Бесплатная электронная книга

УЧУСЬ

jQuery UI Library

Free unaffiliated eBook created from
Stack Overflow contributors.

#jquery-ui

Около

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.

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

глава 1: Начало работы с библиотекой пользовательского интерфейса jQuery

замечания

jQuery UI - это библиотека пользовательского интерфейса JavaScript, построенная поверх jQuery, предлагающая набор взаимодействий, эффектов и виджетов пользовательского интерфейса.

Версии

Версия	Дата выхода
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

Версия	Дата выхода
1.8.14	2011-06-28
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

Добавление сценария пользовательского интерфейса jQuery и базового использования

Чтобы начать работу с библиотекой пользовательского интерфейса jQuery, вам нужно добавить скрипт jQuery, сценарий пользовательского интерфейса jQuery и таблицу стилей jQuery UI в свой HTML.

Сначала загрузите пользовательский интерфейс jQuery; выберите нужные функции на странице загрузки. Разархивируйте вашу загрузку и поместите `jquery-ui.css` и `jquery-ui.js` (и `jquery.js`) в папку, в которой вы можете использовать их из своего HTML (например, с помощью других скриптов и таблиц стилей).

jQuery UI зависит от jQuery, поэтому не забудьте включить `jquery.js` перед `jquery-ui.js` .

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

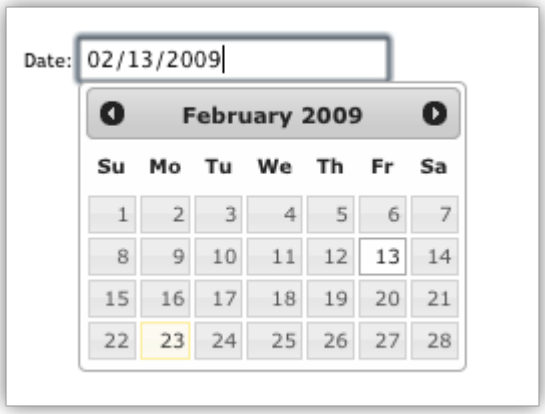
Это оно! Теперь вы можете использовать пользовательский интерфейс jQuery. Например, используйте datepicker со следующим HTML:

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

Затем используйте следующий JavaScript:

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

Который даст вам хорошее всплывающее окно datepicker:



Более того, см. **Официальный «Приступая к работе» gude** .

Настройка пользовательского интерфейса jQuery для первого примера

Интерфейс пользовательского интерфейса jQuery помогает расширять и увеличивать элементы управления пользовательским интерфейсом для библиотеки JavaScript jQuery.

Если вы хотите использовать jQuery UI, вам нужно будет добавить эти библиотеки в свой HTML. Быстрый способ начать использовать источники доступных источников доставки контента:

Библиотеки jQuery

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

Вы можете выбрать много разных тем для jQuery UI и даже Roll Your Own Theme. В этом примере мы будем использовать «Гладкость». Вы добавляете тему через CSS.

jQuery UI CSS

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

Все вместе

Когда вы загрузили или выбрали свой CDN, вы теперь захотите добавить эти библиотеки и таблицы стилей в свой HTML, чтобы ваша веб-страница теперь могла использовать пользовательский интерфейс jQuery и jQuery. Важен порядок загрузки библиотек. Сначала вызовите библиотеку jQuery, а затем вашу библиотеку пользовательского интерфейса jQuery. Поскольку jQuery UI расширяет jQuery, он должен быть вызван после. Ваш HTML может выглядеть примерно так:

```
<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>
```

Прочитайте Начало работы с библиотекой пользовательского интерфейса jQuery онлайн: https://riptutorial.com/ru/jquery-ui/topic/513/начало-работы-с-библиотекой-пользовательского-интерфейса-jquery

глава 2: Datepicker

Examples

инициализация

Datepicker используется для отображения интерактивного селектора дат, который привязан к стандартным полям ввода формы. Он очень легко выбирает дату для ввода задач, а также настраивается.

