



UTN.BA FACULTAD
REGIONAL
BUENOS AIRES
SECRETARÍA DE EXTENSIÓN UNIVERSITARIA FRBA UTN

**Centro de
e-Learning**

Javascript, JQuery y JSON avanzado.



www.sceu.frba.utn.edu.ar/e-learning



Módulo 2

JQUERY UI E

INTERACCIÓN CON EL SERVIDOR



Que es JQuery UI, como instalarla y utilizarla.



Presentación de la Unidad:

En esta unidad aprenderán que es JQuery UI y cuales son los beneficios de su utilización.

Veremos paso a paso como instalarlo y comenzaremos a introducirnos en sus “Widgets”.



Objetivos:

- ❖ Qué es jQueryUI.
- ❖ Cuáles son los pasos que debes realizar para comenzar a usar estas librerías para la creación de interfaces de usuario enriquecidas y compatibles con todos los navegadores.
- ❖ Aprender a utilizar los “Widgets” de JQueryUI.



Temario:

- QUE ES JQUERYUI
- COMO DESCARGALA Y UTILIZARLA
- EJEMPLO 1: DATEPICKER
- OTRO EJEMPLO CON DATEPICKER
- EJEMPLO 2: ACCORDION
- EJEMPLO 3: TABS
- EJEMPLO 4: AUTOCOMPLETE
- EJEMPLO 5: DIALOG
- EJEMPLO 6: SLIDER
- EJEMPLO 7: MENU



CONSIGNAS PARA EL APRENDIZAJE COLABORATIVO

En esta Unidad los participantes se encontrarán con diferentes tipos de consignas que, en el marco de los fundamentos del MEC*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:



1. Los foros asociados a cada una de las unidades.
2. La Web 2.0.
3. Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen las actividades sugeridas y compartan en los foros los resultados obtenidos.

QUE ES JQUERYUI

jQuery UI es un complemento que permite implementar componentes diversos para generar interfaces de usuario en páginas web, además de otras funcionalidades básicas para crear aplicaciones web enriquecidas. Como su propio nombre indica, está basado en el popular framework Javascript y podemos encontrar links, explicaciones, así como demos y descargas a partir del sitio web oficial de jQuery.

Entre los scripts jQuery listos para usar que existen en Internet debemos de comenzar por los proporcionados en jQuery UI, ya que es la distribución oficial de componentes para la creación de interfaces de usuario avanzadas.

Así que comencemos con una serie de pasos iniciales necesarios para comenzar a usar jQuery UI en nuestras páginas web. Luego veremos como implementar funcionalidades y componentes avanzados que nos proporcionan estas librerías.

Podemos empezar por observar la cantidad de funcionalidades listas para usar, así como componentes (widgets) para implementar interfaces de usuario. Todo esto lo encontramos en <http://jqueryui.com/>.

Lo más interesante en esta página son los denominados widgets e interacciones, donde podremos encontrar ejemplos de uso de estas librerías para crear los típicos menús "acordeón", calendarios para seleccionar fechas, ventanas de diálogo, interfaces de pestañas, etc. Iremos viendo ejemplos de cada uno de ellos y cuales son las variantes para cada uno.

COMO DESCARGALA Y UTILIZARLA

Para comenzar realmente a usar jQuery UI necesitamos descargar las librerías, pero cabe decir que este sistema es bastante amplio, por lo que igual nos interesa descargar sólo una parte, para que no resulte demasiado pesado para las personas que visiten nuestro sitio web. Lo bueno es que la página de descargas de jQuery UI tiene un sistema para poder seleccionar sólo aquellos componentes que deseamos utilizar en nuestro sitio, con lo que la descarga se puede optimizar para únicamente contener aquellas cosas que realmente van a ser necesarias en nuestro sitio.

Podemos acceder a la página de descargas en: <http://jqueryui.com/download>

Download Builder

Quick downloads: Stable (Themes) (1.10.3 for jQuery 1.6+) | Legacy (Themes) (1.9.2 for jQuery 1.6+)

All jQuery UI Downloads

Version

☒ 1.10.3 (Stable, for jQuery 1.6+)

☐ 1.9.2 (Legacy, for jQuery 1.6+)

Components

☒ Toggle All

<p>UI Core</p> <p><input checked="" type="checkbox"/> Toggle All</p> <p>A required dependency, contains basic functions and initializers.</p>	<p><input checked="" type="checkbox"/> Core</p> <p><input checked="" type="checkbox"/> Widget</p> <p><input checked="" type="checkbox"/> Mouse</p> <p><input checked="" type="checkbox"/> Position</p>	<p>The core of jQuery UI, required for all interactions and widgets.</p> <p>Provides a factory for creating stateful widgets with a common API.</p> <p>Abstracts mouse-based interactions to assist in creating certain widgets.</p> <p>Positions elements relative to other elements.</p>
<p>Interactions</p> <p><input checked="" type="checkbox"/> Toggle All</p> <p>These add basic behaviors to any element and are used by many components below.</p>	<p><input checked="" type="checkbox"/> Draggable</p> <p><input checked="" type="checkbox"/> Droppable</p> <p><input checked="" type="checkbox"/> Resizable</p> <p><input checked="" type="checkbox"/> Selectable</p> <p><input checked="" type="checkbox"/> Sortable</p>	<p>Enables dragging functionality for any element.</p> <p>Enables drop targets for draggable elements.</p> <p>Enables resize functionality for any element.</p> <p>Allows groups of elements to be selected with the mouse.</p> <p>Enables items in a list to be sorted using the mouse.</p>

Componentes:

Aquí podemos seleccionar/deseleccionar los checkbox de cada uno de los componentes que forman parte de jQuery UI. Algunos tienen dependencias con otros, por lo que observaremos que, al seleccionar algunos componentes, automáticamente se seleccionan sus dependencias.

Aparte de los componentes a descargar, tenemos un par de cosas adicionales que marcar antes de la descarga.

Temas (Themes):

Por una parte tenemos que seleccionar el tema o aspecto que van a tener las interfaces de usuario de jQuery UI. Tenemos decenas de temas ya creados por el propio equipo de las librerías, con aspectos que varían bastante, sobretodo cromáticamente. Además, existe un generador de temas, que podemos utilizar para personalizar aun más el aspecto de las interfaces de usuario. El objetivo es que los menús y demás de este sistema tengan un aspecto que se integre bien con el diseño de nuestra página web.

Versión:

También tendremos que escoger la versión de jQueryUI que queremos utilizar, sabiendo que cada versión está preparada para funcionar con una versión del framework.

Por último podemos apretar el botón de download para descargar un zip con todo lo que necesitaremos para usar jQueryUI.

Una vez descargado nuestro paquete de jQuery UI obtendremos un archivo

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar

comprimido con diversos directorios y archivos, que tiene las siguientes carpetas principales:

Carpeta "css": En esta carpeta se encuentra el CSS y las imágenes para generar el tema o los temas escogidos. Realmente no tenemos por qué tocar esta carpeta en principio para nada, pero contiene cosas que serán fundamentales para que todo se muestre como deseamos.

Carpeta "development-bundle": Esta carpeta contiene una serie de materiales útiles para los desarrolladores que van a utilizar estas librerías. Veremos aquí páginas de documentación, ejemplos de uso y otras cosas interesantes. Nada de lo que hay aquí es necesario en principio para hacer funcionar los componentes jQuery UI, pero son materiales que podrán venirnos bien para aprender.

Carpeta "js": Aquí veremos los scripts Javascript de jQuery y jQuery UI necesarios para que todos los componentes funcionen. Esta carpeta contiene el archivo Javascript con el código de los componentes que habíamos seleccionado al hacer la descarga, además del archivo Javascript con el código de la versión del framework que funciona bien con las librerías.

Con el zip de descarga podremos ya comenzar a implementar alguno de los componentes de jQueryUI. Junto con estas carpetas viene un archivo index.html que tiene como ejemplo algunos de los widgets proporcionados.

Ahora pasaremos a la práctica, mostrando cómo hacer una página web que utilice uno de los componentes disponibles en jQuery UI, que es el Datepicker o seleccionador de fecha por medio de un calendario.

EJEMPLO 1: DATEPICKER

En este primer ejemplo vamos a ver como trabajar con el componente Datepicker y vamos a ver cuales son algunas de las alternativas que tenemos para paramentizarlo. Dentro del archivo ejemplo1.html tenemos distintos ejemplos de su uso vamos a ir viendo y explicando cada uno de ellos.

Caso 1:

El primer caso es el del Datepicker inline, es decir el calendario aparece visible en la página, para ello simplemente definimos un div con un id asociado

```
<div id="datepicker"></div>
```

y luego con el selector de JQuery lo seleccionamos por su id y le asociamos el componente Datepicker a dicho div de la siguiente manera:

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar

```
$( "#datepicker" ).datepicker({  
    inline: true  
});
```

Caso 2:

En el segundo caso podemos ver el comportamiento por defecto, que consiste en asociarle un id a un campo de input y el calendario será displayado una vez que posicionamos el cursor dentro de ese campo de input. De esta forma lo que tenemos en nuestro <body> es una etiqueta de este tipo:

```
<div>Date: <input type="text" id="datepicker2" /></div>
```

Y luego desde JQuery seleccionamos el input por su id y le asociamos el component Datepicker, que ya no aparecerá visible, sino que se mostrará una vez que posicionamos el cursor dentro del campo de input.

```
$( "#datepicker2" ).datepicker({  
});
```

En este caso no estamos estableciendo ningún parámetro sino que estamos utilizando los valores por defecto.

A partir de ahora lo que haremos es ir mostrando distintas opciones a través de ir modificando las opciones o métodos del componente Datepicker.

Caso 3:

En el ejemplo 3 tenemos un componente Datepicker mostrando el menú de año y mes. Para ello utilizamos las siguientes opciones:

```
changeMonth: true,  
changeYear: true
```

En el ejemplo 4 mostramos 3 meses en lugar de 1 y le agregamos un panel con dos botones para cancelar el display del calendario con las siguientes opciones:

```
numberOfMonths: 3,  
showButtonPanel: true
```

En el cuarto ejemplo estamos mostrando la fecha con otro formato distinto al formato por defecto que es mm/dd/yy. En este caso estamos utilizando "dd/mm/yy" Podríamos haber seleccionado: Y hubiera mostrado:

ISO 8601 yy-mm-dd	2013-10-16
d MM, y	16 October, 13

dd-mm-yy

16-10-2013

Por citar algunos ejemplos. El formato de la fecha puede ser una combinación de los siguientes:

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

Caso 4:

El siguiente ejemplo es para restringir un rango de fechas. Para ello utilizamos las opciones `minDate` y `maxDate` de la siguiente manera:

`minDate: -20,`
`maxDate: "+1M +10D"`

lo que significa fecha desde: 20 días antes de la fecha de hoy y fecha hasta: hasta 1 mes y 10 días desde la fecha de hoy. Esas serán las fechas habilitadas en nuestro calendario.

Tanto `minDate` como `maxDate` tienen un valor por defecto = null y aceptan tres tipos de formato para pasarlo como parámetro:

Formato Fecha: una fecha.

Formato Numero: Un número de días desde hoy. Por ejemplo 2 representa 2 días a partir de hoy y -1 significa ayer.

Formato String: Un string en el formato definido anteriormente o una fecha relativa. La fecha relativa debe contener períodos válidos, estos son "y" para años, "m" para meses, "w" para semanas y "d" para días. Por ejemplo "+1m +7d" significa 1 mes y 7

días mas a partir de hoy.

Caso 6:

Seleccionar un período de tiempo desde una fecha hasta otra. Para ello utilizamos el siguiente código:

```
$( "#from" ).datepicker({
    defaultDate: "+1w",
    changeMonth: true,
    numberOfMonths: 3,
    onClose: function( selectedDate ) {
        $( "#to" ).datepicker( "option", "minDate", selectedDate );
    }
});

$( "#to" ).datepicker({
    defaultDate: "+1w",
    changeMonth: true,
    numberOfMonths: 3,
    onClose: function( selectedDate ) {
        $( "#from" ).datepicker( "option", "maxDate", selectedDate );
    }
});
```

La opción `defaultDate` hace que se muestre como default 1 semana a partir del día de la fecha.

La opción `changeMonth` permite seleccionar el mes desde el cual mostrar en el primer calendario mostrado.

La opción `numberOfMonths: 3` muestra 3 calendarios.

Con el método `onClose` establecemos la fecha seleccionada y si es en el caso del `#from` establecemos como fecha mínima en el `#to` la fecha seleccionada en el `#from`, es decir que la fecha hasta no va a ser menos que la fecha desde.

Esta es una lista de las “Options” que podemos modificar en el componente DatePicker:

Options

[altField](#)

[altFormat](#)

[appendText](#)

[autoSize](#)

[beforeShow](#)

[beforeShowDay](#)

[buttonImage](#)

[buttonImageOnly](#)

[buttonText](#)

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar



[calculateWeek](#)
[changeMonth](#)
[changeYear](#)
[closeText](#)
[constrainInput](#)
[currentText](#)
[dateFormat](#)
[dayNames](#)
[dayNamesMin](#)
[dayNamesShort](#)
[defaultDate](#)
[duration](#)
[firstDay](#)
[gotoCurrent](#)
[hideIfNoPrevNext](#)
[isRTL](#)
[maxDate](#)
[minDate](#)
[monthNames](#)
[monthNamesShort](#)
[navigationAsDateFormat](#)
[nextText](#)
[numberOfMonths](#)
[onChangeMonthYear](#)
[onClose](#)
[onSelect](#)
[prevText](#)
[selectOtherMonths](#)
[shortYearCutoff](#)
[showAnim](#)
[showButtonPanel](#)
[showCurrentAtPos](#)
[showMonthAfterYear](#)
[showOn](#)
[showOptions](#)
[showOtherMonths](#)
[showWeek](#)
[stepMonths](#)
[weekHeader](#)
[yearRange](#)
[yearSuffix](#)

Y esta es la lista de métodos que podemos utilizar con el componente:

Methods

[destroy](#)
[dialog](#)

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar

[getDate](#)
[hide](#)
[isDisabled](#)
[option](#)
[refresh](#)
[setDate](#)
[show](#)
[widget](#)

Para modificar el idioma el idioma de nuestro calendario debemos agregar el siguiente script:

```
<script type="text/javascript" src="jquery.ui.datepicker-es.js"></script>
```

Este script se encuentra comentado en el ejemplo1.html descoméntenlo y verán como les cambia a castellano el idioma de todos los datepickers incluidos.

OTRO EJEMPLO CON DATEPICKER

Vamos a ver un ejemplo en el cual utilizamos una función Javascript para deshabilitar fechas en un calendario. El ejemplo completo lo encuentran en la carpeta inhabilitarFechasCalendario adjuntada a los ejemplos.

Para poder inhabilitar fechas en un calendario hacemos uso de la opción `beforeShowDay` con esta opción indicamos que antes de mostrar el día ejecute la función `DisableDays` en la cual inhabilitaremos las fechas que nos interesan.

```
$( "#datepicker" ).datepicker({  
    inline: true,  
    numberOfMonths: 3,  
    beforeShowDay: DisableDays  
});
```

Ahora lo que tenemos que hacer es desarrollar la función `DisableDays` que será la encargada de inhabilitar o no los días.

La función está preparada para recibir un array con las fechas a inhabilitar o el rango de fechas a inhabilitar, entonces cada posición del vector tendrá el siguiente formato: "mm/dd/yyyy" o "mm/dd/yyyy, mm/dd/yyyy" y ésta inhabilitará el día o el rango de fechas indicado.



EJEMPLO 2: ACCORDION

En este ejemplo veremos el clásico acordeón y algunas alternativas trabajando con las options y los métodos que nos brinda. Para más ejemplos de uso visiten <http://jqueryui.com/accordion/>

Las opciones, métodos y eventos disponibles son los siguientes:

Options

[active](#)
[animate](#)
[collapsible](#)
[disabled](#)
[event](#)
[header](#)
[heightStyle](#)
[icons](#)

Methods

[destroy](#)
[disable](#)
[enable](#)
[option](#)
[refresh](#)
[widget](#)

Events

[activate](#)
[beforeActivate](#)
[create](#)

Como les decía veremos algunas alternativas si quieren seguir investigando pueden leer la documentación de la API.

