



# WordPress: usare AJAX per i prodotti delle categorie di WooCommerce

WooCommerce ci permette di creare le nostre categorie per i prodotti. Queste categorie non sono altro che tassonomie appartenenti alla tassonomia madre `product_cat`. Vogliamo realizzare una pagina dedicata ad una categoria in cui vi siano le sottocategorie della categoria scelta disposte su colonne. Quando clicchiamo su una sottocategoria, una richiesta AJAX mostrerà i prodotti nella riga sottostante.

Definiamo il seguente shortcode in `functions.php`:

```
function my_create_tax_shortcode( $atts, $content = null ) {  
  
    extract( shortcode_atts( array (  
        'id' => '' // ID tassonomia genitore  
    ), $atts ) );  
  
    $name = 'product_cat'; // Tassonomia  
    di WooCommerce  
    $termchildren = get_term_children(  
$id, $name ); // Tassonomie discendenti  
    $html = '<div id="items">';
```

```

        $i = 0;

        foreach ( $termchildren as $child ) {

            $term = get_term_by( 'id',
$child, $name );

            // Impostiamo come immagine
in evidenza delle tassonomie
            // l'immagine in evidenza del
primo prodotto di ciascuna tassonomia.

            $args = array(
                'post_type' =>
'product',
                'posts_per_page' =>
1,
                'tax_query' => array(
                    'taxonomy' =>
'product_cat',
                    'field' =>
'slug',
                    'terms' =>
$term->slug
                )
            );
            $loop = new WP_Query( $args
);

            $image = '';
            if( $loop->have_posts() ) {
                while( $loop-
>have_posts() ) {

```

```

$loop-
>the_post();

$pid =
get_the_ID();

if(
has_post_thumbnail( $pid ) ) {

$src
= wp_get_attachment_image_src( get_post_thumbnail_id(
$pid ), 'medium' );

$image = '';
} else {

$placeholder = plugins_url() .
'/woocommerce/assets/images/placeholder.png';

$image = '';
}

}

wp_reset_query();
}

$html .= '<div class="item">
<a href="#" data-term="' . $term->slug . '">' .
$image . '<span>' . $term->name . '</span></a>
</div>';

$i++;
if( $i % 5 == 0 ) {
$html .= '<div
class="item-clear item-ajax"><ul class="products">
</ul></div>';
}
}

$html .= '</div>';

```

```
        return $html;
    }

    add_shortcode( 'my-tax-shortcode',
        'my_create_tax_shortcode' );
```

Lo shortcode accetta come parametro l'ID della categoria genitore. Quindi avremo:

```
[my-tax-shortcode id="95"]
```

Ogni riga contiene 5 categorie e dopo ogni cinque categorie viene inserito l'elemento che conterrà i prodotti. Abbiamo volutamente usato la lista non ordinata con classe `products` perché in questo modo erediterà gli stili già definiti in WooCommerce o nel tema.

Ogni link delle categorie contiene un attributo di dati dove viene salvato lo slug (abbreviazione) della categoria. Questo attributo verrà usato nella richiesta AJAX. Definiamo quindi la nostra action AJAX in `functions.php`:

```
function my_get_products() {

    $slug = $_GET['slug'];

    $args = array(
        'post_type' =>
        'product',
        'product_cat' =>
        $slug,
```

```

'posts_per_page' =>
-1

);

$loop = new WP_Query( $args );
$html = '';

if( $loop->have_posts() ) {
    while( $loop-
>have_posts() ) {
                                $loop-
>the_post();
                                $pid =
get_the_ID();
                                $html .= '<li
class="product">';
                                $html .= '<a
href="' . get_permalink() . '">';
                                $html .=
'<div class="thumbnail-container">';
                                if(
has_post_thumbnail( $pid ) ) {
                                    $src
= wp_get_attachment_image_src( get_post_thumbnail_id(
$pid ), 'medium' );
                                    $html
.= '';
                                } else {

$placeholder = plugins_url() .
'/woocommerce/assets/images/placeholder.png';
                                $html
.= '';

```

```

    }
    $html .=
'</div>';

    $html .=
'<h5>' . get_the_title() . '</h5>';
    $html .=
'</a>';

    $html .=
'</li>';

    }
    wp_reset_query();
}

echo $html;
exit();

}

add_action( 'wp_ajax_my_get_products',
'my_get_products' );
add_action( 'wp_ajax_nopriv_my_get_products',
'my_get_products' );

```

Infine usiamo jQuery per la richiesta AJAX:

```

(function( $ ) {
    $(function() {
        var $items = $( "#items" );
        if( $items.length ) {

```

```

        var url = "http://" +
location.host + "/wp-admin/admin-ajax.php";

        $( "a[data-term]", $items
).click(function( e ) {
                    e.preventDefault();
                    $( ".products",
$items ).slideUp( "fast" ).empty();

                    var $a = $( this );
                    var $parent =
$a.parent();

                    var slug = $a.data(
"term" );

                    var data = {
                        action:
                        slug: slug
                    };

                    $.get( url, data,
function( html ) {
$a.parent().siblings( ".item-ajax" ).
                        find(
                        ".products" ).html( html ).
                        slideDown(
"fast" );
                    });
                });
        }
    });
})( jQuery );

```

