



EBook Gratuito

APPENDIMENTO

jQuery UI Library

Free unaffiliated eBook created from
Stack Overflow contributors.

#jquery-ui

Sommario

Di.....	1
Capitolo 1: Introduzione alla libreria dell'interfaccia utente jQuery.....	2
Osservazioni.....	2
Versioni.....	2
Examples.....	3
Aggiunta dello script dell'interfaccia utente jQuery e utilizzo di base.....	4
Impostazione dell'interfaccia utente jQuery per il primo esempio.....	4
Capitolo 2: Completamento automatico.....	6
Examples.....	6
Semplice esempio.....	6
Capitolo 3: Date picker.....	7
Examples.....	7
Inizializzazione.....	7
Impostazione delle date minime e massime per un datepicker.....	7
Mostra settimana dell'anno.....	7
Imposta un formato data personalizzato.....	7
Mostra il menu a discesa mese e anno.....	9
Capitolo 4: Dialogo.....	10
Sintassi.....	10
Parametri.....	10
Osservazioni.....	13
Examples.....	13
Semplice esempio.....	13
Apri la finestra di dialogo quando si verifica un evento.....	13
Esempio complesso - Interfaccia utente jQuery Creazione dinamica della finestra di dialogo.....	14
Creazione di una finestra di dialogo con barra del titolo a schede.....	17
Finestra di dialogo senza pulsante di chiusura.....	19
Capitolo 5: Fisarmonica.....	20
Sintassi.....	20
Parametri.....	20

Osservazioni.....	20
Examples.....	20
Usa base per fisarmonica.....	20
La fisarmonica distrugge l'uso.....	21
Fisarmonica disabilita l'uso.....	21
La fisarmonica abilita l'uso.....	21
Opzione di fisarmonica.....	21
L'aggiornamento della fisarmonica.....	22
Utilizzo del widget di Fisarmonica.....	22
Capitolo 6: Icone.....	23
Sintassi.....	23
Osservazioni.....	23
Examples.....	23
Utilizzo di base.....	23
Capitolo 7: jquery ui ordinabile.....	24
Examples.....	24
jQuery UI Sortable - Drop Placeholder.....	24
Capitolo 8: Ordinabile.....	26
Sintassi.....	26
Parametri.....	26
Osservazioni.....	29
Examples.....	29
Semplice esempio.....	29
Griglia ordinabile con layout flessibile.....	29
Oggetti stazionari durante il trascinamento.....	30
Ordinabile - Ripristina animate di elementi non accettati.....	31
Capitolo 9: Plugin girevole jQuery UI.....	33
Parametri.....	33
Examples.....	33
Esempio di utilizzo iniziale.....	33
Capitolo 10: Pulsante.....	35
Sintassi.....	35

Parametri.....	35
Examples.....	35
Utilizzo di base.....	35
Capitolo 11: Slider.....	36
Examples.....	36
Semplice esempio.....	36
Range Slider.....	36
Inizializzazione di valori e limiti di valore.....	36
Utilizzando l'evento diapositiva.....	37
Impostazione dei valori e l'evento di modifica.....	37
Capitolo 12: Spinner.....	39
Sintassi.....	39
Parametri.....	39
Osservazioni.....	39
Examples.....	39
Esempio di base.....	39
Capitolo 13: trascinabili.....	40
Examples.....	40
Semplice esempio.....	40
Trascinabile con manico.....	40
Titoli di coda.....	41

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

Capitolo 1: Introduzione alla libreria dell'interfaccia utente jQuery

Osservazioni

jQuery UI è una libreria UI JavaScript, costruita su jQuery, che offre un insieme di interazioni, effetti e widget dell'interfaccia utente.

Versioni

Versione	Data di rilascio
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

Versione	Data di rilascio
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

Aggiunta dello script dell'interfaccia utente jQuery e utilizzo di base

Per iniziare con la libreria dell'interfaccia utente jQuery, devi aggiungere lo script jQuery, lo script dell'interfaccia utente jQuery e il foglio di stile dell'interfaccia utente jQuery nel codice HTML.

Innanzitutto, [scarica l'](#)interfaccia utente di jQuery; scegli le funzionalità necessarie nella pagina di download. Decomprimi il download e metti `jquery-ui.css` e `jquery-ui.js` (e `jquery.js`) in una cartella in cui puoi usarli dal tuo HTML (ad es. Con gli altri tuoi script e fogli di stile).

L'interfaccia utente di jQuery dipende da jQuery, quindi ricorda di includere `jquery.js` prima di `jquery-ui.js`.

