



eBook Gratuit

APPRENEZ

jQuery UI Library

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#jquery-ui

Table des matières

À propos.....	1
Chapitre 1: Démarrer avec la bibliothèque de l'interface utilisateur jQuery.....	2
Remarques.....	2
Versions.....	2
Exemples.....	3
Ajout du script d'interface utilisateur et de l'utilisation de base de jQuery.....	4
Configuration de l'interface utilisateur jQuery pour la première fois Exemple.....	4
Chapitre 2: Accordéon.....	7
Syntaxe.....	7
Paramètres.....	7
Remarques.....	7
Exemples.....	7
Utilisation basique de l'accordéon.....	7
L'accordéon détruit l'usage.....	8
Désactivation de l'accordéon.....	8
Accordéon activer l'utilisation.....	8
Option d'accordéon.....	8
Actualisation de l'accordéon.....	9
Utilisation du widget Accordiong.....	9
Chapitre 3: Autocomplete.....	10
Exemples.....	10
Exemple simple.....	10
Chapitre 4: Bouton.....	11
Syntaxe.....	11
Paramètres.....	11
Exemples.....	11
Utilisation de base.....	11
Chapitre 5: Curseur.....	12
Exemples.....	12
Exemple simple.....	12

Range Slider.....	12
Initialisation des valeurs et des limites de valeur.....	12
Utilisation de l'événement de diapositive.....	13
Définir les valeurs et l'événement de changement.....	13
Chapitre 6: Des icônes.....	15
Syntaxe.....	15
Remarques.....	15
Exemples.....	15
Utilisation de base.....	15
Chapitre 7: Dialogue.....	16
Syntaxe.....	16
Paramètres.....	16
Remarques.....	19
Exemples.....	19
Exemple simple.....	19
Boîte de dialogue ouverte lorsque l'événement se produit.....	19
Exemple complexe - Boîte de dialogue de création dynamique de l'interface utilisateur jQuery.....	20
Création d'une boîte de dialogue avec la barre de titre à onglets.....	23
Dialogue sans bouton de fermeture.....	25
Chapitre 8: Draggable.....	26
Exemples.....	26
Exemple simple.....	26
Draggable avec poignée.....	26
Chapitre 9: Fileur.....	27
Syntaxe.....	27
Paramètres.....	27
Remarques.....	27
Exemples.....	27
Exemple de base.....	27
Chapitre 10: jquery ui classable.....	29
Exemples.....	29
jQuery UI Sortable - Drop Placeholder.....	29

Chapitre 11: Plug-in rotatif de l'interface utilisateur jQuery	31
Paramètres.....	31
Exemples.....	31
Exemple d'utilisation initiale.....	31
Chapitre 12: Sélecteur de date	33
Exemples.....	33
Initialisation.....	33
Définition de dates minimum et maximum pour un datepicker.....	33
Afficher la semaine de l'année.....	33
Définir un format de date personnalisé.....	33
Afficher la liste déroulante mois et année.....	35
Chapitre 13: Triable	36
Syntaxe.....	36
Paramètres.....	36
Remarques.....	39
Exemples.....	39
Exemple simple.....	39
Grille triable avec disposition flexible.....	39
Articles fixes en faisant glisser.....	40
Sortable - Animer le retour d'un élément non accepté.....	41
Crédits	43

À propos

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

Chapitre 1: Démarrer avec la bibliothèque de l'interface utilisateur jQuery

Remarques

jQuery UI est une bibliothèque JavaScript, basée sur jQuery, offrant un ensemble d'interactions, d'effets et de widgets d'interface utilisateur.

Versions

Version	Date de sortie
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

Version	Date de sortie
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

Ajout du script d'interface utilisateur et de l'utilisation de base de jQuery

Pour commencer à utiliser la bibliothèque d'interface utilisateur jQuery, vous devez ajouter le script jQuery, le script d'interface utilisateur jQuery et la feuille de style d'interface utilisateur jQuery à votre code HTML.

Tout d'abord, [téléchargez l'interface utilisateur de jQuery](#); Choisissez les fonctionnalités dont vous avez besoin sur la page de téléchargement. Décompressez votre téléchargement et placez `jquery-ui.css` et `jquery-ui.js` (et `jquery.js`) dans un dossier où vous pourrez les utiliser depuis votre HTML (par exemple avec vos autres scripts et feuilles de style).

jQuery UI dépend de jQuery, n'oubliez donc pas d'inclure `jquery.js` avant `jquery-ui.js`.

