



**Kostenloses eBook**

**LERNEN**

# jQuery UI Library

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#jquery-ui**

# Inhaltsverzeichnis

Über.....	1
<b>Kapitel 1: Erste Schritte mit der jQuery UI Library.....</b>	<b>2</b>
Bemerkungen.....	2
Versionen.....	2
Examples.....	4
Hinzufügen des jQuery-UI-Skripts und grundlegende Verwendung.....	4
Einrichten der jQuery-Benutzeroberfläche für das erste Mal.....	4
<b>Kapitel 2: Akkordeon.....</b>	<b>7</b>
Syntax.....	7
Parameter.....	7
Bemerkungen.....	7
Examples.....	7
Akkordeon Grundlegende Verwendung.....	7
Akkordeon zerstören den Gebrauch.....	8
Akkordeon deaktivieren Verwendung.....	8
Akkordeon aktivieren Verwendung.....	8
Akkordeonoption Verwendung.....	8
Akkordeon-Refresh-Nutzung.....	9
Verwendung des Accordion-Widgets.....	9
<b>Kapitel 3: Autovervollständigung.....</b>	<b>10</b>
Examples.....	10
Einfaches Beispiel.....	10
<b>Kapitel 4: Datumsauswahl.....</b>	<b>11</b>
Examples.....	11
Initialisierung.....	11
Minimale und maximale Datumsangaben für eine Datumsauswahl festlegen.....	11
Zeige Woche des Jahres.....	11
Legen Sie ein benutzerdefiniertes Datumsformat fest.....	11
Monat und Jahr Dropdown anzeigen.....	13
<b>Kapitel 5: Dialog.....</b>	<b>14</b>

Syntax.....	14
Parameter.....	14
Bemerkungen.....	17
Examples.....	17
Einfaches Beispiel.....	17
Öffnet das Dialogfeld, wenn ein Ereignis auftritt.....	17
Komplexes Beispiel - jQuery-UI-Dialog zum dynamischen Erstellen.....	18
Erstellen eines Dialogs mit Tabbed Titelleiste.....	21
Dialog ohne Schließen-Schaltfläche.....	23
<b>Kapitel 6: Icons.....</b>	<b>24</b>
Syntax.....	24
Bemerkungen.....	24
Examples.....	24
Grundlegende Verwendung.....	24
<b>Kapitel 7: jQuery UI Drehbares Plug-In.....</b>	<b>25</b>
Parameter.....	25
Examples.....	25
Erstbenutzungsbeispiel.....	25
<b>Kapitel 8: jquery ui sortierbar.....</b>	<b>27</b>
Examples.....	27
jQuery UI Sortable - Platzhalter löschen.....	27
<b>Kapitel 9: Schieberegler.....</b>	<b>29</b>
Examples.....	29
Einfaches Beispiel.....	29
Bereichsschieberegler.....	29
Werte und Grenzwerte initialisieren.....	29
Verwenden des Folienereignisses.....	30
Einstellungswerte und das Änderungsereignis.....	30
<b>Kapitel 10: Sortierbar.....</b>	<b>32</b>
Syntax.....	32
Parameter.....	32
Bemerkungen.....	35

Examples.....	35
Einfaches Beispiel.....	35
Sortierbares Gitter mit flexiblem Layout.....	36
Stationäre Elemente beim Ziehen.....	36
Sortierbar - Animieren Sie die Rücknahme eines nicht akzeptierten Elements.....	37
<b>Kapitel 11: Spinner.....</b>	<b>40</b>
Syntax.....	40
Parameter.....	40
Bemerkungen.....	40
Examples.....	40
Basisbeispiel.....	40
<b>Kapitel 12: Taste.....</b>	<b>42</b>
Syntax.....	42
Parameter.....	42
Examples.....	42
Grundlegende Verwendung.....	42
<b>Kapitel 13: Ziehbar.....</b>	<b>43</b>
Examples.....	43
Einfaches Beispiel.....	43
Mit Griff ziehbar.....	43
<b>Credits.....</b>	<b>44</b>



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](mailto:info@zzzprojects.com)

---

# Kapitel 1: Erste Schritte mit der jQuery UI Library

## Bemerkungen

Die jQuery-Benutzeroberfläche ist eine JavaScript-Benutzeroberflächenbibliothek, die auf jQuery aufgebaut ist und eine Reihe von Interaktionen, Effekten und Widgets der Benutzeroberfläche bietet.

## Versionen

Ausführung	Veröffentlichungsdatum
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

<b>Ausführung</b>	<b>Veröffentlichungsdatum</b>
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

## Hinzufügen des jQuery-UI-Skripts und grundlegende Verwendung

Um mit der jQuery-UI-Bibliothek zu beginnen, müssen Sie das jQuery-Skript, das jQuery-UI-Skript und das jQuery-UI-Stylesheet zu Ihrem HTML-Code hinzufügen.

Laden Sie zunächst die jQuery-Benutzeroberfläche [herunter](#) . Wählen Sie auf der Downloadseite die gewünschten Funktionen aus. Entpacken Sie Ihren Download, und legen Sie `jquery-ui.css` und `jquery-ui.js` (und `jquery.js` ) in einem Ordner ab, in dem Sie sie in Ihrem HTML `jquery-ui.js` verwenden können (z. B. mit Ihren anderen Skripten und Stylesheets).

