

## Come inserire correttamente scripts e styles in WordPress

*sabato, 05 agosto 2017*

In molti casi potresti avere l'esigenza di modificare l'aspetto grafico del tuo sito creando regole CSS personalizzate, oppure magari di utilizzare funzioni Javascript personalizzate.

Starai pensando: "Beh, semplice, basta inserire gli appositi tag di richiamo nell'header o nel footer!". Sicuramente è così, ma con WordPress ci sono metodi migliori per farlo.



### Sommario

Come lo farebbe un principiante

L'accodamento

Come lo farebbe un esperto WordPress

Le funzioni nel dettaglio

## Come lo farebbe un principiante

Una delle soluzioni che ho visto spesso adottare, in modo non del tutto corretto a mio parere, è quella di inserire semplicemente nel file header.php nella sezione il richiamo a tali files, ad esempio:

```
<script type="text/javascript"
src="wp-content/themes/{nome_del_tema}/js/custom.js"></script>
```

oppure:

```
<link rel="stylesheet" type="text/css"
href="wp-content/themes/{nome_del_tema}/css/custom.css" />
```

Ma WordPress mette a disposizione alcune funzioni che ti permettono di ottenere lo stesso risultato con notevoli vantaggi.

## L'accodamento

Rifletti per un attimo all'utilizzo dei plugins: molti di questi fanno uso di funzioni basate ad esempio su jQuery, ma spesso ne utilizzano versioni differenti. Se ogni plugin includesse i suoi scripts singolarmente, sarebbe il caos ed i conflitti di funzioni abbonderebbero.

Il sistema adottato da WordPress è il cosiddetto **Enqueuing** (accodamento), che fa da "controllore" e, per dirlo in parole semplici, verifica che scripts e styles inclusi nel tuo sito vengano collocati al proprio posto rispettando l'ordine di esecuzione.

L'enqueuing offre diversi vantaggi rispetto all'inclusione "manuale", primo tra tutti il rispetto delle dipendenze: specificando se uno script o un foglio di stile dipendono da un altro, WordPress si occuperà di controllare per noi l'effettiva presenza e l'inclusione dei files nell'ordine corretto.

## Come lo farebbe un esperto WordPress

Vediamo quindi il modo corretto di includere scripts Javascript/jQuery e fogli di stile CSS nel tuo sito. WordPress mette a disposizione un hook, `wp_enqueue_scripts` su cui "agganciare" scripts e fogli di stile da accodare, specificandone dipendenze e posizione.

Riprendendo in esame il caso precedente, per ottenere lo stesso risultato utilizzando l'enqueuing, ti basta inserire nel file `functions.php` del tuo tema, che trovi nello spazio FTP del tuo sito, in:

```
wp-content/themes/{nome_del_tema}/
```

il codice:

```
// Aggiunge files js e css
add_action( 'wp_enqueue_scripts', function() {
    wp_register_style( 'custom-style', get_template_directory_uri() . '/custom.css' );
    wp_enqueue_style( 'custom-style' );

    wp_register_script( 'custom-script', get_template_directory_uri() . '/custom.js',
array( 'jquery' ), false, true );
    wp_enqueue_script( 'theme' );
} );
```

## Le funzioni nel dettaglio

Ti starai chiedendo: ma cosa vuol dire quello che ho appena inserito?

La funzione **wp\_register\_style( \$handle, \$src, \$deps, \$ver, \$media )**, si occupa di registrare il foglio di stile, definendone i parametri fondamentali:

- **\$handle**: il nome univoco con cui WordPress riconosce e richiama il foglio di stile;
- **\$src**: il percorso in cui è presente il file. Abbiamo utilizzato la funzione `get_template_directory_uri()` per ottenere il percorso del tema in uso;
- **\$deps**: un array contenente i nomi univoci (*handle*) dei fogli di stile da cui dipende quello che stai inserendo, così WordPress si assicurerà di caricare i files nell'ordine corretto;
- **\$ver**: l'eventuale versione del foglio di stile, per assicurarsi che la corretta versione del file sia utilizzata indipendentemente dalla cache;
- **\$media**: specifica il media per cui il foglio di stile è stato definito (es: all, screen, print, ecc.)

mentre la funzione **wp\_enqueue\_style( \$handle )** inserisce il foglio di stile appena registrato nell'elenco dell'enqueuing, pronto per essere incluso.

Allo stesso modo funziona l'omologa funzione **wp\_register\_script( \$handle, \$src, \$deps, \$ver, \$in\_footer )**, che varia solamente per l'ultimo parametro:

- **\$in\_footer**: specifica se il file script deve essere visualizzato nell'header (*false*) o nel footer (*true*);