



FREE eBook

LEARNING

advanced-custom- fields

Free unaffiliated eBook created from
Stack Overflow contributors.

#advanced-

custom-

fields

Table of Contents

About.....	1
Chapter 1: Getting started with advanced-custom-fields.....	2
Remarks.....	2
Versions.....	2
Examples.....	2
Installation/Setup.....	2
Chapter 2: Custom Options page.....	4
Syntax.....	4
Parameters.....	4
Examples.....	4
The simplest Options page.....	4
Advanced Options page.....	4
Chapter 3: Date Time Picker.....	6
Introduction.....	6
Examples.....	6
Ouput date-time value.....	6
Use two date-time fields to show a field.....	6
Chapter 4: get_field().....	7
Syntax.....	7
Parameters.....	7
Remarks.....	7
Examples.....	7
Simple Example.....	7
Get Field Value from a Different Post.....	7
Get Field Value from an Options Page.....	8
Check if Field Exists.....	8
Sanitize get_field() output.....	8
Get Field With Radio Buttons.....	9
Chapter 5: the_field().....	10
Syntax.....	10

Parameters.....	10
Remarks.....	10
Examples.....	10
Check if field exists.....	10
Credits.....	11

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [advanced-custom-fields](#)

It is an unofficial and free advanced-custom-fields ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official advanced-custom-fields.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with advanced-custom-fields

Remarks

This section provides an overview of what advanced-custom-fields is, and why a developer might want to use it.

It should also mention any large subjects within advanced-custom-fields, and link out to the related topics. Since the Documentation for advanced-custom-fields is new, you may need to create initial versions of those related topics.

Versions

Standard Version (SV): Version 4.4.11 (10-23-2016)

SV includes the following field types:

Text, text area, number, email, password, WYSIWYG editor, image upload, file upload, select, checkbox, radio buttons, true/false, page link, post object, relationship, taxonomy, user, google maps, date picker, color picker, tab, message, and the ability to create your own custom field type.

Pro Version (PRO): Version 5.4.8 (10-17-2016)

PRO includes everything in SV, with the addition of the following field types:

Repeater, Gallery, Flexible Content, Clone, and Options Pages

PRO bundles both the SV version of the premium add-ons into one plugin. As it contains the same code-base, upgrading from SV to PRO is noted as having 100% compatibility in data.

Examples

Installation/Setup

Standard Version

To install the standard version, go to the Plugin section of your WordPress installation. Search for "Advanced Custom Fields" and install/activate. You will now have access to a new area in the WordPress administrative area labeled "Custom Fields" where you can create your first custom field group.

Pro Version

To install the pro version, download from the "Licenses and Downloads" area of the My Account

section of the ACF website. Then go to the Plugin section of your WordPress installation. Upload the file, and then install/active. You will now have access to a new area in the WordPress administrative area labeled "Custom Fields" where you can create your first custom field group. To register your copy of ACF, use the license key (also available in the "Licenses and Downloads" area of your ACF account area) and paste it in the license field under the following menu area in your WordPress installation: Custom Fields > Updates. You can upgrade an existing license at any time by logging into your ACF account and following the prompts.

Read [Getting started with advanced-custom-fields online](https://riptutorial.com/advanced-custom-fields/topic/7512/getting-started-with-advanced-custom-fields): <https://riptutorial.com/advanced-custom-fields/topic/7512/getting-started-with-advanced-custom-fields>

Chapter 2: Custom Options page

Syntax

- `acf_add_options_page($args);`
- `acf_add_options_sub_page($page);`

Parameters

Parameter	Details
<code>\$args</code>	(mixed) A string for the page title, or an array of settings. If left blank, default settings will be used.
<code>\$page</code>	(mixed) A string for the page title, or an array of settings. If left blank, default settings will be used.

Examples

The simplest Options page

```
if( function_exists('acf_add_options_page') ) {  
    acf_add_options_page();  
}
```

Add the above code to `functions.php` and an options page named 'Options' will appear in your Wordpress admin area. You now need to assign some custom fields to the page.

Advanced Options page

```
if( function_exists('acf_add_options_page') ) {  
  
    acf_add_options_page(array(  
        'page_title'      => 'Theme General Settings',  
        'menu_title'     => 'Theme Settings',  
        'menu_slug'      => 'theme-general-settings',  
        'capability'     => 'edit_posts',  
        'redirect'       => false  
    ));  
  
    acf_add_options_sub_page(array(  
        'page_title'      => 'Theme Header Settings',  
        'menu_title'     => 'Header',  
        'parent_slug'    => 'theme-general-settings',  
    ));  
  
}
```

Add the above code to `functions.php` and an options page named 'Theme Settings' will appear in your Wordpress admin area. So will a sub page named 'Header'.

You now need to assign some custom fields to those pages.

