

WordPress: Diamo Uno Sguardo al Loop

Nei precedenti articoli di questa guida siamo riusciti - partendo completamente da zero - a **sviluppare la parte superiore (l'header) del nostro sito/blog**. Con la lezione di oggi entriamo nel cuore del meccanismo di WordPress, quello che in gergo tecnico è definito il "loop".

Che cos'è questo loop e a cosa serve? Questa sicuramente è una delle tante domande che affiorano nella mente di chi si trova per la prima volta a dissezionare un tema di WordPress nel tentativo di personalizzarlo a proprio piacimento. Vediamo dunque di rispondere a questo interrogativo e di rendere tutto più chiaro.

Indice dei contenuti

1. [Che cosa è il loop e come si utilizza?](#)
 1. [L'avvio del loop](#)
 2. [Generazione degli articoli](#)
 3. [Mostrare il titolo, la data e l'autore](#)
 4. [Mostrare il contenuto dell'articolo](#)
 5. [Mostrare ulteriori dettagli aggiuntivi](#)
2. [Aggiungere la paginazione alla fine del loop](#)
3. [Aggiungere un testo da visualizzare nel caso non ci siano articoli](#)
4. [Conclusione](#)

1. Che cosa è il loop e come si utilizza?

Loop letteralmente significa "esecuzione ripetuta" di "qualcosa"; nel nostro caso si tratta **dell'esecuzione ripetuta "n" volte di un set di istruzioni necessarie alla visualizzazione di un articolo**. Ripetendo queste istruzioni "n" volte saranno visualizzati "n" articoli nel punto in cui è richiamato il loop.

Tornando al tema che stiamo realizzando, supponiamo di volerlo dividere in due colonne subito sotto l'header. Nella colonna sinistra visualizzeremo l'elenco degli ultimi articoli pubblicati mentre nella colonna destra mostreremo una sidebar su cui andremo a inserire (in una delle prossime lezioni) alcune informazioni utili che vogliamo rendere visibili su tutte le pagine del sito/blog. Insomma stiamo definendo il classico layout di un blog.

Ti ricordo che nulla ci vieta di definire in maniera completamente diversa le sezioni: il modo in cui questi contenuti saranno presentati sullo schermo andrà poi definito nel nostro foglio di stile.

Realizziamo dunque il markup HTML necessario per visualizzare la colonna sinistra che dovrà contenere i nostri articoli:

Questa invece la regola da definire nel nostro foglio di stile per visualizzare la sezione *content* a sinistra del nostro layout con una dimensione di 650 pixel (nota: per renderla ancora più visibile provvisoriamente ho aggiunto il solito bordo rosso di un pixel).

Piccola premessa prima di analizzare più nel dettaglio il markup che abbiamo realizzato. Come puoi notare, per processare il loop è stato utilizzato qualche costrutto tipico del linguaggio php, ecco perché per una buona comprensione è consigliabile avere un minimo di nozioni per lo meno sul significato dei costrutti principali di questo linguaggio.

1.1 L'avvio del loop

Ora passiamo ad analizzare più nel dettaglio il markup che abbiamo scritto:

1. Come prima cosa si definisce la sezione che delimita la colonna sinistra (*content*).
2. In seguito eseguiamo quello che nel linguaggio di programmazione è definito un **controllo condizionale**, mediante il costrutto “*if*” del linguaggio *php* verifichiamo la funzione *have_posts()* di WordPress per controllare se esistono post; infatti la funzione restituirà il valore “*true*” (cioè *vero*) nel caso in cui siano presenti post.
3. Se ci sono post è avviato un ciclo *while* (il famoso *loop* di cui si parla) il quale continua a essere eseguito finché la funzione *have_posts()* continua a restituire un valore *true*, cioè fino a quando ci sono post da mostrare.
4. Quando la funzione *have_posts()* restituisce un valore *false* - cioè nel momento in cui non ci sono più post da mostrare - il ciclo viene interrotto.

1.2 Generazione degli articoli

La funzione *the_post()* prende l'elemento (l'articolo/post) corrente del ciclo e lo rende disponibile per l'uso all'interno di questa iterazione. Senza questa funzione molti dei tag utilizzati per stampare le informazioni riguardanti i post non funzionerebbero.

