

## Come creare un widget in WordPress 2.8?



Come avevo già accennato in [questo articolo](#), la nuova versione di **WordPress, la 2.8**, ha notevolmente migliorato la **gestione dei widgets**. Oggi vedremo come è molto più facile **generare un proprio widget** personalizzato utilizzando la nuova classe messa a disposizione dall'ultima versione di WordPress.

Quello che realizzeremo è qualcosa di molto semplice, ma vedrai anche come è possibile fornire agli utenti una **gestione avanzata del widget** tramite la personalizzazione di alcuni suoi aspetti.

**Il widget** che svilupperemo visualizzerà sia l'iscrizione ai Feed del tuo blog tramite FeedBurner che il contatore degli iscritti ai Feed. Tramite i controlli avanzati permetteremo agli utenti un maggiore controllo sulla **visualizzazione del widget** che gli consentirà di gestirne la possibilità di visualizzare o meno l'iscrizione ai Feed, il contatore degli iscritti e di configurarne le relative impostazioni.

Nella foto seguente puoi vedere un'anteprima dell'aspetto finale che avrà il widget:



### Caricare il widget

WordPress fornisce il gancio "widgets\_init" che permette di caricare il tuo widget quando necessario.

### Registrare il widget

Fatto questo non resta che **registrare il widget** all'interno della funzione `my_widget_subscribe_feed()` che hai agganciato nel passaggio precedente.

La **registrazione del widget** si effettua tramite la funzione di WordPress `register_widget('nome_widget')` come nell'esempio seguente:

## Impostare il widget

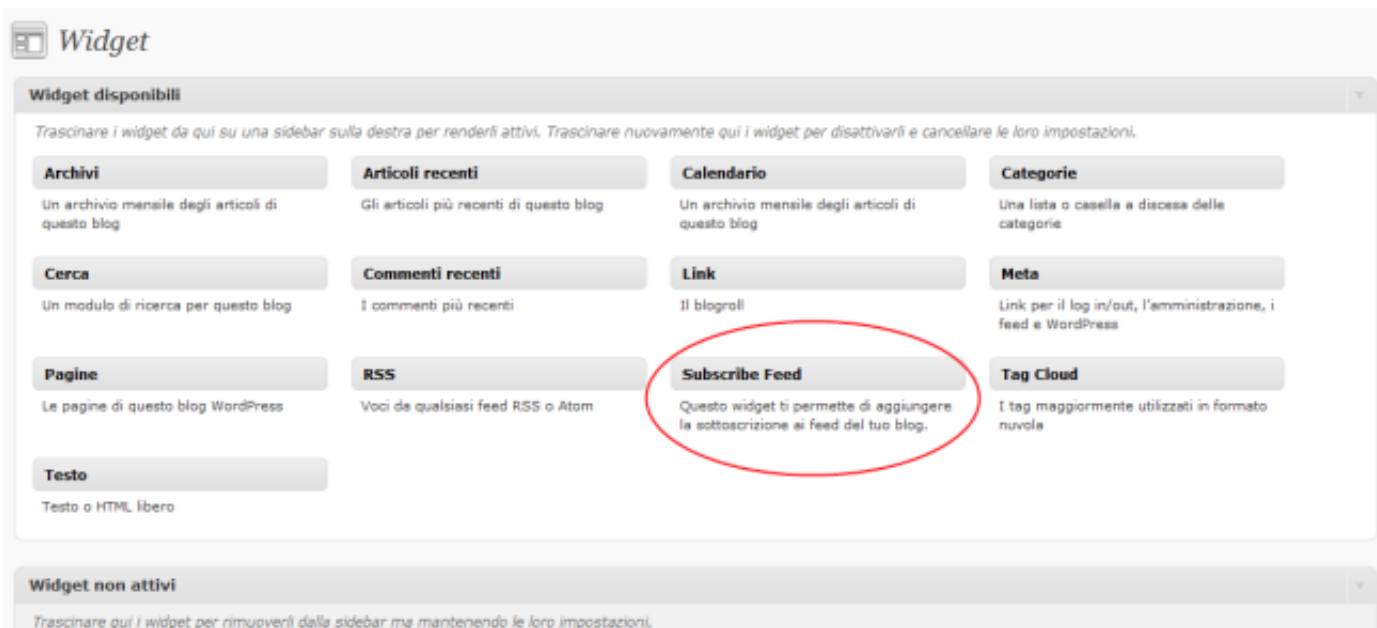
Per **impostare il widget** con la nuova versione di WordPress basta estendere la classe preesistente chiamata **WP\_Widget**. Come tutti i buoni linguaggi ad oggetti, anche PHP permette di estendere una classe tramite altre classi. Do per scontato che si sappia il significato dell'ereditarietà tra classi. Spiegare questo concetto richiederebbe molto tempo, ed esula pertanto dallo scopo di questo articolo.