Die jQuery-Benutzeroberfläche hängt von jQuery ab. `jquery.js` `jquery-ui.js` also daran, `jquery.js` **VOR** `jquery-ui.js` .

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

Das ist es! Sie können jetzt die jQuery-Benutzeroberfläche verwenden. Verwenden Sie den Datepicker beispielsweise mit dem folgenden HTML-Code:

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

Dann benutze folgendes JavaScript:

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

Damit erhalten Sie ein schönes Datepicker-Popup:



Weitere Informationen finden Sie in der [offiziellen Anleitung "Erste Schritte"](#) .

## Einrichten der jQuery-Benutzeroberfläche für das erste Mal

Das jQuery-UI-Framework hilft dabei, die Steuerelemente der Benutzeroberfläche für die

JavaScript-Bibliothek von jQuery zu erweitern und zu erweitern.

Wenn Sie die jQuery-Benutzeroberfläche verwenden möchten, müssen Sie diese Bibliotheken Ihrem HTML-Code hinzufügen. Ein schneller Anfang ist die Verwendung der verfügbaren Codequellen des Content Delivery Network:

## jQuery-Bibliotheken

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

Sie können viele verschiedene Designs für die jQuery-Benutzeroberfläche auswählen und sogar Ihr eigenes Design erstellen. In diesem Beispiel verwenden wir 'Glätte'. Sie fügen das Design über CSS hinzu.

## jQuery UI CSS

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

## Alles zusammenfügen

Nachdem Sie Ihr CDN heruntergeladen oder ausgewählt haben, möchten Sie diese Bibliotheken und Stylesheets jetzt Ihrem HTML-Code hinzufügen, damit Ihre Webseite nun die Benutzeroberfläche von jQuery und jQuery verwenden kann. Die Reihenfolge, in der Sie die Bibliotheken laden, ist wichtig. Rufen Sie zuerst die jQuery-Bibliothek und dann Ihre jQuery-UI-Bibliothek auf. Da jQuery UI jQuery erweitert, muss es danach aufgerufen werden. Ihr HTML-Code kann ungefähr wie folgt aussehen.

```
<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>
```

Erste Schritte mit der jQuery UI Library online lesen: <https://riptutorial.com/de/jquery-ui/topic/513/erste-schritte-mit-der-jquery-ui-library>

# Kapitel 2: Akkordeon

## Syntax

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

## Parameter

Parameter	Detail
aktiv	Typ Boolean oder Integer. Boolean erfordert, dass collapsible den Wert true hat
animieren	Geben Sie Boolean, Number, String oder Object ein
zusammenklappbar	Geben Sie Boolean ein
Veranstaltung	Geben Sie String ein
Header	Typauswahlelement
heightStyle	Geben Sie String ein
Symbole	Geben Sie ein jQuery-UI-Symbolobjekt ein

## Bemerkungen

Weitere Informationen finden Sie hier: <http://api.jqueryui.com/accordion/>

## Examples

### Akkordeon Grundlegende Verwendung

Um ein Akkordeon zu verwenden, müssen Kopfzeilen und Inhalt in den Kopfzeilen im HTML-Code enthalten sein. Dann muss man die `accordion()` Methode der jQuery-Benutzeroberfläche instanzieren.

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

Im HTML:

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



## Akkordeon zerstören den Gebrauch

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

Dadurch wird die Akkordeonfunktionalität vollständig entfernt und der Standard-HTML-Code angezeigt, in dem alle jQuery-UI-Elemente entfernt werden.

Diese Methode akzeptiert keine Argumente.

## Akkordeon deaktivieren Verwendung

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

Diese Methode deaktiviert das Akkordeon, dh die Header sind nicht auswählbar, sodass der Inhalt nur lesbar und statisch ist.

Diese Methode akzeptiert keine Argumente.

## Akkordeon aktivieren Verwendung

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

Diese Methode ermöglicht ein Akkordeon. Dadurch wird ein deaktiviertes Akkordeon aktiviert oder es wird einfach nichts mit einem bereits aktivierten Akkordeon getan.

Diese Methode akzeptiert keine Argumente.

## Akkordeonoption Verwendung

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

Dadurch wird ein PlainObject zurückgegeben, das alle Optionen für das ausgewählte Akkordeon enthält. Dieser enthält alle Werte der Schlüssel, die im Abschnitt Parameter erläutert werden.

Diese Methode verwendet Parameter, die die im Parameter erläuterten grundlegenden optionNames sind. Man kann die Optionen folgendermaßen einstellen:

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

## Akkordeon-Refresh-Nutzung

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

Diese Methode berechnet die Höhe der Akkordeonfelder erneut, wenn im DOM Header oder Inhalt hinzugefügt oder entfernt wurden.

## Verwendung des Accordion-Widgets

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

Diese Methode gibt ein jQuery-Objekt zurück, das das Akkordeon enthält.

Akkordeon online lesen: <https://riptutorial.com/de/jquery-ui/topic/630/akkordeon>

---

# Kapitel 3: Autovervollständigung

## Examples

### Einfaches Beispiel

Die Widgets "Autocomplete" enthalten Vorschläge, während Sie das Feld eingeben.

```
<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' />
```

Autovervollständigung online lesen: <https://riptutorial.com/de/jquery-ui/topic/519/autovervollstandigung>

# Kapitel 4: Datumsauswahl

## Examples

### Initialisierung

Der **Datepicker** wird verwendet, um eine interaktive **Datumsauswahl anzuzeigen**, die an ein Standardformulareingabefeld gebunden ist. Es macht die Datumsauswahl für Eingabeaufgaben sehr einfach und ist auch in hohem Maße konfigurierbar.

Jedes Eingabefeld kann mit jquery-ui datepicker vom Auswahlfeld des Eingabefelds (id, class usw.) mit der **datepicker ()**-Methode wie **folgt gebunden werden**.

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

Live-Demo ist [da](#) .

### Minimale und maximale Datumsangaben für eine Datumsauswahl festlegen

```
<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">
```

### Zeige Woche des Jahres

Der folgende Code zeigt die Woche der Jahreszahl auf der linken Seite des Datepickers. Die Woche beginnt standardmäßig am Montag, kann jedoch mit der `firstDay` Option angepasst werden. Die erste Woche des Jahres enthält den ersten Donnerstag des Jahres nach der Definition von ISO 8601.

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

### Legen Sie ein benutzerdefiniertes Datumsformat fest

**Standard-Datumsformat:** "mm / tt / jj"

Das folgende Beispiel zeigt, wie Sie das Datumsformat bei der Initialisierung mit der Option `dateFormat` einstellen können.

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

Das folgende Beispiel zeigt, wie Sie das Datumsformat nach der Initialisierung mit der Option `dateFormat` einstellen können.

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

Sie können Kombinationen der folgenden verwenden:

```
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
```

Oder vordefinierter Standard:

```
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)
```

Ein Standard-Datumsformat kann auf alle Datumsauswahl angewendet werden. Beispiel:

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

## Monat und Jahr Dropdown anzeigen

jQuery Datepicker bietet zwei Optionen, um Dropdowns für die Auswahl von Monat und Jahr anzeigen zu lassen. Diese Optionen erleichtern die Navigation durch große Zeiträume.

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

Datumsauswahl online lesen: <https://riptutorial.com/de/jquery-ui/topic/520/datumsauswahl>

# Kapitel 5: Dialog

## Syntax

- `$ ("selector") .dialog ("Option", "deaktiviert");` // Option Getter, spezifisch
- `$ ("selector") .dialog ("Option");` // Option Alle abrufen
- `$ ("selector") .dialog ("Option", "deaktiviert", true);` // Option Setter, spezifisch
- `$ ("selector") .dialog ("Option", {disabled: true});` // Option Setter, mehrere
- `$ ("selector") .dialog ("close");` // Trigger schließen
- `$ ("selector") .dialog ({close: function () {}});` // Überladung schließen
- `$ ("selector") .on ("dialogclose", Funktion (Ereignis, ui) {});` // Überladung schließen

## Parameter

Parameter	Beschreibung
<b>Optionen</b>	
Anhängen	(Selector) [Default: "body" ] An welches Element soll der Dialog (und die Überlagerung, falls modal) angehängt werden?
autoOpen	(Boolean) [Default: true ] Wenn true festgelegt ist, wird das Dialogfeld bei der Initialisierung automatisch geöffnet. Bei false bleibt der Dialog ausgeblendet, bis die Methode open () aufgerufen wird.
Tasten	(Objekt / Array) Gibt an, welche Schaltflächen im Dialogfeld angezeigt werden sollen. Der Kontext des Rückrufs ist das Dialogelement. Wenn Sie auf die Schaltfläche zugreifen müssen, ist sie als Ziel des Ereignisobjekts verfügbar.
closeOnEscape	(Boolean) [Standard: true ] Gibt an, ob das Dialogfeld geschlossen werden soll, wenn der Fokus festgelegt ist und der Benutzer die Escape-Taste (ESC) drückt.
closeText	(String) [Standard: "close" ] Gibt den Text für die Schaltfläche "close" . Beachten Sie, dass der geschlossene Text bei Verwendung eines Standarddesigns sichtbar ausgeblendet wird.
dialogClass	(String) Die angegebenen Klassennamen werden dem Dialog für zusätzliche Themen hinzugefügt.
ziehbar	(Boolean) [Default: true ] Wenn diese Einstellung auf true , kann das Dialogfeld von der Titelleiste gezogen werden. Erfordert das Draggable-Widget für die jQuery-Benutzeroberfläche.
Höhe	(Number / String) [Default: "auto" ] Die Höhe des Dialogs.