Любое поле ввода может быть связано с jquery-ui datepicker с помощью селектора поля ввода (id, class и т. Д.) С использованием метода **datepicker (),** подобного этому -

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

Живая демонстрация здесь .

Установка минимальных и максимальных дат для дампикера

```
<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">
```

Показать неделю года

В следующем коде будет отображаться неделя номера года в левой части даты. По умолчанию начало недели в понедельник, но его можно настроить с `firstDay` опции `firstDay` . Первая неделя года содержит первый четверг года, следуя определению ISO 8601.

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

Настройка пользовательского формата даты

Формат даты по умолчанию: «мм / дд / гг»

В следующем примере показано, как вы можете установить формат даты при инициализации с помощью параметра dateFormat.

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

В следующем примере показано, как вы можете установить формат даты после инициализации с помощью параметра dateFormat.

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

Вы можете использовать следующие комбинации:

```
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
```

Или предопределенный стандарт:

```
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)
```

Формат даты по умолчанию может применяться ко всем дампикерам, используя следующий пример:

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

Показать выпадающий месяц и год

jQuery datepicker имеет два варианта, позволяющих отображать раскрывающиеся списки для выбора месяца и года. Эти параметры упрощают навигацию в больших временных рамках.

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

Прочитайте Datepicker онлайн: https://riptutorial.com/ru/jquery-ui/topic/520/datepicker

глава 3: jQuery UI Rotatable Plug-in

параметры

параметр	подробности
справиться	url для пользовательского изображения для дескриптора
угол	начальное вращение элемента.
rotationCenterX	положение, относительно которого элемент будет повернут
rotationCenterY	положение, относительно которого элемент будет повернут
шаг	угол в градусах, который будет вращаться, если удерживать клавишу сдвига.
щелчок	привязывается к градусам.
Начните	срабатывает при начале вращения
стоп	срабатывает при остановке вращения
вращаться	срабатывает, когда объект вращается
wheelRotate	включить / отключить колесо мыши для поворота элемента.

Examples

Пример использования

> jquery-ui-rotatable - это плагин для пользовательского интерфейса jQuery, который работает аналогично Draggable и Resizable, но не является полнофункциональным. По умолчанию он помещает небольшой значок вращения в левом нижнем углу любого элемента, который вы хотите сделать вращаемым.

```
<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>
```

Прочитайте jQuery UI Rotatable Plug-in онлайн: https://riptutorial.com/ru/jquery-ui/topic/1806/jquery-ui-rotatable-plug-in

глава 4: jquery ui сортировать

Examples

jQuery UI Sortable - Drop Placeholder

Этот пример Сортируемого с использованием Placeholder является обычным использованием. Сортировка применяется к группе элементов DOM, что позволяет пользователю перемещать элементы в списке с помощью действий стиля 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>
  <li class="ui-state-default">Item 6</li>
  <li class="ui-state-default">Item 7</li>
</ul>

</body>
</html>
```

Прочитайте jquery ui сортировать онлайн: https://riptutorial.com/ru/jquery-ui/topic/1226/jquery-ui-сортировать

глава 5: Автозаполнение

Examples

Простой пример

Виджеты Autocomplete предоставляют предложения при вводе в поле.

