hash</code> فرا خوانده می شود، ویجت مقدار دهی می شود. این شامل زمانی است که ویجت ایجاد می شود. |
| <code>_off(element, eventName)</code> | این متد گرداننده ی رویداد را از عناصر تعیین شده رها می کند. |
| <code>_on([suppressDisabledCheck] [, element], handlers)</code> | گرداننده های رویداد را در عناصر مخصوصی محدود می کند. انتقال از طریق انتخابگرهای داخل نام های مربوط به رویداد پشتیبانی می شود، به عنوان مثال <code>"click.foo"</code> . |

| | |
|---|---|
| <code>_setOption(key, value)</code> | این متد از متد <code>_setOptions()</code> برای هر گزینه ی مجزایی، فرا خوانده می شود. وضعیت ویجت باید براساس تغییرات آپدیت شود. |
| <code>_setOptions(options)</code> | زمانی که متد <code>option()</code> فراخوانده می شود، این متد نیز فرا خوانده می شود، بدون توجه به فرمی که متد <code>option()</code> از آن فرا خوانده شد. |
| <code>_show(element, option [, callback])</code> | با استفاده از متدهای انیمیشن داخلی یا با استفاده از افکت های <code>custom</code> ، یک عنصر را فوراً نشان می دهد. گزینه <code>show</code> را برای مقادیر ممکن گزینه بررسی کنید. |
| <code>_super([arg] [, ...])</code> | این متد، یک متد هم نام را از ویجت <code>parent</code> ، با آرگومان های تعیین شده، تحریک می کند. به ویژه <code>call()</code> . |
| <code>_superApply(arguments)</code> | متد هم نام را از ویجت <code>parent</code> با ردیفی از آرگومان ها، تحریک می کند. |
| <code>_trigger(type [, event] [, data])</code> | این متد یک رویداد و کال بک مربوط به آن را آغاز می کند. گزینه ای با نامی به همان مقدار برای تایپ، به عنوان یک کال بک تحریک می شود. |
| <code>destroy()</code> | این متد قابلیت ویجت را به طور کل حذف می کند. این امر عنصر را به وضعیت <code>pre-init</code> خود باز خواهد گرداند. |
| <code>disable()</code> | این متد ویجت را غیرفعال می کند. |
| <code>enable()</code> | این متد ویجت را فعال می کند. |
| <code>option(optionName)</code> | این متد مقدار کنونی مربوط به <code>optionName</code> مشخص شده را می گیرد. |
| <code>option()</code> | این متد آبجکتی حاوی جفت های <code>key/value</code> را دریافت می کند که گزینه های <code>hash</code> کنونی مربوط به ویجت را نشان می دهند. |

| | |
|-----------------------------|---|
| option(optionName, value) | این متد مقدار مربوط به گزینه ی widget که با optionName تعیین شده در ارتباط است، تنظیم می کند. |
| option(options) | این متد یک یا چند گزینه را برای ویجت تنظیم می کند. |
| widget() | این گزینه یک آبجکت jQuery را باز می گرداند که حاوی عنصر اصلی یا دیگر عناصر تولید شده مرتبط می باشد. |

Events

| Event Method | Description |
|---------------------|--|
| create(event, ui) | این رویداد با ایجاد یک ویجت آغاز می شود. |

jQueryUI widget factory یک روش آبجکت محور برای مدیریت طول دوره ی (lifecycle) یک ویجت، ارائه

می دهد. این فعالیت ها عبارتند از:

ایجاد و یا از بین بردن ویجت، به عنوان مثال:

```
$( "#elem" ).progressbar();
```

تغییر گزینه های ویجت، به عنوان مثال:

```
$( "#elem" ).progressbar({ value: 20 });
```

فراخوانی های super در ویجت های زیر مجموعه، برای مثال:

```
$( "#elem" ).progressbar( "value" );
```

Or

```
$( "#elem" ).progressbar( "value", 40 );
```

اطلاعیه های رویداد، برای مثال:

```
$( "#elem" ).bind( "progressbarchange", function() {
```

```
    alert( "The value has changed!" );  
});
```

مثال:

اکنون اجازه بدهید در مثال زیر یک **custom widget** ایجاد کنیم. ما یک دکمه ی **widget** ایجاد خواهیم کرد. چگونگی ایجاد گزینه ها، متدها و رویدادها را در یک ویجت در مثال های زیر مشاهده خواهیم کرد:

اجازه بدهید یک **custom widget** ساده ایجاد کنیم:

```
<!doctype html>  
<head>  
  <meta charset="utf-8">  
  <title>jQuery UI Widget - Default functionality</title>  
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">  
  <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>  
</script>  
  $(function () {  
    $.widget("iP.myButton", {  
      _create: function () {  
        this._button = $("<button>");  
        this._button.text("My first Widget Button");  
        this._button.width(this.options.width)  
        this._button.css("background-color", this.options.color);  
        this._button.css("position", "absolute");  
        this._button.css("left", "100px");  
        $(this.element).append(this._button);  
      },  
    });  
    $("#button1").myButton();  
  });  
</script>  
</head>  
<body>  
  <div id="button1"></div>  
</body>  
</html>
```

افزودن گزینه ها به custom widget

در مثال قبل از تابع **_create** برای ایجاد یک **custom control** استفاده کردیم. اما یوزرها به طور کل تمایل دارند که با تنظیم و اصلاح گزینه ها، کنترل را سفارشی سازند. ما می توانیم یک آبجکت **options** تعریف کنیم که مقادیر پیش فرض را برای همه ی گزینه هایی که شما تعریف کردید، تعریف می کند. تابع **_setOption**

برای این هدف استفاده می شود. این تابع برای گزینه های مجزایی که یوزرها تنظیم کرده اند، فرا خوانده می شود. در مورد زیر دکمه های **width** و **background-color** را تنظیم می کنیم.

```
<!doctype html>
<head>
  <meta charset="utf-8">
  <title>jQuery UI Widget - Default functionality</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <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>
  <script>
    $(function () {
      $.widget("iP.myButton", {
        _create: function () {
          this._button = $("<button>");
          this._button.text("My first Widget Button");
          this._button.width(this.options.width);
          this._button.css("background-color", this.options.color);
          this._button.css("position", "absolute");
          this._button.css("left", "100px");
          $(this.element).append(this._button);
        },
        _setOption: function (key, value) {
          switch (key) {
            case "width":
              this._button.width(value);
              break;
            case "color":
              this._button.css("background-color", value);
              break;
          }
        },
      });
      $("#button2").myButton();
      $("#button2").myButton("option", { width: 100, color: "#cedc98" });
    });
  </script>
</head>
<body>
  <div id="button2"></div>
</body>
</html>
```

