

Custom Post Type: come migliorare l'esperienza utente?

Nel precedente articolo abbiamo introdotto i [Custom Post Type \(CPT\)](#): abbiamo visto cosa sono e come generarli. In questo articolo vedremo come adattare perfettamente il nostro *Custom Type* all'interno di Wordpress, così da farlo sembrare un elemento predefinito, come i post o le pagine, e fornire ai nostri clienti un'esperienza utente unica.

Il Custom Type progetto

Tutto il codice che vedremo in questo articolo si riferisce all'esempio del *CPT* progetto che abbiamo iniziato nel [primo articolo di questa serie](#). Quindi in qualsiasi momento mi riferirò sempre al file `wp-content/plugins/yiw/yiw_custom_types.php`.

Le personalizzazioni che ci apprestiamo a realizzare sono:

- l'aggiunta di campi predefiniti associati ad un progetto
- la creazione di una tassonomia 'Skill' da associare ai progetti
- la modifica delle colonne mostrate nell'elenco dei progetti
- la creazione di una pagina/vetrina per i progetti

Custom field: informazioni comuni a tutti i progetti

Ogni progetto che realizziamo ha delle informazioni comuni con gli altri. **Anno di realizzazione, tecnologie utilizzate, link** al sito del progetto possono essere un buon esempio di partenza. Allo stato attuale del plugin, potresti scrivere tutte queste informazioni nel corpo del testo, avendo cura di utilizzare lo stesso formato in tutti i progetti. Ad esempio il contenuto di un progetto potrebbe essere:

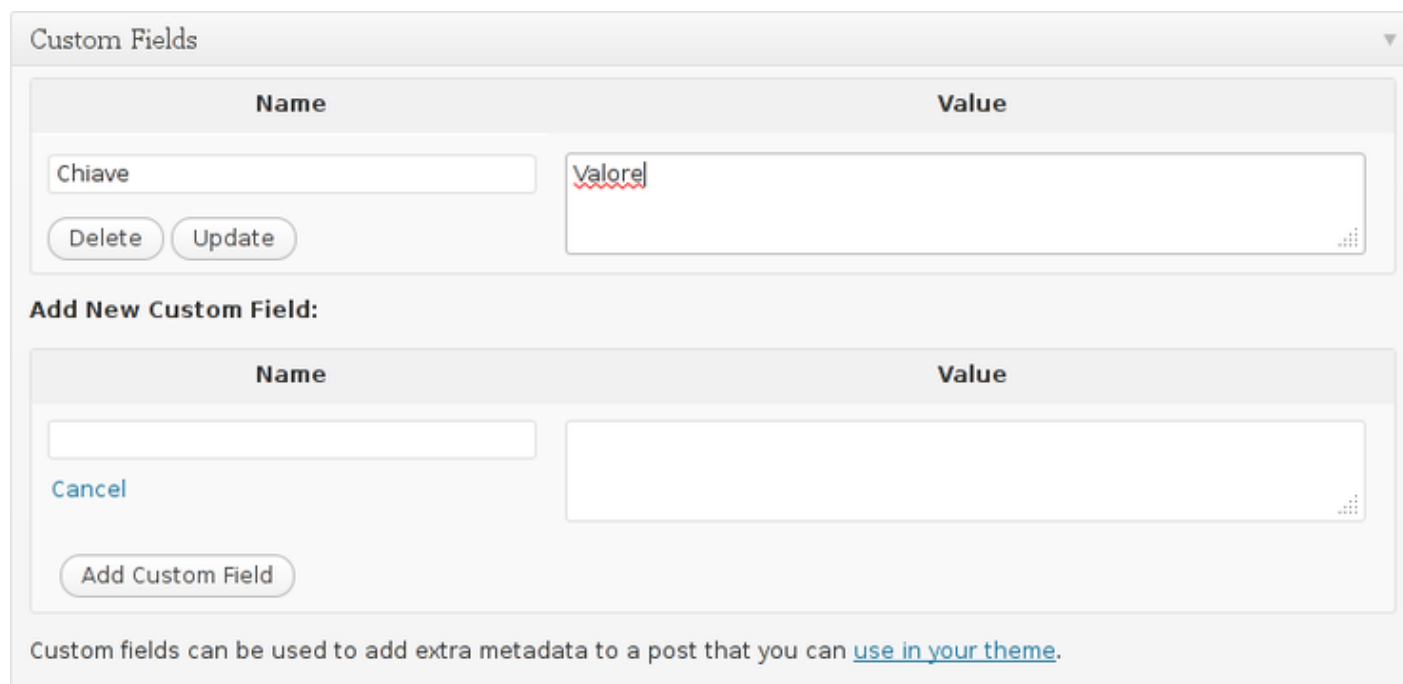
Potrebbe sicuramente funzionare: ma come potremmo, ad esempio, recuperare tutti i progetti dell'anno 2012? Dovremmo esaminare ogni progetto alla ricerca dell'informazione ricercate. Da questo esempio, si può capire come possedere dei dati strutturati è sicuramente meglio rispetto a possedere dati generici.

