



EBook Gratis

APRENDIZAJE jQuery UI Library

Free unaffiliated eBook created from
Stack Overflow contributors.

#jquery-ui

Tabla de contenido

Acerca de.....	1
Capítulo 1: Comenzando con jQuery UI Library.....	2
Observaciones.....	2
Versiones.....	2
Examples.....	3
Agregar el script de jQuery UI y el uso básico.....	4
Configuración de jQuery UI para el primer ejemplo.....	4
Capítulo 2: Acordeón.....	6
Sintaxis.....	6
Parámetros.....	6
Observaciones.....	6
Examples.....	6
Uso básico de acordeón.....	6
Acordeón destruye el uso.....	7
Acordeón deshabilitar el uso.....	7
Acordeón habilitar el uso.....	7
Opción de acordeón.....	7
Uso de actualización de acordeón.....	8
Uso del widget Accordion.....	8
Capítulo 3: Arrastrable.....	9
Examples.....	9
Ejemplo simple.....	9
Arrastrable con asa.....	9
Capítulo 4: Autocompletar.....	10
Examples.....	10
Ejemplo simple.....	10
Capítulo 5: Botón.....	11
Sintaxis.....	11
Parámetros.....	11
Examples.....	11

Uso básico.....	11
Capítulo 6: Deslizador.....	12
Examples.....	12
Ejemplo simple.....	12
Control deslizante de rango.....	12
Inicializando valores y límites de valores.....	12
Usando el evento de diapositivas.....	13
Configuración de valores y el evento de cambio.....	13
Capítulo 7: Diálogo.....	15
Sintaxis.....	15
Parámetros.....	15
Observaciones.....	18
Examples.....	18
Ejemplo simple.....	18
Abrir diálogo cuando se produce el evento.....	18
Ejemplo complejo: la interfaz de usuario de jQuery dinámicamente crea un diálogo.....	19
Creando un diálogo con barra de título tabulada.....	22
Diálogo sin botón de cierre.....	24
Capítulo 8: Hilandero.....	25
Sintaxis.....	25
Parámetros.....	25
Observaciones.....	25
Examples.....	25
Ejemplo básico.....	25
Capítulo 9: Iconos.....	27
Sintaxis.....	27
Observaciones.....	27
Examples.....	27
Uso básico.....	27
Capítulo 10: jQuery ui sortable.....	28
Examples.....	28
jQuery UI Sortable - Drop Placeholder.....	28

Capítulo 11: jQuery UI Plug-in rotativo	30
Parámetros.....	30
Examples.....	30
Ejemplo de uso inicial.....	30
Capítulo 12: Ordenable	32
Sintaxis.....	32
Parámetros.....	32
Observaciones.....	35
Examples.....	35
Ejemplo simple.....	35
Rejilla clasificable con diseño flexible.....	36
Artículos estacionarios al arrastrar.....	36
Ordenable - Animar la reversión de un artículo no aceptado.....	37
Capítulo 13: Selector de fechas	40
Examples.....	40
Inicialización.....	40
Configuración de fechas mínimas y máximas para un selector de fechas.....	40
Mostrar semana del año.....	40
Establecer un formato de fecha personalizado.....	40
Mostrar mes y año desplegable.....	42
Creditos	43

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [jquery-ui-library](#)

It is an unofficial and free jQuery UI Library ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official jQuery UI Library.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Comenzando con jQuery UI Library

Observaciones

jQuery UI es una biblioteca de IU de JavaScript, construida sobre jQuery, que ofrece un conjunto de interacciones de interfaz de usuario, efectos y widgets.

Versiones

Versión	Fecha de lanzamiento
1.7.0	2009-03-06
1.7.1	2009-03-19
1.7.2	2009-06-12
1.7.4	2010-05-04
1.8.0	2010-03-23
1.8.1	2010-05-04
1.8.2	2010-06-07
1.8.4	2010-08-10
1.8.5	2010-09-17
1.8.6	2010-10-02
1.8.7	2010-12-10
1.8.8	2011-01-14
1.8.9	2011-01-21
1.8.10	2011-02-24
1.8.11	2011-03-18
1.8.12	2011-04-23
1.8.13	2011-05-17
1.8.14	2011-06-28

Versión	Fecha de lanzamiento
1.8.15	2011-08-08
1.8.16	2011-08-18
1.8.17	2012-01-10
1.8.18	2012-02-23
1.8.19	2012-04-17
1.8.20	2012-04-30
1.8.21	2012-06-05
1.8.22	2012-07-24
1.8.23	2012-08-15
1.8.24	2012-09-28
1.9.0	2012-10-08
1.9.1	2012-10-25
1.9.2	2012-11-23
1.10.0	2013-01-17
1.10.1	2013-02-15
1.10.2	2013-03-14
1.10.3	2013-05-03
1.10.4	2014-01-17
1.11.0	2014-06-26
1.11.1	2014-08-13
1.11.2	2014-10-16
1.11.3	2015-02-12
1.11.4	2015-03-11

Examples

Agregar el script de jQuery UI y el uso básico

Para comenzar con la biblioteca de jQuery UI, deberá agregar el script jQuery, el script jQuery UI y la hoja de estilo jQuery UI a su HTML.

En primer lugar, [descargue](#) jQuery UI; Elija las características que necesita en la página de descarga. Descomprima su descarga y coloque `jquery-ui.css` y `jquery-ui.js` (y `jquery.js`) en una carpeta donde puede usarlos desde su HTML (por ejemplo, con sus otros scripts y hojas de estilo).

jQuery UI depende de jQuery, así que recuerde incluir `jquery.js` antes que `jquery-ui.js`.

```
<link rel="stylesheet" href="stylesheets/jquery-ui.css">
<script src="scripts/jquery.js"></script>
<script src="scripts/jquery-ui.js"></script>
```

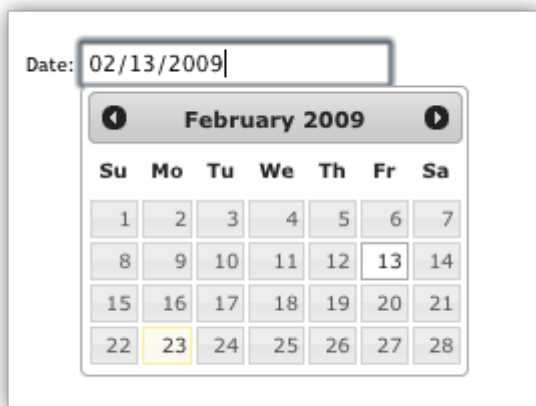
¡Eso es! Ahora puedes usar jQuery UI. Por ejemplo, use el selector de fecha con el siguiente HTML:

```
<input type="text" name="date" id="date">
```

Luego usa el siguiente JavaScript:

```
$("#date").datepicker();
```

Lo que te dará una buena ventana emergente para el datepicker:



Para más información, consulte el [gude oficial de "Cómo empezar"](#).

Configuración de jQuery UI para el primer ejemplo

El marco de la interfaz de usuario de jQuery ayuda a ampliar y aumentar los controles de la interfaz de usuario para la biblioteca de jQuery JavaScript.

