

Come realizzare la view di una HTML Mobile App con tab-bar animata

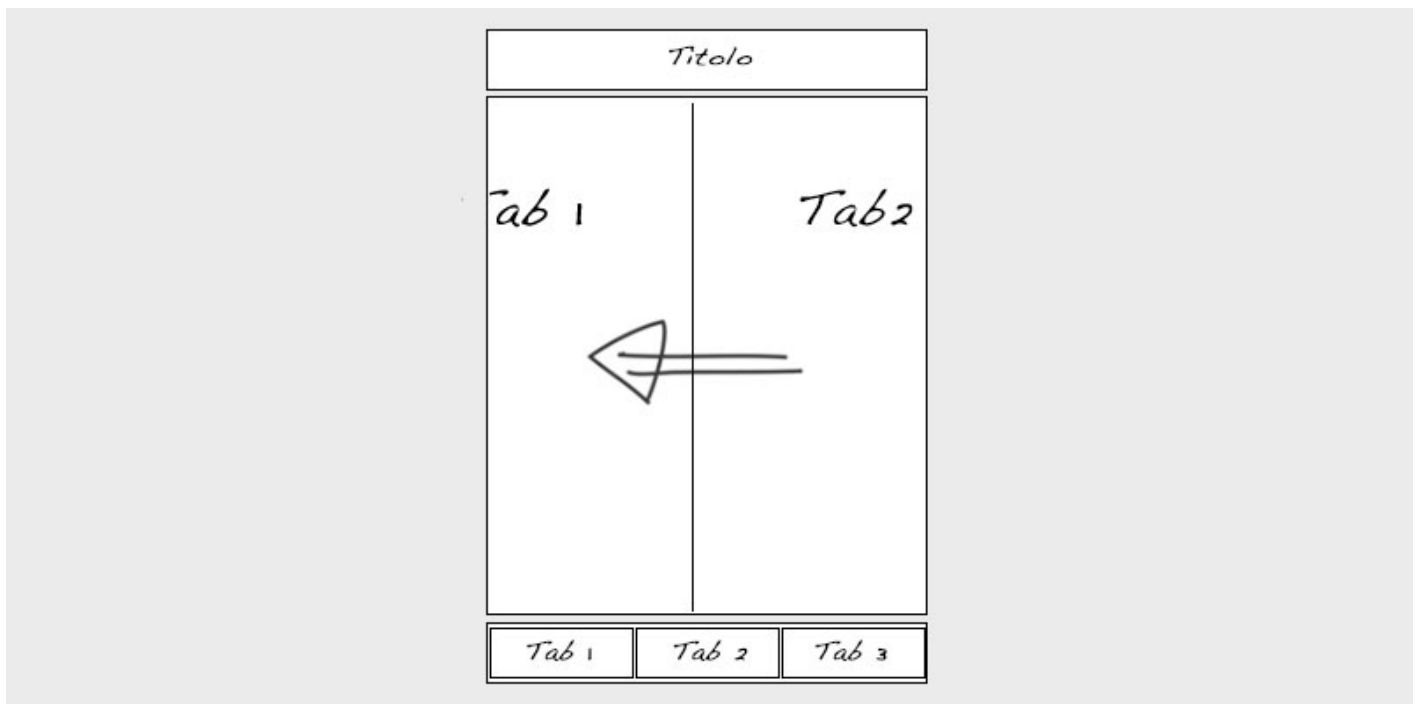
Finora abbiamo parlato di **performance** e **fluidità** di **applicazioni mobile**, in particolare di quali tecniche e consigli adottare per potere **migliorare** le **prestazioni** e limitare in modo considerevole le differenze tra un progetto realizzato in **HTML5**, **CSS3** e **Javascript** e un'app sviluppata **nativamente**.

In questo articolo ti voglio mostrare come usare del **markup HTML5** insieme a **codice CSS3** performante per poter **impostare** una **“classica” view** (quindi una tipica schermata mobile) con una **tab-bar interna**, che richiami un'animazione al momento del passaggio da una scheda ad un'altra.

La struttura della view

L'obiettivo è quello di realizzare una **struttura** che presenti un'**header** all'interno del quale posizioneremo il **titolo** e **due pulsanti**, un **footer** che racchiuda la **tab-bar**, ed una parte **centrale** con le varie **schede** legate ai **pulsanti** della tab-bar.

Per rendere la **user-experience** più accattivante inseriremo un'**animazione** stile **'slide'** nel passaggio tra una scheda e l'altra.



Ovviamente utilizzeremo **animazioni fluide** e performanti, sfruttando anche la tecnica **dell'accelerazione hardware**.

Se sei interessato/a a scoprire come funziona la tecnica dell'accelerazione hardware (o hardware acceleration) e poterla impiegare per realizzare animazioni leggere e che sfruttano la memoria grafica del dispositivo (o GPU), ti consiglio di leggere questo post:

-

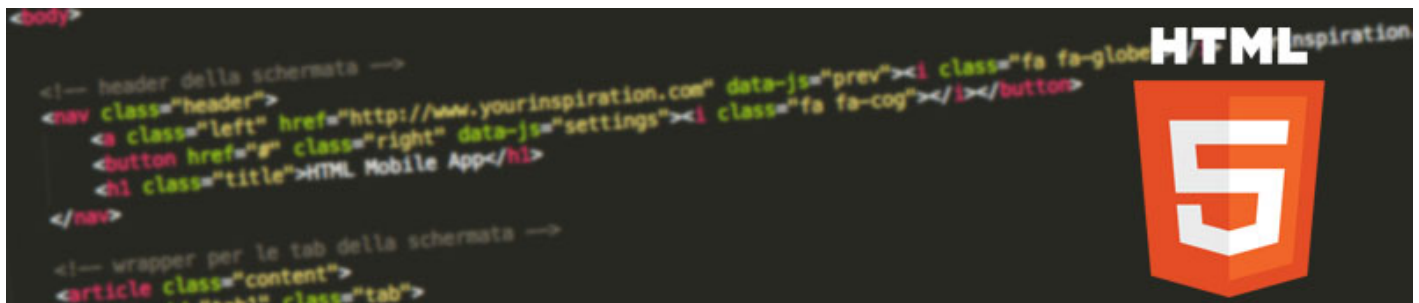
[Mobile App Fluide con Hardware acceleration](#)

Cosa includere nel documento

Per il nostro esperimento includeremo nel documento le seguenti risorse esterne:

- il foglio di stile *'normalize.css'*, per il **reset** su tutti i browser degli elementi HTML
- la libreria [Modernizr](#), utile per ottenere il **nome corretto** dell'evento di fine animazione **CSS**, a seconda del browser usato
- il "solito" **jQuery**, di cui ci serviremo per gestire gli eventi (come il 'click' delle tab) e per la **manipolazione** del **DOM** (essenzialmente aggiungere o rimuovere classi di stile).

Il markup HTML



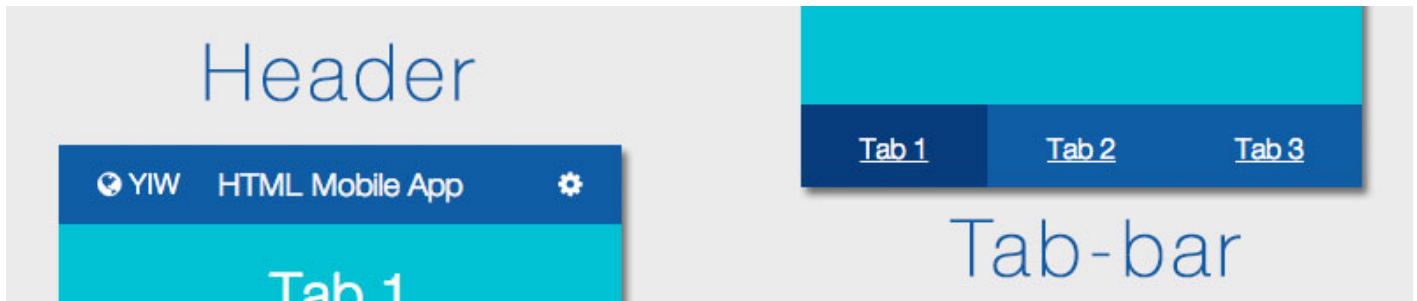
Vediamo subito come impostare gli elementi **DOM** all'interno del nostro documento:

Gli stili CSS3

Per quanto riguarda gli **stili CSS**, come prima cosa facciamo in modo che header e footer abbiano le seguenti caratteristiche:

Posizioniamo quindi il footer in fondo al documento, specificando un tipo di visualizzazione “formato tabella” con la proprietà “*display: table*”.

Vediamo ora come trattare il titolo ed i pulsanti presenti all’interno dell’header:



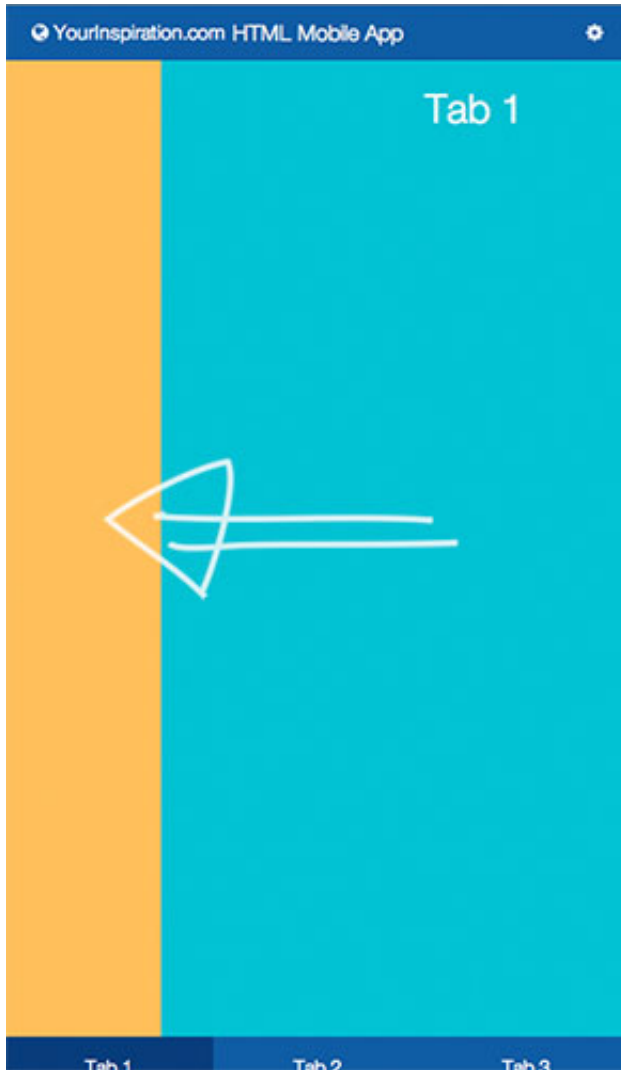
Mentre invece i pulsanti della **tab-bar** avranno i seguenti stili:

Posizioniamo anche il contenuto centrale (tag `<div>` avente **classe di stile** `‘.content’`) della *view* con il valore `‘fixed’` e facciamo in modo che abbia le stesse dimensioni del suo contenitore (nel nostro caso il `<div>`, quindi della finestra stessa), impostando un `padding-top` o `padding-bottom` rispettivamente della stessa altezza di header e footer, nel caso in cui questi siano presenti nel documento allo stesso livello DOM del contenuto.

Vediamo ora gli stili da attribuire alle *tab* dei contenuti.

