

# JQuery



- JQuery es una librería javascript multinavegador que facilita tareas habituales como son acceso al dom, ajax, eventos, etc.
- Es una librería ampliamente usada y fácilmente extensible.
- Tiene gran cantidad de plugins para todo tipo de tareas (formularios, widgets, animaciones...).
-

- JQuery ofrece facilidades para:
  - Selección de elementos.
  - Tratamiento de atributos.
  - Recorrido y modificación de elementos.
  - Facilidades para el tratamiento de css.
  - Facilidades para la gestión de eventos.
  - Ajax.
- Para usar jquery debemos referenciar a la librería:
  - Minified para entorno de producción ya que está reducida al máximo para que el tiempo de descarga sea mínimo.
  - “Normal” para el entorno de desarrollo.

# jQuery

---

- Podemos referenciar jquery de manera local a nuestra aplicación:

`<script src="/lib/jquery/jquery-1.7.1.js" type="text/javascript"></script>`

- También podemos usar algunos de los CDN habituales para la carga de librerías:
- <http://code.jquery.com/jquery-1.7.1.min.js>
- <http://code.jquery.com/jquery-1.7.1.js>
- <http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js>
- <http://ajax.microsoft.com/ajax/jquery/jquery-1.7.1.min.js>

# jQuery

---

- Por defecto al cargar jquery definirá la variable \$ para que podamos usar las funcionalidades que nos ofrece.
- JQuery ofrece muchas opciones para seleccionar elementos del dom. Algunas de las más útiles:
  - Acceso por id:  
`jQuery("#ID");`
  - Acceso por atributo con valor:  
`jQuery("[nombre = valor]");`
  - Acceso por atributo que comienza por valor:  
`jQuery("[nombre ^= valor]");`
  - Acceso por atributo que termina por valor:  
`jQuery("[nombre $= valor]");`
  - Todos los elementos con la clase (css) X:  
`jQuery(".X");`
  - Todos los elementos ocultos:  
`jQuery(":hidden");`
  - Todos los elementos visibles:  
`jQuery(":visible");`
  - Todos los botones:  
`jQuery(":button");`
  - Todos los hijos del padre:  
`jQuery("parent >child");`
  - Todos los div:  
`jQuery("div");`

- Atributos

- Permiten modificar atributos a un conjunto de elementos del DOM.
- Algunos métodos útiles:
  - Añadir una clase o clases al conjunto de elementos:  
`addClass("XXX");`
  - Obtener el valor de un atributo para el primer elemento:  
`attr("nombreAtributo");`
  - Asignar el valor de un atributo para los elementos:  
`attr("nombreAtributo", valor);`
  - Asignar varios valores a la vez, incluso ejecutar funciones:  
`jQuery("[name^=boton]").attr("value",function(index,oldvalue){ return ""+index;});`  
`jQuery("#id").attr({name:"nombre",value:"valor"});`

- Recorrido de elementos
  - jQuery ofrece toda una serie de métodos que facilita el recorrido de elementos:
    - Recorrer todos los elementos DOM de un objeto jQuery ejecutando una función:
      - `each( function(index, element) )`
    - Filtrar los elementos de un objeto jQuery:
      - `filter( selector )`
      - `filter( function(index) )`
    - Obtener el siguiente elemento (que cumpla una condición):
      - `next( [ selector ] )`

- Manipulación

- jQuery ofrece métodos para añadir elementos, eliminarlos, modificar sus propiedades...

- Para obtener el valor de un elemento:

`jQuery("#id").val();`

- Para asignar el valor a un elemento:

`jQuery("#id").val(valor);` o

`jQuery(selector).val(valor);` o

`jQuery(selector).val(function(index,valor){...});`

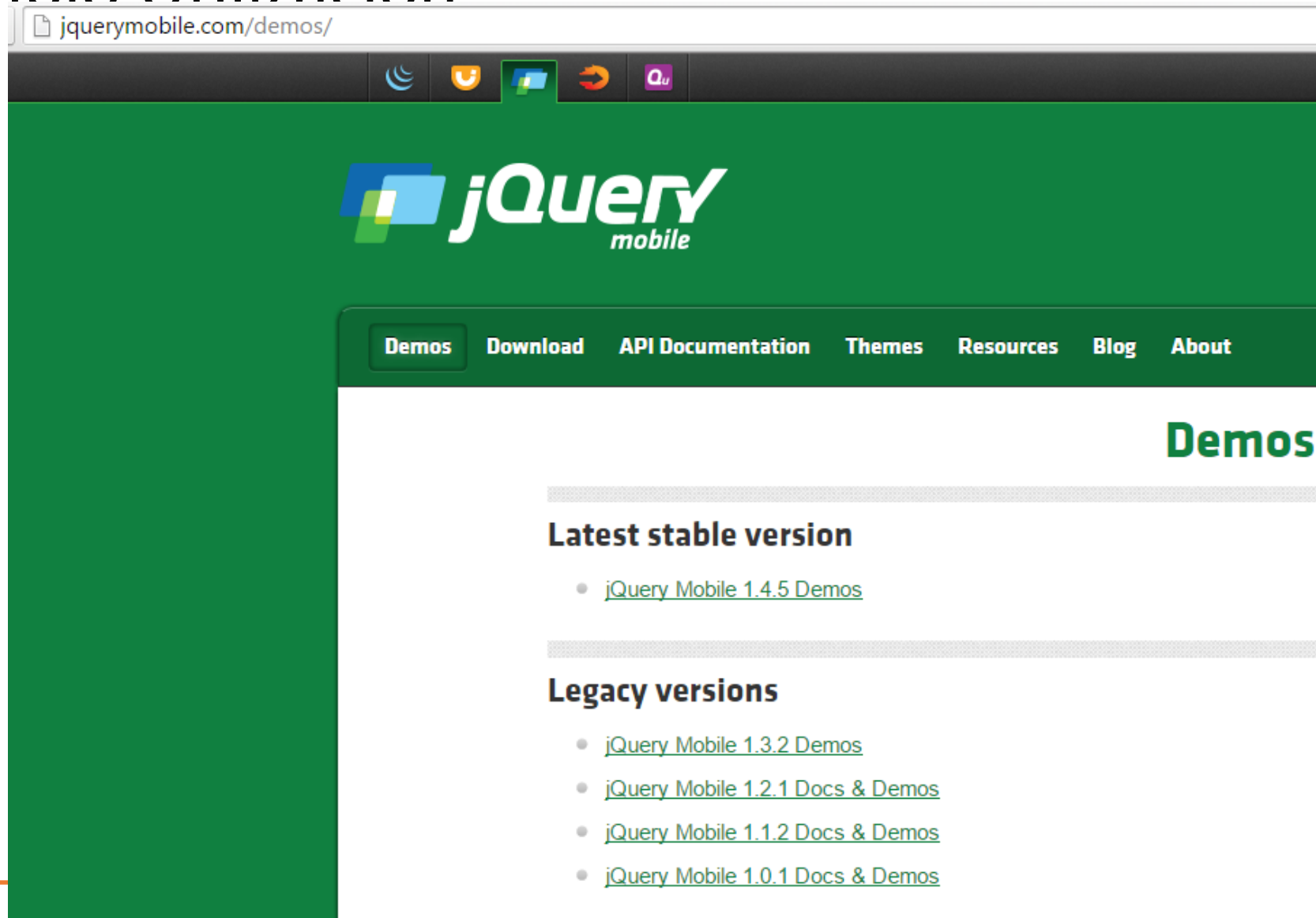


- Eventos

- Algunos métodos útiles que ofrece jQuery para gestionar eventos y event handlers:
  - Para ejecutar una función cuando el DOM está listo (cargado):  
`jQuery(document).ready(function);`
  - Para asignar un evento a un elemento:  
`jQuery(selector).bind("nombreEvento",handler)` o  
`jQuery(selector).bind("nombreEvento",datos,handler);`
- Más métodos interesantes:
  - Para programar un click (de forma más corta que con bind):  
`jQuery(selector).click(function);`
  - Para lanzar un evento click:  
`jQuery(selector).click();`
  - Para parar la propagación de un evento:  
`jQuery(selector).bind("evento",data,function(event){event.stopPropagation(); ...});`



- Ejemplo Utilización:



- Ejemplo Utilización:

The screenshot shows the jQuery UI documentation page. On the left is a navigation menu with links: Home, Introduction, Buttons, Button widget, Checkboxradio widget, Collapsible (set) widget, Contr..., Datep..., Event..., Filterable widget, Flips... widget, Forms, Grids, and Grouping and dividing content. The main content area is titled 'Button markup' and includes the text: 'Add classes to style `a` and `button`' and 'Note that in 1.4 `data-` attributes v... `role="button"`) and `button` elem...'. Below this, there is a section titled 'Basic markup' in green. A blue overlay box with the title 'HTML' contains the following code:

```
1 <a href="#" class="ui-btn">Anchor</a>
2 <button class="ui-btn">Button</button>
```

Below the code, there is a 'View Source' button and a section titled 'Corners' in green.

# Ej

```
index.html
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
  <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
  <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
</head>
<body>

  <div data-role="page" id="page1">
    <div data-role="header">
      <h1>Navegación con JQuery Mobile</h1>
    </div>
    <div data-role="content">
      <div data-role="controlgroup" data-type="horizontal">
        <a href="#page2" data-role="button" data-icon="star" data-iconpos="left" data-theme="a" data-mini="true" data-transition="slide">Pag. 2</a>
        <a href="#page3" data-role="button" data-icon="star" data-iconpos="left" data-theme="b" data-mini="true" data-transition="pop">Pag. 3</a>
        <a href="#page4" data-role="button" data-icon="star" data-iconpos="left" data-theme="c" data-mini="true" data-transition="fade">Pag. 4</a>
      </div>
    </div>
    <div data-role="footer"><h1>PIE DE PÁGINA</h1></div>
  </div>

  <div data-role="page" id="page2">
    <div data-role="header"><h1>Pag. 2</h1></div>
    <div data-role="content">
      <a href="#page1" id="ancla" data-role="button" data-icon="home" data-iconpos="notext" data-transition="pop"></a>
    </div>
    <div data-role="footer"><h1>PIE DE PÁGINA</h1></div>
  </div>

  <div data-role="page" id="page3">
    <div data-role="header"><h1>Pag. 3</h1></div>
    <div data-role="content">
      <a href="#page1" id="ancla" data-role="button" data-icon="home" data-iconpos="notext" data-transition="slideup"></a>
    </div>
    <div data-role="footer"><h1>PIE DE PÁGINA</h1></div>
  </div>

  <div data-role="page" id="page4">
    <div data-role="header"><h1>Pag. 4</h1></div>
    <div data-role="content">
      <a href="#page1" id="ancla" data-role="button" data-icon="home" data-iconpos="notext" data-transition="slidedown"></a>
    </div>
    <div data-role="footer"><h1>PIE DE PÁGINA</h1></div>
  </div>

  <!-- Referencia de Cordova que se agrega a la aplicación cuando se compila. -->
  <script src="cordova.js"></script>
  <script src="scripts/platformOverrides.js"></script>

  <script src="scripts/index.js"></script>
</body>
```

- Ejemplo Utilización:

