

**jQuery Mobile**

Rev Digitale 1.0 del 01/09/2016

Utilizzo delle librerie jQuery Mobile .....	2
L'attributo data-role .....	3
L'attributo data-theme .....	3
Il widget Page .....	4
Il widget ListView .....	5
Il widget Button .....	5
I widget Collapsible e Collapsibleset .....	6
La classe ui-gird .....	7
Collegamento a pagine esterne .....	7

## jQuery Mobile

jQuery Mobile è una libreria per la scrittura di applicazioni per dispositivi mobili (smartphone e tablet). Consente di realizzare applicazioni multi dispositivo, utilizzabili su diverse piattaforme.

Ultime versioni

<u>1.3.2</u>	19 luglio 2013
<u>1.4.1</u>	12 febbraio 2014

Il sito di riferimento è [jquerymobile.com](http://jquerymobile.com).

Il link per l'utilizzo delle librerie è il seguente:

```
<script src="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.js">
```

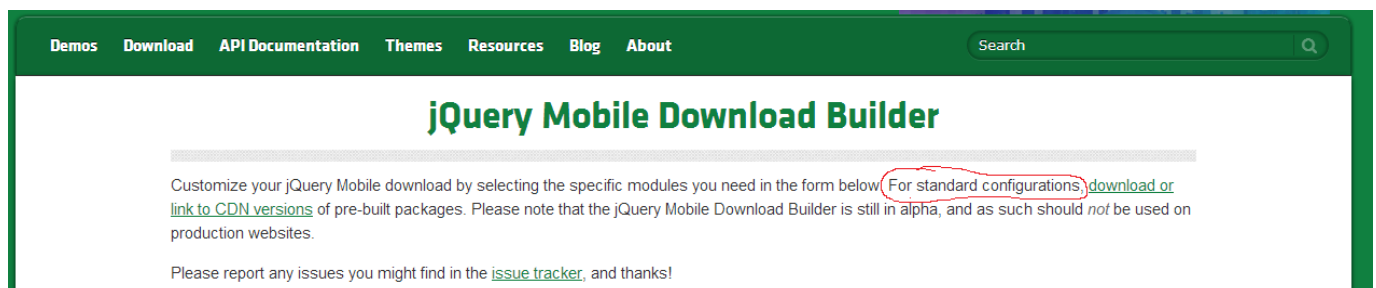
E' anche possibile fare riferimento direttamente alla versione più recente:

```
<script src="http://code.jquery.com/mobile/latest/jquery.mobile.js">
```

Come nel caso dell'interfacce grafiche, anche per il mobile sono disponibili una serie impostazioni CSS comode per realizzare layout efficaci orientati all'uso dei dispositivi mobile, tipicamente ad elevato contrasto quindi facilmente leggibili. Il link è il seguente:

```
<link href="http://code.jquery.com/mobile/latest/jquery.mobile.css"/>
```

Per scaricare la versione 1.3.2 aprire la sezione “**custom download**”



Utilizzare il link iniziale **standard configuration**, e scaricare l'intera cartella

## Widget e Classi

In jQuery Mobile è disponibile un vasto gruppo di widget simili a quelli di jQuery UI. La sintassi di utilizzo è identica, però il nome dei widget è diverso rispetto a quelli di jQueryUI ed anche, a parità di oggetto, Proprietà, Metodi ed Eventi sono differenti.

[jQueryMobile.com](http://jquerymobile.com) / [ApiDocumentation](http://jquerymobile.com/apiDocumentation) contiene

- un elenco completo dei **widget** jQuery Mobile con le relative sintassi
- un elenco completo di tutte le **classi** del framework, con una sezione apposita per la **ui-grid**

Interessante anche la pagina [jQueryMobile.com / Demos](http://jquerymobile.com/demos)

### L'attributo data-role

Per trasformare un elemento HTML nel corrispondente Widget, anziché applicare il costruttore all'interno di una pagina di script, è possibile utilizzare un nuovo attributo HTML5 **data-role** che trasforma l'elemento nel widget corrispondente. Anche tutte le proprietà disponibili per il widget possono essere impostate come attributo HTML5 antepoendo il prefisso data-

Ad esempio il widget **page** dispone della proprietà **dialog** che lo trasforma in una finestra di dialogo. Questa proprietà può essere assegnata anche nel modo seguente:

```
<div data-role="page" data-dialog="true">
```

Provando nel browser semplicemente apre in una nuova scheda.