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

C'est tout! Vous pouvez maintenant utiliser l'interface utilisateur jQuery. Par exemple, utilisez le datepicker avec le code HTML suivant:

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

Ensuite, utilisez le code JavaScript suivant:

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

Ce qui va vous faire un bon popup datepicker:



Pour plus d'informations, consultez le [manuel "Getting started"](#).

Configuration de l'interface utilisateur jQuery pour la première fois Exemple

Le framework d'interface utilisateur jQuery permet d'étendre et d'augmenter les contrôles d'interface utilisateur pour la bibliothèque jQuery JavaScript.

Lorsque vous souhaitez utiliser l'interface utilisateur jQuery, vous devez ajouter ces bibliothèques

à votre code HTML. Pour commencer rapidement, utilisez les sources de code disponibles de Content Delivery Network:

Bibliothèques jQuery

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

Vous pouvez choisir plusieurs thèmes différents pour l'interface utilisateur jQuery et même lancer votre propre thème. Pour cet exemple, nous utiliserons «Smoothness». Vous ajoutez le thème via CSS.

jQuery UI CSS

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

Mettre tous ensemble

Lorsque vous avez téléchargé ou sélectionné votre CDN, vous devez maintenant ajouter ces bibliothèques et feuilles de style à votre code HTML afin que votre page Web puisse désormais utiliser l'interface utilisateur jQuery et jQuery. L'ordre dans lequel vous chargez les bibliothèques est important. Appelez d'abord la bibliothèque jQuery, puis la bibliothèque de votre interface utilisateur jQuery. Comme jQuery UI étend jQuery, il doit être appelé après. Votre code HTML peut ressembler à ce qui suit.

```
<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>
```

Lire Démarrer avec la bibliothèque de l'interface utilisateur jQuery en ligne:

<https://riptutorial.com/fr/jquery-ui/topic/513/demarrer-avec-la-bibliotheque-de-l-interface-utilisateur-jquery>

Chapitre 2: Accordéon

Syntaxe

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

Paramètres

Paramètre	Détail
actif	Tapez Boolean ou Integer, Boolean nécessite d'être réduit pour être vrai
animer	Type Booléen, Nombre, Chaîne ou Objet
pliant	Type booléen
un événement	Type String
entête	Élément sélecteur de type
heightStyle	Type String
icônes	Type d'objet icône de l'interface utilisateur jQuery

Remarques

Plus d'informations peuvent être trouvées ici: <http://api.jqueryui.com/accordion/>

Exemples

Utilisation basique de l'accordéon

Pour utiliser un accordéon, il faut avoir des en-têtes et du contenu dans les en-têtes de leur code HTML. Ensuite, il faut instancier la méthode `accordion()` de l'interface utilisateur jQuery.

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

Dans le HTML:

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

▸ First header

▾ Second header

Second content panel

L'accordéon détruit l'usage

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

Cela supprimera complètement la fonctionnalité d'accordéon et affichera le code HTML par défaut en supprimant tous les éléments jQuery-UI.

Cette méthode ne prend aucun argument.

Désactivation de l'accordéon

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

Cette méthode désactive l'accordéon, c'est-à-dire que les en-têtes ne sont pas sélectionnables, ce qui rend le contenu en lecture seule et statique.

Cette méthode ne prend aucun argument.

Accordéon activer l'utilisation

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

Cette méthode permettra un accordéon. Cela permettra un accordéon désactivé ou ne fera rien sur un accordéon déjà activé.

Cette méthode ne prend aucun argument.

Option d'accordéon

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

Cela renverra un PlainObject donnant toutes les options représentant l'accordéon sélectionné.

Cela contiendra toutes les valeurs des clés expliquées dans la section Paramètres.

Cette méthode prend des paramètres qui sont les options de base expliquées dans le paramètre. On peut définir les options comme ceci:

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

Actualisation de l'accordéon

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

Cette méthode recalcule la hauteur des panneaux d'accordéon si des en-têtes ou du contenu ont été ajoutés ou supprimés dans le DOM.

Utilisation du widget Accordiong

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

Cette méthode retourne un objet jQuery contenant l'accordéon.