```
<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' />
```

Прочитайте Автозаполнение онлайн: https://riptutorial.com/ru/jquery-ui/topic/519/автозаполнение

глава 6: аккордеон

Синтаксис

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

параметры

параметр	подробность
активный	Тип Boolean или Integer, Boolean требует сбрасывания, чтобы быть правдой
оживлять	Тип Boolean, Number, String или Object
разборный	Тип Boolean
событие	Тип String
заголовок	Тип Селекторный элемент
heightStyle	Тип String
иконки	Тип объекта jQuery UI icon

замечания

Более подробную информацию можно найти здесь: http://api.jqueryui.com/accordion/

Examples

Аккордеонное базовое использование

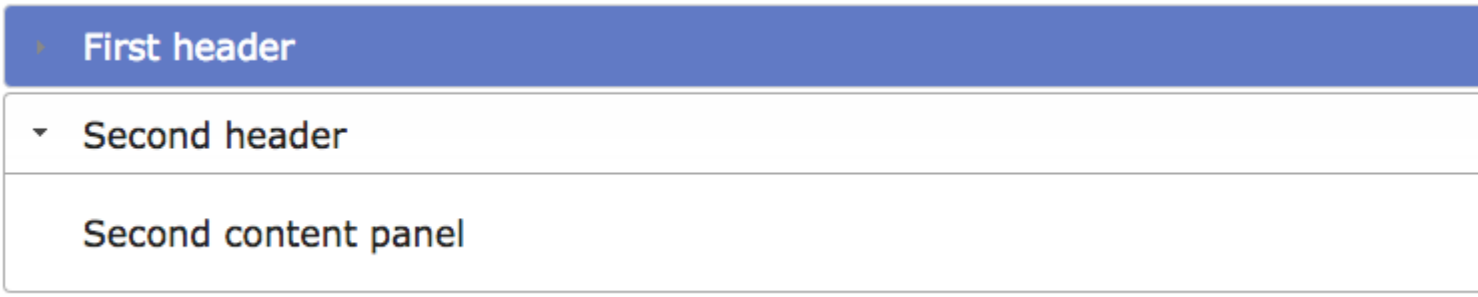
Чтобы использовать аккордеон, в заголовках HTML должны быть заголовки и содержимое. Затем нужно создать экземпляр метода `accordion()` jQuery UI.

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

```
</script>
```

В HTML:

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



Аккордеон уничтожает использование

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

Это полностью удалит функциональность аккордеона и покажет HTML по умолчанию, удалив все элементы jQuery-UI.

Этот метод не принимает никаких аргументов.

Аккордеон отключен Использование

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

Этот метод отключит аккордеон, т. Е. Заголовки не будут выбираться, делая контент только для чтения и статическим.

Этот метод не принимает никаких аргументов.

Аккордеонное разрешение Использование

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

Этот метод позволит использовать аккордеон. Это позволит отключить аккордеон или просто ничего не сделать на уже включенном аккордеоне.

Этот метод не принимает никаких аргументов.

Вариант аккордеона Использование

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

Это вернет объект PlainObject, предоставляющий все параметры, представляющие выбранный аккордеон. Это будет содержать все значения ключей, которые описаны в разделе «Параметры».

Этот метод принимает параметры, которые являются базовыми параметрамиNames, объясняемыми в параметре. Можно задать следующие параметры:

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

Аккордеонное обновление Использование

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

Этот метод рекомпирует высоту панелей аккордеона, если в DOM были добавлены или удалены заголовки или содержимое.

Использование виджета Accordiong

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

Этот метод возвращает объект jQuery, содержащий аккордеон.

Прочитайте аккордеон онлайн: https://riptutorial.com/ru/jquery-ui/topic/630/аккордеон

глава 7: волчок

Синтаксис

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

параметры

параметры	подробность
мин	Минимальное значение
Максимум	Максимальное значение
шаг	Сколько значение увеличивается при нажатии на прядильщик, может быть десятичным
вращение	Может использоваться для проверки значения `ui.value` , `ui.value` и сделать что-то

замечания

Официальный пример

Официальная документация

Examples

Основной пример

Делает ввод чисел немного удобнее, показывая набор стрелок на правой стороне `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">
```

Прочитайте волчок онлайн: https://riptutorial.com/ru/jquery-ui/topic/6637/волчок

глава 8: диалог

Синтаксис

- $ (".selector") .dialog ("option", "disabled"); // Опция Getter, специфическая
- $ (".selector") .dialog ("option"); // Опция Getter all
- $ (".selector") .dialog ("option", "disabled", true); // Опция Setter, специфическая
- $ (".selector") .dialog ("option", {disabled: true}); // Опция Setter, несколько
- $ (".selector") .dialog ("close"); // Триггеры закрываются
- $ (".selector") .dialog ({close: function () {}}); // Закрытие перегрузки
- $ (".selector") .on ("dialogclose", function (event, ui) {}); // Закрытие перегрузки