Caso 1: Funcionamiento por defecto

Como se puede observar en el código lo que tenemos que hacer es definir una etiqueta div a la cual le asignamos un id, en este caso lo hemos llamado accordion y luego tomamos desde JQuery ese elemento div por su id y le aplicamos el componente accordion de la siguiente manera:

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

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar



Aquí vemos el código html que explicáramos mas arriba. Como se ve también el accordion está formado por dos elementos html fundamentales un <h3> y un <div>. En el <h3> se coloca el título que será visible en el acordeón y dentro del <div> a continuación va el texto que será mostrado una vez seleccionado el título del acordeón. Como se puede ver dentro del <div> pueden embeberse otras etiquetas html como pueden ser <p> o en nuestro ejemplo. Podrían agregar otras si fuera necesario.

```
<div id="accordion">
  <h3>Section 1</h3>
  <div>
    <p>Mauris mauris ante, blandit et, ultrices a, suscipit eget, quam.
Integer.....
    </p>
  </div>
  <h3>Section 2</h3>
  <div>
    <p>Sed non urna. Donec et ante. Phasellus eu ligula. Vestibulum sit
amet.....
    </p>
  </div>
  <h3>Section 3</h3>
  <div>
    <p>Nam enim risus, molestie et, porta ac, aliquam ac, risus. Quisque
lobortis.....
    </p>
    <ul>
      <li>List item one</li>
      <li>List item two</li>
      <li>List item three</li>
    </ul>
  </div>
  <h3>Section 4</h3>
  <div>
    <p>
      Cras dictum. Pellentesque habitant morbi tristique senectus...
    </p>
    <p>Suspendisse eu nisl. Nullam ut libero. Integer dignissim consequat
lectus.....
    </p>
  </div>
</div>
```

Caso 2: Resizable

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar

El caso que sigue presenta la información en un espacio limitado dentro del panel de contenido. Por defecto el acordeón adapta su ancho al espacio disponible. Para llenar el espacio vertical asignado por su contenedor, hay que configurar la opción `heightStyle` en `fill`, "llenar", y el texto se ajusta automáticamente al alto de su contenedor principal. El código queda de la siguiente manera:

```
$( "#accordion2" ).accordion({
    heightStyle: "fill"
});
```

Luego se establecen las medidas de mínimas de ancho y alto que tiene que tener el contenedor de la siguiente manera:

```
$(function() {
    $( "#accordion-resizer" ).resizable({
        minHeight: 140,
        minWidth: 200,
        resize: function() {
            $( "#accordion" ).accordion( "refresh" );
        }
    });
});
```

Lo que tenemos que hacer en el `<body>` de nuestro html es colocar el `<div id="accordion2">` dentro del `<div id="accordion-resizer">` de la siguiente manera:

```
<div id="accordion-resizer">
    <div id="accordion2">
        .
        .
        .
        .
    </div>
</div>
```

Y establecer en el CSS las medidas correspondientes al `div id="accordion-resizer"`, luego el acordeón se adaptará a estas medidas.

Caso 3: Sortable (permite el ordenamiento de los títulos)

El código correspondiente es el siguiente:

```
$(function() {
    $( "#accordion" )
    .accordion({
```

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar

```
header: "> div > h3"
})
.sortable({
  axis: "y",
  handle: "h3",
  stop: function( event, ui ) {
    // IE doesn't register the blur when sorting
    // so trigger focusout handlers to remove .ui-state-focus
    ui.item.children( "h3" ).triggerHandler( "focusout" );
  }
});
});
```

En este caso estamos seteando el componente acordeón estableciendo el header como <h3> hijo de <div>. En CSS el signo > es utilizado para designar hijos directos de un padre, así si tenemos

```
<div>
  <h3 id="subtitulo1"></h3>
  <p>
    <h3 id="subtitulo2"></h3>
  </p>
</div>
```

Y definimos una regla de CSS

```
div > h3 {
  background-color: grey;
}
```

Este atributo solo afectará al primer <h3> ya que es hijo directo del div mientras que el segundo h3 es hijo del <p> que lo contiene.

Lo que sigue a continuación es la interacción 'sortable' también perteneciente a JQueryUI y que explicaremos brevemente, es decir que en este ejemplo estamos aplicando una interacción a un widget.

Las propiedades que estamos seteando son:

Axis: que determina si el objeto va a desplazarse en forma horizontal o vertical, en este caso lo hace en forma vertical.

Handle: establecemos el elemento sobre el cual al hacer click se deberá ordenar.

Y el evento stop que se activa cuando se detiene el desplazamiento del elemento a reubicar.