Cuando desee utilizar la interfaz de usuario jQuery, deberá agregar estas bibliotecas a su HTML. Una forma rápida de comenzar es utilizando las fuentes de código disponibles de Content

Delivery Network:

Bibliotecas jQuery

```
https://code.jquery.com/jquery-3.1.0.js
https://code.jquery.com/ui/1.12.0/jquery-ui.js
```

Puedes elegir muchos temas diferentes para jQuery UI e incluso lanzar tu propio tema. Para este ejemplo, utilizaremos 'Suavidad'. Usted agrega el tema a través de CSS.

jQuery UI CSS

```
https://code.jquery.com/ui/1.12.0/themes/smoothness/jquery-ui.css
```

Poniendolo todo junto

Cuando haya descargado o seleccionado su CDN, ahora querrá agregar estas bibliotecas y hojas de estilo a su HTML para que su página web pueda usar jQuery y jQuery UI. El orden en que se cargan las bibliotecas es importante. Llame primero a la biblioteca jQuery y luego a su biblioteca jQuery UI. Como jQuery UI extiende jQuery, debe llamarse después. Su HTML puede ser similar al siguiente.

```
<html>
<head>
  <title>My First UI</title>
  <link rel="stylesheet" href="https://code.jquery.com/ui/1.12.0/themes/smoothness/jquery-
ui.css">
  <script src="https://code.jquery.com/jquery-3.1.0.js"></script>
  <script src="https://code.jquery.com/ui/1.12.0/jquery-ui.js"></script>
  <script>
    $( function() {
      $( "#sortable" ).sortable();
      $( "#sortable" ).disableSelection();
    } );
  </script>
</head>
<body>

<ul id="sortable">
  <li class="ui-state-default">Item 1</li>
  <li class="ui-state-default">Item 2</li>
  <li class="ui-state-default">Item 3</li>
  <li class="ui-state-default">Item 4</li>
  <li class="ui-state-default">Item 5</li>
  <li class="ui-state-default">Item 6</li>
  <li class="ui-state-default">Item 7</li>
</ul>

</body>
</html>
```

Lea Comenzando con jQuery UI Library en línea: <https://riptutorial.com/es/jquery-ui/topic/513/comenzando-con-jquery-ui-library>

Capítulo 2: Acordeón

Sintaxis

- `$(function () {$("#selector").accordion ();});`
- `$(function () {$("#selector").accordion ({active: 2});});`
- `$(function () {$("#selector").accordion ({animate: 200});});`
- `$(function () {$("#selector").accordion ({colapsable: true});});`

Parámetros

Parámetro	Detalle
activo	Escriba Boolean o Integer, Boolean requiere colapsable para ser verdadero
animar	Tipo Booleano, Número, Cadena u Objeto
plegable	Tipo Booleano
evento	Tipo de cadena
encabezamiento	Elemento selector de tipo
heightStyle	Tipo de cadena
iconos	Tipo jQuery UI icon objeto

Observaciones

Puede encontrar más información aquí: <http://api.jqueryui.com/accordion/>

Examples

Uso básico de acordeón

Para usar un acordeón, uno debe tener encabezados y contenido dentro de los encabezados en su HTML. Entonces uno debe instanciar el método del `accordion()` de jQuery UI.

```
<script>
$(function() {
    $( "#accordion" ).accordion();
});
</script>
```

En el HTML:

```
<div id="accordion">
  <h3>First header</h3>
  <div>First content panel</div>
  <h3>Second header</h3>
  <div>Second content panel</div>
</div>
```



Acordeón destruye el uso.

```
$( "#accordion" ).accordion( "destroy" );
```

Esto eliminará la funcionalidad de acordeón por completo y mostrará el código HTML predeterminado eliminando todos los elementos jQuery-UI.

Este método no tiene ningún argumento.

Acordeón deshabilitar el uso

```
$( "#accordion" ).accordion( "disable" );
```

Este método deshabilitará el acordeón, es decir, los encabezados no son seleccionables, lo que hace que el contenido sea de solo lectura y estático.

Este método no tiene ningún argumento.

Acordeón habilitar el uso

```
$( "#accordion" ).accordion( "enable" );
```

Este método permitirá un acordeón. Esto habilitará un acordeón deshabilitado o simplemente no hará nada en un acordeón ya habilitado.

Este método no tiene ningún argumento.

Opción de acordeón

```
var options = $( "#accordion" ).accordion( "option" );
```

Esto devolverá un objeto PlainObject con todas las opciones que representan el acordeón seleccionado. Esto contendrá todos los valores de las claves que se explican en la sección Parámetros.

Este método toma parámetros que son los nombres de opciones básicos explicados en el parámetro. Uno puede configurar las opciones de esta manera:

```
$( "#accordion" ).accordion( "option", "disabled", true );
```

Uso de actualización de acordeón

```
$( "#accordion" ).accordion( "refresh" );
```

Este método vuelve a calcular la altura de los paneles de acordeón si los encabezados o el contenido se agregaron o eliminaron en el DOM.

Uso del widget Accordion

```
var widget = $( "#accordion" ).accordion( "widget" );
```

Este método devuelve un objeto jQuery que contiene el acordeón.

Lea Acordeón en línea: <https://riptutorial.com/es/jquery-ui/topic/630/acordeon>

Capítulo 3: Arrastrable

Examples

Ejemplo simple

Habilitar la funcionalidad de arrastrar en cualquier elemento DOM.

```
<script>
  $(function() {
    $( "#draggable" ).draggable();
  });
</script>
<div id="draggable" class="ui-widget-content">
  <p>Drag me around</p>
</div>
```

Arrastrable con asa.

Puede usar cualquier elemento como control para arrastrar otro elemento alrededor:

```
<script>
  $(function() {
    $( "#draggable" ).draggable({
      handle: ".handle"
    });
  });
</script>
<div id="draggable">
  <span class="handle">Handle</span>
  <div>Content</div>
</div>
```

Violín

Lea Arrastrable en línea: <https://riptutorial.com/es/jquery-ui/topic/522/arrastrable>

Capítulo 4: Autocompletar

Examples

Ejemplo simple

Los widgets de autocompletar proporcionan sugerencias mientras escribe en el campo.

```
<script>
  $(document).ready(function() {
    var tags = ["ask","always", "all", "alright", "one", "foo", "blackberry",
"tweet","force9", "westerners", "sport"];
    $( "#tags" ).autocomplete({
      source: tags
    });
  });
</script>
<input type='text' title='Tags' id='tags' />
```

Lea Autocompletar en línea: <https://riptutorial.com/es/jquery-ui/topic/519/autocompletar>

Capítulo 5: Botón

Sintaxis

- `$(".selector").button();`
- `$(".selector").button({disabled: true});`
- `$(".selector").button({iconos: {primario: "ui-icon-gear", secundario: "ui-icon-triangle-1-s"}});`
- `$(".selector").button({label: "custom label"});`
- `$(".selector").button({text: false});`

Parámetros

Parámetro	Tipo - Detalles - Por defecto
disabled	Boolean : desactiva el botón si se establece en verdadero - false
icons	Object - Iconos para mostrar - {primario: nulo, secundario: null }
label	String - Texto para mostrar en el botón - null
text	Boolean - Si mostrar la etiqueta - true

Examples

Uso básico

Cree un elemento html de entrada (o botón o ancla) y llame al `button()` de jQuery UI.

```
<script>
$(function() {
    $( "#myButton" ).button();
});
</script>
```

HTML

```
<input type="button" value="A button" id="myButton">
```

Lea Botón en línea: <https://riptutorial.com/es/jquery-ui/topic/4600/boton>

Capítulo 6: Deslizador

Examples

Ejemplo simple

Un control deslizante utiliza controladores que se pueden arrastrar para seleccionar valores numéricos. A continuación se muestra un ejemplo de una inicialización de deslizador básico:

```
<script>
  $(function() {
    $( "#slider" ).slider();
  });
</script>
<div id="slider"></div>
```

Control deslizante de rango

Los controles deslizantes de rango proporcionan 2 controladores arrastrables para seleccionar valores numéricos. La inicialización del control deslizante debe proporcionar una opción de `range` establecida en `true` para crear un control deslizante de rango:

```
<script>
  $(function() {
    $( "#range-slider" ).slider({
      range: true
    });
  });
</script>
<div id="range-slider"></div>
```

Inicializando valores y límites de valores

Un elemento deslizante puede tener su valor establecido en la inicialización al proporcionar una opción de `value`. Esta opción es un número:

```
$( "#slider" ).slider({
  value: 5
});
```

Un control deslizante de rango también puede tener sus valores establecidos de esta manera al proporcionar una opción de `values`. Esta opción es una matriz de números:

```
$( "#range-slider" ).slider({
  range: true,
  values: [5, 25]
});
```


Además de proporcionar valores iniciales, el valor mínimo, el valor máximo y el intervalo de manejo se pueden definir con las opciones `min` , `max` y `step` , respectivamente:

```
$( "#range-slider" ).slider({
  range: true,
  min: 0,    // The lowest possible value will be 0
  max: 100,  // The highest possible value will be 100
  step: 5,   // The slider handles will lock in at intervals of 5
  values: [0, 100]
});
```

Usando el evento de diapositivas

El control deslizante proporciona un evento llamado `slide` que se activará **cada vez que el mouse se mueva durante un arrastre de control deslizante** . Esta función contiene una referencia al `event` diapositiva y una referencia al objeto `ui` del control deslizante. El objeto `ui` contiene un objeto jQuery para el controlador que se está moviendo y los valores del control deslizante.

Control deslizante de una sola manija:

```
var value;

$( "#slider" ).slider({
  slide: function(event, ui) {
    value = ui.value;
  }
});
```

Rango deslizante:

```
var lowValue;
var highValue;

$( "#range-slider" ).slider({
  range: true,
  slide: function(event, ui) {
    lowValue = ui.values[0];
    highValue = ui.values[1];
  }
});
```

Nota: el evento `slide` está diseñado para responder al movimiento activo del mouse y no se activará si los valores del control deslizante se cambian mediante programación. Para reaccionar ante estos eventos, usa el evento `change` .

Configuración de valores y el evento de cambio

El control deslizante proporciona un evento llamado `change` que se activará **después de que el mouse complete un arrastre del control deslizante o si los valores se han cambiado mediante programación** . Esta función contiene una referencia al `event` diapositiva y una referencia al objeto `ui` del control deslizante. El objeto `ui` contiene un objeto jQuery para el

controlador que se está moviendo y los valores del control deslizante.

Un ejemplo podría ser tener que mostrar nueva información después de que los valores de un control deslizante hayan sido actualizados por el evento de otro elemento. Usemos un elemento de `select` para la demostración donde el valor del control deslizante se establece programáticamente cuando cambia el valor de la `select` :

HTML

```
<select id="setting">
  <option value="1">Low</option>
  <option value="2">Medium</option>
  <option value="3">High</option>
</select>

<div id="slider"></div>

<div id="display-value"></div>
```

JavaScript

```
$(function() {
  $( "#slider" ).slider({
    min: 0,
    max: 11,
    // This will trigger when the value is programmatically changed
    change: function(event, ui) {
      $( "#display-value" ).text(ui.value);
    }
  });

  $( "#setting" ).change(function () {
    switch ($(this).val()) {
      case "1":
        $( "#slider" ).slider( "value", 3 ); // Sets the value of a slider programmatically
        break;
      case "2":
        $( "#slider" ).slider( "value", 7 ); // Sets the value of a slider programmatically
        break;
      case "3":
        $( "#slider" ).slider( "value", 11 ); // Sets the value of a slider programmatically
        break;
    }
  });
});
```

Nota: En estas circunstancias, el evento de `slide` no se activará y se necesita el evento de `change` . Sin embargo, si los elementos deben reaccionar a los valores del control deslizante que cambian a medida que se arrastra el manejador, el evento de `slide` será necesario.

Lea Deslizador en línea: <https://riptutorial.com/es/jquery-ui/topic/3206/deslizador>

Capítulo 7: Diálogo

Sintaxis

- `$ ("selector") .dialog ("option", "disabled");` // Option Getter, específico
- `$ ("selector") .dialog ("opción");` // Opción Getter all
- `$ ("selector") .dialog ("option", "disabled", true);` // Option Setter, específico
- `$ ("selector") .dialog ("option", {disabled: true});` // Option Setter, multiple
- `$ ("selector") .dialog ("cerrar");` // Los disparadores se cierran
- `$ ("selector") .dialog ({close: function () {}});` // Cerrar sobrecarga
- `$ ("selector") .on ("dialogclose", function (event, ui) {});` // Cerrar sobrecarga

Parámetros

Parámetro	Descripción
Opciones	
añadir a	(Selector) [Valor predeterminado: "body"] El elemento al que se debe agregar el cuadro de diálogo (y la superposición, si es modal).
autoOpen	(Booleano) [Predeterminado: <code>true</code>] Si se establece en verdadero, el cuadro de diálogo se abrirá automáticamente en el momento de la inicialización. Si es falso, el cuadro de diálogo permanecerá oculto hasta que se llame al método <code>open ()</code> .
botones	(Objeto / Array) Especifica qué botones deben mostrarse en el diálogo. El contexto de la devolución de llamada es el elemento de diálogo; Si necesita acceder al botón, está disponible como objetivo del objeto de evento.
cerrarOnEscape	(Booleano) [Predeterminado: <code>true</code>] Especifica si el cuadro de diálogo debe cerrarse cuando está enfocado y el usuario presiona la tecla Escape (ESC).
cerrarTexto	(Cadena) [Predeterminado: "close"] Especifica el texto para el botón de cerrar. Tenga en cuenta que el texto de cierre se oculta visiblemente cuando se utiliza un tema estándar.
dialogClass	(Cadena) El nombre de la clase especificada se agregará al cuadro de diálogo, para temas adicionales.
arrastrable	(Booleano) [Predeterminado: <code>true</code>] Si se establece en <code>true</code> , la barra de título podrá arrastrar el cuadro de diálogo. Requiere que se incluya el widget jQuery UI Draggable.