Parameter	Beschreibung
verbergen	(Bool / Num / Str / Obj) Ob und wie das Ausblenden des Dialogs animiert wird.
maximale Höhe	(Anzahl) [Standard: <code>false</code> ] Die maximale Höhe, bis zu der der Dialog in Pixel geändert werden kann.
maximale Breite	(Anzahl) [Standard: <code>false</code> ] Die maximale Breite, bis zu der der Dialog in Pixel geändert werden kann.
minHöhe	(Anzahl) [Standard: <code>150</code> ] Die minimale Höhe, bis zu der der Dialog in Pixel geändert werden kann.
minWidth	(Anzahl) [Standard: <code>150</code> ] Die minimale Breite, bis zu der der Dialog in Pixel geändert werden kann.
modal	(Boolean) [Standardeinstellung: <code>false</code> ] Wenn der Wert auf <code>true</code> gesetzt ist, weist das Dialogfeld ein modales Verhalten auf. Andere Elemente auf der Seite werden deaktiviert, dh, mit ihnen kann nicht interagiert werden. Modale Dialoge erstellen eine Überlagerung unter dem Dialog, aber über anderen Seitenelementen.
Position	(Objekt) [Standard: <code>{ my: "center", at: "center", of: window }</code> ] Gibt an, wo der Dialog beim Öffnen angezeigt werden soll. Der Dialog behandelt Kollisionen so, dass möglichst viel des Dialogs sichtbar ist.
der Größe veränderbar	(Boolean) [Default: <code>true</code> ] Wenn auf <code>true</code> festgelegt, wird die Größe des Dialogfelds geändert. Erfordert, dass das Widget für die anpassbare jQuery-Benutzeroberfläche enthalten ist.
Show	(Bool / Num / Str / Obj) Ob und wie die Anzeige des Dialogs animiert werden soll.
Titel	(String) Gibt den Titel des Dialogs an. Wenn der Wert null ist, wird das <code>title</code> -Attribut des Dialogquellenelements verwendet.
Breite	(Anzahl) [Standard: <code>300</code> ] Die Breite des Dialogs in Pixel.
<b>Methoden</b>	
schließen	Schließt den Dialog.
zerstören	Entfernt die Dialogfunktion vollständig. Dadurch wird das Element in den Zustand vor dem Initialisieren zurückgesetzt.
Beispiel	Ruft das Instanzobjekt des Dialogs ab. Wenn das Element keine zugeordnete Instanz hat, wird <code>undefined</code> zurückgegeben.
ist offen	Ob der Dialog aktuell geöffnet ist.

Parameter	Beschreibung
moveToTop	Verschiebt den Dialog an den Anfang des Dialogstapels.
öffnen	Öffnet den Dialog.
Möglichkeit	Ruft den aktuell mit dem angegebenen <code>optionName</code> Wert <code>optionName</code> .
Möglichkeit	Ruft ein Objekt ab, das Schlüssel / Wert-Paare enthält, die den aktuellen Dialogoptionshash darstellen.
Möglichkeit	Legt eine oder mehrere Optionen für den Dialog fest.
Widget	Gibt ein jQuery-Objekt zurück, das den generierten Wrapper enthält.
<b>Erweiterungspunkte</b>	
<code>_allowInteraction</code>	(Ereignis) Modale Dialoge erlauben Benutzern nicht die Interaktion mit Elementen hinter dem Dialog. Dies kann für Elemente problematisch sein, die keine untergeordneten Elemente des Dialogs sind, aber absolut positioniert sind, um so zu erscheinen, als wären sie. Die Methode <code>_allowInteraction()</code> bestimmt, ob der Benutzer mit einem bestimmten <code>_allowInteraction()</code> interagieren darf. Daher kann es verwendet werden, um Elemente auf die Positivliste zu setzen, die keine untergeordneten Elemente des Dialogfelds sind, die Benutzer jedoch verwenden können.
<b>Veranstaltungen</b>	
vorSchließen	(event, ui) Wird ausgelöst, wenn ein Dialogfeld geschlossen wird. Wenn Sie den Vorgang abbrechen, wird der Dialog nicht geschlossen.
schließen	(event, ui) Wird ausgelöst, wenn der Dialog geschlossen wird.
erstellen	(event, ui) Wird ausgelöst, wenn der Dialog erstellt wird.
ziehen	(event, ui {position, offset}) Wird ausgelöst, wenn der Dialog gezogen wird.
DragStart	(event, ui {position, offset}) Wird ausgelöst, wenn der Benutzer beginnt, den Dialog zu ziehen.
DragStop	(event, ui {position, offset}) Wird ausgelöst, nachdem der Dialog gezogen wurde.
Fokus	(event, ui) Wird ausgelöst, wenn der Dialog den Fokus erhält.
öffnen	(event, ui) Wird beim Öffnen des Dialogs ausgelöst.
Größe ändern	(event, ui {originalPosition, position, originalSize, size}) Wird

Parameter	Beschreibung
	ausgelöst, während die Größe des Dialogfelds geändert wird.
resizeStart	(event, ui {originalPosition, position, originalSize, size}) Wird ausgelöst, während die Größe des Dialogfelds geändert wird.
resizeStop	(event, ui {originalPosition, position, originalSize, size}) Wird ausgelöst, während die Größe des Dialogfelds geändert wird.

## Bemerkungen

Parameterquelle: <http://api.jqueryui.com/dialog/>

## Examples

### Einfaches Beispiel

Dialog ist ein Fenster, das im Ansichtsfenster überlagert wird.

```
<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>
```

### Öffnet das Dialogfeld, wenn ein Ereignis auftritt

Normalerweise möchten wir die Erstellung des Dialogs von seinem Erscheinungsbild trennen. Dann sind drei Schritte erforderlich.

#### 1. Basiselement erstellen

```
<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. Machen Sie einen Dialog, beachten Sie die Option `autoOpen: false`, die sicherstellt, dass sie zuerst geschlossen wird

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

### 3. Öffnen Sie es bei Bedarf wie auf einen Knopfdruck

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

## Komplexes Beispiel - jQuery-UI-Dialog zum dynamischen Erstellen

Im Allgemeinen basiert der Dialog auf einem `div` im HTML-Code. Manchmal möchten Sie einen Dialog programmgesteuert von Grund auf erstellen. Hier ist ein Beispiel eines komplexen modalen Dialogs, der dynamisch mit interaktiven Funktionen erstellt wird.

### HTML

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

### CSS

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

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

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

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