Lire Accordéon en ligne: <https://riptutorial.com/fr/jquery-ui/topic/630/accordeon>

Chapitre 3: Autocomplete

Exemples

Exemple simple

Les widgets de saisie semi-automatique fournissent des suggestions pendant que vous tapez dans le champ.

```
<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' />
```

Lire Autocomplete en ligne: <https://riptutorial.com/fr/jquery-ui/topic/519/autocomplete>

Chapitre 4: Bouton

Syntaxe

- `$(".selector").button ();`
- `$(".selector").button ({désactivé: true});`
- `$(".selector").button ({icons: {primary: "ui-icon-gear", secondaire: "ui-icon-triangle-1-s"}});`
- `$(".selector").button ({label: "label personnalisé"});`
- `$(".selector").button ({text: false});`

Paramètres

Paramètre	Type - Détails - Par défaut
<code>disabled</code>	Boolean - Désactive le bouton s'il est défini sur <code>true</code> - <code>false</code>
<code>icons</code>	Object - Icônes à afficher - <code>{primary: null, secondary: null }</code>
<code>label</code>	String - Texte à afficher dans le bouton - <code>null</code>
<code>text</code>	Boolean - Boolean si l'étiquette doit être - <code>true</code>

Exemples

Utilisation de base

Créez un élément html en entrée (ou bouton ou ancre) et appelez la méthode `button()` de l'interface utilisateur jQuery.

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

HTML

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

Lire Bouton en ligne: <https://riptutorial.com/fr/jquery-ui/topic/4600/bouton>

Chapitre 5: Curseur

Exemples

Exemple simple

Un contrôle de curseur utilise des poignées déplaçables pour sélectionner des valeurs numériques. Voici un exemple d'initialisation d'un curseur de base:

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

Range Slider

Les curseurs de plage fournissent 2 poignées déplaçables pour sélectionner des valeurs numériques. L'initialisation du curseur doit fournir une option de `range` définie sur `true` pour créer un curseur de plage:

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

Initialisation des valeurs et des limites de valeur

Un élément de curseur peut avoir sa valeur lors de l'initialisation en fournissant une option de `value`. Cette option est un numéro:

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

Un curseur de plage peut également avoir ses valeurs définies de cette manière en fournissant une option de `values`. Cette option est un tableau de nombres:

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


Outre les valeurs initiales, la valeur minimale, la valeur maximale et l'intervalle de traitement peuvent être définis avec les options `min`, `max` et `step`, respectivement:

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

Utilisation de l'événement de diapositive

Le curseur fournit un événement appelé `slide` qui se déclenche **chaque fois que la souris se déplacera au cours d'un curseur**. Cette fonction contient une référence à l' `event` `slide` et une référence à l'objet `ui` `slider`. L'objet `ui` contient un objet jQuery pour le handle en cours de déplacement et la ou les valeurs du curseur.

Curseur à une poignée:

```
var value;

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

Range Slider:

```
var lowValue;
var highValue;

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

Remarque: L'événement de `slide` est conçu pour répondre aux mouvements de la souris et ne se déclenche pas si les valeurs du curseur sont modifiées par programme. Pour réagir à ces événements, utilisez l'événement de `change`.

Définir les valeurs et l'événement de changement

Le curseur fournit un événement appelé `change` qui se déclenche **après que la souris ait effectué un glissement de la poignée du curseur ou si la ou les valeurs ont été modifiées par programme**. Cette fonction contient une référence à l' `event` `slide` et une référence à l'objet `ui` `slider`. L'objet `ui` contient un objet jQuery pour le handle en cours de déplacement et la ou les valeurs du curseur.

Un exemple pourrait être d'avoir à afficher de nouvelles informations après la mise à jour des valeurs d'un curseur par l'événement d'un autre élément. Utilisons un élément `select` pour la démonstration où la valeur du curseur est définie par programmation lorsque la valeur de la `select` change:

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;
    }
  });
});
```

Remarque: Dans ces circonstances, l'événement de `slide` ne se déclenche pas et l'événement de `change` est nécessaire. Toutefois, si des éléments doivent réagir à la modification des valeurs du curseur au fur et à mesure que le descripteur est déplacé, l'événement de `slide` est nécessaire.

Lire Curseur en ligne: <https://riptutorial.com/fr/jquery-ui/topic/3206/curseur>

Chapitre 6: Des icônes

Syntaxe

- `.ui-icon- {type d'icône} - {icon sub description} - {direction}`

Remarques

Les icônes sont également intégrées dans un certain nombre de widgets de l'interface utilisateur jQuery, tels que l' [accordéon](#) , le bouton, le menu.

Exemples

Utilisation de base

Pour une flèche épaisse pointant vers le nord dans une portée, ajoutez les classes `ui-icon` et `ui-icon-arrowthick-1-n` :

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

Pour un triangle pointant vers le sud dans une portée, ajoutez les classes `ui-icon` et `ui-icon-triangle-1-s` :

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

Liste complète des éléments disponibles ici <https://api.jqueryui.com/theming/icons/>

Lire Des icônes en ligne: <https://riptutorial.com/fr/jquery-ui/topic/4633/des-icônes>

Chapitre 7: Dialogue

Syntaxe

- `$ ("selector") .dialog ("option", "disabled");` // Option Getter, spécifique
- `$ ("selector") .dialog ("option");` // Option Getter tous
- `$ ("selector") .dialog ("option", "disabled", true);` // Option Setter, spécifique
- `$ ("selector") .dialog ("option", {disabled: true});` // Option Setter, multiple
- `$ ("selector") .dialog ("close");` // Les déclencheurs se ferment
- `$ ("selector") .dialog ({close: function () {}});` // Fermer la surcharge
- `$ ("selector") .on ("dialogclose", function (event, ui) {});` // Fermer la surcharge

Paramètres

Paramètre	La description
Les options	
ajouter à	(Sélecteur) [Par défaut: "body"] Quel élément le dialogue (et la superposition, si modale) doivent être ajoutés.
autoOpen	(Booléen) [Valeur par défaut: <code>true</code>] Si défini sur <code>true</code> , la boîte de dialogue s'ouvrira automatiquement lors de l'initialisation. Si la valeur est <code>false</code> , la boîte de dialogue reste masquée jusqu'à ce que la méthode <code>open ()</code> soit appelée.
boutons	(Object / Array) Spécifie les boutons à afficher dans la boîte de dialogue. Le contexte du rappel est l'élément de dialogue; Si vous avez besoin d'accéder au bouton, il est disponible en tant que cible de l'objet événement.
fermerenvironnement	(Booléen) [Valeur par défaut: <code>true</code>] Spécifie si la boîte de dialogue doit se fermer lorsqu'elle a le focus et que l'utilisateur appuie sur la touche Échap (ÉCHAP).
closeText	(String) [Valeur par défaut: "close"] Spécifie le texte du bouton de fermeture. Notez que le texte de fermeture est visiblement masqué lors de l'utilisation d'un thème standard.
dialogClass	(String) Le ou les noms de classe spécifiés seront ajoutés à la boîte de dialogue, pour un thème supplémentaire.
draggable	(Booléen) [Défaut: <code>true</code>] Si défini sur <code>true</code> , la boîte de dialogue sera déplaçable par la barre de titre. Nécessite le widget <code>Draggable</code> de l'interface utilisateur jQuery pour être inclus.

Paramètre	La description
la taille	(Number / String) [Valeur par défaut: "auto"] Hauteur de la boîte de dialogue.
cache	(Bool / Num / Str / Obj) Si et comment animer le masquage de la boîte de dialogue.
hauteur maximum	(Number) [Default: false] Hauteur maximale à laquelle la boîte de dialogue peut être redimensionnée, en pixels.
largeur maximale	(Number) [Valeur par défaut: false] Largeur maximale à laquelle la boîte de dialogue peut être redimensionnée, en pixels.
minHeight	(Nombre) [Défaut: 150] Hauteur minimale à laquelle la boîte de dialogue peut être redimensionnée, en pixels.
minWidth	(Nombre) [Défaut: 150] La largeur minimale à laquelle la boîte de dialogue peut être redimensionnée, en pixels.
modal	(Booléen) [Par défaut: false] Si défini sur true, le dialogue aura un comportement modal; les autres éléments de la page seront désactivés, c'est-à-dire qu'ils ne pourront pas interagir avec. Les boîtes de dialogue modales créent une superposition sous la boîte de dialogue, mais au-dessus des autres éléments de la page.
position	(Object) [Valeur par défaut: { my: "center", at: "center", of: window }] Spécifie où la boîte de dialogue doit être affichée lorsqu'elle est ouverte. La boîte de dialogue gèrera les collisions de manière à ce que la plus grande partie de la boîte de dialogue soit visible.
redimensionnable	(Booléen) [Valeur par défaut: true] Si défini sur true, la boîte de dialogue sera redimensionnable. Nécessite le widget jQuery UI Resizable pour être inclus.
montrer	(Bool / Num / Str / Obj) Si et comment animer l'affichage de la boîte de dialogue.
Titre	(String) Spécifie le titre de la boîte de dialogue. Si la valeur est null, l'attribut title de l'élément source de la boîte de dialogue sera utilisé.
largeur	(Number) [Default: 300] La largeur de la boîte de dialogue, en pixels.
Les méthodes	
Fermer	Ferme la boîte de dialogue.
détruire	Supprime complètement la fonctionnalité de dialogue. Cela ramènera l'élément à son état pré-initial.

Paramètre	La description
exemple	Récupère l'objet d'instance de la boîte de dialogue. Si l'élément n'a pas d'instance associée, undefined est renvoyé.
est ouvert	Si la boîte de dialogue est actuellement ouverte.
moveToTop	Déplace la boîte de dialogue en haut de la pile de boîtes de dialogue.
ouvrir	Ouvre la boîte de dialogue.
option	Obtient la valeur actuellement associée à l' <code>optionName</code> spécifiée.
option	Obtient un objet contenant des paires clé / valeur représentant les options de la boîte de dialogue en cours.
option	Définit une ou plusieurs options pour la boîte de dialogue.
widget	Renvoie un objet jQuery contenant le wrapper généré.
Points d'extension	
<code>_allowInteraction</code>	(event) Les boîtes de dialogue modales n'autorisent pas les utilisateurs à interagir avec les éléments situés derrière la boîte de dialogue. Cela peut être problématique pour les éléments qui ne sont pas des enfants du dialogue, mais qui sont absolument positionnés pour apparaître comme s'ils étaient. La méthode <code>_allowInteraction()</code> détermine si l'utilisateur doit être autorisé à interagir avec un élément cible donné; par conséquent, il peut être utilisé pour mettre en liste blanche des éléments qui ne sont pas des enfants du dialogue, mais que vous souhaitez que les utilisateurs puissent utiliser.
Événements	
avantFermer	(event, ui) Déclenché lorsqu'une boîte de dialogue est sur le point de se fermer. En cas d'annulation, la boîte de dialogue ne se ferme pas.
Fermer	(événement, ui) Déclenché lorsque la boîte de dialogue est fermée.
créer	(événement, ui) Déclenché lorsque la boîte de dialogue est créée.
traîne	(événement, ui {position, offset}) Déclenché pendant le glissement de la boîte de dialogue.
dragStart	(event, ui {position, offset}) Déclenché lorsque l'utilisateur commence à faire glisser la boîte de dialogue.
dragStop	(événement, ui {position, offset}) Déclenché après le déplacement de la boîte de dialogue.
concentrer	(événement, ui) Déclenché lorsque la boîte de dialogue gagne du

Paramètre	La description
	focus.
ouvrir	(événement, ui) Déclenché lorsque la boîte de dialogue est ouverte.
redimensionner	(event, ui {originalPosition, position, originalSize, size}) Déclenché pendant le redimensionnement de la boîte de dialogue.
redimensionnerStart	(event, ui {originalPosition, position, originalSize, size}) Déclenché pendant le redimensionnement de la boîte de dialogue.
resizeStop	(event, ui {originalPosition, position, originalSize, size}) Déclenché pendant le redimensionnement de la boîte de dialogue.

Remarques

Source de paramètre: <http://api.jqueryui.com/dialog/>

Exemples

Exemple simple

Le dialogue est une fenêtre superposée positionnée dans la fenêtre d'affichage.

