



jQuery Mobile

Víctor Custodio

Jquery Mobile

“jQuery Mobile es un framework de interfaz gráfica especialmente diseñado para el desarrollo de aplicaciones web para móviles que pretende unificar el diseño de interfaces de usuario para la mayoría de dispositivos móviles del mercado”

- Basado en jQuery
- Framework ligero
- Conseguir una misma sensación de navegación por parte del usuario en la mayoría de los dispositivos móviles

Características de jQuery Mobile

- Basado en el núcleo de jQuery
- Altamente compatible con la gran mayoría de dispositivos
- Tamaño reducido (unos 20kb)
- HTML5
- Mejora progresiva
- Accesibilidad
- Soporte de eventos de ratón y de contacto
- Fácilmente modificable

¿Cómo funciona jQuery Mobile?

1. Páginas en HTML (con pequeñas modificaciones)
2. Transformaciones del DOM
3. Páginas web adaptadas a dispositivos móviles

Incluir JQuery Mobile en nuestras páginas

- `<head>`

```
<!-- Include meta tag to ensure proper rendering and touch zooming -->
<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Include jQuery Mobile stylesheets -->
<link rel="stylesheet" href="https://code.jquery.com/mobile/1.4.5/jquery.
mobile-1.4.5.min.css">

<!-- Include the jQuery library -->
<script src="https://code.jquery.com/jquery-1.11.3.min.js"></script>

<!-- Include the jQuery Mobile library -->
<script src="https://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>

</head>
```

Hola Mundo JQuery Mobile

```
<div data-role="page">  
  <div data-role="header">  
    <h1>Primera Página</h1>  
  </div>
```

```
<div data-role="main" class="ui-content">  
  <p>Hola Mundo JQueryMobile!!</p>  
</div>
```

```
<div data-role="footer">  
  <h1>pie de página</h1>  
</div>  
</div>
```

Hola Mundo JQuery Mobile

- **data-role= "page"** es la página que se muestra en el navegador
- **data-rol = "header"** crea una barra de herramientas en la parte superior de la página (a menudo utilizado para el título o botones de búsqueda)
- **El data-role = "main"** define el contenido de la página, como texto, imágenes, botones, formularios, etc.
- **La clase "ui-content"** añade un relleno y un margen adicionales dentro del contenido de la página
- **El data-role = "footer"** crea una barra de herramientas en la parte inferior de la página
- Dentro de estos contenedores, se puede agregar cualquier elemento HTML - párrafos, imágenes, encabezados, listas, etc.

Añadiendo más páginas y enlazandolas

```
<div data-role="page" id="pageone">
  <div data-role="header">
    <h1>Welcome To My
Homepage</h1>
  </div>
  <div data-role="main" class="ui-
content">
    <p>Welcome! If you click on the
link below, it will take you to Page
Two.</p>
    <a href="#pagetwo">Go to Page
Two</a>
  </div>
  <div data-role="footer">
    <h1>Footer Text</h1>
  </div>
</div>
```

```
<div data-role="page"
id="pagetwo">
  <div data-role="header">
    <h1>Welcome To My
Homepage</h1>
  </div>
  <div data-role="main" class="ui-
content">
    <p>This is Page Two. If you click
on the link below, it will take you to
Page One.</p>
    <a href="#pageone">Go to Page
One</a>
  </div>
  <div data-role="footer">
    <h1>Footer Text</h1>
  </div>
</div>
```


Dialogos

- Agregar **data-dialog = "true"** a la página que desea mostrar como un diálogo:

- ```
<div data-role="page" id="pageone">
 <div data-role="main" class="ui-content">
 Go to Page Two
 </div>
</div>
```

```
<div data-role="page" data-
dialog="true" id="pagetwo">
 <div data-role="main" class="ui-content">
 Go to Page One
 </div>
</div>
```

# Transiciones

- El efecto de transición se puede aplicar a cualquier enlace o presentación de formularios mediante el atributo de transición de datos:  
`<a href="#anylink" data-transition="slide">Slide to Page Two</a>`
- Se pueden añadir tanto a paso de páginas como para abrir dialogos.