Parámetro	Descripción
altura	(Número / Cadena) [Valor predeterminado: "auto"] La altura del cuadro de diálogo.
esconder	(Bool / Num / Str / Obj) Si y cómo animar la ocultación del diálogo.
Altura máxima	(Número) [Predeterminado: false] La altura máxima a la que se puede cambiar el tamaño del diálogo, en píxeles.
anchura máxima	(Número) [Predeterminado: false] El ancho máximo al que se puede cambiar el tamaño del diálogo, en píxeles.
minHeight	(Número) [Valor predeterminado: 150] La altura mínima a la que se puede cambiar el tamaño del diálogo, en píxeles.
minwidth	(Número) [Valor predeterminado: 150] El ancho mínimo al que se puede cambiar el tamaño del diálogo, en píxeles.
modal	(Booleano) [Predeterminado: false] Si se establece en verdadero, el diálogo tendrá un comportamiento modal; otros elementos de la página se desactivarán, es decir, no se pueden interactuar con ellos. Los diálogos modales crean una superposición debajo del diálogo pero sobre otros elementos de la página.
posición	(Objeto) [Valor predeterminado: { my: "center", at: "center", of: window }] Especifica dónde se debe mostrar el diálogo cuando se abre. El diálogo manejará las colisiones de modo que la mayor parte del diálogo sea lo más visible posible.
redimensionable	(Booleano) [Predeterminado: true] Si se establece en verdadero, el cuadro de diálogo será redimensionable. Requiere que se incluya el widget jQuery UI Resizable.
espectáculo	(Bool / Num / Str / Obj) Si y cómo animar la visualización del diálogo.
título	(Cadena) Especifica el título del diálogo. Si el valor es nulo, se utilizará el atributo de título en el elemento de fuente de diálogo.
anchura	(Número) [Predeterminado: 300] El ancho del cuadro de diálogo, en píxeles.
Métodos	
cerrar	Cierra el diálogo.
destruir	Elimina por completo la funcionalidad de diálogo. Esto devolverá el elemento a su estado pre-init.
ejemplo	Recupera el objeto de instancia del diálogo. Si el elemento no tiene una

Parámetro	Descripción
	instancia asociada, se devuelve undefined.
Esta abierto	Si el diálogo está actualmente abierto.
moveToTop	Mueve el diálogo a la parte superior de la pila de diálogos.
abierto	Abre el diálogo.
opción	Obtiene el valor asociado actualmente con el <code>optionName</code> especificado.
opción	Obtiene un objeto que contiene pares clave / valor que representan el hash de las opciones de diálogo actuales.
opción	Establece una o más opciones para el diálogo.
widget	Devuelve un objeto jQuery que contiene el contenedor generado.
Puntos de extensión	
<code>_parteInteracción</code>	(evento) Los diálogos modales no permiten a los usuarios interactuar con elementos detrás del diálogo. Esto puede ser problemático para los elementos que no son elementos secundarios del diálogo, pero que están absolutamente posicionados para aparecer como si lo fueran. El método <code>_allowInteraction()</code> determina si se debe permitir al usuario interactuar con un elemento de destino determinado; por lo tanto, se puede utilizar para incluir en la lista blanca los elementos que no son elementos secundarios del diálogo, pero que desea que los usuarios puedan usar.
Eventos	
antesCerrar	(evento, ui) Se activa cuando un diálogo está a punto de cerrarse. Si se cancela, el diálogo no se cerrará.
cerrar	(evento, ui) Se activa cuando se cierra el cuadro de diálogo.
crear	(evento, ui) Se desencadena cuando se crea el diálogo.
arrastrar	(evento, ui {posición, desplazamiento}) Se dispara mientras se arrastra el diálogo.
dragstart	(evento, ui {posición, desplazamiento}) Se dispara cuando el usuario comienza a arrastrar el diálogo.
arrastrar	(evento, ui {posición, desplazamiento}) Se desencadena después de arrastrar el cuadro de diálogo.
atención	(evento, ui) Se desencadena cuando el diálogo se enfoca.

Parámetro	Descripción
abierto	(evento, ui) Se activa cuando se abre el cuadro de diálogo.
redimensionar	(evento, ui {originalPosition, position, originalSize, size}) Se dispara mientras se redimensiona el diálogo.
reiniciar	(evento, ui {originalPosition, position, originalSize, size}) Se dispara mientras se redimensiona el diálogo.
redimensionar	(evento, ui {originalPosition, position, originalSize, size}) Se dispara mientras se redimensiona el diálogo.

Observaciones

Fuente del parámetro: <http://api.jqueryui.com/dialog/>

Examples

Ejemplo simple

Diálogo es una ventana que se superpone colocada dentro de la ventana gráfica.

```
<script>
  $(function() {
    $( "#dialog" ).dialog();
  });
</script>
<div id="dialog" title="Basic dialog">
  <p>This is the default dialog which is useful for displaying information. The dialog window
  can be moved, resized and closed with the 'x' icon.</p>
</div>
```

Abrir diálogo cuando se produce el evento

Normalmente queremos separar la creación del diálogo de su aspecto. Entonces se necesitan tres pasos.

1. Crear elemento base

```
<div id="dialog" title="Basic dialog">
  <p>This is the default dialog which is useful for displaying information. The dialog window
  can be moved, resized and closed with the 'x' icon.</p>
</div>
```

2. `autoOpen: false` en un diálogo, tenga en cuenta la opción `autoOpen: false` que garantiza que se cerrará al principio

```
$( "#dialog" ).dialog({
```

```
    autoOpen: false
  });
```

3. Ábrelo cuando sea necesario, como en un botón.

```
$( "#dialog" ).dialog( "open" );
```

Ejemplo complejo: la interfaz de usuario de jQuery dinámicamente crea un diálogo

Generalmente, el diálogo se basa en un `div` dentro del HTML. A veces es posible que desee crear un diálogo desde cero, mediante programación. Este es un ejemplo de un diálogo modal complejo creado dinámicamente con funciones interactivas.

HTML

```
<div id="users-contain" class="ui-widget">
  <h1>Existing Users:</h1>
  <table id="users" class="ui-widget ui-widget-content">
    <thead>
      <tr class="ui-widget-header ">
        <th>Name</th>
        <th>Email</th>
        <th>Password</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>John Doe</td>
        <td>john.doe@example.com</td>
        <td>johndoel</td>
      </tr>
    </tbody>
  </table>
</div>
<button id="create-user">Create new user</button>
```

CSS

```
label,
input {
  display: block;
}

input.text {
  margin-bottom: 12px;
  width: 95%;
  padding: .4em;
}

fieldset {
  padding: 0;
  border: 0;
  margin-top: 25px;
}
```

```

h1 {
  font-size: 1.2em;
  margin: .6em 0;
}

div#users-contain {
  width: 350px;
  margin: 20px 0;
}

div#users-contain table {
  margin: 1em 0;
  border-collapse: collapse;
  width: 100%;
}

div#users-contain table td,
div#users-contain table th {
  border: 1px solid #eee;
  padding: .6em 10px;
  text-align: left;
}

.ui-dialog .ui-state-error {
  padding: .3em;
}

.validateTips {
  border: 1px solid transparent;
  padding: 0.3em;
}

```

jQuery

```

$(function() {
  // Define variables for the dialog, form and a regular expression used to verify email
  // addresses in the form
  var dialog, form,
      emailRegex = /^[a-zA-Z0-9.!#$%&'*\+\/=?^_`{|}~-]+@[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?(?:\.[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?)*$/;

  // Function to update tips when an issue in the form is detected
  // t = text to enter as the tip
  function updateTips(t) {
    tips
      .text(t)
      .addClass("ui-state-highlight");
    setTimeout(function() {
      tips.removeClass("ui-state-highlight", 1500);
    }, 500);
  }

  // Function to check the length of text entered into a field
  // o = object reference (object), n = name of field (string), min = minimum number of
  // characters (int), max = maximum number of characters (int)
  function checkLength(o, n, min, max) {
    if (o.val().length > max || o.val().length < min) {
      o.addClass("ui-state-error");
      updateTips("Length of " + n + " must be between " +