```

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

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

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

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

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

```

## jQuery

```

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

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

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

```

```

}

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

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

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

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

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

// Creation of the dialog object
dialog = $("<div>", {
  id: "dialog-form",
  title: "Create New User"
}).dialog({
  autoOpen: false,
  height: 400,
  width: 350,
  modal: true,
  buttons: {
    "Create an account": addUser,
    Cancel: function() {
      dialog.dialog("close");
    }
  },
  close: function() {
    form[0].reset();
    allFields.removeClass("ui-state-error");
  }
});

```

```

    }
  });

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

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

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

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

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

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

```

Arbeitsbeispiel für Referenz: <https://jsfiddle.net/Twisty/LqjuxLu1/>

## Erstellen eines Dialogs mit Tabbed Titelleiste

Gelegentlich möchten wir möglicherweise Dialoge mit mehr als einem Inhaltsbereich anzeigen. Die jQuery-Benutzeroberfläche bietet Registerkarten, die zusammen mit einem Dialog verwendet werden können, um dies zu ermöglichen. Zwar ist es üblicher, dass sich Registerkarten im Inhaltscontainer eines Dialogfelds befinden. In diesem Beispiel wird gezeigt, wie eine Liste von Registerkarten zur Titelleiste des Dialogfelds wird.

### 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");
  });

```

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

Arbeitsbeispiel als Referenz: <https://jsfiddle.net/5x4zz681/>

## Dialog ohne Schließen-Schaltfläche

Wenn Sie den Dialog ohne die Schaltfläche "Schließen" (z. B. die Schaltfläche "x" in der rechten oberen Ecke des Dialogs) anzeigen möchten, möglicherweise, weil Sie den Benutzer zwingen möchten, eine der Optionen oder Schaltflächen im Dialog selbst auszuwählen:

1- Geben Sie Ihrem Dialog eine CSS-Klasse:

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

2- Verbergen Sie die Schließen-Schaltfläche mit diesem CSS:

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

Hinweis: Wenn Sie die gesamte Titelleiste ausblenden möchten, verwenden Sie stattdessen dieses CSS:

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

Alternativ können Sie die Schließen-Schaltfläche im Initialisierungscode des Dialogs ausblenden:

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

Dialog online lesen: <https://riptutorial.com/de/jquery-ui/topic/521/dialog>

---

# Kapitel 6: Icons

## Syntax

- `.ui-icon- {icon type} - {icon Unterbeschreibung} - {direction}`

## Bemerkungen

Die Symbole werden auch in einer Reihe von jQuery UI - Widgets, wie integrierte [Akkordeon](#) , Taste, Menü.

## Examples

### Grundlegende Verwendung

`ui-icon-arrowthick-1-n` für einen dicken Pfeil nach Norden in einer Spanne die Klassen `ui-icon` und `ui-icon-arrowthick-1-n` :

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

Für ein Dreieck, das in einer Spanne nach Süden zeigt, fügen Sie die Klassen `ui-icon` und `ui-icon-triangle-1-s` :

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

Vollständige Liste der verfügbaren Artikel hier <https://api.jqueryui.com/theming/icons/>

**Icons online lesen:** <https://riptutorial.com/de/jquery-ui/topic/4633/icons>

# Kapitel 7: jQuery UI Drehbares Plug-In

## Parameter

Parameter	Einzelheiten
Griff	URL zu einem benutzerdefinierten Bild für den Griff
Winkel	die Startrotation für das Element.
rotationCenterX	Position, um die das Element gedreht wird
rotationCenterY	Position, um die das Element gedreht wird
Schritt	ein Winkel in Grad, auf den die Drehung einrastet, wenn die Umschalttaste gedrückt wird
schnappen	schnappt schrittweise ab.
Start	Wird ausgelöst, wenn die Rotation beginnt
halt	Wird ausgelöst, wenn die Rotation stoppt
drehen	Wird ausgelöst, wenn ein Objekt gedreht wird
wheelRotate	Aktivieren / Deaktivieren des Mausekzes zum Drehen des Elements.

## Examples

### Erstbenutzungsbeispiel

jquery-ui-rotatable ist ein Plugin für die jQuery-Benutzeroberfläche, das auf ähnliche Weise wie Draggable und Resizable funktioniert, ohne dass es alle Funktionen bietet. Standardmäßig wird ein kleines Rotationssymbol links unten in dem Element angezeigt, das Sie drehen möchten.

```
<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.jsdelivrivr.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.jsdelivrivr.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>
```

jQuery UI Drehbares Plug-In online lesen: <https://riptutorial.com/de/jquery-ui/topic/1806/jquery-ui-drehbares-plug-in>

# Kapitel 8: jquery ui sortierbar

## Examples

### jQuery UI Sortable - Platzhalter löschen

Dieses Beispiel für die Sortierung mithilfe eines Platzhalters wird häufig verwendet. Sortable wird auf eine Gruppe von DOM-Elementen angewendet, sodass der Benutzer Elemente in der Liste mit Drag-and-Drop-Aktionen verschieben kann.

```
<!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>
```

jquery ui sortierbar online lesen: <https://riptutorial.com/de/jquery-ui/topic/1226/jquery-ui-sortierbar>

# Kapitel 9: Schieberegler

## Examples

### Einfaches Beispiel

Ein Schieberegler verwendet ziehbare Ziehpunkte, um numerische Werte auszuwählen. Unten sehen Sie ein Beispiel für eine grundlegende Initialisierung des Schiebereglers:

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

### Bereichsschieberegler

Bereichsschieberegler bieten zwei ziehbare Ziehpunkte zum Auswählen numerischer Werte. Die Initialisierung des Schiebereglers muss eine `range` enthalten, die auf `true`, um einen Bereichsschieberegler zu erstellen:

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

### Werte und Grenzwerte initialisieren

Der Wert eines Schiebereglerelements kann bei der Initialisierung durch `value` Wertoption festgelegt werden. Diese Option ist eine Nummer:

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

Die Werte eines Bereichsschiebereglers können auch auf diese Weise festgelegt werden, indem eine `values` bereitgestellt wird. Diese Option ist ein Array von Zahlen:

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

Neben der Angabe von Anfangswerten können der Minimalwert, der Maximalwert und das Handle-Intervall mit den Optionen `min`, `max` bzw. `step` definiert werden:

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

## Verwenden des Folienergebnisses

Der Schieberegler bietet ein Ereignis namens `slide`, das ausgelöst wird, **wenn sich die Maus während des Ziehens des Schiebereglers bewegt**. Diese Funktion enthält einen Verweis auf den Schlitten `event` und einem Verweis auf den Schieber `ui` - Objekt. Das `ui` Objekt enthält ein jQuery-Objekt für das bewegte Handle und die Wert (`e`) des Schiebereglers.

### Einhand-Schieberegler:

```
var value;

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

### Bereichsschieberegler:

```
var lowValue;
var highValue;

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

**Hinweis:** Das `slide` reagiert auf aktive Mausbewegungen und wird nicht ausgelöst, wenn die Schiebereglerwerte programmgesteuert geändert werden. Um auf diese Ereignisse zu reagieren, verwenden Sie das `change`.

## Einstellungswerte und das Änderungsereignis

Der Schieberegler bietet ein Ereignis namens `change`, das ausgelöst wird, **nachdem der Mauszeiger mit dem Schieberegler gezogen wurde oder wenn die Werte programmgesteuert geändert wurden**. Diese Funktion enthält einen Verweis auf den Schlitten `event` und einem Verweis auf den Schieber `ui` - Objekt. Das `ui` Objekt enthält ein jQuery-Objekt für das bewegte Handle und die Wert (`e`) des Schiebereglers.

Ein Beispiel könnte sein, dass neue Informationen angezeigt werden müssen, nachdem die Werte eines Schiebereglers durch das Ereignis eines anderen Elements aktualisiert wurden. Verwenden Sie ein `select` Element zur Demonstration, bei dem der Wert des Schiebereglers programmgesteuert festgelegt wird, wenn sich der Wert des `select` ändert:

## 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;
    }
  });
});
```

**Hinweis:** Unter diesen Umständen wird das `slide` nicht ausgelöst und das `change` ist erforderlich. Wenn Elemente jedoch darauf reagieren müssen, dass sich die Schiebereglerwerte beim Ziehen des Ziehpunkts ändern, ist das `slide` Ereignis erforderlich.

Schieberegler online lesen: <https://riptutorial.com/de/jquery-ui/topic/3206/schieberegler>

# Kapitel 10: Sortierbar

## Syntax

- `$("# sortable"). sortable ({/ * Options Here * /}); // Sortierbar initialisieren`
- `$("# sortable"). sortable ("option", "option_name", option_value); // Option außerhalb des Initialisierers setzen`
- `var value = $("# sortable"). sortable ("option", "option_name"); // Ruft den Wert einer Option ab`

## Parameter

Parameter	Beschreibung
<b>Optionen</b>	
Anhängen	(jQuery, Element, Selector, String) [Standard: "parent"] Das Element, zu dem der Helfer hinzugefügt wird
Achse	(String) [Standard: false] Die Richtungen, in die das Element gezogen werden kann (x oder y).
stornieren	(Auswahl) [Standard: "Eingabe, Textbereich, Schaltfläche, Auswahl, Option"] Startet die Sortierung nicht, wenn Sie mit einem Element beginnen, das mit dem Auswahlelement übereinstimmt
Klassen	(Object) [Default: {}] Geben Sie zusätzliche Klassen an, die beim Hinzufügen der Strukturklassen zu den sortierbaren Elementen hinzugefügt werden sollen. ({ui-sortable-helper: custom_class})
verbinden mit	(Auswahl) [Standard: false] Ermöglicht das Ziehen von Elementen aus einer sortierbaren Tabelle in eine andere
Eindämmung	(Element, Selector, String) [Standard: false] Das Element, auf das Elemente beschränkt sind
Mauszeiger	(String) [Standard: "auto"] Definiert den Typ des Cursors, der beim Sortieren angezeigt werden soll
CursorAt	(Objekt) [Standard: false] Definiert die Position, an der der Helfer aussieht, als würde er verschoben
deaktiviert	(Boolean) [Standard: false] Deaktiviert die Sortierung, wenn true
dropOnEmpty	(Boolean) [Default: true] Wenn falsche Elemente aus dieser

Parameter	Beschreibung
	sortierbaren Tabelle nicht in leere sortierbare Elemente eingefügt werden können
forceHelperSize	(Boolean) [Standard: false] Erzwingt, dass der Helfer eine Größe hat
forcePlaceholderSize	(Boolean) [Standard: false] Erzwingt die Größe des Platzhalters
Gitter	(Array) [Standard: false] Definiert ein Raster, an dem der Helfer ausgerichtet werden soll ([x, y]).
Griff	(Auswahl, Element) [Standard: false] Definiert Elemente, mit denen die Sortierung beginnen kann. Gegenteil zu stornieren
Helfer	(String, Funktion) [Default: "original"] String "original" oder "clone" oder eine Funktion, die das als Helfer zu verwendende Element zurückgibt.
Artikel	(Auswahl) [Standard: "> *"] Definiert die Elemente, die sortierbar sein sollen
Opazität	(Zahl 0,01 bis 1) [Standard: false] Legt die Deckkraft für den Helfer fest
Platzhalter	(String) [Standard: false] Definiert eine oder mehrere Klassen, die auf den Platzhalter angewendet werden sollen
zurückkehren	(Boolean, Number) [Standard: false] Die Zeit, die der Helfer benötigt, um in seine neue Position zu rutschen
scrollen	(Boolean) [Standard: true] Gibt an, wenn an den Rändern der Seite gescrollt werden soll
scrollSensitivity	(Anzahl) [Standard: 20] Legt fest, wie nahe der Cursor an der Seite sein muss, um den Bildlauf zu starten
Scrollgeschwindigkeit	(Anzahl) [Standard: 20] Die Geschwindigkeit, mit der gescrollt werden soll
Toleranz	(String) [Default: "intersect"] Legt fest, welcher Modus bei der Berechnung verwendet werden soll, wenn ein Element über einem anderen liegt ("intersect" oder "pointer").
zIndex	(Ganzzahl) [Standard: 1000] Definiert den Z-Index des Helpers beim Sortieren
<b>Methoden</b>	
stornieren()	Bricht die aktuelle Sortierung ab und bringt die Elemente vor dem

Parameter	Beschreibung
	Sortiervorgang an ihre Position zurück
zerstören()	Entfernt die sortierbare Funktion und versetzt das Element in den Zustand vor der Initialisierung
deaktivieren ()	Deaktiviert die Sortierung
aktivieren()	Aktiviert die Sortierung
Beispiel()	Gibt das sortables-Instanzobjekt zurück
Möglichkeit()	Ruft Schlüsselwertpaare aller Optionen für die Sortierbarkeit ab
Option (String)	Ruft den Wert einer Option ab
Option (String, Any)	Legt den Wert der durch den String angegebenen Option fest
Option (Objekt)	Legt eine oder mehrere Optionen fest, wobei das Objekt aus Schlüsselwertpaaren besteht
Aktualisierung()	Aktualisiert die sortierbaren Optionen, indem alle sortierbaren Elemente geladen werden. Dadurch werden neue Elemente erkannt
refreshPositions ()	Gibt die zwischengespeicherten Positionen der sortierbaren Elemente zurück
serialisieren (Objekt)	Serialisiert die Element-IDs (standardmäßig) in eine Zeichenfolge, die gesendet oder an eine URL angehängt werden kann. Objektoptionen: {Schlüssel: Legt den Schlüssel in der serialisierten Zeichenfolge fest, Attribut: [Standard "ID"] legt fest, dass das Attribut ausdruck angezeigt wird : [Standard: "/(.+)-=_/"] Regex, um das Attribut in Schlüsselwertpaare aufzuteilen}
toArray (Objekt)	Serialisiert die ID der sortierbaren Elemente in ein Array. Das Objekt kann ein Parameterattribut enthalten, das das Attribut enthält, das in das Array eingefügt werden soll
Widget ()	Gibt ein jQuery-Objekt des sortierbaren Elements zurück
<b>Veranstaltungen</b>	
aktivieren (event, ui)	Wird bei einer verbundenen Liste ausgelöst, erhält jede verbundene Liste beim Ziehen den Start
beforeStop (Ereignis, ui)	Wird ausgelöst, bevor die Sortierung stoppt, wenn der Helfer in dieselbe Position wie der Platzhalter verschoben wurde
ändern (event, ui)	Wird ausgelöst, wenn Elemente ihre Position ändern. wenn sich der platzhalter bewegt

Parameter	Beschreibung
erstellen (event, ui)	Wird ausgelöst, wenn sortierbar erstellt wird
deaktivieren (event, ui)	Wird ausgelöst, wenn die Sortierung stoppt. Dies gilt auch für alle verbundenen Listen
out (event, ui)	Wird ausgelöst, wenn das Element aus der sortierbaren Liste verschoben wird
vorbei (event, ui)	Wird ausgelöst, wenn der Artikel in eine sortierbare Liste verschoben wird
empfangen (Ereignis, ui)	Wird ausgelöst, wenn ein Element aus einer verbundenen Liste in eine andere fallen gelassen wurde. Ziel ist die Empfangsliste
entfernen (event, ui)	Wird ausgelöst, wenn ein Element aus einer verbundenen Liste in eine andere fallen gelassen wurde. Ziel ist die gebende Liste
sort (event, ui)	Wird während des Sortierens ausgelöst
start (event, ui)	Wird ausgelöst, wenn die Sortierung beginnt
Stop (Ereignis, ui)	Wird ausgelöst, wenn die Sortierung stoppt
Update (Ereignis, ui)	Wird ausgelöst, wenn die Sortierung stoppt und die DOM-Position aktualisiert wurde

## Bemerkungen

Offizielle Dokumentation [hier](#)

## Examples

### Einfaches Beispiel

Nehmen Sie eine beliebige Liste und fügen Sie dem äußeren Wrapper eine Kennung hinzu ( `ul` , `div` )

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

In deiner JQuery:

```
$(function() {
```

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

Dadurch können alle `li` im `#sortable` Wrapper in die Liste gezogen und abgelegt werden

## Sortierbares Gitter mit flexiblem Layout

Dabei wurde das Flex-Layout mit der Sortierbarkeit verwendet, um ein Raster aus Antwortboxen zu erstellen, die per Drag & Drop verschoben werden können.

### 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;
}
```

## Stationäre Elemente beim Ziehen

In diesem Beispiel wird eine Klasse auf dem Platzhalter verwendet, um daraus eine Zeile zu machen und keinen Platz zu beanspruchen.