# Transiciones

- **Slide:** la que se asigna por defecto. data-transition="slide"
- **Slide Up:** La página se perderá por la parte superior. data-transition="slideup"
- **Slide Down:** La página se perderá por la parte inferior. data-transition="slidedown"
- **Pop:** La página aparece de improvisto. data-transition="pop"
- **Fade:** Hace un efecto de fundido. data-transition="pop"
- **Flip:** La página hace un bonito efecto 3D. transition="flip«
- **Turns :** giro de página parecido al flip.

# Botones

- Un botón en jQuery Mobile se puede crear de tres maneras:
  - Utilizando el elemento `<input>`
  - Utilizando el elemento `<button>` con `class = "ui-btn"`
  - Utilizando el elemento `<a>` con `class = "ui-btn"`
- **Botones de navegación:** Utiliza `<a>` como botones de navegación.

# Grupo de Botones

- Query Mobile proporciona una forma sencilla de agrupar botones.
- Utiliza el atributo **data-role = "controlgroup"** junto con **data-type = "horizontal | vertical"** en un elemento de contenedor, para especificar si agrupar los botones horizontal o verticalmente:

# Ejemplo controlgroup

```
<div data-role="page" id="pageone">
 <div data-role="header">
 <h1>Grouped Buttons</h1>
 </div>

 <div data-role="main" class="ui-content">
 <div data-role="controlgroup" data-type="horizontal">
 <p>Horizontal group:</p>
 Button 1
 Button 2
 Button 3
 </div>

 <div data-role="controlgroup" data-type="vertical">
 <p>Vertical group (default):</p>
 Button 1
 Button 2
 Button 3
 </div>
 </div>

 <div data-role="footer">
 <h1>Footer Text</h1>
 </div>
</div>
```

# Botones

- Distintas clases CSS para botones:
  - **Ui-btn-b**: Cambia el color del botón a un fondo negro con texto blanco (por defecto es el fondo gris con texto en negro).
  - **Ui-corner-all**: Agrega las esquinas redondeadas al botón
  - **Ui-mini**: Hace que el botón sea más pequeño
  - **Ui-shadow**: Agrega sombras al botón
- Si deseas utilizar más de una clase, separa cada clase con un espacio: `class = "ui-btn ui-btn-inline ui-btn-corner-all ui-shadow"`
- De forma predeterminada, los botones `<input>` tienen sombras y esquinas redondeadas. Los elementos `<a>` y `<botón>` no.