### L'attributo data-theme in jQuery Mobile 1.3.2

Un **tema** è un insieme di impostazioni grafiche che definiscono gli aspetti visuali di un'applicazione: dai font alle ombreggiature, dai colori alle texture, dalla forma degli elementi alla loro disposizione.

Il tema predefinito di jQuery Mobile prevede **cinque swatch**, ciascuno individuato da una delle prime cinque lettere dell'alfabeto (a, b, c, d, e). Ogni swatch definisce un determinato livello di **priorità visuale**, cioè di enfasi nella visualizzazione. Così dal livello più alto dello swatch a con forte contrasto (testo bianco su sfondo nero) si va gradatamente verso il livello più basso dello swatch e, che presenta colori più delicati e un minore contrasto.

```
"a" content bianco su sfondo nero,  
"b" content bianco su sfondo blu  
"c" content nero su sfondo grigio chiaro  
"d" content nero su sfondo grigio scuro  
"e" content nero su sfondo giallo
```

I vari swatch assumono un valor differente a seconda degli elementi.

Ad esempio lo **swatch b** :

- assume il colore bianco / sfondo blu per **liste** e **pulsanti**
- assume invece il colore nero / sfondo grigio chiaro per gli elementi **content** e **page**.

Se non diversamente specificato jQuery Mobile 1.3.2 utilizza le seguenti **regole** per assegnare gli swatch ai vari elementi della pagina:

- **header e footer** assumono per default il **theme a**, in modo da risultare più evidenti. **header** e **footer** non ereditano dal contenitore, per cui se si specifica un theme ad esempio nel tag **page**, **header** e **footer** assumeranno comunque il theme **a**.
- il **content** assume per default il **theme c** per massimizzare il contrasto rispetto a header e footer. Nemmeno lui eredita da page
- **listitem** e **pulsanti** assumono lo swatch dell'elemento che li contiene. Ereditano cioè dall'elemento **content** se questi ha un tema esplicitamente definito oppure, se il tema di content non è definito, ereditano dall'elemento **page**, sempre se questo ha un tema esplicitamente definito; in caso contrario ereditano il tema di default dell'elemento content.

Scrivendo su google "jquery mobile themes" si può accedere alla URL

<http://demos.jquerymobile.com/1.2.1/docs/api/themes.html>

che spiega come verrà applicato ciascun swatch ai vari elementi.

## L'attributo `data-theme` in jQuery Mobile 1.4.1

[jquerymobile.com/upgrade-guide/1.4](http://jquerymobile.com/upgrade-guide/1.4)

In jQuery Mobile 1.4.1 gli swatches prestabiliti sono solo più 2:

- **a theme** (light) (nero / grigio o bianco a seconda del contesto)
- **b theme** (dark) (bianco / nero)

Temi colorati personalizzati possono essere creati mediante il nuovo apposito **Theme Roller** disponibile su [jQueryMobile.com / Themes](http://jQueryMobile.com / Themes) che consente di creare **26 swatches** differenti con lettere dalla **a** alla **z**. Terminata la creazione dello swatch (ad esempio sulla finestra c) un pulsante in alto, dopo aver richiesto un filename, consente di salvare il tema creato all'interno di un **filename.css** che dovrà poi essere incluso al progetto rendendo utilizzabile il nuovo swatch con la lettera utilizzata in fase di progetto.

## Il Widget Page

La pagina HTML può essere organizzata in piccole pagine indipendenti, pensate appositamente per piccoli dispositivi come gli smartphone, ciascuna identificata dal widget page.