افزودن متدها به custom widget

در مثال زیر متدهایی را اضافه خواهیم کرد که یوزر می تواند مورد استفاده قرار دهد و برای ساخت در یک چارچوب بسیار ساده هستند. ما یک متد **Move** خواهیم نوشت که دکمه را در یک فاصله ی افقی مشخص،

تغییر می دهد. برای انجام این کار لازم است موقعیت و پراپرتی های سمت چپ را در تابع **create** _ تعیین کنیم:

```
this._button.css("position", "absolute");
```

```
this._button.css("left", "100px");
```

در ادامه ی آن، اکنون یوزر می تواند متد شما را به روش معمول **jQuery UI** فرا بخواند:

```
this._button.css("position", "absolute");
```

```
this._button.css("left", "100px");
```

```
$("#button3").myButton("move", 200);
```

```
<!doctype html>
<head>
  <meta charset="utf-8">
  <title>jQuery UI Widget - Default functionality</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <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>
  <script>
    $(function () {
      $.widget("iP.myButton", {
        _create: function () {
          this._button = $("<button>");
          this._button.text("My first Widget Button");
          this._button.width(this.options.width)
          this._button.css("background-color", this.options.color);
          this._button.css("position", "absolute");
          this._button.css("left", "100px");
          $(this.element).append(this._button);
        },

        move: function (dx) {
          var x = dx + parseInt(this._button.css("left"));
          this._button.css("left", x);
          if (x > 400) { this._trigger("outbounds", {}, { position: x }); }
        }
      });
      $("#button3").myButton();
      $("#button3").myButton("move", 200);
    });
  </script>
</head>
<body>
  <div id="button3"></div>
</body>
</html>
```

افزودن رویدادها به Custom Widget

در این مثال چگونگی ایجاد یک رویداد را توضیح خواهیم داد. برای ایجاد یک رویداد تمام کاری که باید انجام دهید، استفاده از متد **trigger** است. اولین پارامتر نام رویداد و دومین پارامتر آبجکتی است که می خواهید انتقال دهید و سومین پارامتر نیز هر آبجکت **custom event** که می خواهید انتقال دهید، می باشد. در اینجا در حال اجرای رویدادی هستیم که هنگامی اتفاق می افتد که دکمه بیشتر از **x=400** جابه جا می شود. تمام آنچه باید انجام دهید، افزودن به تابع جابه جایی می باشد:

```
if(x<400){ this._trigger("outbounds",{},{position:x});}
```

در این مورد رویداد **outbound** نامیده می شود و یک آبجکت رویداد خالی با یک آبجکت **custom event** منتقل می شود که به سادگی موقعیت را به عنوان تنها پراپرتی خود ارائه می دهد.

کل تابع **move** عبارت است از:

```
move: function(dx) {  
    var x = dx+parseInt(this._button.css("left"));  
    this._button.css("left", x);  
    if(x<400){ this._trigger("outbounds",{  
        {position:x}});  
    }  
}
```

یوزر می تواند عملکرد رسیدگی به رویداد را به سادگی و با تعریف یک گزینه ی هم نام، تنظیم کند.

```
$("#button4").myButton("option",  
    {width: 100,  
    color: "red",  
    outbounds:function(e,ui){  
        alert(ui.position);  
    }});  
<!doctype html>  
<head>  
    <meta charset="utf-8">  
    <title>jQuery UI Widget - Default functionality</title>  
    <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">  
    <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>  
</script>  
$(function () {  
    $.widget("iP.myButton", {  
        _create: function () {  
            this._button = $("<button>");  
            this._button.text("My first Widget Button");  
            this._button.width(this.options.width)  
            this._button.css("background-color", this.options.color);  
        }  
    });  
});
```

```

        this._button.css("position", "absolute");
        this._button.css("left", "100px");
        $(this.element).append(this._button);
    },
    move: function (dx) {
        var x = dx + parseInt(this._button.css("left"));
        this._button.css("left", x);
        if (x > 400) { this._trigger("outbounds", {}, { position: x }); }
    }
});
$("#button4").myButton();
$("#button4").on("mybuttonoutbounds", function (e, ui) {
    alert("out");
});
$("#button4").myButton("move", 500);
});
</script>
</head>
<body>
<div id="button4"></div>
</body>
</html>

```

JqueryUI - Add Class

این بخش متد **addClass()** را توضیح می دهد که یکی از متدهایی است که برای مدیریت افکت های بصری **jQueryUI** استفاده می شود. متد **addClass()** امکان متحرک سازی تغییرات را برای پراپرتی های **CSS** فراهم می کند.

متد **addClass()** گروه های خاصی را به عناصر هماهنگ شده اضافه می کند، در حالیکه همه ی تغییرات استایل را متحرک سازی می کند.

متد addClass() به شکل زیر می باشد

addClass(className [, duration] [, easing] [, complete])

| Parameter | Description |
|-----------|--|
| className | رشته ای حاوی یک یا چند گروه CSS است (به وسیله ی فاصله از یکدیگر مجزا شده اند.) |
| duration | از نوع Number یا String می باشد و تعداد ثانیه های افکت را نشان می دهد. مقدار 0 مستقیما و بدون پیشرفت، عنصر را وارد استایل جدید می کند. مقدار پیش فرض آن 400 است. |
| easing | از نوع String می باشد و رویه ی پیشروی در افکت را نشان می دهد. مقدار پیش فرض آن swing است. |
| complete | یک متد کال بک است که زمانیکه افکت برای یک عنصر کامل شود، برای آن عنصر فراخوانده می شود. |

افزوده شده به ورژن 1.9 از jQuery:

با ورژن 1.9 این متد گزینه ی children را پشتیبانی می کند که عناصر بعدی را نیز متحرک سازی می کند.