```
<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>
```

Boîte de dialogue ouverte lorsque l'événement se produit

Nous voulons généralement séparer la création du dialogue de son apparence. Trois étapes sont alors nécessaires.

1. Créer un élément de 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. Faites-en une boîte de dialogue, notez l'option `autoOpen: false` qui garantit qu'elle sera fermée au début

```
$( "#dialog" ).dialog({
  autoOpen: false
});
```

3. Ouvrez-le si nécessaire, comme sur un clic de bouton

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

Exemple complexe - Boîte de dialogue de création dynamique de l'interface utilisateur jQuery

En général, le dialogue repose sur une `div` dans le code HTML. Parfois, vous pouvez créer une boîte de dialogue à partir de zéro, par programmation. Voici un exemple de dialogue modal complexe créé dynamiquement avec des fonctions interactives.

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-Z0-9-9])?(?:\.[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");
});
});

```

Exemple pratique pour référence: <https://jsfiddle.net/Twisty/LqjuxLu1/>

Création d'une boîte de dialogue avec la barre de titre à onglets

Parfois, nous souhaitons afficher des boîtes de dialogue contenant plusieurs volets de contenu. jQuery UI propose des onglets qui peuvent être utilisés conjointement avec une boîte de dialogue pour rendre cela possible. Bien qu'il soit plus fréquent d'avoir des onglets dans le conteneur de contenu d'une boîte de dialogue, cet exemple montre comment créer une liste d'onglets dans la barre de titre de la boîte de dialogue.