```



```

        min + " and " + max + ".");
    return false;
} else {
    return true;
}
}

// Function to perform regular expression check of text entered in field
// o = object reference (object), regexp = regular expression reference (RegExp Object), n =
name of field
function checkRegexp(o, regexp, n) {
    if (!(regexp.test(o.val()))) {
        o.addClass("ui-state-error");
        updateTips(n);
        return false;
    } else {
        return true;
    }
}

//Function called when form is submitted that will check all the form fields. If all fields
have text and all the text meets the requirements, the data is collected and added back to the
table.
function addUser() {
    var valid = true;
    allFields.removeClass("ui-state-error");

    valid = valid && checkLength(name, "username", 3, 16);
    valid = valid && checkLength(email, "email", 6, 80);
    valid = valid && checkLength(password, "password", 5, 16);

    valid = valid && checkRegexp(name, /^[a-z]([0-9a-z_\s])+$/i, "Username may consist of a-z,
0-9, underscores, spaces and must begin with a letter.");
    valid = valid && checkRegexp(email, emailRegex, "eg. ui@jquery.com");
    valid = valid && checkRegexp(password, /^([0-9a-zA-Z])+$/, "Password field only allow : a-
z 0-9");

    if (valid) {
        $("#users tbody").append("<tr>" +
            "<td>" + name.val() + "</td>" +
            "<td>" + email.val() + "</td>" +
            "<td>" + password.val() + "</td>" +
            "</tr>");
        dialog.dialog("close");
    }
    return valid;
}

// Creation of the dialog object
dialog = $("<div>", {
    id: "dialog-form",
    title: "Create New User"
}).dialog({
    autoOpen: false,
    height: 400,
    width: 350,
    modal: true,
    buttons: {
        "Create an account": addUser,
        Cancel: function() {
            dialog.dialog("close");
        }
    }
});

```

```

    }
  },
  close: function() {
    form[0].reset();
    allFields.removeClass("ui-state-error");
  }
});

// Adding elements to the dialog to be shown
dialog.html("<p class='validateTips'>All form fields are required.</p>");

// Creation of the form object to be shown inside the dialog
form = $("<form>").submit(function(e) {
  e.preventDefault();
  addUser();
}).appendTo(dialog);

// Adding elements to the form, fieldset and fields
form.append("<fieldset>");
var markup = "";
markup += "<label for='name'>Name</label>\r\n";
markup += "<input type='text' name='name' id='name' value='Jane Smith' class='text ui-
widget-content ui-corner-all'>";
markup += "<label for='email'>Email</label><input type='text' name='email' id='email'
value='jane@smith.com' class='text ui-widget-content ui-corner-all'>\r\n";
markup += "<label for='password'>Password</label><input type='password' name='password'
id='password' value='xxxxxxx' class='text ui-widget-content ui-corner-all'>\r\n";
markup += "<input type='submit' tabindex='-1' style='position:absolute; top:-1000px'>\r\n";

// Assigning our fields HTML markup to the fieldset
form.find("fieldset").html(markup);

// Assigning variables to be used for easy reference, post creation and amendment of dynamic
objects
var name = $("#name"),
    email = $("#email"),
    password = $("#password"),
    allFields = $([]).add(name).add(email).add(password),
    tips = $(".validateTips");

// Override the click event of the button to launch our dynamic dialog
$("#create-user").button().on("click", function() {
  dialog.dialog("open");
});
});

```

Ejemplo de trabajo para referencia: <https://jsfiddle.net/Twisty/LqjuxLu1/>

Creando un diálogo con barra de título tabulada

De vez en cuando, es posible que desee mostrar diálogos con más de un panel de contenido. jQuery UI ofrece pestañas que se pueden usar en conjunto con un diálogo para hacer esto posible. Si bien puede ser más común tener pestañas dentro del contenedor de contenido de un cuadro de diálogo, este ejemplo mostrará cómo hacer una lista de pestañas en la barra de título del cuadro de diálogo.

HTML

```

<button id="openButton">
  Open Dialog
</button>
<div id="dialog" style="display:none">
  <div class="ui-tabs">
    <ul>
      <li><a href="#tab_1">Tab 1</a></li>
      <li><a href="#tab_2">Tab 2</a></li>
    </ul>
    <div id="tab_1">
      <p>Tab 1 content...</p>
    </div>
    <div id="tab_2">
      <p>Tab 2 content...</p>
    </div>
  </div>
</div>

```

jQuery

```

$(document).ready(function() {
  // Options to pass to the jQuery UI Dialog
  var options = {
    position: {
      my: "left top",
      at: "left top",
      of: window
    },
    autoOpen: false
  };

  /* Initialization */
  // Initialize the dialog
  var dialog = $("#dialog").dialog(options);

  // Initialize the tabs
  var tabs = $(".ui-tabs").tabs();

  /* Gather Elements Before Rearrangement */
  var closeButton = dialog.siblings(".ui-dialog-titlebar").find(".ui-dialog-titlebar-close");
  var initialTitlebar = dialog.siblings(".ui-dialog-titlebar");

  // Find the list of tabs to make the titlebar, add the ui-dialog-titlebar class, and append
  // the close button
  var tabbedTitlebar = dialog.find(".ui-tabs ul:first").addClass("ui-dialog-
titlebar").append(closeButton);

  /* Arranging */
  // Remove the initialTitlebar
  $(initialTitlebar).remove();

  // Create a new .ui-tabs container for the tabbedTitlebar
  var tabbedTitlebarContainer = $("<div>", {
    class: "ui-tabs"
  }).append(tabbedTitlebar);

  // Prepend the tabbedTitlebarContainer to the dialog container
  dialog.parents(".ui-dialog").prepend(tabbedTitlebarContainer);

  /* Show the Dialog */

```

```
dialog.dialog("open");

var openButton = $("#openButton").button().click(function() {
    dialog.dialog("open");
});
});
```

Ejemplo de trabajo para referencia: <https://jsfiddle.net/5x4zz681/>

Diálogo sin botón de cierre

Si desea mostrar el cuadro de diálogo sin el botón de cerrar (es decir, el botón x en la esquina superior derecha del cuadro de diálogo), tal vez porque quiere forzar al usuario a seleccionar una de las opciones o botones en el cuadro de diálogo:

1- Dale a tu diálogo una clase de CSS:

```
$("#selector").dialog({
    closeOnEscape: false,
    dialogClass: "dialog-no-close",
});
```

2- Ocultar el botón de cerrar usando este CSS:

```
.dialog-no-close .ui-dialog-titlebar-close {display: none; }
```

Nota: Si desea ocultar la barra de título completa, use este CSS en su lugar:

```
.dialog-no-close .ui-dialog-titlebar {display: none; }
```

Alternativamente, puede ocultar el botón de cerrar en el código de inicialización del diálogo:

```
$("#selector").dialog({
    closeOnEscape: false,
    open: function(event, ui) {
        $(".ui-dialog-titlebar-close", $(this).parent()).hide();
    }
});
```

Lea Diálogo en línea: <https://riptutorial.com/es/jquery-ui/topic/521/dialogo>

Capítulo 8: Hilandero

Sintaxis

- `$("#id").spinner();`
- `$("#id").spinner({min: 0, max: 100, step: 5, spin: function(event, ui) {}});`

Parámetros

Parámetros	Detalle
min	Valor mínimo
max	Valor máximo
paso	¿Cuánto aumenta el valor al hacer clic con el botón de giro, puede ser decimal?
girar	Se puede usar para verificar el valor del hilandero, <code>ui.value</code> y hacer algo

Observaciones

[Ejemplo oficial](#)

[Documentacion oficial](#)

Examples

Ejemplo básico

Hace que ingresar números sea un poco más práctico al mostrar un conjunto de flechas en el lado derecho de la `input`.

