get_field()

- Description
- Parameters
- Return
- Examples
  - Get a value from the current post
  - Get a value from a specific post
  - Check if value exists
  - Get a value from different objects
  - Get a value without formatting
  - Get an escaped (HTML safe) field value (since ACF 6.2.6)
  - Escaping the value
- Notes

Description

Returns the value of a specific field.

Intuitive and powerful (much like ACF itself ?), this function can be used to load the value of any field from any location. Please note that each field type returns different forms of data (string, int, array, etc).

Any returned values should be escaped using an appropriate escaping function before they are output.

Parameters

```
get_field($selector, [$post_id = false], [$format_value = true], [$escape_html = false]);
```

- `$selector` *(string) (Required)* The field name or field key.
- `$post_id` *(mixed) (Optional)* The post ID where the value is saved. Defaults to false, which will use the current post.
- `$format_value` *(bool) (Optional)* Whether to apply formatting logic. Defaults to true.
- `$escape_html` *(bool) (Optional)* Since 6.2.6 – return an escaped HTML safe version of the field value. Requires $format_value to be true. Defaults to false.

Return

(mixed) The field value.

Examples

Get a value from the current post

This example shows how to load the value of field ‘text_field’ from the current post.

```
$value = get_field( "text_field" );
```

Get a value from a specific post

This example shows how to load the value of field ‘text_field’ from the post with ID = 123.

```
$value = get_field( "text_field", 123 );
```

Check if value exists

This example shows how to check if a value exists for a field.

```
$value = get_field( "text_field" );

if( $value ) {
    echo wp_kses_post( $value );
} else {
    echo 'empty';
```

```
}
```

Get a value from different objects

This example shows a variety of **$post_id** values to get a value from a post, user, term and option.

```
$post_id = false; // current post
$post_id = 1; // post ID = 1
$post_id = "user_2"; // user ID = 2
$post_id = "category_3"; // category term ID = 3
$post_id = "event_4"; // event (custom taxonomy) term ID = 4
$post_id = "option"; // options page
$post_id = "options"; // same as above

$value = get_field( 'my_field', $post_id );
```

Get a value without formatting

In this example, the field “image” is an image field which would normally return an Image object. However, by passing false as a 3rd parameter to the get_field function, the value is never formatted and returned as is from the Database.

Please note the second parameter is set to **false** to target the current post.

```
$image = get_field('image', false, false);
```

Get an escaped (HTML safe) field value (since ACF 6.2.6)

In this example, the field “wysiwyg” is a wysiwyg field which returns a string containing the contents of the WYSIWYG field. However, by passing true as the fourth parameter, we’ll ask the field to make sure it’s escaped. In the case of the WYSIWYG field, this escaping happens before filters are run on the field, so shortcodes will still work, but any `<script>` or `<iframe>`s saved in the field will be removed.

Please note the second parameter is set to **false** to target the current post, and the third parameter is set to **true** which is required when getting an escaped value, otherwise a `_doing_it_wrong` will be thrown.

```
$escaped_wysiwyg = get_field('wysiwyg', false, true, true);
```

Escaping the value

Before you output any value provided by `get_field`, you should escape it using one of the WordPress escaping functions to ensure your field values aren’t able to be used for malicious purposes. For example using `wp_kses_post`

```
$value = get_field( "text_field" );


if( $value ) {
    echo wp_kses_post( $value );
}
```

Notes

Prior to ACF 5.11, `get_field()` could be used to retrieve the values of options or meta for items not associated with ACF fields. This had security implications as site options could be exposed. `get_field()` also could retrieve values of fields that are no longer registered with ACF.

This was updated in ACF 5.11, see the Updates to ACF Field Functions in 5.11 resource for more info.

□□□ #1
□ □□□ □□□ 2026-08-29 15:44:03 UTC
□ □□□ □□□ 2026-08-29 15:44:03 UTC