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

Questo è tutto! Ora puoi utilizzare l'interfaccia utente di jQuery. Ad esempio, usa il datepicker con il seguente codice HTML:

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

Quindi utilizzare il seguente codice JavaScript:

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

Che ti farà avere un bel popup di Datepicker:



Per ulteriori informazioni, consulta la pagina [ufficiale "Come iniziare"](#).

Impostazione dell'interfaccia utente jQuery per il primo esempio

La struttura dell'interfaccia utente jQuery consente di estendere e aumentare i controlli dell'interfaccia utente per la libreria JavaScript jQuery.

Quando si desidera utilizzare l'interfaccia utente di jQuery, è necessario aggiungere queste librerie al codice HTML. Un modo rapido per iniziare è utilizzare le fonti di codice disponibili di Content

Delivery Network:

librerie jQuery

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

Puoi scegliere diversi temi per l'interfaccia utente di jQuery e persino il Tuo tema. Per questo esempio, useremo "Smoothness". Si aggiunge il tema tramite CSS.

jQuery UI CSS

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

Mettere tutto insieme

Una volta scaricato o selezionato il tuo CDN, ora vorrai aggiungere queste librerie e fogli di stile al tuo codice HTML in modo che la tua pagina web possa ora utilizzare l'interfaccia utente jQuery e jQuery. L'ordine in cui si caricano le librerie è importante. Chiama prima la libreria jQuery e poi la libreria dell'interfaccia utente jQuery. Poiché jQuery UI estende jQuery, deve essere chiamato dopo. Il codice HTML potrebbe essere simile al seguente.

```
<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>
```

Leggi [Introduzione alla libreria dell'interfaccia utente jQuery online](https://riptutorial.com/it/jquery-ui/topic/513/introduzione-alla-libreria-dell-interfaccia-utente-jquery): <https://riptutorial.com/it/jquery-ui/topic/513/introduzione-alla-libreria-dell-interfaccia-utente-jquery>

Capitolo 2: Completamento automatico

Examples

Semplice esempio

I widget di completamento automatico forniscono suggerimenti durante la digitazione nel 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' />
```

Leggi Completamento automatico online: <https://riptutorial.com/it/jquery-ui/topic/519/completamento-automatico>

Capitolo 3: Date picker

Examples

Inizializzazione

Il **datepicker** viene utilizzato per mostrare un selettore di data interattivo che è legato a un campo di input del modulo standard. Rende molto semplice la selezione della data per le attività di input ed è anche altamente configurabile.

Qualsiasi campo di input può essere associato a jquery-ui datepicker dal selettore del campo di input (id, classe ecc.) Utilizzando il metodo **datepicker ()** come questo -

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

La demo live è [qui](#) .

Impostazione delle date minime e massime per un datepicker

```
<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">
```

Mostra settimana dell'anno

Il seguente codice mostrerà il numero della settimana dell'anno sul lato sinistro di DatePicker. Per impostazione predefinita, la settimana inizia lunedì, ma può essere personalizzata utilizzando `firstDay` opzione `firstDay` . La prima settimana dell'anno contiene il primo giovedì dell'anno, seguendo la definizione ISO 8601.

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

Imposta un formato data personalizzato

Formato data predefinito: "mm / gg / aa"

L'esempio seguente mostra come è possibile impostare il formato della data in fase di inizializzazione con l'opzione `dateFormat`.

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

L'esempio seguente mostra come è possibile impostare il formato della data dopo l'inizializzazione con l'opzione `dateFormat`.

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

Puoi usare combinazioni di quanto segue:

```
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 standard predefinito:

```
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)
```

Un formato data predefinito può essere applicato a tutti i datepicker usando il seguente esempio:

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

Mostra il menu a discesa mese e anno

jQuery datepicker ha due opzioni per consentire la visualizzazione di menu a discesa per la selezione di mesi e anni. Queste opzioni facilitano la navigazione attraverso i grandi orizzonti temporali.

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

Leggi Date picker online: <https://riptutorial.com/it/jquery-ui/topic/520/date-picker>

Capitolo 4: Dialogo

Sintassi

- `$ ("selector") .dialog ("option", "disabled");` // Opzione Getter, specifica
- `$ ("selector") .dialog ("opzione");` // Opzione Ottieni tutto
- `$ ("selector") .dialog ("option", "disabled", true);` // Opzione Setter, specifica
- `$ ("selector") .dialog ("option", {disabled: true});` // Opzione Setter, multiplo
- `$ ("selector") .dialog ("close");` // I trigger si chiudono
- `$ ("selector") .dialog ({chiudi: function () {}});` // Chiudi sovraccarico
- `$ ("selector") .on ("dialogclose", function (event, ui) {});` // Chiudi sovraccarico