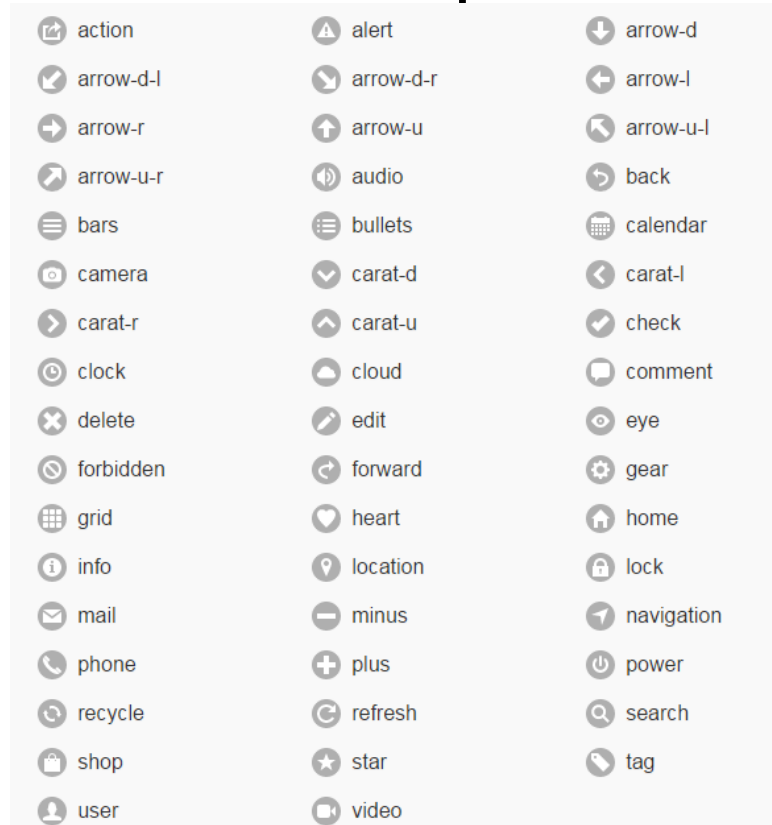
# Botones

- Para crear un botón Atrás, utiliza el atributo **data-rel = "back"** (Nota: esto ignorará el valor href del enlace):
- De forma predeterminada, los botones ocupan todo el ancho de la pantalla. Si desea un botón que sea tan amplio como su contenido, o si desea que aparezcan dos o más botones uno al lado del otro, agrega la class "**ui-btn-inline**":
  - `<a href="#noimporta" data-rel="back" class="ui-btn ui-btn-inline">Atras</a>`



# Iconos

- JQuery Mobile proporciona un conjunto de iconos que harán que tus botones parezcan más atractivos.
- Para usarlos:
  - `Data-icon=nombreIcono`
  - `o añadir al class`
  - `ui-icon-nombreIcono`



# Botones e Iconos

- Para colocar iconos en los botones en el class añadimos **ui-btn-icon-POSICIONBOTON** :

```
<div data-role="main" class="ui-content">
 <a href="#" data-rel="back" class="ui-btn ui-icon-back ui-btn-
icon-left">Back Button
</div>
```

**Posicionboton** puede ser: left, right, top, bottom

# Iconos personalizados

- Podemos usar el icono como cualquier otro, pero con el nombre que le pongamos y que no coincida con el resto:

```
<button class="ui-btn ui-shadow ui-corner-all ui-btn-icon-left ui-icon-myicon">myicon</button>
```

- Con css debemos indicar la imagen del icono de la siguiente forma

```
.ui-icon-myicon:after {
 background-image: url("../_assets/img/icons/21-skull.png");
 /* Make your icon fit */
 background-size: 18px 18px;
}
```

# Popups

- Los popups son similares a los diálogos, ya que ambos superponen una parte de una página. Un cuadro emergente puede ser útil cuando desea mostrar texto pequeño, fotos, mapas u otro contenido.
- Para crear una ventana emergente:
  - Comienza con un elemento `<a>` y un elemento `<div>`.
  - Agrega el atributo `data-rel = "popup"` a `<a>` y el atributo `data-role = "popup"` al `<div>`.
  - A continuación, especifique un id para `<div>` y establezca el href de `<a>` para que coincida con el id especificado.
  - El contenido dentro de `<div>` es el contenido real que aparecerá cuando un usuario haga clic en el vínculo

# Popups

```
<div data-role="main" class="ui-content">
 Show Popup
 <div data-role="popup" id="myPopup">
 <p>This is a simple popup.</p>
 </div>
</div>
```

Personalizando el Popup con atributos en el **enlace que lo abre**:

- **data-transition**: mismas opciones que para pages o dialogs(fade,slide,pop..)
- **data-position**: coloca el popup en algún lugar en particular, sus posibles valores son : window(en el centro de la pantalla), id de otro elemento y origin(encima de donde se origina)
- **data-arrow**: crea una flecha tipo bocadillo en el popup, puede tomar los valores de l,r,t,b

# Popups

Para añadir boton de cerrar al popup, creamos un boton de tipo back y lo colocamos por css:

```
<div data-role="main" class="ui-content">
 Show Popup

 <div data-role="popup" id="myPopup" class="ui-content" data-overlay-theme="b">
 <a href="#" data-rel="back" class="ui-btn ui-corner-all ui-shadow ui-btn ui-icon-delete ui-btn-
icon-notext ui-btn-right">Close
 <p>I have a close button at the top right corner.</p>
 <p>Tip: You can also click outside to close me.</p>
 </div>
</div>
```

Nota que además se ha añadido el atributo data-overlay-theme="b" para oscurecer toda la ventana menos el popup.