`addClass(className [, options])`

| Parameter | Description |
|-----------|---|
| className | یک رشته می باشد که حاوی یک یا چند گروه CSS است (به وسیله ی فاصله از یکدیگر مجزا می شوند.) |
| options | همه ی رشته های انیمیشن را نمایش می دهد. همه ی پراپرتی ها انتخابی هستند. مقادیر ممکن عبارتند از: |


```

</style>
<script type="text/javascript">
$(document).ready(function () {
  $('.button').click(function () {
    if (this.id == "add") {
      $('#animTarget').addClass("myClass", "fast")
    } else {
      $('#animTarget').removeClass("myClass", "fast")
    }
  })
});
</script>
</head>
<body>
<div id=animTarget class="elemClass">
  Hello!
</div>
<button class="button" id="add">Add Class</button>
<button class="button" id="remove">Remove Class</button>
</body>
</html>

```

روی دکمه های **Add Class** و **Remove Class** کلیک کنید تا افکت گروه ها را روی باکس ها مشاهده کنید.

انتقال چند گروه

این مثال انتقال چند گروه را به متد **addClass** نشان می دهد:

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI addClass 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>
<!-- CSS -->
<style>
  .red {
    color: red;
  }
  .big {
    font-size: 5em;
  }
  .spaced {
    padding: 1em;
  }
</style>
<script>
$(document).ready(function () {
  $('.button-1').click(function () {
    $('#welcome').addClass("red big spaced", 3000);
  });
});

```

```

});
</script>
</head>
<body>
  <p id="welcome">Welcome to Tutorials Point!</p>
  <button class="button-1">Click me</button>
</body>
</html>

```

JqueryUI - Color Animation

jQueryUI چند متد مرکزی **jQuery** را طوری گسترش می دهد که می توانید انتقال های مختلف برای یک عنصر را متحرک سازی کنید. یکی از آنها متد **animate** است. **jQueryUI** متد **jQuery animate** را برای پشتیبانی از متحرک سازی رنگ، گسترش می دهد. شما می توانید یکی از چندین پراپرتی **CSS** را رنگ یک عنصر را تعریف می کند، متحرک سازی کنید. در زیر پراپرتی های **CSS** را مشاهده می کنید که متد **animate** از آنها پشتیبانی می کند.

backgroundColor: رنگ زمینه ی عنصر را تنظیم می کند.

borderTopColor: رنگ مربوط به قسمت بالای مرز عنصر را تنظیم می کند.

borderBottomColor: رنگ قسمت پایین مرز عنصر را تنظیم می کند.

borderLeftColor: رنگ سمت چپ مرز عنصر را تنظیم می کند.

borderRightColor: رنگ سمت راست مرز عنصر را تنظیم می کند.

color: رنگ متن عنصر را تنظیم می کند.

outlineColor: رنگ اوت لاین را تنظیم می کند؛ برای تاکید بر روی عنصر استفاده می شود.

ترکیب

در زیر ترکیب مربوط به متد **jQueryUI animate** را مشاهده می کنید:

```

$("#selector").animate(

```



```
{
  backgroundColor: "black",
  color: "white"
}
);
```

می توانید به تعداد دلخواه پراپرتی برای این متد تنظیم کنید که به وسیله ی کاما مجزا شده اند.

مثال:

مثال زیر استفاده از متدهای **addClass()** را توضیح می دهد:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery UI addClass 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>
    .elemClass {
      width: 200px;
      height: 50px;
      background-color: #b9cd6d;
    }
    .myClass {
      font-size: 40px;
      background-color: #ccc;
      color: white;
    }
  </style>
  <script type="text/javascript">
    $(document).ready(function () {
      $('#button-1').click(function () {
        $('#animTarget').animate({
          backgroundColor: "black",
          color: "white"
        })
      })
    });
  </script>
</head>
<body>
  <div id="animTarget" class="elemClass">
    Hello!
  </div>
  <button id="button-1">Click Me</button>
</body>
</html>
```

روی دکمه ی **click me** کلیک کنید و شاهد تغییرات انیمیشن باکس باشید.

JqueryUI - Effect

این فصل در مورد متد **effect()** بحث خواهد کرد که یکی از متدهایی است که برای تنظیم افکت های بصری **jQueryUI** استفاده می شود. متد یک **jQueryUI** انیمیشن را برای برای عناصری به کار می گیرد که نیازی به نمایش دادن یا پنهان کردن آن نیست.

ترکیب

متد **effect()** دارای ترکیب زیر می باشد

`effect(effect [, options] [, duration] [, complete])`

| Parameter | Description |
|-----------|--|
| effect | این پارامتر، رشته ای است که افکتی را نمایش می دهد که برای تغییر استفاده می شود. |
| options | این پارامتر از نوع Object است و تنظیمات خاص افکت و easing را نشان می دهد. علاوه بر این هر افکت دارای مجموعه گزینه های خود می باشد که می تواند در بین چند افکت در جدول افکت های jQueryUI به عنوان افکت مشترک تعیین شود. |
| duration | این پارامتر از نوع عدد یا رشته می باشد و عدد مربوطه زمان افکت را به هزارم ثانیه نشان می دهد. مقدار پیش فرض آن 400 است. |
| complete | وقتی که افکت برای یک عنصر کامل می شود، این یک متد کال بک برای آن عنصر میباشد. |

افکت های jQuery

جدول زیر افکت های مختلفی را توصیف می کند که با متد **effects()** استفاده می شوند.

| Effect | Description |
|---------|---|
| blind | این افکت عنصر را به شیوه ی پنجره ی کور نشان می دهد یا پنهان می کند: با حرکت بالا و پایین پایینی و یا راست و چپ، بسته به مسیر و سبک تعیین شده. |
| bounce | باعث می شود که عنصر در حالت پرش در یک مسیر افقی یا عمودی ظاهر شود، که عنصر را به طور انتخابی پنهان کرده و یا نمایان می سازد. |
| clip | با حرکت مرزهای روبه روی عنصر با یکدیگر، عنصر را پنهان کرده یا نمایان می سازد تا اینکه مرزها در وسط به یکدیگر برسند. |
| drop | عنصر را تنظیم می کند که طوری به نظر برسد که انگار روی صفحه یا از صفحه رها شده است، به این ترتیب آن را پنهان کرده یا نمایان می کند. |
| explode | با تقسیم عنصر به قطعات مختلف که در مسیرهای شعاعی حرکت می کند، انگار که روی صفحه و یا از صفحه پخش می شود، آن را پنهان می کند یا نمایش می دهد. |
| fade | با تنظیم میزان کدري عنصر، آن را پنهان یا آشکار می کند، درست مانند افکت های محو مرکزی عمل می کند، اما بدون گزینه. |
| fold | با تنظیم مرزها به داخل و یا خارج، عنصر را نمایش داده و یا پنهان می کند، سپس همین کار را برای مجموعه ی دیگری از مرزها انجام می دهد. |