Parametri

Parametro	Descrizione
Opzioni	
appendTo	(Selettore) [Predefinito: "body"] A quale elemento deve essere aggiunta la finestra di dialogo (e la sovrapposizione, se modale).
AutoOpen	(Boolean) [Default: <code>true</code>] Se impostato su <code>true</code> , la finestra di dialogo si aprirà automaticamente al momento dell'inizializzazione. Se <code>false</code> , la finestra di dialogo rimarrà nascosta finché non verrà chiamato il metodo <code>open ()</code> .
pulsanti	(Oggetto / Array) Specifica quali pulsanti dovrebbero essere visualizzati nella finestra di dialogo. Il contesto del callback è l'elemento di dialogo; se è necessario accedere al pulsante, è disponibile come destinazione dell'oggetto evento.
closeOnEscape	(Booleano) [Predefinito: <code>true</code>] Specifica se la finestra di dialogo deve chiudersi quando è attivo e l'utente preme il tasto Esc (ESC).
closetext	(String) [Default: "close"] Specifica il testo per il pulsante di chiusura. Nota che il testo vicino è visibilmente nascosto quando si usa un tema standard.
dialogClass	(String) I nomi di classe specificati verranno aggiunti alla finestra di dialogo, per ulteriori tematizzazioni.
trascinabili	(Boolean) [Default: <code>true</code>] Se impostato su <code>true</code> , la finestra di dialogo sarà trascinata dalla barra del titolo. Richiede il widget Draggable dell'interfaccia utente jQuery da includere.
altezza	(Numero / stringa) [Impostazione predefinita: "auto"] L'altezza della

Parametro	Descrizione
	finestra di dialogo.
nascondere	(Bool / Num / Str / Obj) Se e come animare l'occultamento della finestra di dialogo.
altezza massima	(Numero) [Predefinito: <code>false</code>] L'altezza massima a cui è possibile ridimensionare la finestra di dialogo, in pixel.
larghezza massima	(Numero) [Predefinito: <code>false</code>] La larghezza massima a cui è possibile ridimensionare la finestra di dialogo, in pixel.
minHeight	(Numero) [Predefinito: <code>150</code>] L'altezza minima a cui è possibile ridimensionare la finestra di dialogo, in pixel.
minWidth	(Numero) [Predefinito: <code>150</code>] La larghezza minima a cui è possibile ridimensionare la finestra di dialogo, in pixel.
modale	(Boolean) [Default: <code>false</code>] Se impostato su <code>true</code> , la finestra di dialogo avrà un comportamento modale; gli altri elementi sulla pagina saranno disabilitati, cioè non possono essere interagiti con. Le finestre di dialogo modali creano una sovrapposizione sotto la finestra di dialogo ma sopra gli altri elementi della pagina.
posizione	(Oggetto) [Default: <code>{ my: "center", at: "center", of: window }</code>] Specifica dove deve essere visualizzata la finestra di dialogo all'apertura. La finestra di dialogo gestirà le collisioni in modo tale che la maggior parte della finestra di dialogo sia visibile il più possibile.
ridimensionabile	(Boolean) [Default: <code>true</code>] Se impostato su <code>true</code> , la finestra di dialogo sarà ridimensionabile. Richiede il widget ridimensionabile dell'interfaccia utente jQuery da includere.
mostrare	(Bool / Num / Str / Obj) Se e come animare la visualizzazione della finestra di dialogo.
titolo	(String) Specifica il titolo della finestra di dialogo. Se il valore è <code>null</code> , verrà utilizzato l'attributo <code>title</code> sull'elemento <code>source</code> della finestra di dialogo.
larghezza	(Numero) [Predefinito: <code>300</code>] La larghezza della finestra di dialogo, in pixel.
metodi	
vicino	Chiude la finestra di dialogo.
distruggere	Rimuove completamente la funzionalità di dialogo. Ciò restituirà l'elemento al suo stato pre-init.
esempio	Recupera l'oggetto istanza del dialog. Se l'elemento non ha un'istanza