параметры

параметр	Описание
Опции	
добавить в	(Селектор) [По умолчанию: `"body"`] К какому элементу должен быть добавлен диалог (и наложение, если модальный).
AutoOpen	(Boolean) [По умолчанию: `true`] Если установлено значение true, диалоговое окно будет автоматически открываться после инициализации. Если false, диалог останется скрытым до тех пор, пока не будет вызван метод open ().
кнопки	(Object / Array) Определяет, какие кнопки должны отображаться в диалоговом окне. Контекст обратного вызова - это элемент диалога; если вам нужен доступ к кнопке, он доступен как цель объекта события.
closeOnEscape	(Boolean) [По умолчанию: `true`] Указывает, должно ли диалоговое окно закрываться, когда оно имеет фокус, и пользователь нажимает клавишу escape (ESC).
closeText	(String) [По умолчанию: `"close"`] Задает текст кнопки закрытия. Обратите внимание, что закрытый текст явно скрыт при использовании стандартной темы.
dialogClass	(String) Указанные имена классов будут добавлены в диалог для дополнительной тематики.
перетаскиваемый	(Boolean) [По умолчанию: `true`] Если установлено значение `true` ,

параметр	Описание
	диалог будет перетаскиваться в строке заголовка. Требуется включить виджет jQuery UI Draggable.
рост	(Number / String) [По умолчанию: `"auto"`] Высота диалога.
скрывать	(Bool / Num / Str / Obj) Если и как оживить скрытие диалога.
максимальная высота	(Number) [По умолчанию: `false`] Максимальная высота, на которую можно изменить размер диалогового окна, в пикселях.
Максимальная ширина	(Число) [По умолчанию: `false`] Максимальная ширина, на которую можно изменить размер диалогового окна, в пикселях.
MinHeight	(Число) [По умолчанию: `150`] Минимальная высота, на которую можно изменить размер диалогового окна, в пикселях.
MinWidth	(Число) [По умолчанию: `150`] Минимальная ширина, на которую можно изменить размер диалогового окна, в пикселях.
модальный	(Boolean) [По умолчанию: `false`] Если установлено значение true, диалог будет иметь модальное поведение; другие элементы на странице будут отключены, т. е. с ними невозможно взаимодействовать. Модальные диалоги создают надпись ниже диалога, но над другими элементами страницы.
позиция	(Object) [По умолчанию: `{ my: "center", at: "center", of: window }`] Указывает, где диалог должен отображаться при открытии. Диалоговое окно будет обрабатывать конфликты таким образом, чтобы как можно больше диалогов было видно.
изменяемый	(Boolean) [По умолчанию: `true`] Если установлено значение true, диалог будет изменяться. Требуется включить виджет jQuery UI Resizable.
шоу	(Bool / Num / Str / Obj) Если и как оживить показ диалога.
заглавие	(String) Задает заголовок диалога. Если значение равно null, будет использоваться атрибут title в элементе источника диалога.
ширина	(Число) [По умолчанию: `300`] Ширина диалога в пикселях.
методы	
близко	Закрывает диалог.
уничтожить	Полностью удаляет функциональность диалога. Это вернет

параметр	Описание
	элемент обратно в исходное состояние.
пример	Извлекает объект экземпляра диалога. Если элемент не имеет связанного с ним экземпляра, возвращается undefined.
открыт	Открывается ли диалог.
moveToTop	Перемещает диалог вверху диалогового стека.
открыть	Открывает диалог.
вариант	Возвращает значение, связанное с указанным `optionName` .
вариант	Получает объект, содержащий пары ключ / значение, представляющие хеш-параметры текущего диалога.
вариант	Устанавливает один или несколько параметров для диалога.
виджет	Возвращает объект jQuery, содержащий сгенерированную оболочку.
Точки расширения	
_allowInteraction	(событие) Модальные диалоги не позволяют пользователям взаимодействовать с элементами за диалогом. Это может быть проблематично для элементов, которые не являются дочерними элементами диалогового окна, но абсолютно позиционируются так, как будто они есть. Метод `_allowInteraction()` определяет, разрешено ли пользователю взаимодействовать с данным целевым элементом; поэтому его можно использовать для элементов белого списка, которые не являются дочерними элементами диалогового окна, но вы хотите, чтобы пользователи могли их использовать.
События	
BeforeClose	(event, ui) Запускается при закрытии диалогового окна. Если отменено, диалог не будет закрыт.
близко	(event, ui) Запускается, когда диалог закрыт.
Создайте	(event, ui) Запускается при создании диалога.
тащить, тянуть	(event, ui {position, offset}) Запускается, когда диалог перетаскивается.