| | |
|-----------|--|
| highlight | با تغییر لحظه ای رنگ زمینه، توجه را به سمت عنصر فرا می خواند و عنصر را پنهان کرده و یا نمایش می دهد. |
| puff | با تنظیم میزان کدري عنصر، آن را در جای خود باز کرده و یا می بندد. |
| pulsate | قبل از اینکه اطمینان حاصل کند که عنصر همانطور که تعیین شده، نمایش داده می شود یا پنهان می شود، میزان کدري عنصر را تنظیم می کند. |
| scale | عنصر را با یک درصد خاص باز کرده و یا می بندد. |
| shake | عنصر را به عقب و جلو ، افقی یا عمودی، حرکت می دهد. |
| size | طول و عرض عنصر را به مقدار مشخصی تغییر می دهد. مانند مقیاس، به جز اینکه چطور اندازه ی مورد هدف تعیین شده باشد. |
| slide | عنصر را طوری جابه جا می کند که به نظر می رسد روی صفحه یا خارج صفحه اسلاید شده است. |
| transfer | یک عنصر موقت اوت لاین را محرک سازی می کند که باعث می شود عنصر به عنصری دیگر منتقل شود. ظاهر عنصر اوت لاین باید از طریق قوانین CSS برای گروه ui-effects-transfer تعریف شود یا گروه به عنوان یک گزینه تعیین شود. |

مثال:

مثال زیر استفاده از متد **effect()** را با افکت های مختلف لیست شده در جدول بالا توضیح می دهد:

افکت – حرکت

```
<!doctype html>
<html lang="en">
<head>
```

```

<meta charset="utf-8">
<title>jQuery UI effect 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>
<!-- CSS -->
<style>
#box-1 {
    height: 100px;
    width: 100px;
    background: #b9cd6d;
}
</style>
<script>
$(document).ready(function () {
    $('#box-1').click(function () {
        $('#box-1').effect("shake", {
            times: 10,
            distance: 100
        }, 3000, function () {
            $(this).css("background", "#cccccc");
        });
    });
});
</script>
</head>
<body>
<div id="box-1">Click On Me</div>
</body>
</html>

```

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI effect 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>
<!-- CSS -->
<style>
#box-2 {
    height: 100px;
    width: 100px;
    background: #b9cd6d;
}
</style>
<script>
$(document).ready(function () {
    $('#box-2').click(function () {
        $('#box-2').effect({
            effect: "explode",
            easing: "easeInExpo",

```

```

        pieces: 4,
        duration: 5000
    });
});
});
</script>
</head>
<body>
<div id="box-2"></div>
</body>
</html>

```

JqueryUI – Hide

این فصل به متد **hide()** خواهد پرداخت که یکی از متدهایی است که برای تنظیم افکت های بصری **jQueryUI** استفاده می شود. متد **effect()** یک افکت انیمیشنی برای پنهان کردن عناصر به کار می برد.

ترکیب:

متد **hide()** دارای ترکیب زیر است

hide(effect [, options] [, duration] [, complete])

| Parameter | Description |
|-----------|---|
| effect | این پارامتر یک رشته است که نشان می دهد چه افکتی را برای تغییر استفاده کنیم. |
| options | این پارامتر از نوع Object است و تنظیمات خاص افکت و easing را نشان می دهد. علاوه بر این هر افکت دارای مجموعه گزینه های مربوطه خود می باشد که می توانند به عنوان افکت مشترک بین چند افکت توصیف شده در جدول jQueryUI Effects تعیین شوند. |
| duration | این پارامتر از نوع Number یا String است و زمان افکت را به هزارم ثانیه نشان می دهد. مقدار پیش فرض آن 400 است. |

| | |
|----------|--|
| complete | وقتی که افکت برای این عنصر کامل می شود، این پارامتر یک متد کال بک است که برای عنصر فراخوانده می شود. |
|----------|--|

افکت های jQueryUI

جدول زیر افکت های مختلفی را توصیف می کند که می توانند با متد **hide()** استفاده شوند.

افکت – Blind

مثال زیر استفاده از متد **hide()** را با استفاده از افکت **blind** نشان می دهد. روی دکمه ی **Blind Effect**

Hide کلیک کنید تا قبل از پنهان شدن عنصر، افکت **blind** را مشاهده کنید.

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI hide 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>
<!-- CSS -->
<style>
.toggler {
width: 500px;
height: 200px;
}
#button {
padding: .5em 1em;
text-decoration: none;
}
#effect {
width: 240px;
height: 135px;
padding: 0.4em;
position: relative;
}
#effect h3 {
margin: 0;
padding: 0.4em;
text-align: center;
}
</style>
<script>
$(function () {
function runEffect() {
$("#effect").hide("blind", { times: 10, distance: 100 }, 1000, callback);
```

```

});
// callback function to bring a hidden box back
function callback() {
    setTimeout(function () {
        $("#effect").removeAttr("style").hide().fadeIn();
    }, 1000);
};

$("#button").click(function () {
    runEffect();
    return false;
});
});
</script>
</head>
<body>
<div class="toggler">
<div id="effect" class="ui-widget-content ui-corner-all">
<h3 class="ui-widget-header ui-corner-all">Hide</h3>
<p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore.
</p>
</div>
</div>
<a href="#" id="button" class="ui-state-default ui-corner-all">Blind Effect Hide</a>
</body>
</html>

```

افکت – shake

مثال زیر استفاده از متد **shake()** را با افکت **blind** نشان می دهد. روی دکمه ی **Shake Effect Hide** کلیک کنید تا قبل از اینکه عنصر پنهان شود، افکت **shake** را مشاهده کنید:

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery UI hide 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>
<!-- CSS -->
<style>
.toggler-1 {
    width: 500px;
    height: 200px;
}
#button-1 {
    padding: .5em 1em;
    text-decoration: none;
}
#effect-1 {
    width: 240px;

```