Il vantaggio di questo tipo di template rispetto a quello a pagina singola, sta nella maggior **responsività** legata al fatto di caricare tramite Ajax solo la pagina che serve e non l'intera pagina HTML. La navigazione interna è realizzata considerando ogni pagina come un'ancora:

Ciascuna pagina è impostata secondo il tipico model HTML5 contenente diverse sezioni :

**`data-role="header"`** contenente un tag H che rappresenta la barra del titolo  
**`data-role="content"`** che rappresenta il contenuto della pagina (*deprecato*)  
**`data-role="footer"`** che rappresenta il fondo pagina

## Il pulsante Indietro

All'interno dell'elemento con **`data-role = "page"`** relativo ad una pagina secondaria è possibile aggiungere un pulsante di ritorno tramite l'utilizzo dei seguenti attributi :

- L'attributo **`data-add-back-btn="true"`** consente di inserire nella pagina un pulsante di ritorno alla pagina precedente.
- L'attributo **`data-back-btn-text="Back"`** consente di impostare il testo del pulsante di ritorno.

In alternativa è possibile utilizzare, all'interno dell'header, normali collegamenti ipertestuali dotati di icona

```
<div data-role="header">
  <a href="#pagPrincipale" data-icon="back" > back </a>
  <h1> Filmati in Italiano </h1>
  <a href="#" data-icon="gear">Options</a>
</div>
```

Notare che il **`data-role = header`** accetta un unico pulsante a sinistra ed un unico pulsante a destra. Se si desidera avere più pulsanti occorre utilizzare una classe **`ui-bar`** oppure un widget **`nav-bar`** (menù)

```
<div class="ui-bar ui-bar-b">
  <a href="#" data-role="button" data-mini="true"> Button1 </a>
  <a href="#" data-role="button" data-mini="true"> Button2 </a>
</div>
```

Viceversa nel **`data-role = footer`** si possono inserire quanti pulsanti si vuole.

## Il Widget listview

Tipica lista utilizzata nei dispositivi mobili (barre con a destra una icona a forma di freccia).

```
<ul data-role="listview">
  <li><a href="audi.html">Audi</a></li>
  <li><a href="bmw.html">BMW</a></li>
  <li><a href="acura.html">Cadillac</a></li>
</ul>
```

Se invece di **UL** si usa **OL** le voci della lista saranno automaticamente numerate.

### Esempio:

```
<div id="indice" data-role="page" data-theme="b">
  <div data-role="header">
    <h1> Storia del computer </h1>
  </div>

  <div data-role="content">
    <ul data-role="listview" data-inset="true">
      <li> <a href="#indiceIta"> Filmati in italiano </a></li>
      <li> <a href="#indiceEng"> Filmati in inglese </a></li>
    </ul>
  </div>

  <div data-role="footer">
    <p> &copy; Itis Vallauri Fossano </p>
  </div>
</div>
```

L'attributo **data-inset="true"** fa sì che le barre della lista non occupino tutta la pagina nella sua larghezza, ma abbiano un margine sinistro e destro e gli angoli arrotondati.

## Il Widget button

L'attributo **data-mini="true"** fa sì che il pulsante / oggetto venga visualizzato con una grafica leggermente ridotta particolarmente adatta a dispositivi di piccole dimensioni.

## Il Widget collapsible

Un singolo elemento diventa espandibile se si imposta l'attributo **data-role="collapsible"**. Il primo tag successivo al tag `collapsible` dovrà essere un tag `H` il cui contenuto è quello che verrà mostrato sulla barra con davanti un `+`. Clickando sul `+` verrà mostrato anche il contenuto successivo (nell'esempio i tag `p` e `img`).

```
<div data-role="collapsible">
  <h3> Il Bar Caffetteria </h3>
  <p> Immagini del Bar Caffetteria del London Cafè </p>
  
</div>
```

Inserendo i vari elementi espandibili all'interno di un tag `<div data-role="collapsible-set">` **l'apertura dei vari blocchi diventa alternativa**: aprendo un blocco viene automaticamente chiuso il blocco precedente.

