

GABRIELE ROMANATO

WordPress: creare thumbnail senza immagini in evidenza

Se cercate un modo semplice per generare thumbnail su post in cui non avete impostato un'immagine in evidenza nelle opzioni del post, Get The Image di *Justin Tadlock* è la soluzione che fa per voi. Questo semplice plugin, scritto con in mente la retrocompatibilità con le versioni di WordPress precedenti alla 2.9, esegue eccellentemente il suo compito. Vediamone il funzionamento dietro le quinte.

Caso 1: la thumbnail è già impostata

In questo caso (disponibile solo in WordPress 2.9+) il plugin usa la seguente funzione:

```
function image_by_the_post_thumbnail( $args = array() ) {  
  
    /* Check for a post image ID (set by WP as a custom  
field). */  
    $post_thumbnail_id = get_post_thumbnail_id(  
$args['post_id'] );  
  
    /* If no post image ID is found, return false. */  
    if ( empty( $post_thumbnail_id ) )  
        return false;  
  
    /* Apply filters on post_thumbnail_size because this  
is a default WP filter used with its image feature. */  
    $size = apply_filters( 'post_thumbnail_size',  
$args['size'] );  
  
    /* Get the attachment image source. This should  
return an array. */  
    $image = wp_get_attachment_image_src(  
$post_thumbnail_id, $size );
```

```

        /* Get the attachment excerpt to use as alt text. */
        $alt = trim( strip_tags( get_post_field(
'post_excerpt', $post_thumbnail_id ) ) );

        /* Return both the image URL and the post thumbnail
ID. */
        return array( 'url' => $image[0], 'post_thumbnail_id'
=> $post_thumbnail_id, 'alt' => $alt );
    }

```

La funzione restituisce l'URL dell'immagine conservato nella libreria media, l'ID del post di riferimento e il testo alternativo dell'immagine.

Caso 2: thumbnail non disponibile

Se la thumbnail non è disponibile, il plugin prova ad estrarre la prima immagine dal contenuto del post:

```

function image_by_scan( $args = array() ) {

    /* Search the post's content for the <img /> tag and
get its URL. */
    preg_match_all( '|<img.*?src=[\'"](.*)[\'"].*?>|i',
get_post_field( 'post_content', $args['post_id'] ), $matches
);

    /* If there is a match for the image, return its URL.
*/
    if ( isset( $matches ) && $matches[1][0] )
        return array( 'url' => $matches[1][0] );

    return false;
}

```

La funzione `image_by_scan()` in questo caso restituisce l'URL dell'immagine ottenuto usando le espressioni regolari.

Visualizzare l'immagine

L'immagine viene visualizzata con la seguente funzione che restituisce una stringa HTML:

```
function display_the_image( $args = array(), $image = false )
{

    /* If there is no image URL, return false. */
    if ( empty( $image['src'] ) )
        return false;

    /* Extract the arguments for easy-to-use variables.
*/
    extract( $args );

    /* If there is alt text, set it. Otherwise, default
to the post title. */
    $image_alt = ( ( !empty( $image['alt'] ) ) ) ?
$image['alt'] : apply_filters( 'the_title', get_post_field(
'post_title', $post_id ) ) );

    /* If there is a width or height, set them as HTML-
ready attributes. */
    $width = ( ( $width ) ? ' width="' . esc_attr( $width
) . '"' : '' );
    $height = ( ( $height ) ? ' height="' . esc_attr(
$height ) . '"' : '' );

    /* Loop through the custom field keys and add them as
classes. */
    if ( is_array( $meta_key ) ) {
        foreach ( $meta_key as $key )
            $classes[] = str_replace( ' ', '-',
strtolower( $key ) );
    }

    /* Add the $size and any user-added $image_class to
the class. */
    $classes[] = $size;
```

```

    $classes[] = $image_class;

    /* Join all the classes into a single string and make
    sure there are no duplicates. */
    $class = join( ' ', array_unique( $classes ) );

    /* If there is a $post_thumbnail_id, apply the WP
    filters normally associated with get_the_post_thumbnail(). */
    if ( !empty( $image['post_thumbnail_id'] ) )
        do_action( 'begin_fetch_post_thumbnail_html',
        $post_id, $image['post_thumbnail_id'], $size );

    /* Add the image attributes to the <img /> element.
    */
    $html = '';

    /* If $link_to_post is set to true, link the image to
    its post. */
    if ( $link_to_post )
        $html = '<a href="' . get_permalink( $post_id
        ) . '" title="' . esc_attr( apply_filters( 'the_title',
        get_post_field( 'post_title', $post_id ) ) ) . '">' . $html .
        '</a>';

    /* If there is a $post_thumbnail_id, apply the WP
    filters normally associated with get_the_post_thumbnail(). */
    if ( !empty( $image['post_thumbnail_id'] ) )
        do_action( 'end_fetch_post_thumbnail_html',
        $post_id, $image['post_thumbnail_id'], $size );

    /* If there is a $post_thumbnail_id, apply the WP
    filters normally associated with get_the_post_thumbnail(). */
    if ( !empty( $image['post_thumbnail_id'] ) )
        $html = apply_filters( 'post_thumbnail_html',
        $html, $post_id, $image['post_thumbnail_id'], $size, '' );

    return $html;
}

```

Come si può notare, la funzione scritta da Justin è abbastanza "intelligente" da riuscire ad impostare dei valori predefiniti qualora i valori passati come parametri non fossero impostati.