```

height: 135px;
padding: 0.4em;
position: relative;
}
#effect-1 h3 {
margin: 0;
padding: 0.4em;
text-align: center;
}
</style>
<script>
$(function () {
function runEffect() {
$("#effect-1").hide("shake", { times: 10, distance: 100 }, 1000, callback);
};
// callback function to bring a hidden box back
function callback() {
setTimeout(function () {
$("#effect-1").removeAttr("style").hide().fadeIn();
}, 1000);
};
// set effect from select menu value
$("#button-1").click(function () {
runEffect();
return false;
});
});
</script>
</head>
<body>
<div class="toggler-1">
<div id="effect-1" class="ui-widget-content ui-corner-all">
<h3 class="ui-widget-header ui-corner-all">Hide</h3>
<p>
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore.
</p>
</div>
</div>
<a href="#" id="button-1" class="ui-state-default ui-corner-all">Shake Effect Hide</a>
</body>
</html>

```

JqueryUI - Remove Class

این فصل در مورد متد **removeClass()** توضیح خواهد داد که یکی از متدهایی است که برای افکت های بصری **jQueryUI** استفاده می شود. متد **removeClass()** گروه های به کار گرفته از عناصر را حذف می کند. متد **removeClass()** گروه های خاصی را حذف می کند تا عناصر را در هنگام متحرک سازی همه ی تغییرات، هماهنگ کند.

افزوده شده به ورژن 1.0 از jQueryUI:

متد `removeClass()` با ترکیب اصلی خود به شکل زیر می باشد

`removeClass(className [, duration] [, easing] [, complete])`

| Parameter | Description |
|-----------|---|
| className | این پارامتر یک String حاوی یک یا چند گروه CSS است (به وسیله ی فاصله از هم جدا شده اند) که حذف می شوند. |
| duration | این پارامتر از نوع number یا string است و زمان افکت را به هزارم ثانیه نشان می دهد. یک مقدار 0، عنصر را مستقیما و بدون پیشروی، وارد سبک جدید می کند. مقدار پیش فرض آن 400 است. |
| easing | این پارامتر از نوع String است که راه پیشروی را در افکت نشان می دهد. مقدار پیش فرض آن swing است. |
| complete | یک متد بازگشتی است که هنگامی که افکت برای یک عنصر کامل است، برای هر عنصر فراخوانده می شود. |

افزوده شده به ورژن 1.9 از jQueryUI:

با ورژن 1.9، این متد یک گزینه ی **children** را پشتیبانی می کند که همچنین عناصر بعدی را نیز متحرک سازی می کند.

`removeClass(className [, options])`

| Parameter | Description |
|-----------|-------------|
|-----------|-------------|

| | |
|-----------|---|
| className | این پارامتر یک String حاوی یک یا چند گروه CSS است (به وسیله ی فاصله از هم جدا شده اند). |
| options | <p>این پارامتر همه ی تنظیمات انیمیشن را نمایش می دهد. همه ی پراپرتی ها انتخابی هستند. مقادیر ممکن عبارتند از:</p> <p>duration: از نوع Number یا String می باشد و زمان افکت را به هزارم ثانیه نشان می دهد. مقدار 0 عنصر را مستقیما و بدون پیشروی، وارد سبک جدید می کند. مقدار پیش فرض آن 400 است.</p> <p>easing: از نوع String است و روش پیشروی در افکت را نمایش می دهد. مقدار پیش فرض آن swing است.</p> <p>complete: یک متد بازگشتی است که هنگامی که افکت برای یک عنصر کامل است، برای هر عنصر فراخوانده می شود.</p> <p>children: از نوع Boolean است و نشان می دهد که آیا انیمیشن باید به طور اضافه برای همه ی عناصر هماهنگ شده ی بعدی، به کار گرفته شود. مقدار پیش فرض آن false است.</p> <p>queue: از نوع Boolean یا String است و نشان می دهد که آیا انیمیشن را در ردیف افکت ها قرار دهیم. مقدار پیش فرض آن true است.</p> |

مثال

مثال زیر استفاده از متدهای **removeClass()** را توضیح می دهد:

Passing single class

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
```

```

<title>jQuery UI removeClass 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>
    .elemClass {
        width: 200px;
        height: 50px;
        background-color: #b9cd6d;
    }
    .myClass {
        font-size: 40px;
        background-color: #ccc;
        color: white;
    }
</style>
<script type="text/javascript">
    $(document).ready(function () {
        $('.button').click(function () {
            if (this.id == "add") {
                $('#animTarget').addClass("myClass", "fast")
            } else {
                $('#animTarget').removeClass("myClass", "fast")
            }
        })
    });
</script>
</head>
<body>
    <div id=animTarget class="elemClass">
        Hello!
    </div>
    <button class="button" id="add">Add Class</button>
    <button class="button" id="remove">Remove Class</button>
</body>
</html>

```

روی دکمه های **Add Class** و **Remove Class** کلیک کنید تا افکت گروه ها را روی باکس مشاهده کنید.

JqueryUI - Show

این فصل به بررسی متود **show()** خواهد پرداخت که یکی از متودهایی است که برای تنظیم افکت های بصری **jQueryUI** استفاده می شود. متود **show()** یک آیتم را با استفاده از افکت نمایش داده شده، نشان می دهد.

متود **show()** حالت بصری عناصر در هم پیچیده را با استفاده از افکت تعیین شده، تغییر وضعیت می دهد.

ترکیب:

متود show() دارای ترکیب زیر است

```
.show( effect [, options ] [, duration ] [, complete ] )
```

| Parameter | Description |
|-----------|--|
| effect | این پارامتر یک String است که برای تغییر استفاده می شود. این یک String بوده و افکت ها را برای استفاده در هنگام تنظیم قابلیت رویت عنصر، نمایش می دهد. افکت ها رد جدول ارائه شده اند. |
| options | این پارامتر از نوع Object می باشد و تنظیمات خاص افکت و easing را نشان می دهد. علاوه بر این هر افکت دارای مجموعه ای از گزینه های خود می باشد که می توانند همراه با چند افکت در جدول <i>jQueryUI Effects</i> توصیف شوند. |
| duration | این پارامتر از نوع Number یا String است و طول مدت اجرای انیمیشن را تعیین خواهد کرد. مقدار پیش فرض آن 400 است. |
| complete | این یک متود بازگشتی است که وقتی افکت برای هر عنصر کامل می شود، فراخوانده می شود. |

افکت های jQueryUI

جدول زیر افکت های مختلفی را توضیح می دهد که می توانند با متود **effects()** استفاده شوند.