Quindi, per iniziare, aggiungiamo alcune informazioni predefinite per tutti i progetti:

- l'anno di realizzazione del progetto
- un link al sito del progetto

La soluzione che Wordpress ha fornito per associare informazioni aggiuntive ai singoli post è

l'introduzione dei *Campi personalizzati (Custom Fields)*. Un campo personalizzato è una coppia chiave-valore arbitraria che può essere associata ad un post durante la scrittura.



Custom Fields

Name	Value
Chiave	Valore

Delete Update

Add New Custom Field:

Name	Value
<input type="text"/>	<input type="text"/>

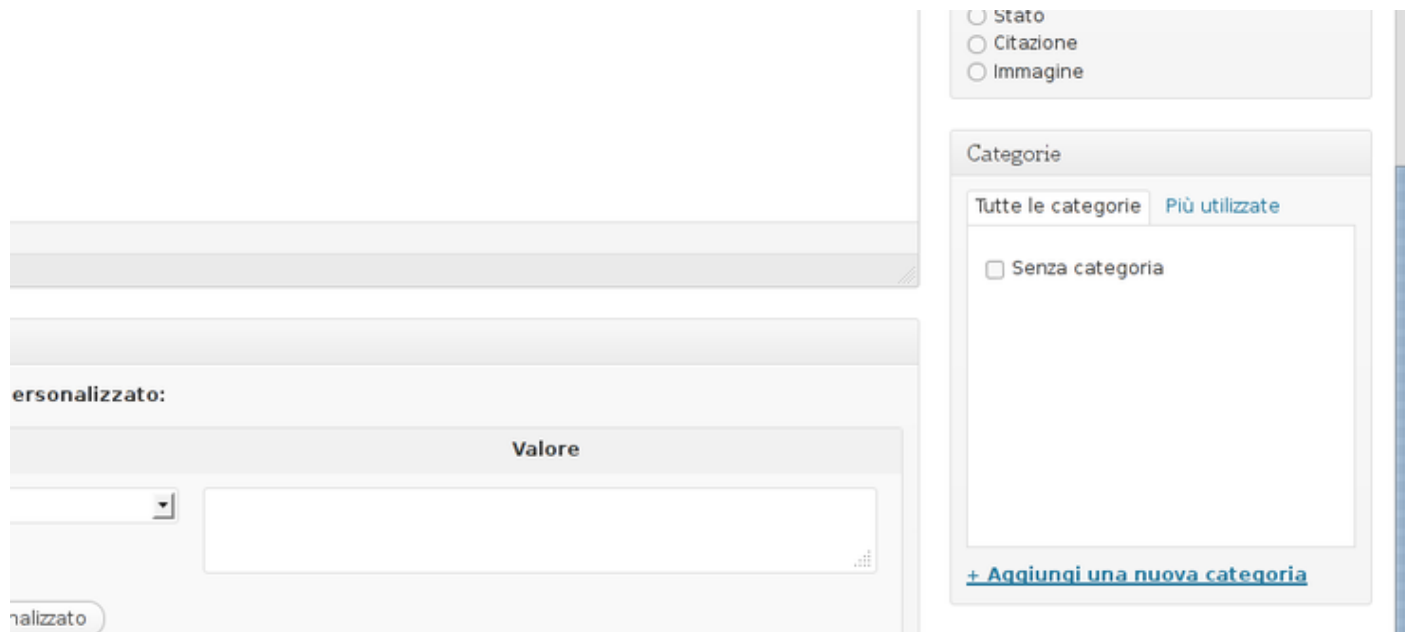
Cancel

Add Custom Field

Custom fields can be used to add extra metadata to a post that you can [use in your theme](#).