EJEMPLO 3: TABS

Este componente de JQuery UI con el que podemos implementar una navegación mediante pestañas es muy fácil de insertar en nuestras páginas web. Para ello:

- Dentro de un 'DIV' que actúa como contenedor creamos una lista con enlaces a partir de los cuales se crearán las pestañas, junto con otros 'DIV' con los paneles que asociaremos a ellas.
- Para indicar qué panel o 'DIV' debe mostrarse cuando hagamos click en una pestaña, escribimos como valor de su atributo href el 'id' de uno de los anteriores 'DIV' precedido de almohadilla (#).
- Al cargarse la página creamos el componente llamando al método tabs().

Algunas de sus propiedades mas utilizadas son:

- active: activar un determinado panel.
- disabled: desactivar pestañas.

Algunos de sus métodos más útiles:

- enable() / disable(): activar/desactivar las pestañas.
- destroy(): elimina el componente.

Les dejo la lista completa de propiedades, métodos y eventos disponibles para este componente, de todos ellos pueden encontrar explicación y ejemplos de uso en la documentación de la API.

Options

[active](#)
[collapsible](#)
[disabled](#)
[event](#)
[heightStyle](#)
[hide](#)
[show](#)

Methods

[destroy](#)
[disable](#)
[enable](#)
[load](#)
[option](#)
[refresh](#)
[widget](#)

Events

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar

[activate](#)
[beforeActivate](#)
[beforeLoad](#)
[create](#)
[load](#)

En el ejemplo3.html vemos varios casos de utilización de este componente. El mas sencillo de todos es el **Caso 1** en el cual solo aplicamos el componente tabs() al <div id="tabs">.

El **Caso 2** es similar al anterior con la única diferencia que establecemos que funcione ante el evento mouseover a través del siguiente código:

```
$(function() {  
    $('#tabs2').tabs({  
        event: "mouseover"  
    });  
});
```

En el **Caso 3** ponemos en práctica uno de los eventos definidos dentro del componente que es beforeActivate, el mismo es ejecutado previamente a la activación de la pestaña.

En el ejemplo lo que hacemos es evaluar cual es la pestaña activa esto lo hacemos a través de la línea:

```
var tabactual=$(this).tabs('option', 'active');
```

Y luego si la pestaña es la número 1 (las tabs se enumeran de 0 a n al igual que los arrays) y no fue chequeado el checkbox destinado a aceptar los términos mostramos el mensaje de alerta y retornamos false, eso hará que no me permita seleccionar las pestaña número 3.

```
if (tabactual==1 && !$('#acepto').prop('checked')) {  
    alert("Debe aceptar los terminos");  
    return false;  
}
```

El resto de los eventos también funcionan de manera similar a este ejemplo que hemos visto, es cuestión de ver, según lo que queremos lograr, cual es el evento que mas nos conviene utilizar y de que manera debemos programarlo.

Los demás eventos son:

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar



activate: se ejecuta una vez que la pestaña se activa

beforeLoad: se ejecuta antes de cargar el contenido remoto de una pestaña después de beforeActive

create: se ejecuta cuando las pestañas son creadas.

load: luego de que una pestaña remota fue cargada

Más adelante veremos como cargar la información de una pestaña en forma remota, es decir acudiendo al servidor para obtener su contenido, pero para eso debemos ver el concepto de Ajax primero.

EJEMPLO 4: AUTOCOMPLETE

El componente autocomplete brinda sugerencias mientras el usuario está tipeando en un campo de input.

La lista de propiedades, métodos y eventos que se encuentra a continuación la encuentran en la documentación de la API:

Options

[appendTo](#)

[autoFocus](#)

[delay](#)

[disabled](#)

[minLength](#)

[position](#)

[source](#)

Methods

[close](#)

[destroy](#)

[disable](#)

[enable](#)

[option](#)

[search](#)

[widget](#)

Events

[change](#)

[close](#)

[create](#)

[focus](#)

[open](#)

[response](#)

[search](#)

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar

select

En el ejemplo4.html que es el funcionamiento por defecto del componente, las sugerencias son lenguajes de programación, los mismos son cargados en un array simple de Javascript y proporcionado al componente a través de la propiedad source. En cuanto al html correspondiente lo único que tenemos que definir es un campo de input con un id asociado, en este caso tenemos `<input id="tags" />` que será al que le asociemos el componente autocomplete.

