

Tarea JQuery

Nivel Básico

Objetivo: Crear una **tabla** de HTML **editable**

Conocimientos previos: Tablas de HTML

Las tablas permiten mostrar datos en forma tabular:

Código

```
<table>
  <tr>
    <th>Nombre</th>
    <th>Domicilio</th>
  </tr>
  <tr>
    <td>Juanito Perez</td>
    <td>Montevideo DF</td>
  </tr>
  <tr>
    <td>Jose Arcadio</td>
    <td>Macondo</td>
  </tr>
</table>
```

Muestra en Pantalla

Nombre	Domicilio
Juanito Perez	Montevideo DF
Jose Arcadio	Macondo

- **<table>** Indica que comienza la **tabla**
- **<tr>** Indica que comienza un **renglón**
- **<td>** Define las **celdas** de cada renglón, un renglón tiene tantos **<td>** como columnas tenga la tabla
- **<th>** Similar a **<td>**, excepto que se usa para definir las celdas del **título de la tabla**

IMPORTANTE: Atributo **contentEditable="true"**

Este atributo es parte de **HTML5** y permite que al aplicarlo a cualquier caja (div, span, p, table etc.) automáticamente el contenido de esta se puede **editar**. Puede comprobar esto al abrir el archivo **tarea_basico.html** y note que en las tablas se puede editar el contenido automáticamente al darle click y escribir texto.

Instrucciones:

1. Abra el archivo tarea_basico.html
2. Por medio de jQuery escuche el evento click en el botón **#boton-agregar**
3. Cuando ocurra el click, usando jQuery obtenga la tabla y **agregue un nuevo renglón** con dos columnas por medio del siguiente HTML:

```
<tr>
  <td contentEditable='true'>Nombre</td>
  <td contentEditable='true'>dirección</td>
</tr>
```

Observación: No cambie el HTML de arriba.

Sugerencia para el punto 3: Para realizar inserción de código html, al final de la etiqueta table, use el método append.

Note que al pasar el ratón encima de alguna de las celdas, estas cambian de color por medio del pseudo selector **:hover** ¿Recuerda este selector de CSS de la segunda tarea? Note que si da click en la celda, puede escribir en esta, es decir el contenido es editable!!

Nota histórica: Anteriormente hacer contenido editable requería hacer uso de técnicas de manipulación del DOM. En HTML5 gracias a la propiedad **contentEditable** esto ya no es necesario.

Nivel Intermedio

Objetivo: Crear una **tabla** de HTML **editable** y aprender a usar los métodos **parents**, **remove** y **on** de jQuery.

Conocimientos previos: Tablas de HTML

Instrucciones:

1. Abra el archivo tarea_intermedio.html
2. Use el siguiente código para escuchar el evento **click** de las cajas que tengan la **clase “.agregar”** (la imagen con el icono verde):

```
$("#body").on("click", ".agregar", function(evento) {
    //CUERPO DE LA FUNCION
});
```

Observación: La estructura anterior es equivalente al **click**, **excepto** que esta estructura **se usa** cuando se quiere escuchar eventos en cajas que se agregan dinámicamente.

3. Agregue en el cuerpo de la función del paso 3 usando jQuery la lógica necesaria para **agregar el siguiente renglón en la tabla**:

```
<tr>
  <td contentEditable="true">Juanito Perez</td>
  <td contentEditable="true">Montevideo Distrito Federal</td>
  <td>
    
    
  </td>
</tr>
```

Sugerencia: Recuerde el método de inserción **append** de jQuery. Escape las dobles cadenas con \" o use comillas sencillas ‘

4. Use el siguiente código para escuchar el evento **click** de las cajas que tengan la **clase “.remover”** (la imagen con el icono rojo):

```
$("body").on("click", ".remover", function(evento) {
    //CUERPO DE LA FUNCION
});
```

5. Obtenga por medio de la propiedad target del evento, el objeto puro del DOM y conviértalo a un objeto de jQuery

```
var botoncito = $(evento.target);
```

6. Del objeto jQuery del paso anterior invoque la siguiente línea:

```
objetojQueryPaso5.parents("tr").remove();
```

Observaciones: El método **parents** obtiene el padre de la caja actual que coincida con el selector de CSS indicado, en este caso, busca por el padre cuya etiqueta sea <tr>

El método **remove** al invocarlo en un objeto de jQuery destruye a la caja HTML asociada a dicho objeto.

Sugerencia para el punto 3: Recuerde el uso de los métodos de inserción en el DOM, ¿Cuál de estos métodos le permite insertar contenido **dentro** de una caja y al **final**?

Note que al pasar el ratón encima de alguna de las celdas, estas cambian de color por medio del pseudo selector **:hover** ¿Recuerda este selector de CSS de la segunda tarea?. Note que si da click en la celda, puede escribir en esta, es decir el contenido es editable!!

Nota histórica: Anteriormente hacer contenido editable requería hacer uso de técnicas de manipulación del DOM. En HTML5 gracias a la propiedad **contentEditable** esto ya no es necesario.