L'utilizzo dei campi personalizzati "nudi e crudi" però comporta diversi problemi: infatti bisogna ricordarsi quali chiavi associare a quali *Post Type* e aggiungerli manualmente ad ogni elemento. Si può capire come questo procedimento può diventare un incubo se il numero di campi personalizzati è elevato.

Una soluzione più efficace sarebbe quella di avere un form con dei campi preimpostati nella schermata di scrittura del *Post Type*, in modo da inserire i dati nel form, senza preoccuparci di dover scegliere il nome del campo. Questo è proprio quello che faremo, sfruttando i *meta_box* di Wordpress. Un *meta_box* è un elemento presente nella schermata di scrittura del post, in cui è possibile inserire del contenuto personalizzato. Wordpress definisce una serie di *meta_box* predefiniti, come ad esempio quello per l'aggiunta delle categorie:



Per aggiungere un box nella pagina di scrittura di un post si utilizza la funzione `add_meta_box`. Questa funzione accetta i seguenti parametri:

- un id arbitrario univoco
- il titolo del box da aggiungere
- il nome della funzione generatrice del contenuto da inserire nel box
- il *Post Type* a cui applicare il box: può essere uno di default (*post*, *page*, ecc.) oppure un *Custom Post Type*
- la posizione in cui aggiungere il box: può essere un valore tra *normal*, *advanced* e *side* chee posizionano il box rispettivamente sotto l'editor, sotto l'editor in ultima posizione, e a lato dell'editor.
- la priorità del box, ovvero se posizionarlo più in basso o più in alto rispetto agli altri box
- dei parametri da passare alla funzione che genera il contenuto del box

Proviamo ad applicare questa funzione al nostro esempio:

Grazie a questo codice sarà creato un box nella parte laterale della schermata di scrittura dei nostri progetti, con il contenuto generato dalla funzione `yiw_progetti_box`. Come puoi notare, come parametro di questa funzione viene passato l'oggetto rappresentante il post che stiamo modificando: ci sarà utile per recuperare delle informazioni, come ad esempio l'id.

Ora aggiungiamo il form che ci permetterà di aggiungere l'anno di realizzazione e il link al sito del progetto. Per farlo, modifica la funzione `yiw_progetti_box` come segue:

Andiamo ad analizzare questo codice: vengono creati due campi di input per testo, con le relative etichette. Le parti importanti sono:

- l'attributo name dei due campi: sarà questo l'identificatore per recuperare il campo personalizzato nella visualizzazione del progetto
- l'attributo value: quando si modifica un progetto è desiderabile sapere se esistono già dei valori associati ai campi personalizzati. La funzione `get_post_meta` ci permette di recuperare questi valori (restituendo una stringa vuota se non esistono). Nota che come parametri abbiamo passato l'id del post, l'attributo name associato al campo personalizzato e true, che indica di restituire solo il primo valore.

Ricaricata la pagina di editing del progetto, questo è il risultato finale:

in vendita su ThemeForest.



The screenshot shows a WordPress interface for editing a project. At the top, it displays the publication date and time: "Pubblicato il: 2 ottobre 2012 @ 07:27" with a "Modifica" link. Below this are two buttons: "Spostare nel cestino" (in red) and "Aggiorna" (in a blue rounded rectangle). The main section is titled "Dettagli progetto" and contains a heading "Aggiungi i dettagli del progetto:". There are two input fields: "Link al sito" with the value "http://themeforest.net/item/sommerce-s" and "Anno" with the value "2012".

Ora non ci resta che salvare il contenuto del form e associarlo al progetto. Per questo compito utilizzeremo la funzione `update_post_meta`:

Per utilizzare questa funzione occorre dunque passare l'id del post, il nome del campo personalizzato da modificare, ed il nuovo valore. La caratteristica interessante di questa funzione è che se non esiste alcun valore per quel campo personalizzato, lo aggiunge, mentre se già esiste lo modifica: questo ci consente di non distinguere tra i due casi. Non ci resta dunque che agganciare la funzione di memorizzazione dei campi personalizzati al salvataggio dei progetti:

Quindi, come detto in precedenza, salviamo i due valori passati nel form (effettuando un minimo di validazione).

Come catalogare i nostri progetti? La tassonomia *skill*

[Tassonomia](#) non è sicuramente una parola che capita di sentire tutti i giorni. Eppure, se hai l'abitudine di scrivere per il tuo blog ne sfrutti i principi, magari senza saperlo.

La tassonomia è la disciplina della classificazione. Immagina di avere delle biglie colorate e tante scatole, ognuna con il nome di un colore diverso. Posizionando le biglie nello scatolo con il colore corrispondente stai utilizzando una tassonomia basata sul colore.

Tornando al mondo di Wordpress, ad un post vengono solitamente aggiunti due elementi: le categorie e i tag. Queste sono due tassonomie, che ci permettono di catalogare con precisione tutti i nostri articoli.

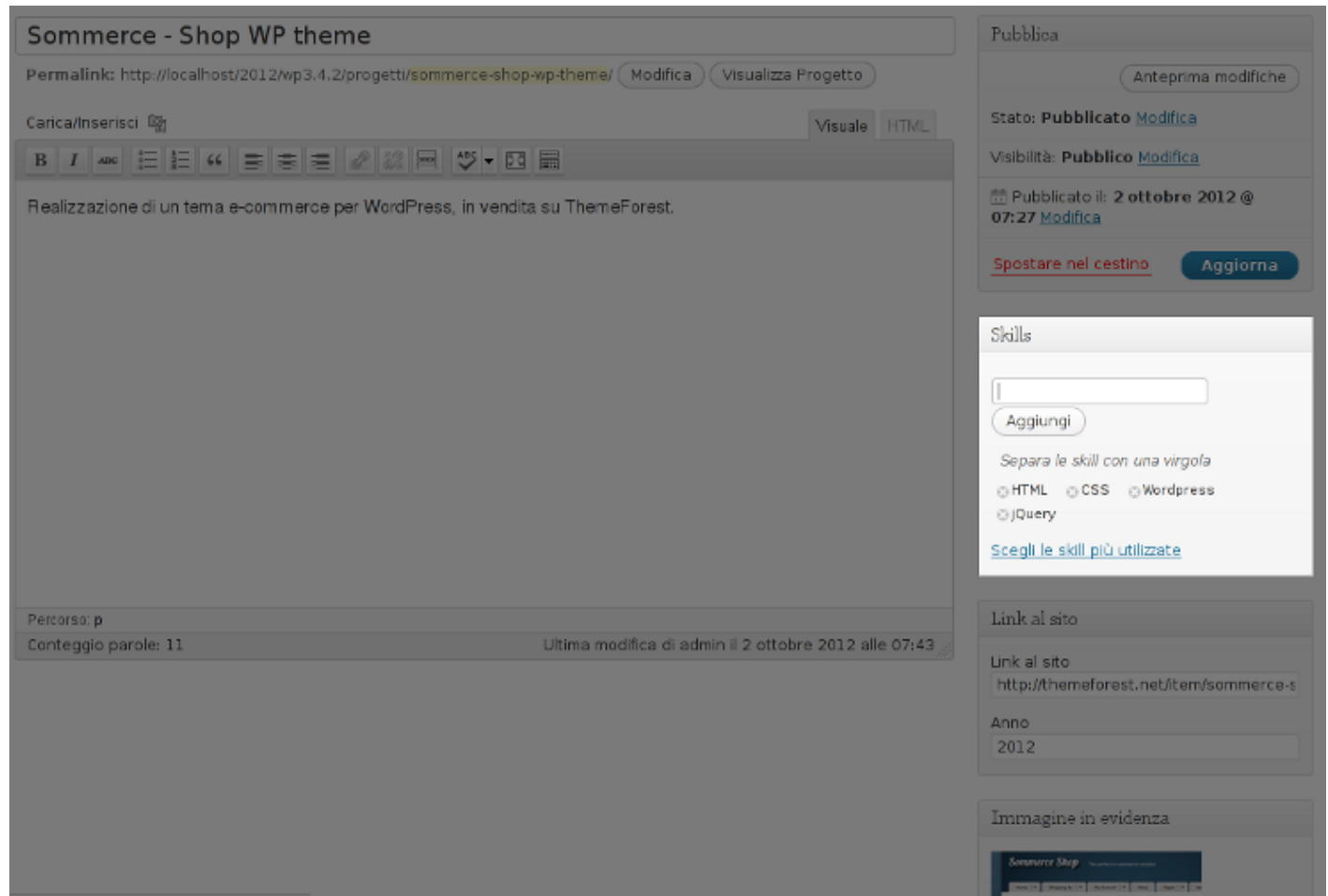
Wordpress consente di creare delle tassonomie personalizzate, da associare ai (*CPT*) (oppure ai *Post Type* predefiniti, come post e pagine). Ad esempio, se avessimo un *CPT* "Ricette", potremmo voler aggiungere una tassonomia basata sugli ingredienti, oppure potremmo voler classificare un *CPT* "Film" per genere.