Parametro	Descrizione
	associata, viene restituito undefined.
è aperto	Se la finestra di dialogo è attualmente aperta.
moveToTop	Sposta la finestra di dialogo nella parte superiore della pila di finestre di dialogo.
Aperto	Apri la finestra di dialogo.
opzione	Ottiene il valore attualmente associato <code>optionName</code> specificato.
opzione	Ottiene un oggetto contenente coppie chiave / valore che rappresentano l'hash delle opzioni di dialogo corrente.
opzione	Imposta una o più opzioni per la finestra di dialogo.
widget di	Restituisce un oggetto jQuery contenente il wrapper generato.
Punti di estensione	
<code>_allowInteraction</code>	(evento) Le finestre di dialogo modali non consentono agli utenti di interagire con gli elementi dietro la finestra di dialogo. Questo può essere problematico per elementi che non sono figli della finestra di dialogo, ma sono posizionati in modo assoluto per apparire come se lo fossero. Il metodo <code>_allowInteraction()</code> determina se l'utente deve poter interagire con un determinato elemento di destinazione; pertanto, può essere utilizzato per autorizzare elementi che non sono figli della finestra di dialogo, ma che si desidera che gli utenti siano in grado di utilizzare.
eventi	
BeforeClose	(evento, ui) Attivato quando una finestra di dialogo sta per chiudere. Se annullato, la finestra di dialogo non verrà chiusa.
vicino	(evento, ui) Attivato quando la finestra di dialogo è chiusa.
creare	(evento, ui) Attivato quando viene creata la finestra di dialogo.
trascinare	(evento, ui {posizione, sfalsamento}) Attivato mentre la finestra di dialogo viene trascinata.
dragStart	(event, ui {position, offset}) Attivato quando l'utente inizia a trascinare la finestra di dialogo.
DragStop	(evento, ui {posizione, offset}) Attivato dopo che la finestra di dialogo è stata trascinata.
messa a fuoco	(evento, ui) Attivato quando il dialogo diventa attivo.

Parametro	Descrizione
Aperto	(evento, ui) Attivato quando viene aperta la finestra di dialogo.
ridimensionare	(evento, ui {originalPosition, position, originalSize, size}) Attivato mentre la finestra di dialogo viene ridimensionata.
resizeStart	(evento, ui {originalPosition, position, originalSize, size}) Attivato mentre la finestra di dialogo viene ridimensionata.
resizeStop	(evento, ui {originalPosition, position, originalSize, size}) Attivato mentre la finestra di dialogo viene ridimensionata.

Osservazioni

Fonte dei parametri: <http://api.jqueryui.com/dialog/>

Examples

Semplice esempio

La finestra di dialogo è una finestra sovrapposta posizionata all'interno del viewport.

```
<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>
```

Apri la finestra di dialogo quando si verifica un evento

Di solito vogliamo separare la creazione del dialogo dal suo aspetto. Quindi sono necessari tre passaggi.

1. Crea un 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. Crea una finestra di dialogo, nota l'opzione `autoOpen: false` che assicura che verrà chiusa all'inizio

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

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

3. Aprilo quando necessario, come su un pulsante

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

Esempio complesso - Interfaccia utente jQuery Creazione dinamica della finestra di dialogo

Generalmente, il dialogo si basa su un `div` all'interno dell'HTML. A volte potresti voler creare una finestra di dialogo da zero, a livello di programmazione. Ecco un esempio di una finestra di dialogo modale complessa creata dinamicamente con funzioni interattive.

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");
});
});

```

Esempio di lavoro per riferimento: <https://jsfiddle.net/Twisty/LqjuxLu1/>

Creazione di una finestra di dialogo con barra del titolo a schede

Occasionalmente, potremmo voler visualizzare finestre di dialogo con più di un pannello di contenuti. L'interfaccia utente di jQuery offre schede che possono essere utilizzate in tandem con una finestra di dialogo per renderlo possibile. Anche se potrebbe essere più comune disporre di schede all'interno del contenitore del contenuto di una finestra di dialogo, questo esempio mostrerà come creare un elenco di schede nella barra del titolo della finestra di dialogo.

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");
});
});
```

Esempio di lavoro per riferimento: <https://jsfiddle.net/5x4zz681/>

Finestra di dialogo senza pulsante di chiusura

Se ti piace mostrare la finestra di dialogo senza il pulsante Chiudi (cioè il pulsante x nell'angolo in alto a destra della finestra di dialogo), forse perché vuoi forzare l'utente a selezionare una delle opzioni o pulsanti nella finestra di dialogo stessa:

1- Dai al tuo dialogo una classe CSS:

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

2- Nascondi il pulsante di chiusura usando questo CSS:

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

Nota: se si desidera nascondere l'intera barra del titolo, utilizzare invece questo CSS:

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

In alternativa, puoi nascondere il pulsante Chiudi nel codice di inizializzazione della finestra di dialogo:

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

Leggi Dialogo online: <https://riptutorial.com/it/jquery-ui/topic/521/dialogo>

Capitolo 5: Fisarmonica

Sintassi

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

Parametri

Parametro	Dettaglio
attivo	Digita Boolean o Integer, Boolean richiede che sia possibile comprimere
animare	Digita Boolean, Number, String o Object
collassabile	Digita booleano
evento	Digita stringa
intestazione	Digita l'elemento del selettore
heightStyle	Digita stringa
icone	Digita oggetto icona jQuery UI

Osservazioni

Ulteriori informazioni possono essere trovate qui: <http://api.jqueryui.com/accordion/>

Examples

Uso base per fisarmonica

Per utilizzare una fisarmonica, è necessario avere intestazioni e contenuti all'interno delle intestazioni nel loro codice HTML. Quindi è necessario istanziare il metodo `accordion()` dell'interfaccia utente di jQuery.