Ognuna di esse viene posizionata in modo assoluto all’interno del contenitore e, come abbiamo fatto in precedenza, nel caso in cui siano presenti all’interno del documento header o footer, modifichiamo le **proprietà top** o **bottom** di ciascuna *tab* per definire l’area in cui deve essere visualizzato il loro contenuto.

Impostiamo ora gli stili per “preparare” ogni scheda per l’**animazione di entrata** e di uscita, definendo su quali proprietà applicare le trasformazioni, la classe della tab correntemente visualizzata, le classi che “**scatenano**” le animazioni, e dichiarando la dinamica di quest’ultime attraverso l’istruzione “**keyframe**”:



Per approfondire il discorso di transizioni ed animazioni **CSS3** impiegate nello sviluppo di interfacce mobile e fluide e performanti, ti consiglio di leggere questi due articoli:

-

[HTML5 Mobile app: transizioni ed animazioni CSS3: parte 1](#)

- [HTML5 Mobile app: transizioni ed animazioni CSS3: parte 2](#)

La gestione della tab-bar attraverso il Javascript

Una volta definito il **markup** e gli stili necessari per il posizionamento dei vari elementi nel documento, e quelli per le animazioni delle schede della tab-bar, non ci resta che gestire il click dei pulsanti presenti nel footer in modo da innescare le animazioni di passaggio da una scheda all'altra.

Come prima cosa grazie a **Modernizr** riprendiamo il nome dell'evento di *fine-animazione* in base al browser usato dall'utente, che ci servirà per richiamare una funzione quando l'effetto è giunto al termine:

Poi rendiamo attivo il primo pulsante del **footer** e visibile la prima scheda, aggiungendo ad entrambi gli elementi la classe `'active'`:

Il passo successivo sarà quello di gestire l'evento di `'click'` su ogni pulsante del footer:



In altre parole abbiamo agito secondo questa sequenza:

- 1.

aggiungiamo la classe `'active'` al pulsante appena cliccato e la rimuoviamo dagli altri

2.

impostiamo a *'true'* la variabile *'animating'* che ci serve per evitare di innescare un'animazione mentre un'altra è già in corso

3.

aggiungiamo la classe *'active'* anche alla scheda che deve essere mostrata

4.

aggiungiamo alla scheda di uscita e quella di entrata rispettivamente le classi *'exit'* ed *'enter'*, in modo da innescare entrambe le animazioni

5.

attraverso il metodo jQuery *'one()'*, che ci permette di eseguire una funzione per un evento solo la prima volta che questo si verifica, ci assicuriamo che l'ultima delle animazioni di entrata o di uscita a giungere al termine, esegua la funzione *'onEnd'*

6.

Grazie alla funzione *'onEnd'* rimuoviamo le classi *'exit'* ed *'enter'* alle schede appena animate in modo tale che esse siano disponibili per un'ulteriore animazione, rimuoviamo la classe *'active'* dalla scheda appena nascosta, ed impostiamo il valore *'false'* per la variabile di controllo *'animating'*.

Ecco finalmente la nostra *HTML5 Mobile App* di base con tanto di Tab-bar al suo interno; puoi vedere la demo o scaricare i sorgenti direttamente dai seguenti link:

Conclusioni

Utilizzando questa tecnica unita a molte altre ho creato per i miei clienti oltre 10 applicazioni

mobile, tra cui due progetti personali: [Tint Weather App e Dieta SI o NO?](#).

Se anche tu sei un web designer e vuoi realizzare la tua applicazione mobile da distribuire nei vari stores, forse sarai interessato al mio e-book: [HTML Mobile Accelerato](#).

Grazie ad esso verrai a conoscenza di tutte quelle tecniche che renderanno la tua mobile app fluida e reattiva come quelle native.

E tu cosa ne pensi? Hai mai usato queste tecniche per realizzare una applicazione mobile in **HTML5, CSS3 e Javascript?**

Se hai dei suggerimenti e consigli da aggiungere lascia un tuo commento!