## Il Widget collapsible-set

del **collapsible-set** è l'equivalente dell'accordion di jQueryUI, cioè rappresenta un insieme di elementi collapsible. In fase di definizione del **collapsible-set** è possibile gestire i seguenti attributi:

**data-theme** il Tema (a, b, c, d, e) da assegnare alle varie barre collapsible

**data-content-theme** il Tema (a, b, c, d, e) da assegnare alle aree di espansione, che vengono comunque mostrate con stile ribaltato rispetto a quello base (cioè es da bianco/blu a blu/bianco)  
Il valore di default è "nessun-tema", cioè testo nero su sfondo bianco.

**data-collapsed-icon** Icona di apertura mostrata ad area chiusa, da utilizzare al posto del `+` che rappresenta il default. Es `arrow-r` (freccia right)

**data-expanded-icon** Icona di chiusura mostrata ad area aperta, da utilizzare in alternativa al `-` che rappresenta il default. Es `arrow-d` (freccia down)

**data-iconpos** Posizione dell'icona di apertura / chiusura dell'area. Per default l'icona viene posizionata sulla sinistra. Volendo visualizzarla a destra occorre assegnare il valore **"right"** a questo attributo

## La classe `ui-grid`

In jQuery Mobile sono definite diverse classi di stile che consentono di trasformare un generico tag `div` in uno specifico blocco di contenuto. La classe `ui-grid` trasforma il tag `div` in una griglia che per default occupa il 100% della larghezza, non ha bordi né sfondi, né margini né padding.

```
<div id="grid" class="ui-grid-b">
```

<b>ui-grid</b>	Il tag <code>div</code> assume la forma di una griglia. Lo spazio a disposizione della griglia verrà diviso equamente fra il numero di colonne impostato. <b>ui-grid-a</b> indica una griglia a 2 colonne <b>ui-grid-b</b> indica una griglia a 3 colonne <b>ui-grid-c</b> indica una griglia a 4 colonne <b>ui-grid-d</b> indica una griglia a 5 colonne
<b>ui-block</b>	I tag <code>div</code> con associata questa classe vengono posizionati nella colonna corrispondente: <b>ui-block-a</b> 1° colonna <b>ui-block-b</b> 2° colonna <b>ui-block-c</b> 3° colonna
<b>ui-bar</b>	Questa classe si applica ad un semplice tag di contenuto (normalmente il contenuto della singola cella), da sola o in abbinamento alla precedente, ed aggiunge un padding intorno al testo. <b>ui-bar-a</b> Applica alla cella lo stile di font e sfondi relativi al tema a <b>ui-bar-b</b> Applica alla cella lo stile di font e sfondi relativi al tema b <b>ui-bar-c</b> Applica alla cella lo stile di font e sfondi relativi al tema c

## Pagine esterne

Con questa tecnica di avere tutte le pagine organizzate mediante diversi tag `<div>` appartenenti alla stessa pagina, se un utente necessita del refresh di una certa 'sottopagina', questo andrebbe a forzare il refresh di tutte le 'sottopagine' presenti nello stesso contenitore, con un eccessivo quanto inutile spreco di risorse. Per questo il caricamento delle pagine viene automaticamente eseguito da jQuery in modo **asincrono** tramite richieste Ajax. Cioè al momento del download viene caricata soltanto la sottopagina principale, mentre la pagina 2 verrà richiamata tramite Ajax soltanto al momento della necessità.

Questa modalità può essere disattivata (per pagine relative ad uno stesso dominio) impostando l'attributo `data-ajax = false`.

Se non si modifica questo attributo, se ad un certo punto sorge la necessità di dover richiamare una pagina esterna con una normale richiesta sincrona http, (pagina appartenente allo stesso dominio o a dominio diverso), dopo `href`, all'interno del tag `a`, occorre aggiungere l'attributo `rel = "external"`.

**Metodi comuni applicabili a qualsiasi selettore**

---

```
$("selector").buttonMarkup({theme: "b"});
```

Consente di cambiare il tema dell'oggetto puntato dal selettore.

**Metodi statici della classe \$.mobile**

---

```
$.mobile.changePage( "#pagMokai", { transition: "slideup"});
```

Consente di cambiare pagina. Accetta come parametro sia una pagina esterna, sia un link ad un'ancora interna.