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

Nel codice HTML:

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



La fisarmonica distrugge l'uso

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

Questo rimuoverà completamente la funzionalità della fisarmonica e mostrerà l'HTML predefinito rimuovendo tutti gli elementi dell'interfaccia jQuery.

Questo metodo non accetta argomenti.

Fisarmonica disabilita l'uso

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

Questo metodo disabiliterà la fisarmonica, ovvero le intestazioni non sono selezionabili rendendo il contenuto di sola lettura e statico.

Questo metodo non accetta argomenti.

La fisarmonica abilita l'uso

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

Questo metodo consentirà una fisarmonica. Ciò consentirà una fisarmonica disabilitata o semplicemente non fare nulla su una fisarmonica già abilitata.

Questo metodo non accetta argomenti.

Opzione di fisarmonica

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

Ciò restituirà un oggetto PlainObject che fornisce tutte le opzioni che rappresentano la fisarmonica selezionata. Questo conterrà tutti i valori dei tasti che sono spiegati nella sezione Parametri.

Questo metodo prende i parametri che sono le opzioni di base di base spiegate nel parametro. Si possono impostare le opzioni in questo modo:

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

L'aggiornamento della fisarmonica

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

Questo metodo ricalcola l'altezza dei pannelli di fisarmonica se intestazioni o contenuti sono stati aggiunti o rimossi nel DOM.

Utilizzo del widget di Fisarmonica

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

Questo metodo restituisce un oggetto jQuery contenente la fisarmonica.

Leggi Fisarmonica online: <https://riptutorial.com/it/jquery-ui/topic/630/fisarmonica>

Capitolo 6: Icone

Sintassi

- `.ui-icon- {tipo di icona} - {icon sub description} - {direction}`

Osservazioni

Le icone sono anche integrate in una serie di widget dell'interfaccia utente di jQuery, come ad esempio [fisarmonica](#) , pulsante, menu.

Examples

Utilizzo di base

Per una freccia spessa che punta a nord in un intervallo, aggiungi le classi `ui-icon` e `ui-icon-arrowthick-1-n` :

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

Per un triangolo che punta a sud in uno span, aggiungi classi `ui-icon` e `ui-icon-triangle-1-s` :

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

Elenco completo degli articoli disponibili qui <https://api.jqueryui.com/theming/icons/>