Per il nostro esempio, il *CPT* "Progetti", aggiungeremo le skill che sono state necessarie per completare quel progetto. Lo schema di creazione di una tassonomia personalizzata ricalca quasi fedelmente quello della creazione di un *CPT*:

La funzione cardine è `register_taxonomy` che accetta come parametri:

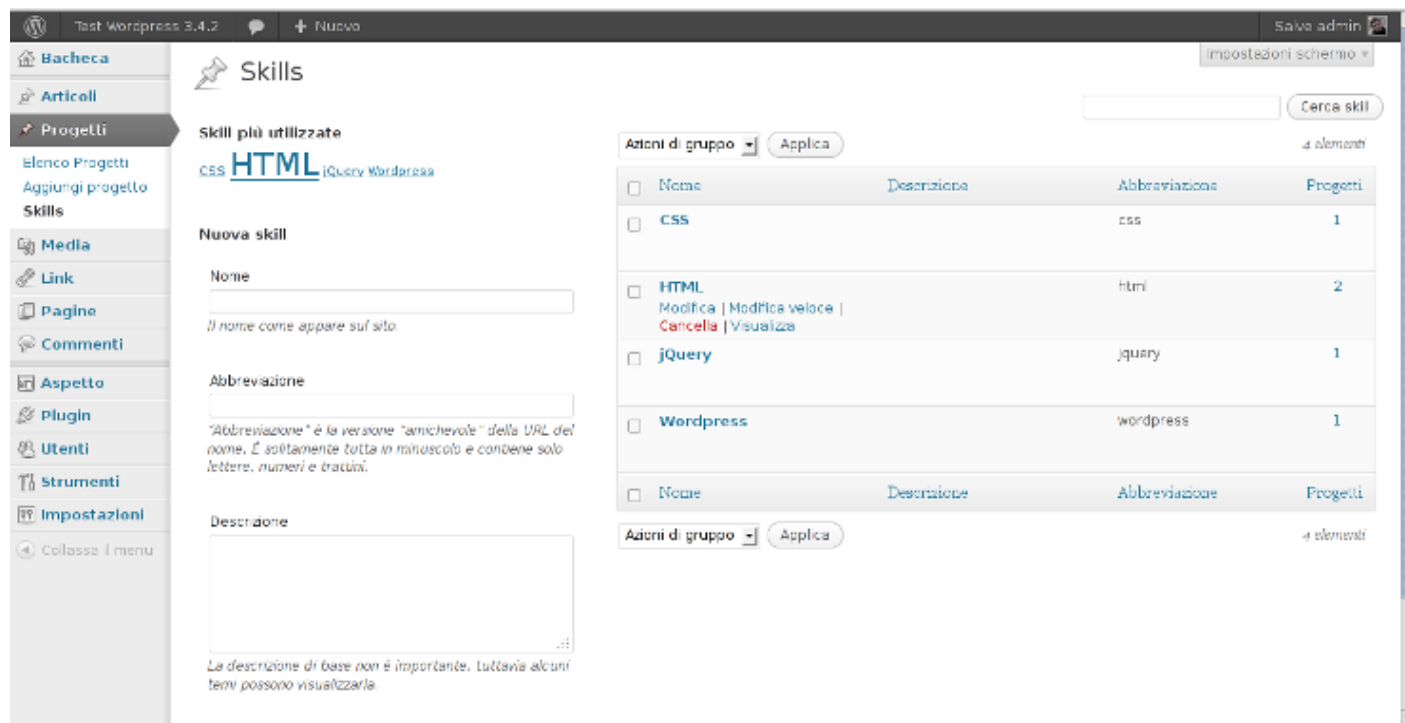
1. Il nome della nuova tassonomia
2. Il nome del *Post Type* a cui associarla
3. Un array di argomenti (come per il *CPT*)

In questo array di argomenti passiamo un elenco di etichette, che verranno utilizzate da Wordpress per la generazione del pannello di amministrazione della tassonomia. Ci sono [molti altri parametri](#), ma non li tratteremo in questa sede, accontentandoci dei valori predefiniti.