HTML

```
<link rel="stylesheet" href="//code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css">
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
<script src="https://code.jquery.com/ui/1.12.0/jquery-ui.js"></script>
<script src="/resources/demos/external/jquery-mousewheel/jquery.mousewheel.js"></script>
<script>
  $( function() {
    var spinner = $( "#spinner" ).spinner();
  } );
</script>

<input id="spinner" name="value">
```

Lea Hiladero en línea: <https://riptutorial.com/es/jquery-ui/topic/6637/hiladero>

Capítulo 9: Iconos

Sintaxis

- `.ui-icon- {tipo de icono} - {sub descripción del icono} - {dirección}`

Observaciones

Los iconos también se integran en una serie de widgets de jQuery UI, como el [acordeón](#) , el botón, el menú.

Examples

Uso básico

Para una flecha gruesa que apunta hacia el norte en un tramo, agregue las clases `ui-icon` y `ui-icon-arrowthick-1-n` :

```
<span class="ui-icon ui-icon-arrowthick-1-n"></span>
```

Para un triángulo que apunte hacia el sur en un tramo, agregue las clases `ui-icon` y `ui-icon-triangle-1-s` :

```
<span class="ui-icon ui-icon-triangle-1-s"></span>
```

Lista completa de artículos disponibles aquí <https://api.jqueryui.com/theming/icons/>

Lea Iconos en línea: <https://riptutorial.com/es/jquery-ui/topic/4633/iconos>

Capítulo 10: jQuery ui sortable

Examples

jQuery UI Sortable - Drop Placeholder

Este ejemplo de Sortable que utiliza un marcador de posición es un uso común. Sortable se aplica a un grupo de elementos DOM, lo que permite al usuario mover elementos en la lista a través de las acciones de estilo Drag'n Drop.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Sortable - Drop Placeholder</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.0/themes/base/jquery-ui.css">
  <style>
#sortable {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 60%;
}
#sortable li {
  margin: 0 5px 5px 5px;
  padding: 5px;
  font-size: 1.2em;
  height: 1.5em;
}
html>body #sortable li {
  height: 1.5em; line-height: 1.2em;
}
.ui-state-highlight {
  height: 1.5em;
  line-height: 1.2em;
}
</style>
<script src="https://code.jquery.com/jquery-3.1.0.js"></script>
<script src="https://code.jquery.com/ui/1.12.0/jquery-ui.js"></script>
<script>
$( function() {
  $( "#sortable" ).sortable({
    placeholder: "ui-state-highlight"
  }).disableSelection();
});
</script>
</head>
<body>

<ul id="sortable">
  <li class="ui-state-default">Item 1</li>
  <li class="ui-state-default">Item 2</li>
  <li class="ui-state-default">Item 3</li>
  <li class="ui-state-default">Item 4</li>
  <li class="ui-state-default">Item 5</li>
</ul>
```



```
<li class="ui-state-default">Item 6</li>
<li class="ui-state-default">Item 7</li>
</ul>

</body>
</html>
```

Lea jQuery ui sortable en línea: <https://riptutorial.com/es/jquery-ui/topic/1226/jquery-ui-sortable>

Capítulo 11: jQuery UI Plug-in rotativo

Parámetros

Parámetro	Detalles
encargarse de	url a una imagen personalizada para el mango
ángulo	La rotación inicial del elemento.
rotationCenterX	Posición sobre la cual se girará el elemento.
centro de rotación	Posición sobre la cual se girará el elemento.
paso	un ángulo en grados al que se ajustará la rotación si se mantiene presionada la tecla de cambio.
chasquido	Se ajusta a paso en grados.
comienzo	se dispara cuando comienza la rotación
detener	se dispara cuando la rotación se detiene
girar	se dispara cuando el objeto está siendo girado
wheelRotate	activar / desactivar la rueda del ratón para rotar el elemento.

Examples

Ejemplo de uso inicial

jquery-ui-rotatable es un complemento para jQuery UI que funciona de manera similar a Draggable y Resizable, sin ser tan completo. De forma predeterminada, coloca un pequeño icono de rotación en la parte inferior izquierda del elemento que desee rotar.

```
<html>
  <head>
    <title>My Rotatable</title>
    <link rel="stylesheet" href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-
ui.css">
    <link rel="stylesheet"
href="//cdn.jsdelivr.net/jquery.ui.rotatable/1.0.1/jquery.ui.rotatable.css">
    <script src="http://code.jquery.com/jquery-1.11.3.js"></script>
    <script src="http://code.jquery.com/ui/1.11.4/jquery-ui.js"></script>
    <script
src="//cdn.jsdelivr.net/jquery.ui.rotatable/1.0.1/jquery.ui.rotatable.min.js"></script>
    <script>
```