Leggi Icone online: <https://riptutorial.com/it/jquery-ui/topic/4633/icone>

Capitolo 7: jquery ui ordinabile

Examples

jQuery UI Sortable - Drop Placeholder

Questo esempio di Sortable usando un segnaposto è un uso comune. Sortable si applica a un gruppo di elementi DOM, consentendo all'utente di spostare gli oggetti nella lista tramite le azioni di stile 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>
```

Leggi jquery ui ordinabile online: <https://riptutorial.com/it/jquery-ui/topic/1226/jquery-ui-ordinabile>

Capitolo 8: Ordinabile

Sintassi

- `$ ("# ordinabile"). ordinabile ({/ * Opzioni qui * /}); // Inizializzabile ordinabile`
- `$ ("# ordinabile"). sortable ("option", "option_name", option_value); // Imposta l'opzione all'esterno dell'inizializzatore`
- `var value = $ ("# ordinabile"). sortable ("option", "option_name"); // Ottiene il valore di un'opzione`

Parametri

Parametro	Descrizione
Opzioni	
appendTo	(jQuery, Element, Selector, String) [Default: "parent"] L'elemento a cui è aggiunto l'helper
asse	(String) [Default: false] Le direzioni che l'oggetto può essere trascinato (x o y)
Annulla	(Selettore) [Default: "input, textarea, button, select, option"] Non inizia l'ordinamento se inizi su un elemento che corrisponde al selettore
classi	(Oggetto) [Predefinito: {}] Specificare classi aggiuntive da aggiungere agli elementi tipibili quando vengono aggiunte le classi strutturali. ({ui-sortable-helper: custom_class})
connesso con	(Selettore) [Predefinito: falso] Consente di trascinare elementi da un ordinabile all'altro
contenimento	(Elemento, Selettore, Stringa) [Default: falso] L'elemento a cui gli oggetti sono vincolati
cursore	(Stringa) [Impostazione predefinita: "auto"] Definisce il tipo di cursore da visualizzare durante l'ordinamento
cursorAt	(Oggetto) [Predefinito: falso] Definisce la posizione in cui l'helper ha l'aspetto del suo spostamento
Disabilitato	(Boolean) [Default: false] Disabilita l'ordinamento se vero
dropOnEmpty	(Boolean) [Predefinito: true] Se non è possibile inserire elementi falsi

Parametro	Descrizione
	da questo ordinabile in sibili vuoti
forceHelperSize	(Boolean) [Default: false] Forza l'helper ad avere una dimensione
forcePlaceholderSize	(Boolean) [Default: false] Forza il segnaposto a una dimensione
griglia	(Matrice) [Default: falso] Definisce una griglia per far scattare l'helper su ([x, y])
maniglia	(Selettore, Elemento) [Default: falso] Definisce gli elementi su cui può iniziare l'ordinamento. Opposto di cancellare
aiutante	(String, Function) [Default: "original"] String "original" o "clone", o funzione che restituisce l'elemento da utilizzare come helper.
elementi	(Selettore) [Default: "> *"] Definisce gli elementi che dovrebbero essere ordinabili
opacità	(Numero da 0,01 a 1) [Default: falso] Definisce l'opacità per l'helper
segnaposto	(String) [Default: false] Definisce una classe o classi da applicare al segnaposto
ritornare	(Boolean, Number) [Default: false] Il tempo che impiega l'helper a scivolare nella sua nuova posizione
scorrere	(Boolean) [Default: true] Indica se scorrere verso i bordi della pagina
scrollSensitivity	(Numero) [Predefinito: 20] Definisce quanto vicino al bordo della pagina deve essere il cursore per iniziare a scorrere
Scrollspeed	(Numero) [Default: 20] La velocità con cui scorrere
tolleranza	(Stringa) [Default: "intersecare"] Definisce quale modalità utilizzare quando si calcola quando un elemento si trova su un altro ("intersecare" o "puntatore")
zIndex	(Intero) [Predefinito: 1000] Definisce l'indice z dell'helper durante l'ordinamento
metodi	
Annula()	Annulla l'ordinamento corrente e riporta gli elementi nella loro posizione prima che inizi l'ordinamento
distruggere()	Rimuove la funzionalità ordinabile e riporta l'elemento alla sua inizializzazione dello stato
disattivare()	Disabilita l'ordinabile

Parametro	Descrizione
abilitare()	Abilita l'ordinabile
esempio()	Restituisce l'oggetto istanza Sables
opzione()	Ottiene coppie di valori chiave di tutte le opzioni per l'ordinabile
Opzione (String)	Ottiene il valore di un'opzione
opzione (String, Any)	Imposta il valore dell'opzione specificata dalla stringa
Opzione (Object)	Imposta una o più opzioni con l'oggetto essendo coppie di valori chiave di opzioni
ricaricare()	Aggiorna le opzioni ordinabili ricaricando tutti gli oggetti ordinabili. Questo fa sì che nuovi elementi vengano riconosciuti
refreshPositions ()	Rrefrisce le posizioni nascoste degli oggetti ordinabili
serializzare (Object)	Serializza gli elementi item (per impostazione predefinita) in una stringa che può essere inviata o aggiunta ad un url Opzioni oggetto: {chiave: imposta la chiave nella stringa serializzata, attributo: [Default "id"] imposta l'attributo da guardare, espressione : [Predefinito: "/(.+)-=_/"] regex per dividere l'attributo in coppie di valori chiave}
toArray (Object)	Serializza l'id degli oggetti ordinabili in un array. L'oggetto può contenere un attributo parametro che ha l'attributo da inserire nell'array predefinito è id
Widget ()	Restituisce un oggetto jQuery dell'elemento ordinabile
eventi	
attivare (evento, ui)	Attivato quando connesso elenco, ogni elenco connesso su trascinamento lo riceve
beforeStop (evento, ui)	I trigger prima dell'ordinamento si interrompono quando l'helper viene spostato nella stessa posizione del segnaposto
cambia (evento, ui)	Attivato quando gli elementi cambiano posizione, ad es. quando il segnaposto si muove
creare (evento, ui)	Attivato quando viene creato ordinabile
disattiva (evento, ui)	Attivato quando l'ordinamento si interrompe. Questo vale anche per tutte le liste collegate
fuori (evento, ui)	Attivato quando l'oggetto viene spostato fuori dall'elenco ordinabile

Parametro	Descrizione
sopra (evento, ui)	Attivato quando l'oggetto viene spostato in una lista ordinabile
ricevere (evento, ui)	Attivato quando un elemento da un elenco collegato è stato rilasciato in un altro. Target è la lista di ricezione
rimuovi (evento, ui)	Attivato quando un elemento da un elenco collegato è stato rilasciato in un altro. L'obiettivo è la lista di donazione
sort (evento, ui)	Attivato durante l'ordinamento
inizio (evento, ui)	Attivato quando inizia l'ordinamento
stop (evento, ui)	Attivato quando l'ordinamento si interrompe
aggiornamento (evento, ui)	Attivato quando l'ordinamento si interrompe e la posizione DOM viene aggiornata

Osservazioni

Documentazione ufficiale [qui](#)

Examples

Semplice esempio

Prendi una lista e aggiungi un identificatore al wrapper esterno (`ul` , `div`)

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

Nella tua jquery:

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

Questo consentirà a tutti la `li` nel `#sortable` wrapper per essere trascinati nella lista

Griglia ordinabile con layout flessibile

Questo ha utilizzato il layout flessibile con l'ordinabile per creare una griglia di scatole reattive che

possono essere spostate trascinandole e rilasciandole.

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;
}
```

