

Slider e menù responsive: ecco due elementi da avere nel vostro sito (Parte 1: Il menù)

Sempre più spesso, navigando per il web, rimaniamo a bocca aperta osservando quel classico **template Wordpress** con uno **slider** a tutto schermo e le solite frasi d'effetto che si usano per favorire la cosiddetta conversione. Per non parlare di quei menù che magicamente si adattano a tutte le risoluzioni.

Bene, ora, se stai ridendo per il "magicamente", probabilmente conoscerai il responsive design e nella guida di oggi andremo proprio a vedere come applicarlo a due elementi, ovvero a uno **slider** e a un **menù**, per avere la vostra (quasi) completa landing o splash page.

La guida si dividerà in due parti: la prima, dove andremo a creare il menù, e la seconda, dove andremo a vedere come implementare lo slider.

In questo articolo ci occuperemo solo della prima parte.

Premessa: per il corretto funzionamento del **menù**, è necessario importare nel nostro file di stile il Google Font "**Lato**", utilizzato nell'esempio. L'import fortunatamente è un'operazione molto semplice, basterà inserire all'inizio del vostro foglio di stile la seguente dichiarazione:

Non basta però. Per far apparire l'icona dello **span toggle**, dentro i **tag head** del vostro **foglio HTML** dovrete inserire la dichiarazione per implementare **FontAwesome**, ovvero:

Impostiamo la struttura del menù

Iniziamo ora a costituire quello che sarà il menù responsive della nostra pagina, utilizzando il seguente **codice HTML**:

Non c'è menù senza **logo**: è proprio per questo che sono andato a racchiudere i due elementi in uno comune, chiamato **header**, che, come suggerisce il nome, rappresenterà la testata della pagina.

Ma la cosa che potrebbe suonare fuori posto è lo **span** di **classe toggle**.

"A cosa serve?" direte voi? È presto detto. Laddove la risoluzione fosse inferiore o uguale a 768px, lo span permetterà al menù, che nel frattempo avrà cambiato forma, di apparire cliccando sul suddetto, grazie ad un toggle creato in **jQuery**.

Arrivati a questo punto, starete sicuramente pensando che stia anticipando troppo le cose, non è vero? Per schiarirci le idee andiamo a scoprire il nuovo punto, ovvero il **codice CSS relativo** al menù, che troverete qui di seguito:

Tralasciando la parte che inizia da “**span**” fino a “**span:hover::before**” (che serve solo per avere un effetto molto carino all’hover delle **voci del menù**, e fidatevi di me se vi dico che si fa prima a mostrarlo che a spiegarlo), andiamo ad analizzare il codice sopra proposto.

Partiamo dal toggle. Visto che, nella parte che vi ho appena detto di ignorare, allo span abbiamo attribuito sia un **border** che un **padding**, andremo subito a rimuovere questi attributi dal nostro **bottone**, grazie alla dichiarazione della classe **.toggle** assieme al corrispettivo **.toggle:hover**.

Al tag invece, andremo ad attribuirgli un **float: right** al fine di posizionare il nostro menù sulla destra dello schermo, in antitesi al logo che al contrario sarà posizionato all’estrema sinistra.

Infine, una volta fatto ciò, posizioniamo i nostri elementi lì sulla stessa riga grazie all’utilizzo di **float: left** e personalizziamo in base ai nostri gusti l’elemento **nav a**.

Se vi state chiedendo se in questo tutorial ci sia qualcosa che non sapevate già, non temete, perché ora arriva la parte davvero interessante: l’utilizzo delle **media queries**.

Davvero mai utilizzate? In breve, queste ultime hanno la capacità di **eseguire il detect** della **larghezza** della **finestra** del **browser**, in modo da potervi dare risposte dinamiche al **resize** della stessa sulla base dei vari **breakpoint** (un **breakpoint** è appunto un “**punto di rottura**”; nel codice che vediamo sopra sono rappresentati dalle varie **min -width** e **max-width**) precedentemente fissati.

Nel nostro caso, abbiamo sfruttato questa funzionalità principalmente per tre cose:

- Modificare in base alla larghezza della pagina la corrispettiva larghezza del contenitore del menù, di default impostata a 1170px;
- Modificare l’aspetto del **menù**, dandogli una **posizione assoluta** (con larghezza 100%), e ordinando le sue voci in modo da apparire una sotto l’altra, rimuovendo il **float: left**;
- Mostrare il **toggle** e nascondere il menù o viceversa.

Eccovi mostrato il risultato di ciò che abbiamo fatto fin ora, ovvero:

- il menù in versione desktop

YOUR LOGO

HOME SERVIZI CHI SIAMO CONTATTACI

- il menù in versione mobile

YOUR LOGO



...e facciamolo funzionare!

