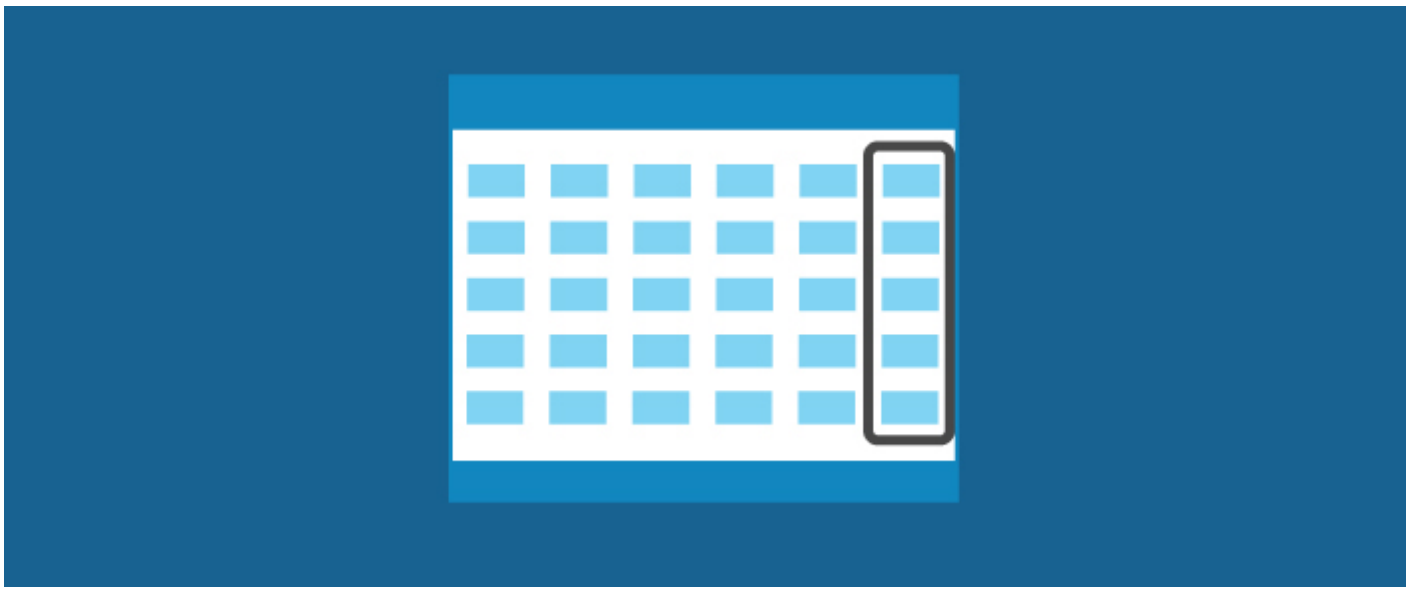
Aggiungere una colonna personalizzata in WordPress

domenica, 25 Dicembre 2016

Spesso, soprattutto nel caso di custom post types, potresti avere la necessità di visualizzare nella tabella contenente i posts nel backend di WordPress una nuova colonna contenente, ad esempio, un campo personalizzato che possa aiutarti nella gestione dei contenuti e velocizzare il tuo lavoro.



Sommario

Un esempio pratico: l'ID

Ipotizza, a scopo dimostrativo, di voler aggiungere alla tabella che contiene l'elenco degli articoli di WordPress, che trovi in:

Articoli > Tutti gli articoli

una nuova colonna che visualizzi l'ID univoco dell'articolo.

Aggiungere la colonna

Il primo passo da compiere, è far capire a WordPress la tua intenzione e creare una nuova colonna che si aggiunga a quelle fornite di default. Per farlo, basta aggiungere al file functions.php del tuo tema, che trovi in:

wp-content/themes/{nome_del_tema}/

il seguente codice:

```
// Aggiunge la colonna ID
add_filter( 'manage_posts_columns', function( $columns ) {
    $columns['id'] = __( 'ID', 'dominio' );
    return $columns;
} );
```

Sfruttando l'hook manage_post_columns, non fai altro che aggiungere un nuovo elemento all?array che contiene le colonne da visualizzare.

Ricaricando la pagina, vedrai la nuova colonna appena creata. Ma come fare a gestirne il contenuto?

Riempiamo la colonna appena creata

Ora non ti resta che fare in modo che le celle della nuova colonna siano riempite con il giusto contenuto. Per farlo, è necessario aggiungere al file functions.php del tuo tema il seguente codice:

```
// Aggiunge il contenuto della colonna ID
add_filter( 'manage_posts_custom_column', function( $column_name, $post_id ) {
 if ( $column_name === 'id' ) {
  echo get_the_ID();
 }
}, 10, 2 );
```

Con l'aiuto di un altro filtro, manage_posts_custom_column, hai appena creato una funzione che restituisce l'ID dell'articolo e lo assegna alla colonna dal nome id.

Ricaricando nuovamente la pagina, vedrai che ogni articolo presente nell'elenco, ha in corrispondenza della colonna id l'identificativo dell?articolo stesso così.

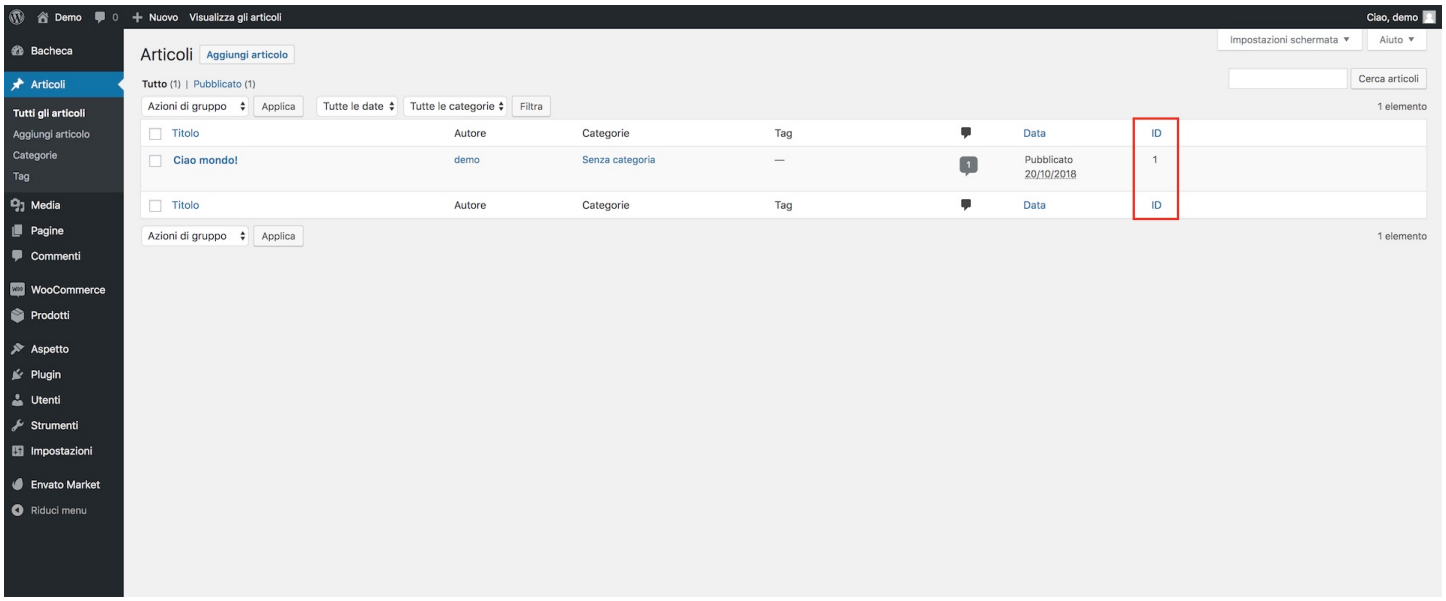
Rendere la colonna ordinabile

Per completare il lavoro, ecco un piccolo valore aggiunto. Se volessi ordinare la tua lista in base ai valori presenti nella colonna appena creata, ti basta aggiungere al file functions.php:

```
// Rende la colonna ordinabile
add_filter( 'manage_edit-post_sortable_columns', function( $columns ) {
 $columns[ 'id' ] = 'id';
 return $columns;
} );
```

che non fa altro che aggiungere la colonna all'elenco delle colonne su cui abilitare la funzione di ordinamento presente nel backend di WordPress.

Ed ecco finalmente il risultato finale:



Custom post types

Naturalmente, puoi aggiungere una colonna personalizzata non solo agli articoli, ma anche ad eventuali custom post types, semplicemente utilizzando gli hooks adatti.
Così, l'hook manage_posts_columns diventa:

```
manage_${post_type}_posts_columns
```

e l'hook manage_posts_custom_column diventa:

```
manage_{$post_type}_posts_custom_column
```

sostituendo alla variabile ${post_type} il nome assegnato al tuo custom post type, ed il gioco è fatto.