jQuery e Wordpress: farli cooperare una volta per tutte

Come probabilmente ben saprete - e mi rivolgo ai lettori che già hanno avuto esperienza con il suddetto **CMS** - Wordpress ha una gestione davvero difficile per quanto riguarda l**'inclusione degli script** .

In questa guida andremo a vedere **come implementare uno script jQuery** su **Wordpress**, partendo dalle pratiche più semplici, che però potrebbero riservare qualche brutta sorpresa, fino ad arrivare a quelle maggiormente complesse ma più funzionali.

1. Inclusione dello script dentro i tag head

La prima cosa che salta in mente e che viene naturale provare a mettere in pratica, è quella di aggiungere lo script in questione nel file ***header.php***, subito prima la chiamata della funzione ***“wp_head()”***, con una semplice inclusione del tipo:

dove ***“script.js”*** sarà il nostro script, che si troverà dentro la cartella che abbiamo chiamato ***“js”***. La funzione ***get_stylesheet_directory_uri()***, invece, servirà a richiamare l’**url** della cartella dove si trova il nostro tema.

Vi assicuro che è un gran risparmio di fatica, difatti, senza l’utilizzo della suddetta funzione, si sarebbe dovuto scrivere qualcosa come:

Sembra facile, vero? In realtà, questo primo metodo, nonostante sia molto semplice e possa apparire anche efficace, molto probabilmente risulterà non funzionante nell’80% dei casi, motivo per cui è bene imparare a utilizzare il metodo di lavoro spiegato nel prossimo paragrafo, tra l’altro previsto anche dalla documentazione ufficiale di Wordpress.

2. Registrazione dello script nel file function.php

Premesso che oggi non parleremo di come creare un file function.php, andiamo a vedere il codice PHP necessario per far riconoscere lo script a Wordpress.

Il file ***function.php,*** come suggerisce il nome, permette di aggiungere funzioni molto potenti al nostro tema, come ad esempio la **registrazione** (ovvero l’***implementazione*** dentro i ***tag head***) di file di stile e di script javascript.

Ricordandoci che naturalmente **jQuery** è un framework JS, e quindi i metodi di registrazione sono i medesimi, scopriamo come usare le funzioni ***wp_register_script e wp_enqueue_script*** per

inserire i nostri script jQuery, analizzando il codice che segue:

Innanzitutto, come possiamo vedere, si crea una nuova funzione, in questo caso denominata "**aggiungi_script**", all'interno della quale si troverà il nostro codice.

Successivamente, aggiungiamo la condizione **if !is_admin()**, che ha lo scopo di non far eseguire lo script se ci troviamo nel **backend** (**area di amministrazione**) di Wordpress, onde evitare eventuali problemi aggiuntivi.

Infine, proprio dentro la condizione **if**, aggiungiamo due funzioni:

1) **wp_register_script()**, con all'interno diversi parametri:

- il **nome** dello script, nel nostro caso "script";
- la **posizione** dello script;
- la **libreria** da cui dipende lo script, nel nostro caso ovviamente sarà jQuery;
- la **versione** dello script;
- un valore **boolean**, che se impostato su false caricherà lo script all'interno dell'header; se, al contrario, il valore sarà "**true**", lo script sarà incluso nel footer;

2) **wp_enqueue_script()**, dove sarà specificato solo il nome dello script da "**mettere in coda**" ("**enqueue**" significa proprio questo) all'interno dell'head.

Questa due funzioni, come già detto, avranno il compito di **registrare** lo script, anche se per farlo funzionare, e quindi includerlo dentro la pagina html, come si può notare, servirà la funzione **add_action()** con due parametri: **l'azione da eseguire e la funzione correlata.**

3. Usare un namespace per il simbolo $

Se siete arrivati fino a questo punto, e lo script risulta di fatto implementato anche se non ancora funzionante, allora quasi sicuramente il vostro problema può dipendere dal simbolo del dollaro, o meglio **$**, che istanzia qualsiasi oggetto, fungendo da alias per la dicitura "**jQuery**".

Nonostante sia un simbolo molto comodo, spesso può creare problemi se si usano altri **framework** quali **Prototype** o simili.

Usiamo perciò un metodo molto comodo, che tra l'altro non prevede nemmeno l'utilizzo del ***noConflict*** (altro argomento che esula da questo tutorial): stiamo parlando dell'assegnazione di un ***namespace*** a **$**.

L'operazione, anche per chi non l'avesse mai fatta, è davvero basilare, difatti il codice per lo script

sarà il seguente:

Da questo momento, ogni alias **$** sarà cambiato in “**jQuery**”, per assicurarvi l’assenza di conflitti.

Esempio

Immaginiamo ora di voler centrare una pagina di login dove centrare due elementi: un **logo**, di classe **.logo**, ed un **form** contenente **diversi elementi input**, di **classe *.center***.

Il nostro obiettivo sarà quello di centrare il logo orizzontalmente, mentre il form, sia orizzontalmente che verticalmente.

Ovviamente, potremmo provare con un po’ di codice jQuery, ovvero:

Proviamo quindi con il punto 1, e se tutto andrà come previsto, lo script risulterà non funzionante come si può osservare con l’immagine di seguito:

Al contrario, seguendo i punti 2 e 3, il risultato sarà certamente quello desiderato, rappresentato dall’immagine sottostante:

Il tutorial termina qui.

Come sempre, se avete dubbi o per caso il vostro codice dovesse non ancora funzionare, chiedete pure nei commenti.