# Barra de Navegacion

- Una barra de navegación consiste en un grupo de enlaces que están alineados horizontalmente, normalmente dentro de un encabezado o pie de página.
- La barra está codificada como una lista desordenada de enlaces envueltos dentro de un elemento `<div>` que tiene el atributo `data-role = "navbar"`:

```
<div data-role="page" id="pageone">
 <div data-role="header">
 <h1>Welcome To My Homepage</h1>
 <div data-role="navbar">

 Home
 Page Two
 Search

 </div>
 </div>
 <div data-role="main" class="ui-content">
 <p>My Content..</p>
 </div>
 <div data-role="footer">
 <h1>My Footer</h1>
 </div>
</div>
```

- De forma predeterminada, los enlaces dentro de una barra de navegación se convierten automáticamente en un botón (no es necesario para `class = "ui-btn"` o `data-role = "button"`).

# Barra de Navegacion

- Para añadir iconos a los botones de navegación podemos usar los atributos data-icon y data-iconpos:

```
<div data-role="page" id="pageone">
 <div data-role="header">
 <h1>Welcome To My Homepage</h1>
 <div data-role="navbar" data-iconpos="left">

 Home
 Page Two
 Search

 </div>
 </div>
 <div data-role="main" class="ui-content">
 <p>contenido.</p>
 </div>
 <div data-role="footer">
 <h1>My Footer</h1>
```



# Barra de Navegacion

- Cuando un enlace en la barra de navegación es pulsado / clicado, obtiene el aspecto seleccionado (presionado).
  - Para lograr este aspecto sin tener que clicar el enlace, utilice la `class = "ui-btn-active"`.
  - Además si tienes varias páginas, es posible que desees que aparezca "seleccionado" cada botón que representa la página en la que se encuentra el usuario. Para ello, agrega la clase "ui-state-persist" a los enlaces, así como la clase "ui-btn-active":
    - `<li><a href="#anylink" class="ui-btn-active ui-state-persist">Home</a></li>`

# Panel lateral

- Los paneles de jQuery Mobile se deslizan hacia fuera desde el lado izquierdo o derecho de la pantalla.
- Para crear un panel, agregue el atributo data-role = "panel" a un elemento <div> y especifique un id.
  - Agregue cualquier marca HTML dentro de esta <div> que desee mostrar en su panel: contenido adicional.
- Para acceder al panel, cree un enlace que apunte al id del panel <div>. Cuando un usuario hace clic en el enlace, el panel se abrirá.
- Los paneles deben definirse antes o después del header, content y footer (no por medio)

# Panel lateral

```
<div data-role="page" id="pageone">
 <div data-role="panel" id="myPanel">
 <h2>Panel Header</h2>
 <p>You can close the panel by clicking outside the panel, pressing the Esc key or by swiping.</p>
 </div>
```

```
<div data-role="header">
 <h1>Page Header</h1>
</div>
```

```
<div data-role="main" class="ui-content">
 <p>Click on the button below to open the Panel.</p>
 Open Panel
</div>
```

```
<div data-role="footer">
 <h1>Page Footer</h1>
</div>
</div>
```

# Panel lateral

- Puede controlar el modo de visualización del panel con el atributo de data-display:

```
<div data-role="panel" id="overlayPanel" data-display="overlay">
```

```
<div data-role="panel" id="revealPanel" data-display="reveal" data-position="right">
```

```
<div data-role="panel" id="pushPanel" data-display="push">
```

- Overlay-> despliega el panel sobre el contenido de la pagina
- Push-> anima ambos, panel y contenido al mismo tiempo
- Reveal-> por defecto, parecido a push.