| Effect | Description |
|-----------|---|
| blind | عنصر را به روش یک window blind پنهان می کند یا نمایش می دهد: با حرکت لبه ی پایین به سمت بالا یا پایین، یا راست و چپ، بستگی به مسیر تعیین شده و سبک مورد نظر دارد. |
| bounce | باعث می شود که عنصر در حالت جهش در یک مسیر افقی یا عمودی، نمایش داده شود و به طور انتخابی عنصر را پنهان می کند یا آن را نمایش می دهد. |
| clip | مرزهای عنصر در مسیرهای عکس حرکت می دهد تا اینکه در وسط به یکدیگر برسند و یا بالعکس، به این صورت عنصر را نشان می دهد یا آن را پنهان می کند. |
| drop | با نمایش عنصر به نحوی که به نظر برسد روی صفحه قرار می گیرد یا از صفحه خارج می شود، عنصر را نمایش می دهد یا آن را پنهان می کند. |
| explode | با تقسیم عنصر به چند قطعه، که در مسیرهای شعاعی حرکت می کنند تا آنکه از صفحه خارج شده یا به آن وارد می شوند، عنصر را نشان می دهد یا آن را پنهان می کند. |
| fade | با تنظیم میزان کدورتی عنصر، آن را پنهان کرده یا نمایش می دهد. این همان افکت محو شدن مرکزی است، اما بدون گزینه. |
| fold | با تنظیم مرزهای متضاد به داخل یا خارج و سپس انجام همان کار برای دیگر مرزها، عنصر را پنهان کرده یا نمایش می دهد. |
| highlight | با تغییر لحظه ای رنگ زمینه ی عنصر، در هنگام نمایش دادن یا پنهان کردن عنصر، توجه را به سمت عنصر جلب می کند. |
| puff | در حالیکه میزان محو عنصر را تنظیم می کند، آن را در مکان خود گسترش داده یا می بندد. |
| pulsate | میزان محو بودن عنصر را قبل از اینکه عنصر نمایش داده شود یا پنهان شود، تنظیم می کند. |

| | |
|----------|---|
| scale | عنصر را با یک درصد خاص گسترش داده یا می بندد. |
| shake | عنصر را به عقب و جلو یا به صورت افقی و عمودی، حرکت می دهد. |
| size | عنصر را با یک طول و عرض خاص تغییر اندازه می دهد. مانند مقیاس، به جز اینکه اندازه ی هدف چگونه تعیین شده است. |
| slide | عنصر را طوری جابه جا می کند که به نظر می رسد روی صفحه یا پشت صفحه قرار دارد. |
| transfer | یک طرح کلی موقتی از عنصر را محرک سازی می کند که به نظر می رسد عنصر به یک عنصر دیگر تغییر کرد. ظاهر عنصر طرح باید از طریق قوانین CSS برای گروه ui-effects-transfer تعریف شود، یا گروه به عنوان یک گزینه تعیین شود. |

مثال

مثال زیر استفاده از متود **show()** را توضیح می دهد.

این مثال استفاده از متود **show()** را با افکت **shake** توضیح می دهد.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI show 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>
  <!-- CSS -->
  <style>
    .toggler {
      width: 500px;
      height: 200px;
    }
    #button {
      padding: .5em 1em;
    }
  </style>
```

```

text-decoration: none;
}

#effect {
width: 240px;
height: 135px;
padding: 0.4em;
position: relative;
}

#effect h3 {
margin: 0;
padding: 0.4em;
text-align: center;
}
</style>
<script>
$(function () {
// run the currently selected effect
function runEffect() {
// run the effect
$("#effect").show("shake", { times: 10, distance: 100 }, 1000, callback);
};
//callback function to bring a hidden box back
function callback() {
setTimeout(function () {
$("#effect-visible").removeAttr("style").fadeOut();
}, 1000);
};
$("#button").click(function () {
runEffect();
return false;
});
$("#effect").hide();
});
</script>
</head>
<body>
<div class="toggler">
<div id="effect" class="ui-widget-content ui-corner-all">
<h3 class="ui-widget-header ui-corner-all">Show</h3>
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore.
</p>
</div>
</div>
<a href="#" id="button" class="ui-state-default ui-corner-all">Run Effect</a>
</body>
</html>

```

نمایش با افکت Blind

مثال زیر استفاده از متود **show()** را با افکت **child** توضیح می دهد.


```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI show 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>
  <!-- CSS -->
  <style>
    .toggler {
      width: 500px;
      height: 200px;
    }
    #button {
      padding: .5em 1em;
      text-decoration: none;
    }
    #effect {
      width: 240px;
      height: 135px;
      padding: 0.4em;
      position: relative;
    }
    #effect h3 {
      margin: 0;
      padding: 0.4em;
      text-align: center;
    }
  </style>
  <script>
    $(function () {
      // run the currently selected effect
      function runEffect() {
        // run the effect
        $("#effect").show("blind", { times: 10, distance: 100 }, 1000, callback);
      };
      //callback function to bring a hidden box back
      function callback() {
        setTimeout(function () {
          $("#effect-visible").removeAttr("style").fadeOut();
        }, 1000);
      };
      // set effect from select menu value
      $("#button").click(function () {
        runEffect();
        return false;
      });
      $("#effect").hide();
    });
  </script>
</head>
<body>
  <div class="toggler">
    <div id="effect" class="ui-widget-content ui-corner-all">
      <h3 class="ui-widget-header ui-corner-all">Show</h3>

```