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

Exemple de travail pour référence: <https://jsfiddle.net/5x4zz681/>

Dialogue sans bouton de fermeture

Si vous souhaitez afficher la boîte de dialogue sans le bouton Fermer (c.-à-d. Le bouton x dans le coin supérieur droit de la boîte de dialogue), vous voudrez peut-être forcer l'utilisateur à sélectionner l'une des options ou boutons de la boîte de dialogue:

1- Donnez à votre boîte de dialogue une classe CSS:

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

2- Cacher le bouton Fermer en utilisant ce CSS:

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

Remarque: Si vous souhaitez masquer la totalité de la barre de titre, utilisez plutôt ce CSS:

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

Vous pouvez également masquer le bouton Fermer dans le code d'initialisation de la boîte de dialogue:

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

Lire Dialogue en ligne: <https://riptutorial.com/fr/jquery-ui/topic/521/dialogue>

Chapitre 8: Draggable

Exemples

Exemple simple

Activer la fonctionnalité déplaçable sur tout élément DOM.

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

Draggable avec poignée

Vous pouvez utiliser n'importe quel élément pour gérer un autre élément:

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

Violon

Lire Draggable en ligne: <https://riptutorial.com/fr/jquery-ui/topic/522/draggable>

Chapitre 9: Fileur

Syntaxe

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

Paramètres

Paramètres	Détail
min	Valeur minimum
max	Valeur maximum
étape	Combien la valeur augmente en cliquant sur spinner, peut être décimal
tourner	Peut être utilisé pour vérifier la valeur de spinner, <code>ui.value</code> et faire quelque chose

Remarques

[Exemple officiel](#)

[Documentation officielle](#)

Exemples

Exemple de base

Permet de saisir des chiffres un peu plus facilement en affichant un ensemble de flèches à droite de l' `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">
```

Lire Fileur en ligne: <https://riptutorial.com/fr/jquery-ui/topic/6637/fileur>

Chapitre 10: jquery ui classable

Exemples

jQuery UI Sortable - Drop Placeholder

Cet exemple du triable utilisant un espace réservé est un usage courant. Sortable est appliqué à un groupe d'éléments DOM, permettant à l'utilisateur de déplacer des éléments dans la liste via des actions de style 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>
```

Lire jquery ui classable en ligne: <https://riptutorial.com/fr/jquery-ui/topic/1226/jquery-ui-classable>

Chapitre 11: Plug-in rotatif de l'interface utilisateur jQuery

Paramètres

Paramètre	Détails
manipuler	URL vers une image personnalisée pour la poignée
angle	la rotation de départ pour l'élément.
rotationCenterX	position sur laquelle l'élément sera tourné
centre de rotation	position sur laquelle l'élément sera tourné
étape	un angle en degrés auquel la rotation s'accroche si la touche Maj est enfoncée.
casser	s'enclenche pour passer en degrés.
début	déclenché lorsque la rotation commence
Arrêtez	déclenché lorsque la rotation s'arrête
tourner	déclenché lorsque l'objet est en rotation
roueRotate	activer / désactiver la molette de la souris pour faire pivoter l'élément.

Exemples

Exemple d'utilisation initiale

jquery-ui-rotatable est un plugin pour l'interface utilisateur jQuery qui fonctionne de la même manière que Draggable et Resizable, sans être aussi complet. Par défaut, il place une petite icône de rotation en bas à gauche de tout élément que vous souhaitez faire pivoter.

```
<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>
```

Lire Plug-in rotatif de l'interface utilisateur jQuery en ligne: <https://riptutorial.com/fr/jquery-ui/topic/1806/plug-in-rotatif-de-l-interface-utilisateur-jquery>

Chapitre 12: Sélecteur de date

Exemples

Initialisation

Le **datepicker** est utilisé pour afficher un sélecteur de date interactif lié à un champ de saisie de formulaire standard. Cela facilite grandement la sélection de la date pour les tâches de saisie et est également hautement configurable.

Tout champ de saisie peut être lié avec jquery-ui datepicker par le sélecteur du champ de saisie (id, class etc.) en utilisant la méthode **datepicker ()** comme ceci -

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

La démo en direct est [ici](#) .

Définition de dates minimum et maximum pour 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">
```

Afficher la semaine de l'année

Le code suivant affichera le numéro de la semaine de l'année sur le côté gauche du datepicker. Par défaut, la semaine commence le lundi, mais elle peut être personnalisée à l'aide de l'option `firstDay` . La première semaine de l'année contient le premier jeudi de l'année, conformément à la définition ISO 8601.

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

Définir un format de date personnalisé

Format de date par défaut: "mm / jj / aa"

L'exemple suivant montre comment définir le format de date lors de l'initialisation avec l'option `dateFormat`.

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

L'exemple suivant montre comment définir le format de date après l'initialisation avec l'option `dateFormat`.

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

Vous pouvez utiliser des combinaisons des éléments suivants:

```
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
```

Ou standard prédéfini:

```
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 format de date par défaut peut être appliqué à tous les datepickers à l'aide de l'exemple suivant:

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

Afficher la liste déroulante mois et année

jQuery datepicker a deux options pour permettre l'affichage des listes déroulantes pour la sélection du mois et de l'année. Ces options facilitent la navigation dans les grands délais.