Quindi il prossimo passaggio sarà quello di creare una nuova classe estendendo la classe già esistente `WP_Widget` nel modo seguente:

Adesso si imposta la prima funzione che servirà a rendere unico il widget e permetterà di crearne le sue impostazioni. Nota che il nome della classe e il nome della funzione devono essere uguali.

La breve **descrizione del widget** visualizzata sul pannello di amministrazione di WordPress viene definita assegnandola all'array con chiave "*description*" alla riga numero tre, mentre il titolo viene assegnato nel momento in cui si crea il widget alla riga numero nove.

Sotto un'immagine che visualizza il widget nel pannello di amministrazione di WordPress con il titolo e la breve descrizione che gli abbiamo assegnato:



## Visualizzare il widget

La prossima funzione che andrai a creare all'interno della classe principale *Subscribe\_Feed* ti permetterà di **visualizzare il contenuto del widget** nel punto in cui questo verrà aggiunto sul tuo Blog dal Pannello di Amministrazione di WordPress. Questa in pratica è la funzione che stamperà a monitor il widget.

Non avendo ancora inserito i controlli del widget per il momento questa porzione di codice ti potrebbe sembrare poco chiara, ma non avere timore, una volta chiarito anche il funzionamento dei controlli avanzati del tuo widget tutto sarà molto più semplice.

E' importante utilizzare le variabili *\$before\_widget*, *\$after\_widget*, *\$before\_title* e *\$after\_title*. Queste variabili sono fornite dal tema, il modo in cui viene **visualizzato il widget** dovrebbe sempre essere gestito dal tema.

## Modificare e salvare le impostazioni del widget

Adesso si crea una funzione che ti permette di salvare le modifiche effettuate dall'utente. La funzione non fa altro che prendere i nuovi valori passati dall'utente e salvarli.

## Creare la gestione avanzata del widget

In WordPress 2.8, con la **nuova classe dei widget**, è molto più facile crearne i **controlli avanzati** per la personalizzazione di alcuni aspetti.

Le funzioni `get_field_id ()` e `get_field_name ()` servono rispettivamente ad assegnare un "id" e un "nome" in modo del tutto automatico ai vari campi che compongono il form e gestiscono da sole la maggior parte del lavoro sporco cosicché ci si possa concentrare sulle cose più importanti.

Per prima cosa potresti creare alcune impostazioni predefinite da utilizzare nel caso in cui l'utente non dovesse selezionare nulla nella gestione avanzata del widget:

Dopo di che passerai alla costruzione del form che conterrà i campi che permetteranno la gestione avanzata del widget:

Nella prima parte del modulo permetti all'utente di inserire il Titolo del Widget da visualizzare nel punto in cui sarà posizionato.

Nella seconda parte consenti all'utente di selezionare tramite un checkbox se visualizzare o meno l'iscrizione ai feed e di inserirne l'URL.

Nella terza ed ultima parte del modulo tramite un checkbox permetti all'utente di scegliere se visualizzare o meno il contatore dei Feed e di configurarne il relativo codice.

Nell'immagine che segue puoi vedere un'anteprima dei controlli avanzati che abbiamo realizzato:

**Subscribe Feed: Iscriviti ai Feed** ▼

Title:

Visualizzare Iscrizione tramite FeedBurner?

Url FeedBurner:

Visualizzare FeedCount?

Codice FeedCount :

[Rimuovi](#) | [Chiudi](#) [Salva](#)

Chiudiamo la funzione e la classe del widget e abbiamo finito il lavoro.

## Conclusione

Come hai potuto vedere in poco più di cinque minuti con la **nuova classe** messa a disposizione dalla **nuova versione di WordPress** siamo riusciti a **realizzare il nostro widget**. Quello che abbiamo realizzato è un widget abbastanza semplice che puoi prendere come esempio per realizzarne altri molto più complessi.

Se non hai voglia di riscrivere il codice passo dopo passo, puoi scaricare il file che trovi di seguito contenente l'esempio che abbiamo appena sviluppato. Tutto il codice è stato ben commentato per renderne più semplice la comprensione e permetterti di riutilizzarlo come modello per lo sviluppo di altri eventuali widget.

## Approfondimenti

Se vuoi approfondire ulteriormente l'argomento ti segnalo alcune risorse che potrebbero tornarti utili:

- [WordPress Codex - Widgets API](#)
- [WordPress Codex - New Widgets API](#)