

## Custom Post Type: come personalizzare il pannello di amministrazione?

Prosegue il nostro viaggio alla scoperta dei *Custom Post Type* di Wordpress: in questa lezione vedremo come personalizzare il pannello di amministrazione ed in particolare l'elenco dei post. Questa tecnica, unita a quelle esaminate [nel precedente articolo](#), ci permetterà di ottenere un'interfaccia completa per la gestione del nostro *Custom Post Type*.

Alla creazione del nostro *CPT*, Wordpress ha generato per noi tutta l'interfaccia per creare, elencare, modificare e cancellare i nostri progetti. In particolare è stato creato l'elenco degli elementi del *CPT*, simile a quello che esiste per i post. Ovviamente quello predefinito non include nessuno dei campi aggiuntivi del nostro *CPT*.

<input type="checkbox"/> Titolo	Data
<input type="checkbox"/> Sommerce - Shop WP theme	35 minuti fa Pubblicato
<input type="checkbox"/> Impero WordPress Theme	30 minuti fa Pubblicato
<input type="checkbox"/> Titolo	Data

Come vedi, l'elenco dei post ha molte più informazioni rispetto all'elenco dei progetti. Poniamo rimedio a questo problema inserendo alcune informazioni aggiuntive che abbiamo associato ai nostri progetti: l'anno di realizzazione e le skill necessarie.

Per fare questo dobbiamo "intercettare" il momento in cui Wordpress genera la tabella contenente i nostri progetti e modificare i titoli ed il contenuto. Iniziamo con la modifica dei titoli. Come hai visto [nella guida allo sviluppo di un plugin di Maurizio](#), Wordpress fornisce dei filtri per la manipolazione dei contenuti: alla nostra funzione viene passato del contenuto, che possiamo modificare e poi restituire.

Noi utilizzeremo il filtro `manage_{$postType}_posts_columns`, avendo cura di sostituire `$postType` con progetto (ovvero con l'identificativo scelto per il *CPT*):

Questo codice ci permette di sostituire completamente le intestazioni delle colonne della pagina di amministrazione. Infatti puoi notare che ignoriamo completamente l'array in input `$old_columns` per costruirne uno da zero con i nostri valori personalizzati. Le chiavi di questo array rappresentano l'identificativo della colonna, mentre i valori saranno visualizzati come intestazione. Otterremo in questo modo cinque colonne: una checkbox, un'immagine, il nome del progetto, le skill associate al progetto e l'anno di realizzazione. Nota che mentre `cb` e `title` sono campi predefiniti di Wordpress e

come tali automaticamente riempiti, `img`, `skills` e `anno` sono chiavi personalizzate a cui dovremo fornire il contenuto.

Il risultato raggiunto finora è il seguente:

<input type="checkbox"/> Immagine	Nome Progetto	Skills	Anno
<input type="checkbox"/>	Sommerce - Shop WP theme		
<input type="checkbox"/>	Impero WordPress Theme		
<input type="checkbox"/> Immagine	Nome Progetto	Skills	Anno

A questo punto dobbiamo inserire i contenuti nelle colonne che abbiamo aggiunto. Per fare questo utilizzeremo l'hook di Wordpress `manage_{$postType}_posts_custom_column` che richiama una funzione su ogni colonna della tabella, consentendoci di modificarne il contenuto (ricorda che a `$postType` va sostituito l'identificativo del *CPT*):

Prima di esaminare il codice per la gestione delle colonne, voglio fare una piccola digressione sulla chiamata a `add_action`. Finora abbiamo visto sempre invocazioni del tipo:

mentre questa è del tipo:

Come puoi vedere dal codice, il terzo parametro stabilisce la priorità di esecuzione della funzione (azioni con priorità più bassa vengono eseguite prima), mentre l'ultimo indica il numero di parametri che accetta la funzione da invocare. Finora abbiamo sempre utilizzato la prima forma perché le funzioni richiedevano solo un parametro. Dato che questo *hook* richiede due parametri, siamo stati costretti a specificare l'intera firma della funzione. Possiamo tornare ora alla gestione delle nostre colonne personalizzate.


