

WordPress: perché é importante conoscere i valori di ritorno delle funzioni core

Per evitare errori nell'uso delle funzioni di WordPress occorre conoscerne i valori di ritorno.

Consideriamo il seguente codice:

```
global $post;
$id = $post->ID;

$terms = get_the_terms( $id, 'nometassonomia' );

foreach( $terms as $term ) {
    echo $term->name . "\n";
}
```

L'errore di fondo sta nel fatto che non abbiamo considerato i valori di ritorno della funzione `get_the_terms()` nel caso in cui non esistano termini associati al post o la tassonomia specificata non esista.

Quindi:

```
global $post;
$id = $post->ID;
```

```
$terms = get_the_terms( $id, 'nometassonomia' );

if ( $terms && ! is_wp_error( $terms ) ) {

    foreach( $terms as $term ) {
        echo $term->name . "\n";
    }
}
```

La sezione **Returns** (valori di ritorno) della documentazione di WordPress per questa funzione afferma infatti che:

Array of term objects on success; false if the post contains no terms from the given taxonomy; false if the post doesn't exist; and a `Class_Reference/WP_Error` object if an invalid taxonomy is entered.

La funzione `is_wp_error()` verifica che il valore restituito sia o meno un'istanza della classe core `WP_Error`. Questa classe può essere usata anche per gestire i nostri errori:

```
function my_func( $param ) {
    if( !$param ) {
        return new WP_Error( 'codice-errore',
'Messaggio di errore' );
    }
}

$return = my_func();
if( is_wp_error( $return ) ) {
    echo $return->get_error_message(); // 'Messaggio
```

```
di errore'  
}
```

Il costruttore della classe accetta tre parametri:

1. una stringa rappresentante il codice dell'errore
2. il messaggio di errore associato al codice
3. i dati associati all'errore sotto forma di array

Potete trovare maggiori informazioni su questa classe nella documentazione ufficiale.