параметр	Описание
dragStart	(event, ui {position, offset}) Запускается, когда пользователь начинает перетаскивать диалог.
dragStop	(event, ui {position, offset}) Запущено после того, как диалог был перетащен.
фокус	(event, ui) Запускается, когда диалог получает фокус.
открыть	(event, ui) Запускается при открытии диалога.
изменить размер	(event, ui {originalPosition, position, originalSize, size}) Запускается при изменении размера диалогового окна.
resizeStart	(event, ui {originalPosition, position, originalSize, size}) Запускается при изменении размера диалогового окна.
resizeStop	(event, ui {originalPosition, position, originalSize, size}) Запускается при изменении размера диалогового окна.

замечания

Параметр Источник: http://api.jqueryui.com/dialog/

Examples

Простой пример

Диалог - это окно, которое накладывается на окно просмотра.

```
<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>
```

Открыть диалог при возникновении события

Обычно мы хотим отделить создание диалога от его внешнего вида. Затем необходимо выполнить три шага.

1. Создание базового элемента

```
<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. Сделайте это диалогом, обратите внимание на параметр `autoOpen: false` который гарантирует, что он будет закрыт сперва

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

3. Откройте его, если необходимо, например, нажатием кнопки

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

Комплексный пример - jQuery UI Динамически созданный диалог

Как правило, диалог опирается на `div` внутри HTML. Иногда вы можете создавать диалоговое окно с нуля, программно. Ниже приведен пример сложного модального диалога, созданного динамически с помощью интерактивных функций.

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>johndoe1</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");
  });
});
```

Рабочий пример для справки: https://jsfiddle.net/Twisty/LqjuxLu1/

Создание диалога с вкладкой заголовка

Иногда мы можем отображать диалоги с более чем одной областью содержимого. Пользовательский интерфейс jQuery предлагает вкладки, которые можно использовать в тандеме с диалогом, чтобы сделать это возможным. Хотя может быть более распространено иметь вкладки в контейнере содержимого диалога, в этом примере будет показано, как сделать список вкладок в строке заголовка диалогового окна.

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

Рабочий пример для справки: https://jsfiddle.net/5x4zz681/

Диалог без кнопки закрытия

Если вам нравится показывать диалог без кнопки закрытия (т. Е. Кнопку x в правом верхнем углу диалога), возможно, потому, что вы хотите заставить пользователя выбрать один из параметров или кнопок в самом диалоге:

1- Дайте вашему диалогу класс CSS:

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

2- Скрыть кнопку закрытия с помощью этого CSS:

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

Примечание. Если вы хотите скрыть всю строку заголовка, используйте вместо этого CSS:

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

Кроме того, вы можете скрыть кнопку закрытия в коде инициализации диалога:

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

Прочитайте диалог онлайн: https://riptutorial.com/ru/jquery-ui/topic/521/диалог

глава 9: Иконки

Синтаксис

- .ui-icon- {icon type} - {icon sub description} - {direction}

замечания

Значки также интегрированы в ряд виджетов jQuery UI, таких как аккордеон , кнопка, меню.