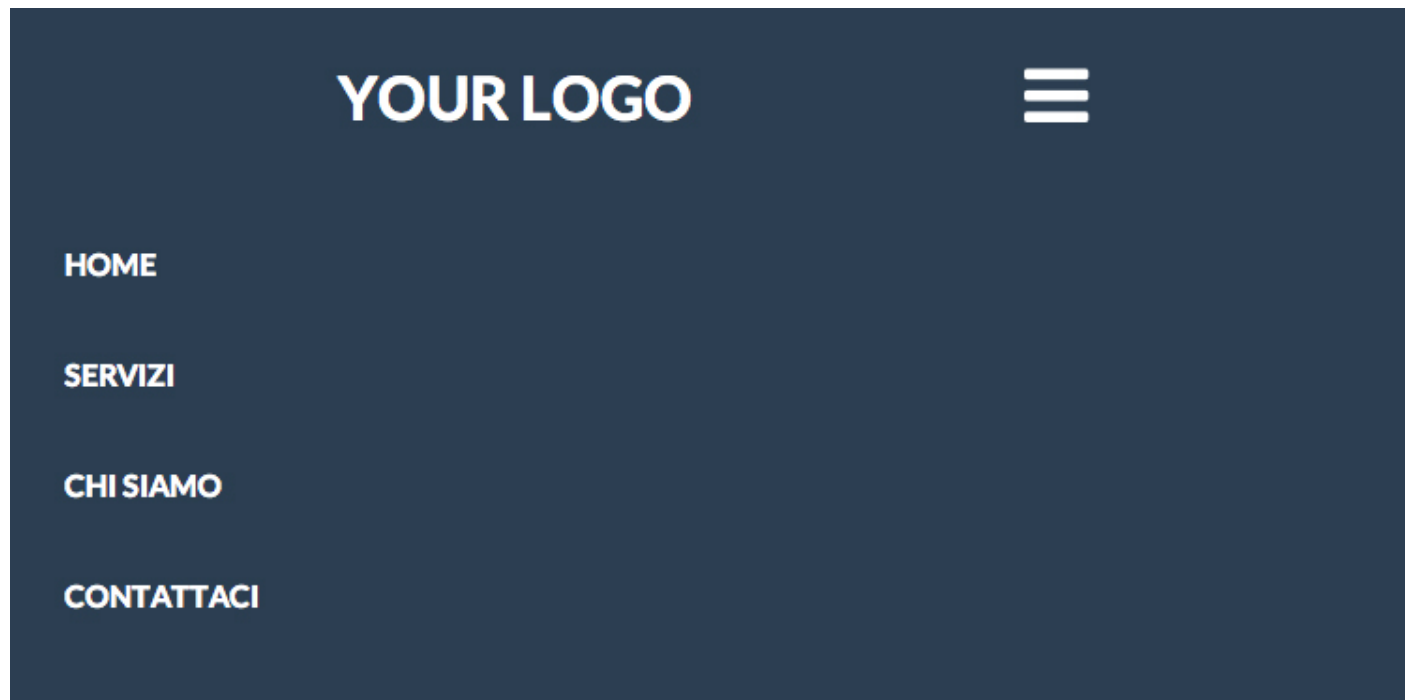
Ora tocca al codice **jQuery**, semplicissimo come sempre, che permetterà al nostro menù di fare il suo dovere anche sui dispositivi mobile.

Potete vedere quest'ultimo proprio qui sotto:

Ora è arrivato il momento di introdurre un **concetto essenziale**: il **toggle**. Ne abbiamo parlato tanto sopra, ma chi non l'ha mai usato effettivamente potrebbe non sapere cosa sia. Sostanzialmente, è un semplice **"bottonone"** che a ogni click su di esso interagisce con un determinato elemento in base al suo stato attuale.

Nel nostro caso, abbiamo usato un gradevole effetto di **slideToggle** per mostrare al **click** del **mouse** (o alla pressione del dito su uno **smartphone**) il nostro menù o nascondere a seconda dei casi.

Per finire in bellezza, eccovi l'immagine del menù **"aperto"**:



Conclusioni

Siamo arrivati alla fine anche di questo articolo. In un mondo che va sempre più verso il **responsive design**, con oltre il 35% di accessi ad internet da parte di dispositivi mobile, creare un menù che si adatti ad ogni situazione (e lo dico visto che il menù è una delle parti fondamentali di un sito) è sempre un **must**.

Come sempre, fatemi sapere nei commenti se il tutorial è stato di vostro gradimento e se riscontrate problemi.

L'esempio completo lo trovate in [questa pagina](#) (per vedere il menù responsive occorrerà effettuare un resize della finestra sotto i 768px), mentre potrete scaricare tutto il codice sorgente da [questo file](#).