Oggetti stazionari durante il trascinamento

Questo esempio utilizza una classe sul segnaposto per trasformarla in una riga e far sì che non occupi spazio.

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;
}
```

Ordinabile - Ripristina animate di elementi non accettati

Esempio di lavoro: <https://jsfiddle.net/Twisty/4f5yh3pa/7/>

Annullare e Ripristinare un ordinabile non è fortemente documentato. Gli aiuti mostrano come lo spostamento di un oggetto da una lista a un'altra lista connessa può essere annullato condizionalmente. per impostazione predefinita, questo non è animato da ordinabile, questo esempio include un'animazione.

Risultato: l'elenco n. 2 accetta solo articoli con una classe `acceptable` . Entrambe le liste possono essere ordinate in modo naturale altrimenti.

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();
});

```

Leggi Ordinabile online: <https://riptutorial.com/it/jquery-ui/topic/1415/ordinabile>

Capitolo 9: Plugin girevole jQuery UI

Parametri

Parametro	Dettagli
maniglia	url a un'immagine personalizzata per l'handle
angolo	la rotazione iniziale per l'elemento.
rotationCenterX	posizione su cui ruotare l'elemento
rotationCenterY	posizione su cui ruotare l'elemento
passo	un angolo in gradi in cui la rotazione scatterà se si tiene premuto il tasto Maiusc.
schiocco	scatta a passo in gradi.
inizio	innescato all'avvio della rotazione
Stop	innescato quando la rotazione si ferma
ruotare	innescato quando l'oggetto viene ruotato
wheelRotate	abilita / disabilita la rotellina del mouse per ruotare l'elemento.

Examples

Esempio di utilizzo iniziale

jquery-ui-rotatable è un plugin per l'interfaccia utente di jQuery che funziona in modo simile a Draggable e Resizable, senza essere così completo. Per impostazione predefinita, mette un'icona piccola rotazione in basso a sinistra di qualsiasi elemento che si desidera rendere ruotabile.

```
<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>
```

Leggi Plugin girevole jQuery UI online: <https://riptutorial.com/it/jquery-ui/topic/1806/plugin-girevole-jquery-ui>

Capitolo 10: Pulsante

Sintassi

- `$(".selector").button ();`
- `$(".selector").button ({disabled: true});`
- `$(".selector").button ({icons: {primary: "ui-icon-gear", secondario: "ui-icon-triangle-1-s"}});`
- `$(".selector").button ({label: "etichetta personalizzata"});`
- `$(".selector").button ({text: false});`

Parametri

Parametro	Tipo - Dettagli - Predefinito
<code>disabled</code>	Boolean - Disabilita il pulsante se impostato su <code>true</code> - <code>false</code>
<code>icons</code>	Object - Icone da visualizzare - <code>{primario: null, secondario: null }</code>
<code>label</code>	String - Testo da mostrare nel pulsante - <code>null</code>
<code>text</code>	Boolean : se mostrare l'etichetta - <code>true</code>

Examples

Utilizzo di base

Crea un elemento html di input (o pulsante, o ancora) e il metodo `button()` call dell'interfaccia utente di jQuery.