Examples

Основное использование

Для толстой стрелки, указывающей на север в промежутке, добавьте классы `ui-icon` и `ui-icon-arrowthick-1-n` :

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

Для треугольника, указывающего на юг в промежутке, добавьте классы `ui-icon` и `ui-icon-triangle-1-s` :

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

Полный список доступных здесь предметов https://api.jqueryui.com/theming/icons/

Прочитайте Иконки онлайн: https://riptutorial.com/ru/jquery-ui/topic/4633/иконки

глава 10: кнопка

Синтаксис

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

параметры

параметр	Тип - Подробности - По умолчанию
`disabled`	`Boolean` - отключает кнопку, если установлено значение true - `false`
`icons`	`Object` - значки для отображения - {primary: null, secondary: `null` }
`label`	`String` - текст, отображаемый в кнопке - `null`
`text`	`Boolean` - показать этикетку - `true`

Examples

Основное использование

Создайте элемент html ввода (или кнопки или якоря) и метод `button()` вызова `button()` для пользовательского интерфейса jQuery.

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

HTML

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

Прочитайте кнопка онлайн: https://riptutorial.com/ru/jquery-ui/topic/4600/кнопка

глава 11: Перетаскиваемый

Examples

Простой пример

Включить перетаскиваемые функции для любого элемента DOM.

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

Перетаскиваемый с ручкой

Вы можете использовать любой элемент в качестве дескриптора для перетаскивания другого элемента:

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

скрипка

Прочитайте Перетаскиваемый онлайн: https://riptutorial.com/ru/jquery-ui/topic/522/перетаскиваемый

глава 12: ползунок

Examples

Простой пример

Управление ползунком использует перетаскиваемые ручки для выбора числовых значений. Ниже приведен пример базовой инициализации ползунка:

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

Ползунок диапазона

Ползунки диапазона обеспечивают две перетаскиваемые ручки для выбора числовых значений. Инициализация ползунка должна предусматривать параметр `range` установленный в `true` чтобы создать слайдер диапазона:

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

Инициализация значений и значений

Элемент слайдера может иметь свое значение, установленное при инициализации, предоставляя параметр `value` . Этот параметр - это номер:

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

Ползунок диапазона также может иметь свои значения, заданные таким образом, предоставляя параметр `values` . Этот параметр представляет собой массив чисел:

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

В дополнение к предоставлению начальных значений минимальное значение, максимальное значение и интервал обработки могут быть определены с помощью параметров `min` , `max` и `step` соответственно:

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

Использование слайд-события

Ползунок обеспечивает событие, называемое `slide` которое будет срабатывать **всякий раз, когда мышь перемещается во время перетаскивания ручки ползунка** . Эта функция содержит ссылку на `event` слайда и ссылку на объект slider `ui` . Объект `ui` содержит объект jQuery для перемещаемого элемента и значения (-ов) ползунка.

Односторонний слайдер:

```
var value;

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

Ползунок диапазона:

```
var lowValue;
var highValue;

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

Примечание. Событие `slide` предназначено для реагирования на активное движение мыши и не запускается, если значения ползунка изменены программно. Чтобы реагировать на эти события, используйте событие `change` .

Установка значений и событие изменения

Ползунок предоставляет событие, называемое `change` которое будет вызываться **после того, как мышь завершит перетаскивание рукоятки ползунка или если значение (ы)**

было изменено программно . Эта функция содержит ссылку на `event` слайда и ссылку на объект slider `ui` . Объект `ui` содержит объект jQuery для перемещаемого элемента и значения (-ов) ползунка.