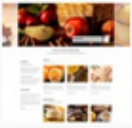
Come detto sopra, la funzione da invocare richiede due parametri: uno è il nome della colonna esaminata, mentre l'altro è l'ID del post analizzato. A noi interessa gestire solo le colonne non predefinite di Wordpress, quindi `img`, `skills` e `anno`. Modifichiamo quindi la nostra funzione:

In questo modo non ci resta che inserire il codice adatto per gestire i vari elementi. Per quanto riguarda l'immagine associata al progetto, utilizziamo la funzione `get_the_post_thumbnail`, assicurandoci che ne esista una con `has_post_thumbnail`; al contrario mostriamo un messaggio appropriato:

Per quanto riguarda le *skill*, utilizzeremo come [nel precedente articolo](#) la funzione `get_the_terms` (che restituisce un array con tutti i termini della tassonomia associati al post), mostrando un messaggio in caso non ne esista nessuna:

Per mostrare l'anno di realizzazione del progetto, invece, utilizzeremo la funzione `get_post_meta`, passando l'identificatore del *custom field*:

E questo è il risultato finale:

<input type="checkbox"/> Immagine	Nome Progetto	Skills	Anno
<input type="checkbox"/> 	<a href="#">Sommerce - Shop WP theme</a>	Css HTML jQuery Wordpress	2012
<input type="checkbox"/> 	<a href="#">Impero WordPress Theme</a>	HTML	2011
<input type="checkbox"/> Immagine	Nome Progetto	Skills	Anno

## Ordinare secondo colonne

Come puoi notare dall'immagine, la colonna "Nome Progetto" è resa come un link: questo significa che è possibile utilizzare il nome del progetto come criterio per ordinare l'elenco dei post. Aggiungiamo anche l'ordinamento per anno di realizzazione del progetto.

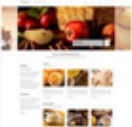
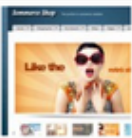
Per ottenere le colonne che permettono di ordinare l'elenco dei post, Wordpress utilizza una funzione chiamata `get_sortable_columns`, che restituisce un array contenente i nomi di tutte le colonne. È possibile *intercettare* la chiamata a questa funzione, per aggiungere le colonne di cui necessitiamo (se stai pensando ad un filtro, sei sulla buona strada). Il nome del filtro di cui necessitiamo è un po' difficile da trovare nella documentazione ufficiale di Wordpress: si chiama `manage_edit-$postType_sortable_columns` e riceve in ingresso l'array di colonne "ordinabili". Non ci resta che aggiungere l'anno di realizzazione del progetto a questo array e restituirlo modificato:

Ricaricando la pagina con l'elenco dei progetti, potrai notare che anche l'intestazione "Anno" è diventata un link e, cliccando, l'ordine dei progetti viene effettivamente modificato. Però Wordpress non conosce nulla del nostro *custom field* quindi si limita a ordinare i progetti utilizzando come criterio la data di pubblicazione. Quello che dobbiamo fare è associare un criterio di ordinamento

alla nostra colonna, in modo che Wordpress diventi 'conscio' del suo contenuto (in questo caso bisogna ordinare numericamente gli anni).

Per fare questo, utilizzeremo l'*hook* `pre_get_posts` che, come suggerisce il nome, viene attivato prima di ogni recupero di post dal database. Alla funzione invocata da questo *hook* viene passato un oggetto di tipo `WP_Query`, attraverso cui possiamo modificare l'ordine dei post.

Come puoi [leggere dalla documentazione](#), l'oggetto `WP_Query` possiede un attributo chiamato `orderby` che memorizza il criterio con cui sono ordinati i post recuperati dal database. Possiamo modificare questo ordine, imponendo che sia basato sul nostro custom field "Anno" (il cui identificatore è `yiw_progetti_anno`) e che l'ordine deve essere di tipo numerico:

<input type="checkbox"/> Immagine	Nome Progetto	Skills	Anno
<input type="checkbox"/> Nessuna immagine!	<b>Progetto senza anno</b>	Nessuna skill!	Anno non specificato!
<input type="checkbox"/> Nessuna immagine!	<b>Progetto di prova II</b>	Nessuna skill!	2010
<input type="checkbox"/> Nessuna immagine!	<b>Progetto di prova</b> <a href="#">Modifica</a>   <a href="#">Modifica veloce</a>   <a href="#">Cestina</a>   <a href="#">Visualizza</a>	Nessuna skill!	2008
<input type="checkbox"/> 	<b>Impero WordPress Theme</b>	HTML	2011
<input type="checkbox"/> 	<b>Sommerce - Shop WP theme</b>	Css HTML jQuery Wordpress	2012
<input type="checkbox"/> Immagine	Nome Progetto	Skills	Anno

## Conclusione

Con questo articolo si conclude il nostro viaggio attraverso i *Custom Post Type* di Wordpress.

Come hai potuto vedere nel corso dello sviluppo del nostro esempio "Progetti", siamo partiti da una semplice copia della funzionalità dei post, per arrivare alla gestione completa di un portfolio, con tanto di pannello di amministrazione e pagine dedicate, ed il tutto in poche righe di codice.

Non sarà la soluzione perfetta per ogni evenienza, ma la considero una valida (e veloce)

alternativa alla creazione di un'applicazione ad hoc. Tu che ne pensi? La useresti per un cliente?