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

HTML

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

Leggi Pulsante online: <https://riptutorial.com/it/jquery-ui/topic/4600/pulsante>

Capitolo 11: Slider

Examples

Semplice esempio

Un controllo a scorrimento utilizza le maniglie trascinabili per selezionare valori numerici. Di seguito è riportato un esempio di inizializzazione di un cursore di base:

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

Range Slider

I cursori di intervallo forniscono 2 maniglie trascinabili per selezionare valori numerici. L'inizializzazione del dispositivo di scorrimento deve fornire un'opzione di `range` impostata su `true` per creare un dispositivo di scorrimento per intervalli:

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

Inizializzazione di valori e limiti di valore

Un elemento di slider può avere il suo valore impostato sull'inizializzazione fornendo un'opzione di `value`. Questa opzione è un numero:

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

Un cursore di intervallo può anche avere i suoi valori impostati in questo modo fornendo un'opzione di `values`. Questa opzione è una matrice di numeri:

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

Oltre a fornire i valori iniziali, è possibile definire il valore minimo, il valore massimo e l'intervallo di gestione con le opzioni `min`, `max` e `step`, rispettivamente:

```
$( "#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]
});
```

Utilizzando l'evento diapositiva

Il dispositivo di scorrimento fornisce un evento chiamato `slide` che si attiverà **ogni volta che il mouse si muoverà durante il trascinamento di una maniglia del dispositivo di scorrimento**. Questa funzione contiene un riferimento `event` diapositiva e un riferimento all'oggetto `ui` del cursore. L'oggetto `ui` contiene un oggetto jQuery per l'handle da spostare e il valore (`i`) del cursore.

Cursore a maniglia singola:

```
var value;

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

Cursore della gamma:

```
var lowValue;
var highValue;

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

Nota: l'evento `slide` scopo di rispondere al movimento del mouse attivo e non si innesca se i valori del cursore vengono modificati a livello di codice. Per reagire a questi eventi, usa l'evento `change`.

Impostazione dei valori e l'evento di modifica

Il dispositivo di scorrimento fornisce un evento chiamato `change` che si attiverà **dopo che il mouse ha completato il trascinamento di una maniglia del dispositivo di scorrimento o se i valori sono stati modificati a livello di codice**. Questa funzione contiene un riferimento `event` diapositiva e un riferimento all'oggetto `ui` del cursore. L'oggetto `ui` contiene un oggetto jQuery per l'handle da spostare e il valore (`i`) del cursore.

Un esempio potrebbe essere la visualizzazione di nuove informazioni dopo che i valori di un cursore sono stati aggiornati dall'evento di un altro elemento. Utilizziamo un elemento `select` per la dimostrazione in cui il valore del cursore viene impostato a livello di codice quando cambia il valore della `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: è in queste circostanze che l'evento di `slide` non si attiva e l'evento di `change` è necessario. Tuttavia, se gli elementi devono reagire ai valori del cursore che cambiano man mano che la maniglia viene trascinata, sarà necessario l'evento `slide` .

Leggi Slider online: <https://riptutorial.com/it/jquery-ui/topic/3206/slider>

Capitolo 12: Spinner

Sintassi

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

Parametri

parametri	Dettaglio
min	Valore minimo
max	Valore massimo
passo	Quanto può aumentare il valore di un clic sullo spinner, può essere decimale
rotazione	Può essere usato per controllare il valore dello spinner, <code>ui.value</code> e fare qualcosa

Osservazioni

[Esempio ufficiale](#)

[Documentazione ufficiale](#)

Examples

Esempio di base

Rende i numeri in entrata un po' più facili mostrando una serie di frecce sul lato destro `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">
```

Leggi Spinner online: <https://riptutorial.com/it/jquery-ui/topic/6637/spinner>

Capitolo 13: trascinabili

Examples

Semplice esempio

Abilita funzionalità trascinabili su qualsiasi elemento DOM.

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

Trascinabile con manico

Puoi utilizzare qualsiasi elemento come maniglia per trascinare un altro elemento intorno:

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

[Violino](#)

Leggi trascinabili online: <https://riptutorial.com/it/jquery-ui/topic/522/trascinabili>

Titoli di coda

S. No	Capitoli	Contributors
1	Introduzione alla libreria dell'interfaccia utente jQuery	Community , J F , jkdev , JonasCz , Twisty
2	Completamento automatico	J F
3	Date picker	cteski , J F , ni8mr , Nishant123 , Peter Tirrell , Pradeep , smdsgn , VincenzoC
4	Dialogo	J F , Jonathan Michalik , Racil Hilan , Theodore K. , Twisty
5	Fisarmonica	Dipen Shah , Madalina Taina
6	Icone	Theodore K.
7	jquery ui ordinabile	Andrew Mcghie , Twisty
8	Ordinabile	Alon Eitan , Andrew Mcghie , M B , Twisty
9	Plugin girevole jQuery UI	Twisty
10	Pulsante	Theodore K.
11	Slider	Jonathan Michalik
12	Spinner	Alon Eitan , Andrew Mcghie , depperm
13	trascinabili	empiric , J F