Одним из примеров может быть необходимость отображения новой информации после того, как значения ползунка были обновлены событием другого элемента. Давайте используем элемент `select` для демонстрации, где значение ползунка программно устанавливается, когда изменяется значение `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;
    }
  });
});
```

Примечание. В этих обстоятельствах событие `slide` не запускается и событие `change` не требуется. Однако, если элементы должны реагировать на значения ползунка, изменяющиеся при перетаскивании дескриптора, потребуется событие `slide` .

Прочитайте ползунок онлайн: https://riptutorial.com/ru/jquery-ui/topic/3206/ползунок

глава 13: Сортируемый

Синтаксис

- $ ("# sortable"). sortable ({/ * Параметры здесь * /}); // Инициализация сортировки
- $ ("# sortable"). sortable ("option", "option_name", option_value); // Установить опцию за пределами инициализатора
- var value = $ ("# sortable"). sortable ("option", "option_name"); // Возвращает значение параметра

параметры

параметр	Описание
Опции	
добавить в	(jQuery, Element, Selector, String) [По умолчанию: «parent»] Элемент, который добавляется помощником в
ось	(String) [По умолчанию: false] Указания, по которым элемент можно перетащить (x или y)
отменить	(Селектор) [По умолчанию: «вход, текстовое поле, кнопка, выбор, опция»] Не запускает сортировку, если вы начинаете с элемента, соответствующего селектору
классы	(Object) [По умолчанию: {}] Укажите дополнительные классы для добавления элементов сортировки при добавлении структурных классов. ({ui-sortable-helper: custom_class})
соединить с	(Селектор) [По умолчанию: false] Позволяет перемещать элементы из одной сортируемой страницы в другую
политика сдерживания	(Элемент, Селектор, Строка) [По умолчанию: false] Элемент, для которого элементы ограничены
курсор	(String) [По умолчанию: «авто»] Определяет тип курсора, который будет отображаться при сортировке
cursorAt	(Object) [По умолчанию: false] Определяет позицию, по которой помощник выглядит как перемещаемый из

параметр	Описание
отключен	(Boolean) [По умолчанию: false] Отключает сортировку, если true
dropOnEmpty	(Boolean) [По умолчанию: true] Если ложные элементы из этой сортировки не могут быть помещены в пустые сортировки
forceHelperSize	(Boolean) [По умолчанию: false] Заставляет помощника иметь размер
forcePlaceholderSize	(Boolean) [По умолчанию: false] Заставляет местозаполнитель иметь размер
сетка	(Array) [По умолчанию: false] Определяет сетку для привязки хелпера к ([x, y])
справиться	(Selector, Element) [По умолчанию: false] Определяет элементы, которые могут начинаться с сортировки. Напротив отмены
помощник	(String, Function) [По умолчанию: «original»] Строка «original» или «clone» или функция, которая возвращает элемент, который будет использоваться в качестве помощника.
Предметы	(Селектор) [По умолчанию: "> *"] Определяет элементы, которые следует сортировать
помутнение	(Число от 0,01 до 1) [По умолчанию: false] Определяет непрозрачность для помощника
заполнитель	(String) [По умолчанию: false] Определяет класс или классы, которые должны применяться к заполнителю
возвращаться	(Boolean, Number) [По умолчанию: false] Время, когда помощник переходит в новое положение
свиток	(Boolean) [По умолчанию: true] Прокручивать ли по краям страницы
scrollSensitivity	(Number) [По умолчанию: 20] Определяет, насколько близко к краю страницы должен быть курсор, чтобы начать прокрутку
Scrollspeed	(Номер) [По умолчанию: 20] Скорость прокрутки
толерантность	(String) [По умолчанию: «intersect»] Определяет, какой режим использовать при расчете, когда один элемент находится над другим («пересекаться» или «указатель»)

параметр	Описание
ZIndex	(Integer) [По умолчанию: 1000] Определяет z-индекс помощника при сортировке
методы	
отменить()	Отменяет текущий сортировку и возвращает элементы обратно в их позицию до начала сортировки
уничтожить ()	Удаляет сортируемую функциональность и возвращает элемент в исходную инициализацию состояния
отключить ()	Отключает сортировку
включить()	Позволяет сортировать
пример()	Возвращает объект экземпляра sortables
опция ()	Получает пары ключевых значений всех параметров сортируемого
опция (String)	Возвращает значение параметра
option (String, Any)	Устанавливает значение параметра, заданного строкой
Опция (объект)	Устанавливает одну или несколько опций, когда объект является парными парами значений параметров
обновить ()	Обновляет параметры сортировки, перезагружая все сортируемые элементы. Это приводит к признанию новых элементов
refreshPositions ()	Убирает позиции кэша сортируемых элементов
сериализации (Объект)	Сериализует элементы ids (по умолчанию) в строку, которая может быть отправлена или добавлена к URL-адресу. Параметры объекта: {key: устанавливает ключ в сериализованной строке, атрибут: [Default "id"] устанавливает атрибут для просмотра, выражение : [По умолчанию: "/(.+)-=_/"] regex для разделения атрибута на пары значений ключа}
ToArray (Объект)	Сериализует идентификатор сортируемых элементов в массив. Объект может содержать атрибут параметра, который имеет атрибут для ввода в массив по умолчанию - id
виджет ()	Возвращает объект jQuery сортируемого элемента

параметр	Описание
События	
активировать (event, ui)	Запускаемый при подключении список, каждый подключенный список при перетаскивании начинает принимать его
beforeStop (событие, ui)	Триггеры перед сортировкой останавливаются, когда помощник переключается на ту же позицию, что и заполнитель
change (event, ui)	Запускается, когда элементы меняются, т.е. когда местозаполнитель перемещается
create (event, ui)	Срабатывается при сортировке
деактивировать (событие, ui)	Запускается при остановке сортировки. Это относится ко всем связанным спискам
out (событие, ui)	Запускается, когда элемент перемещается из сортируемого списка
over (event, ui)	Запускается, когда элемент перемещается в сортируемый список
получать (событие, ui)	Запускается, когда элемент из подключенного списка удаляется в другой. Цель - это список получателей
remove (event, ui)	Запускается, когда элемент из подключенного списка удаляется в другой. Цель - список
sort (event, ui)	Запущено во время сортировки
start (event, ui)	Запуск при старте сортировки
stop (event, ui)	Запуск при остановке сортировки
update (event, ui)	Срабатывается при остановке сортировки и обновлении позиции DOM

замечания

Официальная документация здесь

Examples

Простой пример

Возьмите любой список и добавьте идентификатор во внешнюю оболочку (`ul` , `div`)

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

В вашем jquery:

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

Это позволит перетаскивать и `#sortable` все `li` в `#sortable` wrapper в списке

Сортируемая сетка с гибкой компоновкой

Это использовало гибкую компоновку с сортировкой для создания сетки чувствительных ящиков, которые можно перемещать путем перетаскивания.

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;
}
```