The screenshot shows the WordPress admin interface for a theme page titled "Sommerce - Shop WP theme". The main content area contains the text "Realizzazione di un tema e-commerce per WordPress, in vendita su ThemeForest." and a meta box at the bottom showing "Conteggio parole: 11" and "Ultima modifica di admin il 2 ottobre 2012 alle 07:43". The right sidebar contains several meta boxes: "Pubblica" (with "Anteprima modifiche" button), "Stato: **Pubblicato** Modifica", "Visibilità: **Pubblico** Modifica", "Pubblicato il: **2 ottobre 2012 @ 07:27** Modifica", "Spostare nel cestino" and "Aggiorna" buttons, "Skills" (with a search box, "Aggiungi" button, and a list of skills: HTML, CSS, Wordpress, JQuery), "Link al sito" (with "http://themeforest.net/item/sommerce-s" and "Anno 2012"), and "Immagine in evidenza" (with a thumbnail for "Sommerce Shop").

Oltre ad associare la nuova tassonomia ai nostri progetti, Wordpress genera l'interfaccia utente per gestire ed elencare le skill, oltre al box (simile a quello per i tag) nella pagina di scrittura dei progetti.



The screenshot shows a WordPress admin interface for a 'Skills' page. On the left is a sidebar with navigation links. The main area features a search bar and a table of skills. The table lists skills like CSS, HTML, jQuery, and Wordpress with their respective abbreviations and project counts. There are also 'Azioni di gruppo' and 'Applica' buttons for managing the table.

Page template: una vetrina per i nostri progetti

Nell'articolo precedente abbiamo visto come elencare i progetti sfruttando la pagina archivio generata automaticamente da Wordpress. Questa pagina, però, essendo generata automaticamente, non può essere a conoscenza dei campi aggiuntivi che abbiamo associato ai nostri progetti. Dunque è necessario creare una pagina *ad hoc* che mostri tutte le informazioni per ogni singolo progetto.

Per questo scopo utilizziamo la funzionalità *Page Template* che Wordpress ci mette a disposizione. Come spiegato [nella guida di Nando](#) basta inserire nella cartella del nostro tema un file con un commento di intestazione simile al seguente per creare un nuovo template di pagina, che verrà mostrato nella menu *Modello* del box *Attributi pagina*:

Innanzitutto, recuperiamo l'header e il footer del nostro tema, tramite le funzioni apposite:

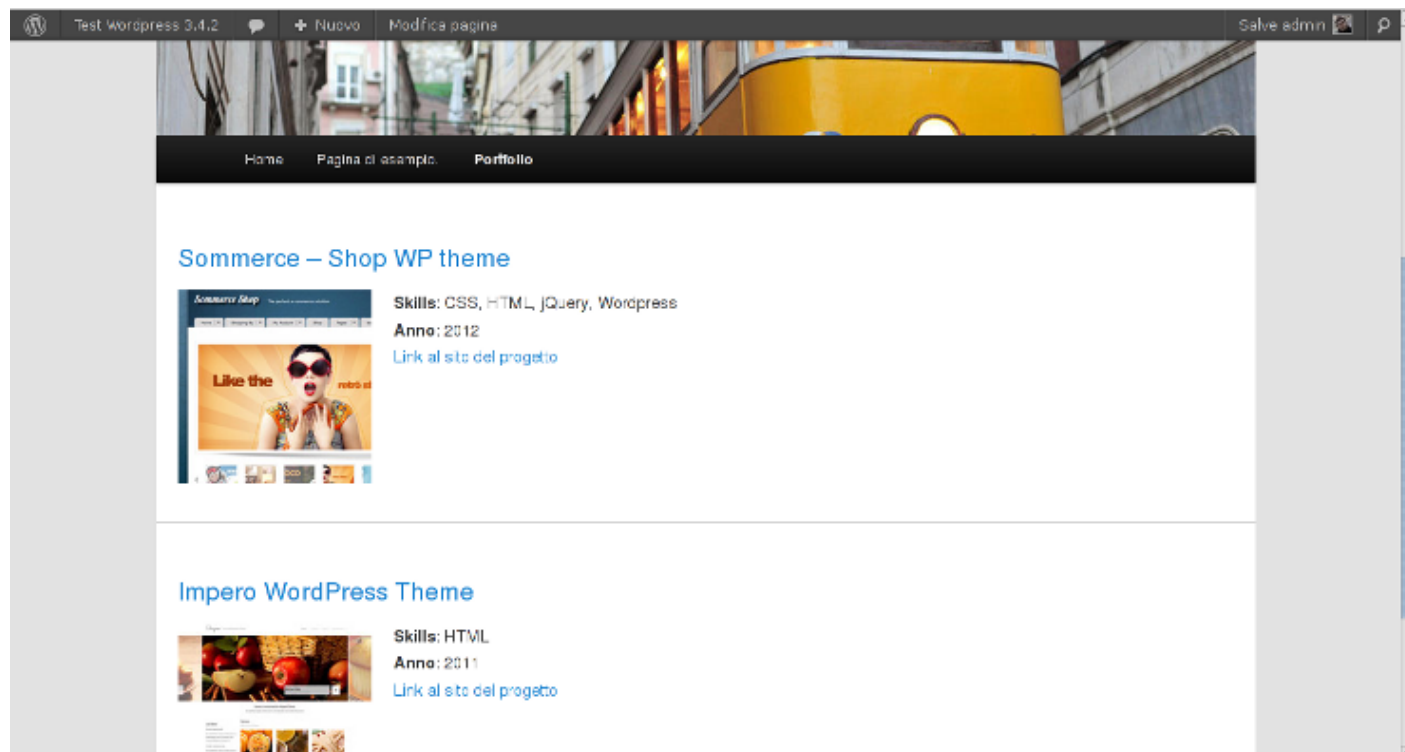
A questo punto non ci resta che creare un [loop](#) personalizzato per elencare i nostri progetti. Utilizzeremo la classe `WP_Query` che accetta come argomento un array contenente i parametri per la creazione del loop:

Abbiamo richiesto, tramite WP_Query, i post di tipo progetto, ordinandoli per il valore del campo personalizzato yiw_progetti_anno, senza alcun limite (questo è il significato del -1 alla voce posts_per_page). Quindi abbiamo mostrato le informazioni recuperate attraverso le solite funzioni del loop. Dobbiamo solo aggiungere le skill, l'anno ed il link al sito, e la vetrina per i nostri progetti sarà pronta.

In effetti, dovresti già sapere come fare per recuperare anno e link al sito: devi utilizzare la funzione get_post_meta, così come abbiamo fatto per recuperare i dati nei *custom box*. La funzione che ci consente di recuperare le skill è invece get_the_terms. Questa funzione ha la seguente firma:

get_the_terms accetta come parametri l'id del post in esame ed il nome della tassonomia di cui recuperare i termini e restituisce un array di elementi della tassonomia associati al post, se esistono, e false altrimenti. Possiamo applicare queste nozioni per recuperare le skill associate ad ogni progetto:

Lascerò che sia tu a completare il nostro portfolio come esercizio. Il risultato che ho ottenuto io è il seguente:



Ma le mie capacità grafiche sono limitate (ecco perché sto seguendo [la guida di Sara](#)): sono sicuro che tu riuscirai a fare di meglio.

Conclusioni

In questo articolo abbiamo visto come aggiungere dei campi predefiniti a tutti i nostri progetti, come catalogarli utilizzando una tassonomia personalizzata ed infine abbiamo realizzato un *Page Template* per elencare tutti i nostri progetti e le nuove informazioni associate.

Nel prossimo articolo vedremo come personalizzare l'elenco dei progetti, aggiungendo le colonne relative a skill e anno di realizzazione; un metodo alternativo per mostrare i nostri progetti attraverso uno *shortcode*, e parleremo dei *Single template* che ci consentiranno di personalizzare ancora di più il nostro *Custom Type*

Puoi scaricare tutto il codice relativo agli esempi in questa serie dal [repository di github](#)