Una volta che le **informazioni relative ai post** sono rese disponibili, il template può iniziare a mostrare i dati sulla pagina per renderli visibili agli utenti. Per il momento, avendo inserito solo la stringa “*contenuto post*” e non avendo utilizzato ancora nessun tag per la visualizzazione delle informazioni relative ai post, la nostra pagina si presenterà come mostrato in Figura 1.

1.3 Mostrare il titolo, la data e l'autore

Adesso proviamo a sostituire la scritta statica che abbiamo inserito all'interno del loop (...*contenuto post*...) con alcuni tag che ci restituiranno le informazioni inerenti i post che vogliamo

mostrare.

Per visualizzare il titolo del post basta utilizzare la funzione [`the_title\(\)`](#) come mostrato sotto:

Se adesso andiamo a vedere la pagina del nostro blog dovremmo vedere il titolo del post così come mostrato in Figura 2.

Se desideriamo rendere il titolo del post linkabile in modo da collegarlo al contenuto dell'articolo, possiamo utilizzare la funzione [`the_permalink\(\)`](#):

Adesso il titolo del nostro post è diventato un collegamento ipertestuale che punta al contenuto dell'articolo.

Che ne dici, mostriamo anche la data di pubblicazione e il nome dell'autore dell'articolo?

Subito sotto il titolo, aggiungiamo queste due righe richiamando le funzioni messe a disposizione da WordPress [`the_time\(\)`](#) e [`the_author\(\)`](#) per mostrare rispettivamente data di pubblicazione e autore dell'articolo:

Definiamo la classe data nel nostro foglio di stile per formattare la presentazione di questi contenuti:

A questo punto la nostra pagina dovrebbe mostrarsi così come visualizzato in Figura 3.

1.4 Mostrare il contenuto dell'articolo

La funzione [`the_content\(\)`](#) serve a mostrare il contenuto dell'articolo. C'è una cosa però a cui è necessario prestare attenzione: la funzione in questione visualizza nel documento il contenuto dell'articolo fino al tag `<!--`, nel caso in cui questo fosse stato usato durante la scrittura dell'articolo.

Per quanti non lo sapessero, il tag `<!--` permette di definire una porzione di articolo da visualizzare come eventuale incipit/introduzione dello stesso (Figura 4). Così facendo la parte dell'articolo inserita subito dopo questo tag verrà mostrata successivamente, quando - e se - l'utente deciderà di leggerne l'intero contenuto cliccato sul relativo collegamento ipertestuale.

Aggiungendo sul nostro template la funzione `the_content()` subito dopo la data e l'autore, nella nostra home page visualizzeremo il contenuto dell'articolo fino al tag `more`, con un testo (passato come parametro alla funzione) che invita a proseguire alla lettura e che collega in modo automatico all'articolo.

Andiamo a vedere il contenuto della nostra home page che comincia a prendere forma (Figura 5).

1.5 Mostrare altri dettagli aggiuntivi

Vogliamo aggiungere qualche altro dettaglio riguardante i nostri articoli?

Magari subito dopo l'incipit ci farebbe comodo visualizzare le categorie cui l'articolo è stato assegnato o il numero di commenti ricevuti. Mediante l'utilizzo delle funzioni [`the_category\(\)`](#) e [`comments_popup_link\(\)`](#) possiamo soddisfare le nostre esigenze:

Alla funzione `the_category()` abbiamo passato come parametro una virgola (,); nel caso in cui all'articolo fossero state assegnate più categorie questo sarà il carattere separatore utilizzato per dividere sullo schermo le varie categorie l'una dall'altra.

Alla funzione `comments_popup_link()` sono stati passati tre parametri:

1. il primo rappresenta il testo da visualizzare nel caso in cui l'articolo non avesse nessun commento;
2. il secondo rappresenta il testo da visualizzare nel caso in cui l'articolo avesse un unico commento;
3. il terzo ed ultimo parametro rappresenta il testo da mostrare nel caso in cui l'articolo avesse più di un commento.

Aggiungiamo qualche altra regola al nostro foglio di stile in modo da definire la formattazione degli altri elementi che stiamo utilizzando per mostrare gli articoli del nostro blog:

Nella Figura 6 puoi vedere un'anteprima di come si presenta adesso la nostra home page.