Стационарные элементы при перетаскивании

В этом примере используется класс на заполнителе, чтобы превратить его в строку и сделать его занятым без места.

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;
}
```

Сортировка - ожидание возврата непринятого элемента

Рабочий пример: https://jsfiddle.net/Twisty/4f5yh3pa/7/

Отмена и возврат сортировки не задокументированы. Помогает показать, как перемещение элемента из одного списка в другой связанный список может быть условно отменено. по умолчанию это не анимируется сортировкой, этот пример включает анимацию.

Результат: List # 2 принимает только элементы, которые имеют класс `acceptable` . Оба списка могут быть отсортированы естественным образом в противном случае.

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();
});
```

Прочитайте Сортируемый онлайн: https://riptutorial.com/ru/jquery-ui/topic/1415/сортируемый

кредиты

S. No	Главы	Contributors
1	Начало работы с библиотекой пользовательского интерфейса jQuery	Community, J F, jkdev, JonasCz, Twisty
2	Datepicker	cteski, J F, ni8mr, Nishant123, Peter Tirrell, Pradeep, smdsgn, VincenzoC
3	jQuery UI Rotatable Plug-in	Twisty
4	jquery ui сортировать	Andrew Mcghie, Twisty
5	Автозаполнение	J F
6	аккордеон	Dipen Shah, Madalina Taina
7	волчок	Alon Eitan, Andrew Mcghie, depperm
8	диалог	J F, Jonathan Michalik, Racil Hilan, Theodore K., Twisty
9	Иконки	Theodore K.
10	кнопка	Theodore K.
11	Перетаскиваемый	empiric, J F
12	ползунок	Jonathan Michalik
13	Сортируемый	Alon Eitan, Andrew Mcghie, M B, Twisty