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

Lire Sélecteur de date en ligne: <https://riptutorial.com/fr/jquery-ui/topic/520/selecteur-de-date>

Chapitre 13: Triable

Syntaxe

- `$ ("# sortable") .table ({/ * Options ici * /}); // Initialiser Sortable`
- `$ ("# sortable"). sortable ("option", "nom_option", valeur_option); // Définir l'option hors initialiseur`
- `var value = $ ("# sortable"). sortable ("option", "nom_option"); // Obtient la valeur d'une option`

Paramètres

Paramètre	La description
Les options	
ajouter à	(jQuery, Element, Selector, String) [Valeur par défaut: "parent"] L'élément auquel l'ajout est ajouté
axe	(Chaîne) [Défaut: false] La direction dans laquelle l'élément peut être déplacé (x ou y)
Annuler	(Sélecteur) [Par défaut: "input, textarea, button, select, option"] Ne commence pas le tri si vous commencez sur un élément correspondant au sélecteur
Des classes	(Object) [Default: {}] Spécifiez des classes supplémentaires à ajouter aux éléments triables lors de l'ajout des classes structurelles. ({ui-sortable-helper: custom_class})
se connecter avec	(Sélecteur) [Par défaut: false] Permet de déplacer les éléments d'un triable vers un autre
endiguement	(Élément, Sélecteur, Chaîne) [Par défaut: false] Élément auquel les éléments sont limités
le curseur	(Chaîne) [Défaut: "auto"] Définit le type de curseur à afficher lors du tri
curseurAt	(Object) [Valeur par défaut: false] Définit la position à laquelle l'aide semble avoir été déplacée
désactivée	(Booléen) [Défaut: faux] Désactive le tri si vrai
dropOnEmpty	(Booléen) [Défaut: true] Si de faux éléments de ce triables ne

Paramètre	La description
	peuvent pas être placés dans des triables vides
forceHelperSize	(Booléen) [Défaut: false] Force la taille de l'assistant
ForcePlaceholderSize	(Booléen) [Défaut: false] Force la taille de l'espace réservé
la grille	(Array) [Défaut: false] Définit une grille pour aligner l'assistant ([x, y])
manipuler	(Sélecteur, Élément) [Par défaut: false] Définit les éléments que le tri peut démarrer. Opposé à annuler
assistant	(String, Function) [Valeur par défaut: "original"] Chaîne "original" ou "clone", ou fonction qui renvoie l'élément à utiliser comme aide.
articles	(Sélecteur) [Par défaut: "> *"] Définit les éléments qui doivent être triables
opacité	(Nombre 0.01 à 1) [Défaut: false] Définit l'opacité de l'assistant
espace réservé	(Chaîne) [Valeur par défaut: false] Définit une classe ou des classes à appliquer à l'espace réservé
revenir	(Boolean, Number) [Valeur par défaut: false] Le temps nécessaire à l'aide pour entrer dans sa nouvelle position
faire défiler	(Booléen) [Défaut: true] Défilement des bords de la page
scrollSensibilité	(Nombre) [Défaut: 20] Définit à quelle distance du bord de la page le curseur doit être pour commencer le défilement
vitesse de défilement	(Nombre) [Défaut: 20] La vitesse à laquelle défiler
tolérance	(String) [Valeur par défaut: "intersection"] Définit le mode à utiliser lors du calcul lorsqu'un élément se trouve au-dessus d'un autre ("intersection" ou "pointeur")
zIndex	(Entier) [Défaut: 1000] Définit l'index z de l'assistant lors du tri
Les méthodes	
Annuler()	Annule le tri en cours et renvoie les éléments à leur position avant le début du tri
détruire()	Supprime la fonctionnalité triable et renvoie l'élément à son état de préinitialisation
disable ()	Désactive le triable