```

<p>
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore.
</p>
</div>
</div>
<a href="#" id="button" class="ui-state-default ui-corner-all">Run Effect</a>
</body>
</html>

```

JqueryUI - Switch Class

این فصل به بررسی متد **switchClass()** خواهد پرداخت که یک گروه جدید و مفید برای اصلاح متد **switchClass()** است، برای جابه جایی از یک گروه **CSS** به گروه دیگر و محرک سازی از یک وضعیت به وضعیت دیگر.

ترکیب:

افزوده شده به ورژن 1.0 از jQueryUI

متد **switchClass()** دارای ترکیب اصلی زیر می باشد

`switchClass(removeClassName, addClassName [, duration] [, easing] [, complete])`

| Parameter | Description |
|-----------------|--|
| removeClassName | این پارامتر یک String است که نام CCS گروه یا لیست space-delimited از نام های گروه را نشان می دهد که باید اضافه شده، حذف شده و یا toggle شوند. |
| addClassName | از نوع String است و نام یک یا چند گروه را ارائه می دهد (که با فاصله از یکدیگر مجزا شده اند)، گروه هایی که قرار است به گروه هماهنگ شده با هر عنصر افزوده شوند. |
| duration | از نوع Number یا String است و به طور انتخابی طول دوره ی افکت را slow، normal، fast یا به هزارم ثانیه نشان می دهد. اگر حذف شود، متد <code>animate()</code> پیش فرض را تعیین می کند. مقدار پیش فرض آن 400 است. |

| | |
|----------|---|
| easing | نام عملکرد easing که به متد animate() منتقل می شود. |
| complete | یک متد بازگشتی است که وقتی متد برای یک عنصر کامل می شود، برای آن عنصر فراخوانده می شود. |

افزوده شده به ورژن 1.9 از jQuery

با ورژن 1.9، این متد گزینه ی **Children** را پشتیبانی می کند که همچنین عناصر بعدی را نیز متحرک سازی خواهد کرد.

`switchClass(removeClassName, addClassName [, options])`

| Parameter | Description |
|-----------------|--|
| removeClassName | این پارامتر یک String است که نام گروه CSS یا لیست فضای محدود از نام گروه (space-delimited list) که باید به آن اضافه شود، حذف شود یا toggle شود، را نشان می دهد. |
| addClassName | از نوع String است و نام یک یا چند گروه را ارائه می دهد (که با فاصله از یکدیگر مجزا شده اند)، گروه هایی که قرار است به گروه هماهنگ شده با هر عنصر افزوده شوند. |
| options | این پارامتر همه ی تنظیمات مربوط به انیمیشن را نشان می دهد. همه ی پراپرتی ها انتخابی هستند. مقادیر ممکن عبارتند از: duration: از نوع Number یا String می باشد و زمان اجرای انیمیشن را تعیین می کند. مقدار پیش فرض آن 400 است. easing: یک string است که عملکرد را برای استفاده برای تغییر آسان می کند. مقدار پیش فرض آن swing است. |

complete: یک متد بازگشتی است که هنگامی که افکت برای یک عنصر کامل است، برای هر عنصر فراخوانده می شود.

children: از نوع Boolean است و نشان می دهد که آیا انیمیشن باید به طور اضافه برای همه ی عناصر هماهنگ شده ی بعدی، به کار گرفته شود.

queue: از نوع Boolean یا String است و نشان می دهد که آیا انیمیشن را در ردیف افکت ها قرار دهیم.

مثال:

مثال زیر استفاده از متد **switchClass()** را توضیح می دهد:

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Switch Class 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>
  <!-- CSS -->
  <style>
    .LargeClass {
      font-family: Arial;
      font-size: large;
      font-weight: bold;
      color: Red;
    }

    .NormalClass {
      font-family: Arial;
      font-size: small;
      font-weight: bold;
      color: Blue;
    }
  </style>
</script>
$(function () {
  $('#btnSwitch').click(function () {
    $(".NormalClass").switchClass("NormalClass", "LargeClass", 'fast');
    $(".LargeClass").switchClass("LargeClass", "NormalClass", 'fast');
    return false;
  });
});
```

```

});
</script>
</head>
<body>
  <div class="NormalClass">
    Tutorials Point Rocks!!!
  </div>
  <div class="NormalClass">
    Welcome to Tutorials Point!!!
  </div>
  <br />
  <input type="button" id="btnSwitch" value="Switch Class" />
</body>
</html>

```

بر روی دکمه ی Switch Class کلیک کنید تا افکت گروه ها را روی باکس مشاهده کنید.

JqueryUI- toggle

در این بخش به بررسی متد **toggle()** از افکت های بصری **jQueryUI** خواهیم پرداخت. متد **toggle()** متدهای **show()** یا **hide()** را با هم تغییر می دهد، بستگی به این دارد که آیا عنصر پنهان شده یا نه.

ترکیب:

متد **toggle()** دارای ترکیب زیر می باشد

toggle(effect [, options] [, duration] [, complete])

| Parameter | Description |
|-----------|--|
| effect | این پارامتر یک String است که بر روی استفاده از انتقال تاثیر دارد. یک رشته می باشد و افکت را برای استفاده در هنگام تنظیم قابلیت رویت عنصر، نشان می دهد. افکت ها در جدول بعدی ارائه می شوند. |

| | |
|----------|--|
| options | این پارامتر از نوع Object است و تنظیمات خاص افکت و <u>easing</u> را نشان می دهد. علاوه بر این هر افکت دارای مجموعه گزینه های خود می باشد که می تواند بین چند افکت توصیف شده در جدول jQueryUI Effects مشترک باشد. |
| duration | این پارامتر از نوع Number یا String است و طول مدت اجرای انیمیشن را تعیین می کند. مقدار پیش فرض آن 400 است. |
| complete | یک متد بازگشتی است که وقتی متد برای یک عنصر کامل می شود، برای آن عنصر فراخوانده می شود. |

افکت های jQueryUI

جدول زیر افکت های مختلفی را توصیف می کند که با متد **effects()** مورد استفاده قرار می گیرند:

| Effect | Description |
|-----------|---|
| blind | عنصر را به روش یک window blind پنهان می کند یا نمایش می دهد: با حرکت لبه ی پایین به سمت بالا یا پایین، یا راست و چپ، بستگی به مسیر تعیین شده و سبک مورد نظر دارد. |
| bounce | باعث می شود که عنصر در حالت جهش در یک مسیر افقی یا عمودی، نمایش داده شود و به طور انتخابی عنصر را پنهان می کند یا آن را نمایش می دهد. |
| clip | مرزهای عنصر در مسیرهای عکس حرکت می دهد تا اینکه در وسط به یکدیگر برسند و یا بالعکس، به این صورت عنصر را نشان می دهد یا آن را پنهان می کند. |
| drop | با نمایش عنصر به نحوی که به نظر برسد روی صفحه قرار می گیرد یا از صفحه خارج می شود، عنصر را نمایش می دهد یا آن را پنهان می کند. |
| explode | با تقسیم عنصر به چند قطعه، که در مسیرهای شعاعی حرکت می کنند تا آنکه از صفحه خارج شده یا به آن وارد می شوند، عنصر را نشان می دهد یا آن را پنهان می کند. |
| fade | با تنظیم میزان کدورت عنصر، آن را پنهان کرده یا نمایش می دهد. این همان افکت محو شدن مرکزی است، اما بدون گزینه. |
| fold | با تنظیم مرزهای متضاد به داخل یا خارج و سپس انجام همان کار برای دیگر مرزها، عنصر را پنهان کرده یا نمایش می دهد. |
| highlight | با تغییر لحظه ای رنگ زمینه ی عنصر، در هنگام نمایش دادن یا پنهان کردن عنصر، توجه را به سمت عنصر جلب می کند. |
| puff | در حالیکه میزان محو عنصر را تنظیم می کند، آن را در مکان خود گسترش داده یا می بندد. |

| | |
|----------|--|
| pulsate | میزان محو بودن عنصر را قبل از اینکه عنصر نمایش داده شود یا پنهان شود، تنظیم می کند. |
| scale | عنصر را با یک درصد خاص گسترش داده یا می بندد. |
| shake | عنصر را به عقب و جلو یا به صورت افقی و عمودی، حرکت می دهد. |
| size | عنصر را با یک طول و عرض خاص تغییر اندازه می دهد. مانند مقیاس، به جز اینکه اندازه ی هدف چگونه تعیین شده است. |
| slide | عنصر را طوری جابه جا می کند که به نظر می رسد روی صفحه یا پشت صفحه قرار دارد. |
| transfer | Animates a transient outline element that makes the element appear to transfer to another element. The appearance of the outline element must be defined via CSS rules for the ui-effects-transfer class, or the class specified as an option. |