```
$(function(){
  $('#target').rotatable();
});
</script>
</head>
<body>
<div id="target">Rotate me!</div>
</body>
</html>
```

Lea jQuery UI Plug-in rotativo en línea: <https://riptutorial.com/es/jquery-ui/topic/1806/jquery-ui-plugin-rotativo>

Capítulo 12: Ordenable

Sintaxis

- `$("# sortable"). sortable ({/ * Options Here * /}); // Inicializar Sortable`
- `$("# sortable"). sortable ("option", "option_name", option_value); // Establecer la opción fuera del inicializador`
- `valor var = $("# sortable"). sortable ("option", "option_name"); // Obtiene el valor de una opción`

Parámetros

Parámetro	Descripción
Opciones	
añadir a	(jQuery, Element, Selector, String) [Predeterminado: "parent"] El elemento al que se agrega el ayudante
eje	(Cadena) [Predeterminado: falso] Las direcciones en las que se puede arrastrar el elemento (x o y)
cancelar	(Selector) [Predeterminado: "input, textarea, button, select, option"] No se inicia la clasificación si se inicia en un elemento que coincida con el selector
clases	(Objeto) [Valor predeterminado: {}] Especifique clases adicionales para agregar a los elementos ordenables cuando se agregan las clases estructurales. ({ui-sortable-helper: custom_class})
conectar con	(Selector) [Predeterminado: falso] Permite arrastrar elementos de un ordenable a otro
contención	(Elemento, Selector, Cadena) [Predeterminado: falso] El elemento al que se limitan los elementos
cursor	(Cadena) [Predeterminado: "auto"] Define el tipo de cursor que se muestra al ordenar
cursorAt	(Objeto) [Predeterminado: falso] Define la posición desde la cual el ayudante parece que se está moviendo
discapacitado	(Booleano) [Predeterminado: falso] Desactiva la clasificación si es verdadero

Parámetro	Descripción
dropOnEmpty	(Booleano) [Predeterminado: verdadero] Si los elementos falsos de este ordenable no pueden colocarse en ordenadores vacíos
forceHelperSize	(Booleano) [Predeterminado: falso] Obliga al ayudante a tener un tamaño
forcePlaceholderSize	(Booleano) [Predeterminado: falso] Hace que el marcador de posición tenga un tamaño
cuadrícula	(Array) [Predeterminado: falso] Define una cuadrícula para ajustar el ayudante a ([x, y])
encargarse de	(Selector, Elemento) [Predeterminado: falso] Define los elementos en los que puede comenzar la clasificación. Opuesto a cancelar
ayudante	(Cadena, Función) [Predeterminado: "original"] Cadena "original" o "clon", o función que devuelve el elemento que se usará como ayudante.
artículos	(Selector) [Predeterminado: "> *"] Define los elementos que deben ordenarse
opacidad	(Número 0.01 a 1) [Predeterminado: falso] Define la opacidad del ayudante
marcador de posición	(Cadena) [Predeterminado: falso] Define una clase o clases que se aplicarán al marcador de posición
revertir	(Booleano, Número) [Predeterminado: falso] El tiempo que tarda el ayudante en deslizarse en su nueva posición
voluta	(Booleano) [Predeterminado: verdadero] Si se desplaza cuando está en los bordes de la página
scrollSensitivity	(Número) [Predeterminado: 20] Define qué tan cerca del borde de la página debe estar el cursor para comenzar a desplazarse
Velocidad de desplazamiento	(Número) [Predeterminado: 20] La velocidad a la que se desplaza
tolerancia	(Cadena) [Predeterminado: "intersecar"] Define qué modo usar cuando se calcula cuándo un elemento está sobre otro ("intersecar" o "puntero")
índice de z	(Entero) [Predeterminado: 1000] Define el índice z del ayudante al ordenar
Métodos	

Parámetro	Descripción
cancelar()	Cancela la clasificación actual y devuelve los elementos a su posición antes de que comience la clasificación
destruir()	Elimina la funcionalidad clasificable y devuelve el elemento a su estado inicialización inicial
inhabilitar()	Desactiva el ordenable
habilitar()	Habilita el ordenable
ejemplo()	Devuelve el objeto de instancia sortable
opción()	Obtiene pares de valores clave de todas las opciones para el ordenable.
opción (String)	Obtiene el valor de una opción
Opción (String, Cualquiera)	Establece el valor de la opción especificada por la Cadena.
opción (Objeto)	Establece una o más opciones con el objeto siendo pares de valores clave de opciones
refrescar()	Actualiza las opciones de ordenación volviendo a cargar todos los elementos ordenables. Esto hace que los nuevos elementos sean reconocidos
refreshPositions ()	Actualiza las posiciones almacenadas en caché de los elementos que se pueden ordenar
serializar (Objeto)	Serializa los identificadores de los elementos (de manera predeterminada) en una cadena que se puede enviar o anexar a una url Opciones de objeto: {clave: establece la clave en la cadena serializada, atributo: [Id. Predeterminada] establece el atributo a mirar, expresión : [Valor predeterminado: "/(.+)-=_/"] expresiones regulares para dividir el atributo en pares de valores clave}
toArray (objeto)	Serializa el id de los elementos que se pueden ordenar en una matriz. El objeto puede contener un atributo de parámetro que tiene el atributo para colocar en la matriz por defecto es id
widget ()	Devuelve un objeto jQuery del elemento clasificable.
Eventos	
activar (evento, ui)	Se dispara cuando la lista está conectada, cada lista conectada en el inicio de arrastre la recibe

Parámetro	Descripción
beforeStop (evento, ui)	Los disparadores antes de la clasificación se detienen cuando el ayudante ha sido desplazado a la misma posición que el marcador de posición
cambio (evento, ui)	Se dispara cuando los elementos cambian de posición, es decir. cuando el marcador de posición se mueve
crear (evento, ui)	Se dispara cuando se puede ordenar
desactivar (evento, ui)	Se dispara cuando la clasificación se detiene. Esto va a todas las listas conectadas también
fuera (evento, ui)	Se desencadena cuando el elemento se mueve fuera de la lista clasificable
sobre (evento, ui)	Se dispara cuando el elemento se mueve a una lista clasificable
recibir (evento, ui)	Se desencadena cuando un elemento de una lista conectada se ha colocado en otro. El objetivo es la lista de receptores.
eliminar (evento, ui)	Se desencadena cuando un elemento de una lista conectada se ha colocado en otro. El objetivo es la lista de dar.
ordenar (evento, ui)	Activado durante la clasificación
inicio (evento, ui)	Se dispara cuando comienza la clasificación
detener (evento, ui)	Se dispara cuando la clasificación se detiene.
actualizar (evento, ui)	Se dispara cuando la clasificación se detiene y la posición DOM se ha actualizado

Observaciones

Documentación oficial [aquí](#)

Examples

Ejemplo simple

Tome cualquier lista y agregue un identificador a la envoltura externa (`ul` , `div`)

```
<ul id="sortable">
<li>Item 1</li>
<li>Item 2</li>
<li>Item 3</li>
<li>Item 4</li>
```

```
</ul>
```

En tu jQuery:

```
$(function() {  
    $('#sortable').sortable({  
        //pass all options in here  
    });  
});
```

Esto permitirá arrastrar y soltar en la lista todos los `li` en el `#sortable` contenedor `#sortable`

Rejilla clasificable con diseño flexible

Esto usó el diseño flexible con el ordenable para crear una cuadrícula de cajas sensibles que se pueden mover arrastrando y soltando.

HTML

```
<div id="sortable">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
  <div>4</div>  
  <div>5</div>  
</div>
```

JS

```
$(function() {  
    $('#sortable').sortable({  
        //pass all options in here  
    });  
});
```

CSS

```
#sortable{  
    width: 500px;  
    display: flex;  
    flex-wrap: wrap;  
}  
#sortable div {  
    margin: 10px;  
    background-color: #f00;  
    flex-basis: 100px;  
    height: 100px;  
}
```

Artículos estacionarios al arrastrar

Este ejemplo utiliza una clase en el marcador de posición para convertirlo en una línea y hacer

que no ocupe espacio.

HTML

```
<div id="sortable">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
```

JS

```
$("#sortable").sortable({
  placeholder: 'placeholder',
  helper: 'clone',
  start: function(event, ui){
    ui.item.show();
  }
});
```

CSS

```
#sortable div{
  background-color: #f00;
  width: 50px;
  height: 50px;
  margin: 10px;
  padding: 0px;
}
#sortable div.placeholder{
  height: 4px;
  margin: -7px 10px;
}
```

Ordenable - Animar la reversión de un artículo no aceptado

Ejemplo de trabajo: <https://jsfiddle.net/Twisty/4f5yh3pa/7/>

La cancelación y la reversión de una orden de clasificación no están fuertemente documentadas. La ayuda muestra cómo se puede cancelar condicionalmente el traslado de un elemento de una lista a otra lista conectada. de forma predeterminada, esto no está animado por ordenable, este ejemplo incluye una animación.

Resultado: la Lista # 2 solo acepta artículos que tienen una clase de `acceptable` . Ambas listas se pueden clasificar de forma natural de lo contrario.

HTML