Nota: para cambiar el lado por el que aparece el panel podemos usar el atributo data-position

# Listas

- Las listview en jQuery Mobile son listas estándar de HTML; Ordenado (<ol>) y no ordenado (<ul>).
- Para crear una lista, aplique el data-role="listview" al elemento <ol> o <ul>. Para hacer que los elementos sean seleccionable, especifique un enlace dentro de cada elemento de lista (<li>):

# Listas

```
<div data-role="page" id="pageone">
 <div data-role="main" class="ui-content">
 <h2>Ordered List:</h2>
 <ol data-role="listview" data-inset =
"true">
 List Item
 List Item
 List Item

 <h2>Unordered List:</h2>
 <ul data-role="listview">
 List Item
 List Item
 List Item

 </div>
</div>
```

Para diseñar sus listas con esquinas redondeadas y algún margen, utilice el atributo **data-inset = "true"**:

# Listas

- Elementos divisores de Listas: se utilizan para organizar y agrupar elementos en categorías/secciones.
- Se especifican añadiendo el data-role="list-divider" como atributo del <li> divisor:

```
<div data-role="page" id="pageone">
 <div data-role="main" class="ui-content">
 <h2>List Dividers</h2>
 <ul data-role="listview" data-inset="true">
 <li data-role="list-divider">Europe
 Norway
 Germany
 <li data-role="list-divider">Asia
 India
 Thailand
 <li data-role="list-divider">Africa
 Zimbabwe
 Uganda

 </div>
</div>
```

- Existen Divisores automaticos para listas, los dividen de forma alfabetica, para usarlo añadir el atributo data-autodividers="true"

# Listas

- Si no quieres el icono que aparece a la derecha de los items de la lista, añade el atributo `data-icon="false"` a cada `<li>`, si quieres otro icono pon `data-icon="nombre del otro icono"`.
- Para agregar un icono estándar de 16x16px a su lista, agregue un elemento `<img>` dentro del enlace con una clase de `"ui-li-icon"`:

```
<ul data-role="listview">
 USA

```

- Si no añades la clase, JQuery Mobile automáticamente le pondrá tamaño 80x80.



# Listas

- Para crear una **lista dividida** con una barra divisora vertical, coloque dos enlaces dentro del elemento `<li>`.
- JQuery Mobile colocará automáticamente el segundo enlace en el lado derecho de la lista con un icono de flecha hacia la derecha. Y el texto dentro del enlace (si lo hay) se mostrará con un hover:

```
<ul data-role="listview">

 Some Text


```

- Las **burbujas de conteo** se utilizan para mostrar números asociados con elementos de la lista, como mensajes en un buzón:

```
<ul data-role="listview" data-inset="true">
 Inbox25
 Sent432
 Trash7

```

# Listas filtradas

- Todos los elementos que tienen más de un elemento hijo, pueden ser filtrados.
- Cómo crear un campo de búsqueda:
  - El elemento que desea que sea filtrado, debe incluir el filtro `data-filter = "true"`.
  - Cree un elemento `<input>` con un id especificado y añada el atributo `data-type = "search"`. Esto creará un campo de búsqueda básico. Envuelva el `<input>` en un formulario y agregue la `class="ui-filterable"` al elemento `<form>` - esto ajustará el margen entre el campo de búsqueda y el elemento filtrable.
  - A continuación, agregue el atributo `data-input` al elemento filtrable. Su valor debe coincidir con el id del elemento `<input>`.

```
<form class="ui-filterable">
 <input id="myFilter" data-type="search">
</form>
```

```
<ul data-role="listview" data-filter="true" data-input="#myFilter">
 Adele
 Billy
 Calvin