### 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;
}
```

## Sortierbar - Animieren Sie die Rücknahme eines nicht akzeptierten Elements

Arbeitsbeispiel: <https://jsfiddle.net/Twisty/4f5yh3pa/7/>

Das Abbrechen und Rückgängigmachen einer Sortierung ist nicht streng dokumentiert. Die Hilfe zeigt, wie das Verschieben eines Elements von einer Liste in eine andere verbundene Liste bedingt abgebrochen werden kann. Standardmäßig ist dies nicht durch Sortierung animiert. Dieses Beispiel enthält eine Animation.

Ergebnis: Liste Nr. 2 akzeptiert nur Elemente, die eine `acceptable` Klasse haben. Beide Listen können ansonsten natürlich sortiert werden.

## 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();
});
```

Sortierbar online lesen: <https://riptutorial.com/de/jquery-ui/topic/1415/sortierbar>

# Kapitel 11: Spinner

## Syntax

- `$("#id").spinner();`
- `$("#id").spinner({min: 0, max: 100, Schritt: 5, Spin: Funktion (Ereignis, ui) {}});`

## Parameter

Parameter	Detail
Mindest	Mindestwert
max	Höchster Wert
Schritt	Wie viel der Wert durch Klicken auf den Drehknopf steigt, kann dezimal sein
Dreh	Kann verwendet werden, um den <code>ui.value</code> , den <code>ui.value</code> zu überprüfen und etwas zu tun

## Bemerkungen

Offizielles [Beispiel](#)

Offizielle [Dokumentation](#)

## Examples

### Basisbeispiel

Ermöglicht die Eingabe von Zahlen durch die Anzeige von Pfeilen auf der rechten Seite der `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">
```

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

# Kapitel 12: Taste

## Syntax

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

## Parameter

Parameter	Typ - Details - Standard
<code>disabled</code>	<code>Boolean</code> - Deaktiviert die Schaltfläche, wenn sie auf <code>true</code> gesetzt ist - <code>false</code>
<code>icons</code>	<code>Object</code> - anzuzeigende Symbole - <code>{primär: null, sekundär: null }</code>
<code>label</code>	<code>String</code> - In der Schaltfläche <code>null</code> <code>Text</code> - <code>null</code>
<code>text</code>	<code>Boolean</code> - Ob das Label angezeigt werden soll - <code>true</code>

## Examples

### Grundlegende Verwendung

Erstellen Sie ein Eingabe- (oder Schaltflächen- oder Anker-) HTML-Element und rufen Sie die `button()` Methode der jQuery-Benutzeroberfläche auf.

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

### HTML

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

Taste online lesen: <https://riptutorial.com/de/jquery-ui/topic/4600/taste>

---

# Kapitel 13: Ziehbar

## Examples

### Einfaches Beispiel

Aktivieren Sie die ziehbare Funktion für ein beliebiges DOM-Element.

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

### Mit Griff ziehbar

Sie können jedes Element als Ziehpunkt verwenden, um ein anderes Element herumzuziehen:

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

---

## Geige

Ziehbar online lesen: <https://riptutorial.com/de/jquery-ui/topic/522/ziehbar>

# Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit der jQuery UI Library	<a href="#">Community</a> , <a href="#">J F</a> , <a href="#">jkdev</a> , <a href="#">JonasCz</a> , <a href="#">Twisty</a>
2	Akkordeon	<a href="#">Dipen Shah</a> , <a href="#">Madalina Taina</a>
3	Autovervollständigung	<a href="#">J F</a>
4	Datumsauswahl	<a href="#">cteski</a> , <a href="#">J F</a> , <a href="#">ni8mr</a> , <a href="#">Nishant123</a> , <a href="#">Peter Tirrell</a> , <a href="#">Pradeep</a> , <a href="#">smdsgn</a> , <a href="#">VincenzoC</a>
5	Dialog	<a href="#">J F</a> , <a href="#">Jonathan Michalik</a> , <a href="#">Racil Hilan</a> , <a href="#">Theodore K.</a> , <a href="#">Twisty</a>
6	Icons	<a href="#">Theodore K.</a>
7	jQuery UI Drehbares Plug-In	<a href="#">Twisty</a>
8	jquery ui sortierbar	<a href="#">Andrew Mcghie</a> , <a href="#">Twisty</a>
9	Schieberegler	<a href="#">Jonathan Michalik</a>
10	Sortierbar	<a href="#">Alon Eitan</a> , <a href="#">Andrew Mcghie</a> , <a href="#">M B</a> , <a href="#">Twisty</a>
11	Spinner	<a href="#">Alon Eitan</a> , <a href="#">Andrew Mcghie</a> , <a href="#">depperm</a>
12	Taste	<a href="#">Theodore K.</a>
13	Ziehbar	<a href="#">empiric</a> , <a href="#">J F</a>