```
<div class="ui-widget">
  <ul id="sortable1" class="connectedSortable">
    <li class="ui-state-default acceptable">Item 1</li>
    <li class="ui-state-default">Item 2</li>
```

```

    <li class="ui-state-default">Item 3</li>
    <li class="ui-state-default">Item 4</li>
    <li class="ui-state-default">Item 5</li>
</ul>
<ul id="sortable2" class="connectedSortable">
    <li class="ui-state-default">Item 6</li>
    <li class="ui-state-default acceptable">Item 7</li>
</ul>
</div>

```

CSS

```

.ui-widget {
    position: relative;
}

.connectedSortable {
    border: 1px solid #eee;
    width: 142px;
    min-height: 20px;
    list-style-type: none;
    margin: 0;
    padding: 5px 0 0 0;
    float: left;
    margin-right: 10px;
}

#sortable1 {
    background: #fff;
}

#sortable2 {
    background: #999;
}

.connectedSortable li {
    margin: 0 5px 5px 5px;
    padding: 5px;
    font-size: 1.2em;
    width: 120px;
}

```

JavaScript

```

$(function() {
    $(".connectedSortable").sortable({
        connectWith: ".connectedSortable",
        receive: function(e, ui) {
            var $self = $(this);
            var $item = ui.item;
            var $sender = ui.sender;
            // Restrict condition to only one specific list if desired
            if ($(e.target).attr("id") == "sortable2") {
                if ($item.hasClass("acceptable")) {
                    // Item Accepted
                    console.log($self.attr("id") + " accepted item from: #" + $sender.attr("id") + " > "
+ $item.text());
                } else {
                    // Item Rejected
                }
            }
        }
    });

```

```
        console.log($self.attr("id") + " rejected item from: #" + $sender.attr("id") + " > "
+ $item.text());
        // Animate the return of the items position
        $item.css("position", "absolute").animate(ui.originalPosition, "slow", function() {
            // Return the items position control to it's parent
            $item.css("position", "inherit");
            // Cancel the sortable action to return it to it's origin
            $sender.sortable("cancel");
        });
    }
}
}
}).disableSelection();
});
```

Lea Ordenable en línea: <https://riptutorial.com/es/jquery-ui/topic/1415/ordenable>

Capítulo 13: Selector de fechas

Examples

Inicialización

El **planificador de evento** se utiliza para mostrar un selector de fecha interactiva que está vinculado a un campo de formulario de entrada estándar. Facilita la selección de la fecha para las tareas de entrada y también es altamente configurable.

Cualquier campo de entrada se puede enlazar con el selector de fecha jquery-ui mediante el selector del campo de entrada (id, clase, etc.) utilizando el método **datepicker ()** como este:

```
<input type="text" id="datepicker">
<script>
    $("#datepicker").datepicker();
</script>
```

Demostración en vivo está [aquí](#) .

Configuración de fechas mínimas y máximas para un selector de fechas

```
<script>
$( ".inclas" ).datepicker({
    minDate: new Date(2007, 1 - 1, 1)
    maxDate: new Date(2008, 1 - 1, 1)
});
</script>

<input type="text" id="datepick" class="inclas">
```

Mostrar semana del año

El siguiente código mostrará el número de la semana del año en el lado izquierdo del selector de fecha. Por defecto, el inicio semanas el lunes, pero se puede personalizar usando `firstDay` opción. La primera semana del año contiene el primer jueves del año, según la definición de la norma ISO 8601.

```
<input type="text" id="datepicker">
<script>
    $("#datepicker").datepicker({
        showWeek: true
    });
</script>
```

Establecer un formato de fecha personalizado

Formato de fecha predeterminado: "mm / dd / aa"

El siguiente ejemplo muestra cómo puede configurar el formato de fecha en la inicialización con la opción `dateFormat`.

```
<input type="text" id="datepicker">
<script>
  $("#datepicker").datepicker({
    dateFormat: "yy-mm-dd"
  });
</script>
```

El siguiente ejemplo muestra cómo puede configurar el formato de fecha después de la inicialización con la opción `dateFormat`.

```
<input type="text" id="datepicker">
<script>
  $("#datepicker").datepicker( "option", "dateFormat", "yy-mm-dd" );
</script>
```

Puede utilizar combinaciones de los siguientes:

```
d - day of month (no leading zero)
dd - day of month (two digit)
o - day of the year (no leading zeros)
oo - day of the year (three digit)
D - day name short
DD - day name long
m - month of year (no leading zero)
mm - month of year (two digit)
M - month name short
MM - month name long
y - year (two digit)
yy - year (four digit)
@ - Unix timestamp (ms since 01/01/1970)
! - Windows ticks (100ns since 01/01/0001)
'...' - literal text
' - single quote
anything else - literal text
```

O estándar predefinido:

```
ATOM - 'yy-mm-dd' (Same as RFC 3339/ISO 8601)
COOKIE - 'D, dd M yy'
ISO_8601 - 'yy-mm-dd'
RFC_822 - 'D, d M y' (See RFC 822)
RFC_850 - 'DD, dd-M-y' (See RFC 850)
RFC_1036 - 'D, d M y' (See RFC 1036)
RFC_1123 - 'D, d M yy' (See RFC 1123)
RFC_2822 - 'D, d M yy' (See RFC 2822)
RSS - 'D, d M y' (Same as RFC 822)
TICKS - '!'
TIMESTAMP - '@'
W3C - 'yy-mm-dd' (Same as ISO 8601)
```

Se puede aplicar un formato de fecha predeterminado a todos los seleccionadores de fechas usando el siguiente ejemplo:

```
<script>
  $.datepicker.setDefaults({
    dateFormat: "yy-mm-dd"
  });
</script>
```

Mostrar mes y año desplegable

jQuery datepicker tiene dos opciones para permitir mostrar desplegables para la selección de mes y año. Estas opciones hacen que la navegación a través de grandes marcos de tiempo sea más fácil.

```
<input type="text" id="datepicker">
<script>
  $("#datepicker").datepicker({
    changeMonth: true, // shows months dropdown
    changeYear: true  // shows years dropdown
  });
</script>
```

Lea Selector de fechas en línea: <https://riptutorial.com/es/jquery-ui/topic/520/selector-de-fechas>

Creditos

S. No	Capítulos	Contributors
1	Comenzando con jQuery UI Library	Community , J F , jkdev , JonasCz , Twisty
2	Acordeón	Dipen Shah , Madalina Taina
3	Arrastrable	empiric , J F
4	Autocompletar	J F
5	Botón	Theodore K.
6	Deslizador	Jonathan Michalik
7	Diálogo	J F , Jonathan Michalik , Racil Hilan , Theodore K. , Twisty
8	Hiladero	Alon Eitan , Andrew Mcghie , depperm
9	Iconos	Theodore K.
10	jQuery ui sortable	Andrew Mcghie , Twisty
11	jQuery UI Plug-in rotativo	Twisty
12	Ordenable	Alon Eitan , Andrew Mcghie , M B , Twisty
13	Selector de fechas	cteski , J F , ni8mr , Nishant123 , Peter Tirrell , Pradeep , smdsgn , VincenzoC