```

- Si no quieres filtrar por el texto de dentro del li, puedes especificar el valor por el que se filtrará con el atributo `data-filtertext` en cada elemento:

```
<li data-filtertext="fav">Adele
```

# Formularios

- Para trabajar con formularios jQuery Mobile:
  - El elemento `<form>` debe tener un método y un atributo de acción
  - Cada elemento de formulario debe tener un atributo "id" único. El identificador debe ser único en las páginas del sitio. Esto se debe a que el modelo de navegación de una sola página de jQuery Mobile permite que muchas "páginas" diferentes estén presentes al mismo tiempo
  - Cada elemento de formulario debe tener un label. Establezca el atributo `for` del label para que coincida con el id del elemento:

```
<form method="post" action="/action_page_post.php">
 <label for="fname">First name:</label>
 <input type="text" name="fname" id="fname">
</form>
```

# Formularios

- En Jquery Mobile puedes usar los siguientes controles:
  - Text Inputs
  - Search Inputs
  - Radio Buttons
  - Checkboxes
  - Select Menus
  - Sliders
  - Flip Toggle Switches

# Formularios

- Típico uso:

- Utilizar placeholder y ocultar los labels:

```
<label for="fname" class="ui-hidden-accessible">First name:</label>
<input type="text" name="fname" id="fname" placeholder="First name...">
```

- Añadir boton de borrar el campo a la derecha del input

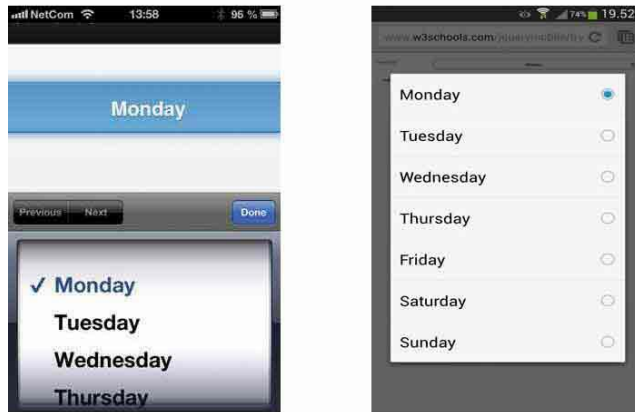
```
<label for="fname">First name:</label>
<input type="text" name="fname" id="fname" data-clear-btn="true">
```

- Para que las etiquetas y los elementos de formulario se vean correctamente en pantallas más amplias, envuelva un elemento `<div>` o `<fieldset>` con la clase "ui-field-contain" alrededor del elemento label / form:

```
<div class="ui-field-contain">
 <label for="fname">First name:</label>
 <input type="text" name="fname" id="fname">
 <label for="lname">Last name:</label>
 <input type="text" name="lname" id="lname">
</div>
```

# Selects

Por defecto el select se verá en cada plataforma móvil con un estilo:



- Si queremos que en todas las plataformas se vea igual deberemos desactivar esta función:  
`<select name="day" id="day" data-native-menu="false">`
- Si queremos un select con multiples selecciones , añadimso ela tribute multiple:  
`<select name="day" id="day" multiple data-native-menu="false">`

# Flip Toggle Switch

- Para crear un flip, use `<input type = "checkbox">` y especifique un rol de datos de "flipswitch":

```
<form method="post" action="/action_page_post.php">
 <label for="switch">Flip toggle switch checkbox:</label>
 <input type="checkbox" data-role="flipswitch" name="switch" id="switch">
</form>
```

- Si quieres cambiar el texto por defecto de los flips:
  - `<input type="checkbox" data-role="flipswitch" name="switch" id="switch" data-on-text="Si" data-off-text="No">`

-

# Themes

- JQuery Mobile ofrece dos temas de estilo diferentes, "a" y "b" - cada uno con colores diferentes para botones, barras, bloques de contenido, etc.
- Para personalizar la apariencia de su aplicación, utilice el atributo data-theme y asigne el atributo con una letra:

```
<div data-role="page" data-theme="a|b">
```