Paramètre	La description
activer()	Active le triable
exemple()	Renvoie l'objet d'instance sortable
option()	Obtient des paires de valeurs clés de toutes les options pour le triable
option (chaîne)	Obtient la valeur d'une option
option (String, Any)	Définit la valeur de l'option spécifiée par la chaîne
option (objet)	Définit une ou plusieurs options avec l'objet étant des paires d'options de valeur clé
rafraîchir()	Actualise les options triables en rechargeant tous les éléments triables. Cela entraîne la reconnaissance de nouveaux éléments
refreshPositions ()	Règle les positions en cache des éléments triables
sérialiser (objet)	Sérialise les identifiants des éléments (par défaut) en une chaîne pouvant être soumise ou ajoutée à une URL. Options de l'objet: {clé: définit la clé dans la chaîne sérialisée, attribut: : [Default: "/(.+)-=_/"] regex pour diviser l'attribut en paires de valeurs de clé}
toArray (Object)	Sérialise les éléments triables en un tableau. L'objet peut contenir un attribut de paramètre qui a l'attribut à mettre dans le tableau
widget ()	Retourne un objet jQuery de l'élément triable
Événements	
activer (événement, ui)	Déclenché lorsque la liste est connectée, toutes les listes connectées lors du lancement par glisser le reçoivent
beforeStop (événement, interface utilisateur)	Déclenche avant que le tri ne s'arrête lorsque l'aide est déplacée vers la même position que l'espace réservé
changement (événement, ui)	Déclenché lorsque les éléments changent de position, c.-à-d. quand l'espace réservé se déplace
créer (événement, ui)	Déclenché lorsque le tri est créé
désactiver (événement, ui)	Déclenché lors du tri Cela va aussi à toutes les listes connectées
out (événement, ui)	Déclenché lorsque l'élément est déplacé hors de la liste triable

Paramètre	La description
terminé (événement, ui)	Déclenché lorsque l'élément est déplacé dans une liste triable
recevoir (événement, ui)	Déclenché lorsqu'un élément d'une liste connectée est déposé dans un autre. La cible est la liste de réception
supprimer (événement, ui)	Déclenché lorsqu'un élément d'une liste connectée est déposé dans un autre. La cible est la liste de dons
trier (événement, ui)	Déclenché pendant le tri
commencer (événement, ui)	Déclenché au début du tri
arrêter (événement, ui)	Déclenché lors du tri
mise à jour (événement, ui)	Déclenché lorsque le tri s'arrête et que la position du DOM a été mise à jour

Remarques

Documentation officielle [ici](#)

Exemples

Exemple simple

Prenez n'importe quelle liste et ajoutez un identifiant à l'enveloppe externe (`ul` , `div`)

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

Dans votre jquery:

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

Cela permettra à tous les `li` dans le wrapper `#sortable` d'être glissé et déposé dans la liste

Grille triable avec disposition flexible

Ceci a utilisé la disposition flexible avec le triable pour créer une grille de boîtes réactives qui peuvent être déplacées en les faisant glisser.

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;
}
```

Articles fixes en faisant glisser

Cet exemple utilise une classe sur l'espace réservé pour le transformer en ligne et l'empêcher de prendre de la place.

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;
}
```

Sortable - Animer le retour d'un élément non accepté

Exemple de travail: <https://jsfiddle.net/Twisty/4f5yh3pa/7/>

L'annulation et le retour à un tri ne sont pas fortement documentés. L'aide montre comment le déplacement d'un élément d'une liste vers une autre liste connectée peut être annulé de manière conditionnelle. par défaut, ceci n'est pas animé par triable, cet exemple inclut une animation.

Résultat: la liste n ° 2 n'accepte que les éléments ayant une classe `acceptable` . Les deux listes peuvent être triées naturellement autrement.

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();
});

```

Lire Triable en ligne: <https://riptutorial.com/fr/jquery-ui/topic/1415/triable>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec la bibliothèque de l'interface utilisateur jQuery	Community , J F , jkdev , JonasCz , Twisty
2	Accordéon	Dipen Shah , Madalina Taina
3	Autocomplete	J F
4	Bouton	Theodore K.
5	Curseur	Jonathan Michalik
6	Des icônes	Theodore K.
7	Dialogue	J F , Jonathan Michalik , Racil Hilan , Theodore K. , Twisty
8	Draggable	empiric , J F
9	Fileur	Alon Eitan , Andrew Mcghie , depperm
10	jquery ui classable	Andrew Mcghie , Twisty
11	Plug-in rotatif de l'interface utilisateur jQuery	Twisty
12	Sélecteur de date	cteski , J F , ni8mr , Nishant123 , Peter Tirrell , Pradeep , smdsgn , VincenzoC
13	Triable	Alon Eitan , Andrew Mcghie , M B , Twisty