مثال

مثال زیر استفاده از متد **toggle()** را با افکت های مختلف لیست شده در جدول بالا توضیح می دهد:

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Toggle 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>
  <!-- CSS -->
  <style>
    .toggler {
      width: 500px;
      height: 200px;
    }
  </style>
```



```

#button {
  padding: .5em 1em;
  text-decoration: none;
}
#effect {
  width: 240px;
  height: 135px;
  padding: 0.4em;
  position: relative;
}
#effect h3 {
  margin: 0;
  padding: 0.4em;
  text-align: center;
}
</style>
<script>
$(function () {
  function runEffect() {
    $("#effect").toggle('explode', 300);
  };
  $("#button").click(function () {
    runEffect();
    return false;
  });
});
</script>
</head>
<body>
<div class="toggler">
  <div id="effect" class="ui-widget-content ui-corner-all">
    <h3 class="ui-widget-header ui-corner-all">Toggle</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore.
    </p>
  </div>
  <div>
    <a href="#" id="button" class="ui-state-default ui-corner-all">Toggle</a>
  </div>
</body>
</html>

```

روی دکمه ی **Toggle** کلیک کنید تا چگونگی نمایش داده شدن یا پنهان شدن گروه ها را مشاهده کنید.

JqueryUI - Toggle Class

این بخش در مورد متد **toggleClass()** بحث خواهد کرد که یک گروه مفید جدید برای ترمیم می باشد. متد **toggleClass()** یک یا چند گروه از هر عنصر را در مجموعه ی هماهنگ شده ی عناصر اضافه یا حذف می کند.

متد (toggleClass) دارای ترکیب پایه ی خود به شکل زیر می باشد

افزوده شده به ورژن 1.0 از jQueryUI

toggleClass(className [, switch] [, duration] [, easing] [, complete])

| Parameter | Description |
|-----------|---|
| className | این پارامتر یک String است که نام CCS گروه یا لیست space-delimited از نام های گروه را نشان می دهد که باید اضافه شده، حذف شده و یا toggle شوند. |
| switch | از نوع Boolean می باشد و اگر تعیین شده باشد، متد (toggleClass) را وادار می کند تا در صورت تنظیم بودن بر روی true، گروه را اضافه کند و در صورت تنظیم بودن بر روی false، گروه را حذف کند. |
| duration | از نوع Number یا String است و به طور انتخابی طول دوره ی افکت را fast، normal، slow یا به هزارم ثانیه نشان می دهد. اگر حذف شود، متد (animate) پیش فرض را تعیین می کند. مقدار پیش فرض آن 400 است. |
| easing | نام عملکرد easing که به متد (animate) منتقل می شود. |
| complete | یک متد بازگشتی است که وقتی متد برای یک عنصر کامل می شود، برای آن عنصر فراخوانده می شود. |

افزوده شده به ورژن 1.9 از jQueryUI

با ورژن 1.9 ، اکنون این متد گزینه ی **children** را پشتیبانی می کند که همچنین عناصر بعدی را نیز متحرک سازی خواهد کرد.

toggleClass(className [, switch] [, options])

| Parameter | Description |
|-----------|---|
| className | این پارامتر یک String است که نام گروه CSS یا لیست فضای محدود از نام گروه (space-delimited list) که باید به آن اضافه شود، حذف شود یا toggle شود، را نشان می دهد. |
| switch | از نوع Boolean میباشد و اگر تعیین شده باشد، متد toggleClass() را وادار می کند تا در صورت تنظیم بودن بر روی true، گروه را اضافه کند و در صورت تنظیم بودن بر روی false، گروه را حذف کند. |
| options | این پارامتر همه ی تنظیمات مربوط به انیمیشن را نشان می دهد. همه ی پراپرتی ها انتخابی هستند. مقادیر ممکن عبارتند از: duration: از نوع Number یا String می باشد و زمان اجرای انیمیشن را تعیین می کند. مقدار پیش فرض آن 400 است. easing: یک string است که عملکرد را برای استفاده برای تغییر آسان می کند. مقدار پیش فرض آن swing است. complete: یک متد بازگشتی است که هنگامی که افکت برای یک عنصر کامل است، برای هر عنصر فراخوانده می شود. children: از نوع Boolean است و نشان می دهد که آیا انیمیشن باید به طور اضافه برای همه ی عناصر هماهنگ شده ی بعدی، به کار گرفته شود. |

queue: از نوع Boolean یا String است و نشان می دهد که آیا انیمیشن را در ردیف افکت ها قرار دهیم.

مثال

مثال زیر استفاده از متد `toggleClass()` را توضیح می دهد:

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Switch Class 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>
  <!-- CSS -->
  <style>
    .class1 {
      border-width: 10px;
      border-color: grey;
      background-color: #cedc98;
      color: black;
    }
  </style>
  <script>
    function toggle() {
      $("#para").toggleClass("class1", 1000);
    }
  </script>
</head>
<body>
  <button onclick=toggle()> Toggle </button>
  <p id="para" style=border-style:solid> Welcome to Tutorials Point </p>
</body>
</html>
```

روی دکمه ی **Toggle** کلیک کنید تا گروه های **CSS** را که برای متن تغییر یافته اند، مشاهده کنید.

در پایان ضمن تشکر از انتخاب شما، امیدواریم مطالب این کتاب برای شما مفید بوده باشد.

علاوه بر این می‌توانید پیشنهادها و انتقادات خود را از طریق رایانامه Book.tahlildadeh@gmail.com با ما در میان بگذارید.