- Nota :Este atributo puede usarse en casi todas las etiquetas html con las que trabaja JQuery Mobile. En elementos que pueden colapsarse se utiliza en cambio data-content-theme. Para enlaces mostrados como botones podemos añadir la clase css ui-btn-b en lugar de ui-btn



# Themes

- JQueryMobile nos proporciona una herramienta para crear nuestros propios temas:  
<http://themeroller.jquerymobile.com/>
- En <https://jqmgallery.com> podemos ver ideas para crear nuestro tema, incluso diseñar nuestra aplicación con JQuery Mobile.
- También podemos usar themes de terceros. Aquí algunos buenos temas descargables:

<http://www.gajotres.net/top-10-best-looking-free-jquery-mobile-themes/>

# Eventos JQuery Mobile

Con el uso de JQuery Mobile nos encontramos con nuevos eventos que podemos manejar.

# Pagecreate

- En lugar de esperar a que se cargue todo el documento, podemos esperar a que se cargue solamente la página deseada, reaccionando a este evento.
- A todos los eventos de JQuery Mobile se les asigna listener con el método on.
- `<script>`  
    `$(document).on("pagecreate","#pageone",function(){`  
        *// jQuery events go here...*  
    `});`  
    `</script>`

# Tap y Swipe

- Tap-> cuando un usuario toca un elemento:  

```
$("#p").on("tap",function(){
 $(this).hide();
});
```
- Taphold -> cuando toca un elemento y lo mantiene por un segundo
- Swipe -> se activa cuando el usuario pasa un elemento horizontalmente por más de 30px:
  - **Swipeleft**
  - **swipteright**

# Scroll

- Scrollstart-> lanzado cuando el usuario comienza hacer scroll en la página.
- Scrollstop-> lanzado cuando el usuario para de hacer scroll en la página.
- Se tienen que escuchar sobre el objeto document:

```
$(document).on("scrollstart",function(){
 alert("Started scrolling!");
});
```
- Nota: Los dispositivos iOS congelan la manipulación DOM durante el scrolling, lo que significa que no es posible cambiar algo cuando el usuario hace scrolling. Sin embargo, el equipo jQuery está trabajando en una solución para esto.

# Orientationchange

- Cuando el usuario hace girar el dispositivo móvil vertical u horizontalmente.
- Se debe escuchar sobre el objeto window:  

```
$(window).on("orientationchange",function(event){
 alert("Orientation is: " +
 event.orientation);
});
```

# Eventos sobre las páginas

- también podemos usar eventos para cuando pasemos de una página a la siguiente.
- Las transiciones de página implican dos páginas: una página "from" y una página "to" - estas transiciones animan el cambio de la página activa actual (fromPage) a una nueva página (toPage).
- En un ejemplo vemos los 4 eventos:

```
$(document).on("pagebeforeshow","#pagetwo",function(){ // When entering pagetwo
 alert("pagetwo is about to be shown");
});
$(document).on("pageshow","#pagetwo",function(){ // When entering pagetwo
 alert("pagetwo is now shown");
});
$(document).on("pagebeforehide","#pagetwo",function(){ // When leaving pagetwo
 alert("pagetwo is about to be hidden");
});
$(document).on("pagehide","#pagetwo",function(){ // When leaving pagetwo
 alert("pagetwo is now hidden");
});
```

# Eventos sobre las páginas

- Carga de páginas externas:
  - Si dentro de una página cargamos otra página externa, se disparan 2 eventos. El primero es `pagecontainerbeforeload`, y el segundo será `pagecontainerload (success)` o `pagecontainerloadfailed (fail)`.

```
$(document).on("pagecontainerload",function(event,data){
 alert("pageload event fired!\nURL: " + data.url);
});
$(document).on("pagecontainerloadfailed",function(event,data){
 alert("Sorry, requested page does not exist.");
});
```