En el ejemplo4b.html tenemos el mismo array Javascript con lenguajes de programación pero en este caso se permite la selección múltiple de valores en un mismo campo. Una vez seleccionado un campo de la lista de sugerencias comienza el autocompletado para el segundo valor ingresado y así sucesivamente. Este es un ejemplo que muestra cómo utilizar la propiedad source junto con algunos eventos para permitir autocompletar varios valores en un solo campo.

Veamos un poco mas en detalle el código

```
$( "#tags" )
    // don't navigate away from the field on tab when selecting an item
    .bind( "keydown", function( event ) {
        if ( event.keyCode === $.ui.keyCode.TAB &&
            $( this ).data( "ui-autocomplete" ).menu.active ) {
            event.preventDefault();
        }
    })
    .autocomplete({
        minLength: 0,
        source: function( request, response ) {
            // delegate back to autocomplete, but extract the last term
            response( $.ui.autocomplete.filter(
                availableTags, extractLast( request.term ) ) );
        },
        focus: function() {
            // prevent value inserted on focus
            return false;
        },
        select: function( event, ui ) {
            var terms = split( this.value );
            // remove the current input
            terms.pop();
            // add the selected item
            terms.push( ui.item.value );
            // add placeholder to get the comma-and-space at the end
            terms.push( "" );
            this.value = terms.join( ", " );
        }
    });
```

```
        return false;  
    }  
});
```

El evento select ocurre en el momento que la sugerencia es seleccionada, allí lo que hacemos es tomar el dato del input actual y agregarle la coma y el espacio.

La propiedad source puede recibir un array, un string o una función, este es el caso del ejemplo. La función recibe un objeto de la petición, con un único término propiedad, que se refiere al valor actual de la entrada de texto y una respuesta, que espera un solo argumento: los datos para sugerir al usuario.

El evento focus ocurre cuando un nos movemos a un ítem de la lista, la acción por defecto del evento es reemplazar el valor del input por el ítem de la lista seleccionado, en este caso lo que estamos haciendo es impedir ese comportamiento por defecto.

Hasta aquí los ejemplos correspondientes al componente autocomplete. Pueden encontrar mas ejemplos en la sección Demos de JQueryui: <http://jqueryui.com/autocomplete/#default>.

En la unidad siguiente veremos como obtener los datos en forma remota en lugar de tomarlos de un array como en estos dos ejemplos, para eso debemos hacer una introducción al formato JSON.

EJEMPLO 5: DIALOG

Mediante este widget tenemos la posibilidad de crear muy fácilmente ventanas dentro de una página web. Sus propiedades principales son:

- **buttons:** permite indicar los botones que se mostrarán en la ventana así como las acciones que realizarán cuando se haga click en ellos.
- **height / width:** alto y ancho de la ventana en píxels.
- **maxHeight / maxWidth:** alto y ancho máximos que podrá tener al ser redimensionada.
- **minHeight / minWidth:** alto y ancho mínimos que podrá tener si es redimensioada.
- **modal:** permite crear una ventana modal (no se podrá interactuar con otros elementos de la página web hasta que sea cerrada).
- **position:** posición de la ventana ('center', 'left', 'right', 'top', 'bottom').
- **resizable:** cuando su valor es 'true' podrá ser redimensionada por el usuario.
- **show / hide:** mostrar y ocultar la ventana.
- **title:** permite indicar un título.

Entre otros dispone también de los siguientes métodos:

- `open()` / `close()`: abrir y cerrar la ventana.
- `isOpen()`: devuelve 'true' si la ventana se encuentra abierta.
- `destroy()`: elimina la ventana.

Veremos dos ejemplos sencillos, el funcionamiento por defecto, que consiste en una ventana que se muestra una vez cargada la página y un ejemplo con ventanas modales (no permiten interactuar con otros elementos de la página hasta ser cerradas).

Dichos ejemplos se encuentran en los scripts `ejemplo5.html` y `ejemplo5b.html`

Pueden consultar la lista completa de propiedades, métodos y eventos en la documentación de la API:

Options

[appendTo](#)
[autoOpen](#)
[buttons](#)
[closeOnEscape](#)
[closeText](#)
[dialogClass](#)
[draggable](#)
[height](#)
[hide](#)
[maxHeight](#)
[maxWidth](#)
[minHeight](#)
[minWidth](#)
[modal](#)
[position](#)
[resizable](#)
[show](#)
[title](#)
[width](#)

Methods

[close](#)
[destroy](#)
[isOpen](#)
[moveToTop](#)
[open](#)
[option](#)
[widget](#)

Events

[beforeClose](#)

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar



[close](#)
[create](#)
[drag](#)
[dragStart](#)
[dragStop](#)
[focus](#)
[open](#)
[resize](#)
[resizeStart](#)
[resizeStop](#)

Pueden ver mas ejemplos en la sección Demos de JQueryUI:
<http://jqueryui.com/dialog/#default>.