Per quanto riguarda la visualizzazione degli articoli possiamo dire che ci siamo. Adesso diamo gli ultimi ritocchi alla pagina prima di terminare con la lezione odierna.

2. Aggiungere la paginazione alla fine del loop

Per impostazione predefinita WordPress visualizza gli ultimi dieci articoli sulla home page del blog, questa impostazione può comunque essere facilmente modificata dal pannello di amministrazione (*Impostazioni* -> *Lettura*) indicando il numero di articoli che desideriamo visualizzare. Nel caso in cui nell'home page del blog sono presenti più articoli di quelli indicati, questi vengono suddivisi in più pagine in modo automatico direttamente da WordPress.

A questo punto è indispensabile visualizzare una paginazione alla fine del loop che ci fornisca la possibilità di navigare da una pagina all'altra qualora questo fosse necessario. A tal proposito WordPress mette a disposizione delle [funzioni di navigazione per le paginazioni dei post](#).

Aggiungiamo queste righe di codice subito dopo la fine del loop:

3. Aggiungere un testo da visualizzare nel caso non ci siano articoli

Dobbiamo prevedere anche il caso in cui, nel nostro blog, non sia presente nessun articolo. All'inizio di questa lezione abbiamo detto che prima dell'avvio del loop verifichiamo se sono presenti degli articoli mediante la funzione `have_posts()`:

In caso affermativo si procede all'avvio del loop e alla visualizzazione degli articoli, ma nel caso la verifica restituisse un valore negativo?

Per ovviare a questa particolare condizione, aggiungiamo prima della chiusura del costrutto `if` (`endif`), un messaggio alternativo da mostrare nel caso in cui la condizione iniziale non fosse vera:

4. Conclusione

Fino a questo momento sul nostro blog abbiamo un unico articolo creato da WordPress in modo automatico in fase d'installazione. Prima di concludere proviamo ad aggiungere un nuovo articolo (Figura 7) di prova per vedere come questo sarà visualizzato sul template che abbiamo finora realizzato (Figura 8).

Ho tolto il bordo rosso che avevo utilizzato inizialmente esclusivamente per fini didattici e adesso il nostro template, come puoi vedere, inizia a prendere forma.

Per aiutarti ulteriormente nella comprensione di queste lezioni ti metto a disposizione i file del tema che abbiamo finora sviluppato.

Oggi abbiamo affrontato la parte più complessa della realizzazione di un tema per WordPress, il loop. Se sei riuscito a comprenderne il meccanismo senza grandi difficoltà WordPress non avrà più segreti per te, potrà essere sezionato e ricostruito in qualsiasi momento secondo le tue effettive esigenze.

Il mio consiglio comunque è sempre quello di consultare l'ottima documentazione sul sito ufficiale per approfondire tutte le funzioni affrontate in questa lezione in modo da comprenderne appieno le potenzialità e sfruttarle dunque nel modo più idoneo alle nostre esigenze. Nella prossima lezione andremo a definire la sidebar del nostro blog, la colonna laterale che generalmente si ripete su tutte le pagine del sito. Sei pronto?

Indice degli articoli trattati in questa guida

Per agevolarti la consultazione della guida alla fine di ogni articolo riporterò l'indice comprendente tutti gli argomenti trattati.

1. [Come Installare WordPress in Locale?](#)
2. [WordPress: Diamo Uno Sguardo al Pannello di Amministrazione](#)
3. [Come Creare un Template per WordPress Partendo da Zero?](#)
4. [WordPress: Come Realizzare l'Header del Template? \(Parte 1\) | \(Parte 2\)](#)
5. [Diamo Uno Sguardo al Loop di WordPress](#)
6. [Aggiungiamo la Sidebar al nostro tema](#)
7. [WordPress: come rendere dinamici Header e Sidebar del template?](#)
8. [Gerarchia dei template e loro utilizzo: come creare un template per gli articoli?](#)
9. [Come Creare Template Personalizzati per le Pagine di WordPress?](#)
10. [WordPress: predisporre la sidebar per l'uso di widget](#)
11. [Personalizzare il Template dei Commenti di WordPress \(Parte 1\) | \(Parte 2\)](#)