Nivel Avanzado

Objetivo: Crear una **tabla** de HTML **editable** y aprender a usar los métodos **parents**, **remove**, **on** y **siblings** de jQuery y tener una introducción a **jQuery UI**
Conocimientos previos: Tablas de HTML

Instrucciones (Igual que intermedio hasta el paso 6)

1. Abra el archivo `tarea_avanzado.html`
2. Use el siguiente código para escuchar el evento **click** de las cajas que tengan la **clase** **“agregar”** (la imagen con el icono verde):

```
$("#body").on("click", ".agregar", function(evento) {  
    //CUERPO DE LA FUNCION  
});
```

Observación: La estructura anterior es equivalente al **click**, **excepto** que esta estructura **se usa** cuando se quiere escuchar eventos en cajas que se agregan dinámicamente (más de esto en la siguiente clase).

3. Agregue en el cuerpo de la función del paso 3 usando jQuery la lógica necesaria para **agregar el siguiente renglón en la tabla:**

```
<tr>
```

```

<td contentEditable="true">Juanito Perez</td>
<td contentEditable="true">Montevideo Distrito Federal</td>
<td>
    
    
    
</td>
</tr>

```

Sugerencia: Recuerde los métodos de **inserción** de jQuery. Escape las dobles cadenas con \"

- Use el siguiente código para escuchar el evento **click** de las cajas que tengan la **clase** **“.remover”** (la imagen con el icono rojo):

```

$("body").on("click", ".remover", function(evento) {
    //CUERPO DE LA FUNCION
});

```

- En el cuerpo de la función del paso 4, obtenga por medio de la propiedad **target** del evento, el objeto puro del DOM y conviértalo a un objeto de jQuery
- Del objeto jQuery del paso anterior invoque la siguiente línea:

```
objetojQueryPaso5.parents("tr").remove();
```

Observaciones: El método **parents** obtiene el padre de la caja actual que coincida con el selector de CSS indicado, en este caso, busca por el padre cuya etiqueta sea **<tr>**

El método **remove** al invocarlo en un objeto de jQuery destruye a la caja HTML asociada a dicho objeto.

- Escuche el evento click en el icono con la clase **“.ver-mapa”** con el siguiente código:

```

$("body").on("click", ".ver-mapa", function(evento) {
    //CUERPO DE LA FUNCION
});

```

8. En el cuerpo de la función del paso 7, obtenga de la propiedad target el objeto puro del DOM y conviértalo a un objeto de jQuery.
9. Use el objeto jQuery del paso anterior para obtener el valor de la **Dirección** de un renglón en particular con la siguiente línea(guarde el resultado en una variable):

```
objetojQueryPaso9.parents("td").siblings(".direccion").html();
```

Observaciones: Con el método **parents** obtenemos el padre del icono “.ver-mapa” que sea una celda (td)

El método **siblings** permite obtener las etiquetas que estén al mismo nivel (hermanos) que cumplan con cierta regla de CSS. En este caso se están obteniendo todos los hermanos de la celda (las demás etiquetas <td> contendidas en el renglon) cuyo clase sea “.direccion”

10. A continuación, por medio de jQuery agregue **dentro** de la caja **#contenedor-mapa** el siguiente html (recuerde escapar las comillas dobles o usar comillas simples):

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

11. Agregue el siguiente código para crear una ventana emergente por medio de jQuery UI:

```
$("#contenedor-mapa").dialog({  
    modal : true,  
    width : "auto",  
    close : function() {  
        $("#mapa").remove();  
    }  
});
```

Observaciones: **jQuery UI** <http://jqueryui.com/> es un framework basado en jQuery que permite construir componentes reutilizables, en este caso una ventana, la cual se creara con el contenido de la caja **#contenedor-mapa**

jQuery UI ya está importado en su tarea, busque los comentarios C1 y C2 en el archivo de la tarea.

En el código anterior la propiedad **modal:true** indica que cuando se abra la ventana todo el contenido detrás de la ventana deberá bloquearse.

La propiedad **width:"auto"** ajusta automáticamente el ancho de la ventana a la caja

La propiedad **close** permite indicar un **callback** que se ejecutará al cerrar la ventana. El callback en este caso destruye al mapa.

12. A continuación del paso 11, obtenga la caja #mapa y en esta cree un mapa usando **goMap**, haga que dicho mapa tenga la dirección que obtuvo en el paso 9 con un **zoom** de 19.

Notas Finales

Observe que al pasar el ratón encima de alguna de las celdas, estas cambian de color por medio del pseudo selector **:hover** ¿Recuerda este selector de CSS de la segunda tarea? Note que si da click en la celda, puede escribir en esta, es decir el contenido es editable!!

Anteriormente hacer contenido editable requería hacer uso de técnicas de manipulación del DOM. En HTML5 gracias a la propiedad **contentEditable** esto ya no es necesario.