EJEMPLO 6: SLIDER

El componente Slider de JQuery UI permite realizar la selección de un valor numérico. Sus principales propiedades son:

- disabled: desactiva el componente.
- max: valor máximo admitido.
- min: valor mínimo permitido.
- orientation: configura la orientación del componente ('vertical' u 'horizontal').
- step: configura el incremento y decremento a aplicar cuando se utiliza el slider.
- value: permite establecer y obtener el valor actual.

En cuanto a sus métodos, entre otros disponemos de:

- destroy(): elimina el componente.
- enable() / disable(): activar/desactivar el slider.
- value(): devuelve o establece el valor actual.

Asimismo nos serán útiles los siguientes eventos:

- change: se produce tras soltar el botón y antes de producirse el evento 'stop'.
- slide: cuando se mueve el deslizador sin soltar el botón.
- stop: se lanza después de soltar el botón y antes de producirse el evento 'change'.

El ejemplo6.html nos ayudará a comprender mejor su funcionamiento. Ejecútenlo en el navegador y verán como se va actualizando en el log (<div id="log">) los tres eventos que están siendo ejecutados, que hemos descripto anteriormente y como se



van actualizando los valores en cada uno de los campos de input #mostrar y #mover.

Como verán el objetivo fundamental del slider es la selección de un rango de valores que puede ser utilizado para acotar una búsqueda en una tabla según el rango de valores por ejemplo.

La lista completa de propiedades, métodos y eventos que listamos a continuación la pueden encontrar en la documentación de la API:

Options

[animate](#)

[disabled](#)

[max](#)

[min](#)

[orientation](#)

[range](#)

[step](#)

[value](#)

[values](#)

Methods

[destroy](#)

[disable](#)

[enable](#)

[option](#)

[value](#)

[values](#)

[widget](#)

Events

[change](#)

[create](#)

[slide](#)

[start](#)

[stop](#)

Si quieren ver mas ejemplos de su uso pueden consultar la sección de Demos de JQueryUI: <http://jqueryui.com/slider/#default>.

EJEMPLO 7: MENU

Este widget permite insertar un menú en nuestra página web. Algunas de sus propiedades son:

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar

- **disabled:** si su valor es 'true' se desactivará el menú.
- **icons:** permite definir iconos a mostrar en los submenús.
- **position:** configura la posición de los submenús con respecto a su menú padre.

En cuanto a sus métodos, disponemos entre otros de:

- **collapse():** cerrar el submenú que se encuentre activo.
- **destroy():** elimina el menú.
- **enable() / disable():** activar/desactivar el menú.
- **expand():** abre el submenú asociado al elemento que se encuentre activo.
- **focus():** activar un determinado elemento de menú.

Veamos un ejemplo cuyo código pueden encontrar en `ejemplo7.html` el cual explicaremos a continuación:

- Creamos un 'DIV' con una lista para los elementos principales del menú, y otras sublistas para los diferentes niveles.
- Definimos una clase CSS `ui-menu` para definir del ancho tanto de los menús como de los submenús.
- Fíjate también en que en la lista usamos la clase CSS `ui-state-disabled` para desactivar elementos del menú.
- Al cargarse la página se llama al método `menu()` para mostrarlo.

Pueden consultar la lista completa de propiedades métodos y eventos que enumeramos a continuación en la documentación de la API: <http://api.jqueryui.com/menu/>

Options

[disabled](#)

[icons](#)

[menus](#)

[position](#)

[role](#)

Methods

[blur](#)

[collapse](#)

[collapseAll](#)

[destroy](#)

[disable](#)

[enable](#)

[expand](#)

[focus](#)

[isFirstItem](#)

[isLastItem](#)

[next](#)

[nextPage](#)

[option](#)

[previous](#)

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar



[previousPage](#)

[refresh](#)

[select](#)

[widget](#)

Events

[blur](#)

[create](#)

[focus](#)

[select](#)