All supported arguments of both functions:

```
$args [or $page] = array(

    /* (string) The title displayed on the options page. Required. */
    'page_title' => 'Options',

    /* (string) The title displayed in the wp-admin sidebar. Defaults to page_title */
    'menu_title' => '',

    /* (string) The slug name to refer to this menu by (should be unique for this menu).
    Defaults to a url friendly version of menu_slug */
    'menu_slug' => '',

    /* (string) The capability required for this menu to be displayed to the user. Defaults to
    edit_posts.
    Read more about capability here: http://codex.wordpress.org/Roles_and_Capabilities */
    'capability' => 'edit_posts',

    /* (int|string) The position in the menu order this menu should appear.
    WARNING: if two menu items use the same position attribute, one of the items may be
    overwritten so that only one item displays!
    Risk of conflict can be reduced by using decimal instead of integer values, e.g. '63.3'
    instead of 63 (must use quotes).
    Defaults to bottom of utility menu items */
    'position' => false,

    /* (string) The slug of another WP admin page. if set, this will become a child page. */
    'parent_slug' => '',

    /* (string) The icon class for this menu. Defaults to default WordPress gear.
    Read more about dashicons here: https://developer.wordpress.org/resource/dashicons/ */
    'icon_url' => false,

    /* (boolean) If set to true, this options page will redirect to the first child page (if a
    child page exists).
    If set to false, this parent page will appear alongside any child pages. Defaults to true
    */
    'redirect' => true,

    /* (int|string) The '$post_id' to save/load data to/from. Can be set to a numeric post ID
    (123), or a string ('user_2').
    Defaults to 'options'. Added in v5.2.7 */
    'post_id' => 'options',

    /* (boolean) Whether to load the option (values saved from this options page) when
    WordPress starts up.
    Defaults to false. Added in v5.2.8. */
    'autoload' => false,

);
```

Read Custom Options page online: <https://riptutorial.com/advanced-custom-fields/topic/7531/custom-options-page>

Chapter 3: Date Time Picker

Introduction

The Date Time Picker allows a user to choose both a date (month/date/year) as well as a time of day (hours/minutes/seconds). A user can then output the date format and return format based on [PHP date()][1] acceptable methods. This is helpful for querying custom post types like events, tour dates, or even a single custom field. [1]: <http://php.net/manual/en/function.date.php>

Examples

Output date-time value

Display the date by itself (custom field is `tour_date`):

```
<p>Tour Date <?php the_field('tour_date');?></p>
```

Use two date-time fields to show a field

Let's say you want to show a text field (`announcement`) based on the time (a scheduled notification message for instance).

You need two date-time fields. In our example, one is called `start_date` and the other `end_date`.

```
<?php
$DateNow = date('Y-m-d H:i:s');
$DateStart = get_field('start_date', false, false);
$DateEnd = get_field('end_date', false, false);

if($DateNow > $DateStart && $DateNow < $DateEnd) {
    echo the_field('announcement');
}
?>
```

Read Date Time Picker online: <https://riptutorial.com/advanced-custom-fields/topic/8354/date-time-picker>

Chapter 4: get_field()

Syntax

- `get_field($field_name, $post_id, $format_value);`

Parameters

Parameter	Details
<code>\$field_name</code>	(string) The name of the field you are requesting. When setting up your field group, this is the "Field Name" setting.
<code>\$post_id</code>	(integer) This is the ID of the post that you are requesting the field name from. This can also be 'options' or a taxonomy reference. This can be left blank when inside of the WordPress loop, if requesting the current post/page ID.
<code>\$format_value</code>	Determine if you want to format the value loaded from the database. Defaults to true. Not required.

Remarks

get_field() vs. the_field()

`the_field()` automatically displays the field value on your page, where `get_field()` does not. Use `get_field()` when assigning field values to variables, or when manipulating the returned content in your code.

Examples

Simple Example

```
<?php echo get_field('my_field_name'); ?>
```

This will echo the value of the field "my_field_name" from the current post.

Get Field Value from a Different Post

```
<?php echo get_field('my_field_name', 123); ?>
```

This will echo the value of "my_field_name" from the post with 123 as its ID.

Get Field Value from an Options Page

```
<?php echo get_field('my_field_name', 'option'); ?>
```

This will echo the value of "my_field_name" from the options page created via ACF.

Check if Field Exists

```
<?php if( get_field('my_field_name') ){ ?>
    <?php echo get_field('my_field_name'); ?>
<?php }; ?>
```

This will only show the field if content exists depending on content type (i.e., an image is uploaded to the field, text is entered, it is selected, etc.).

Sanitize get_field() output

It's a good idea to sanitize get_field() output, especially when using Advanced Custom Fields fields front end (with acf_form()). Otherwise your site is likely vulnerable to cross-site scripting attacks (XSS).

The following function lets you use

```
echo get_field_escaped('my_custom_field', $post_id, true);
```

instead of

```
echo get_field('my_custom_field', $post_id, true);
```

The function uses esc_html as default, but let's you change this as a fourth parameter

```
echo get_field_escaped('url', $post_id, true, 'esc_url');
```

Add the following to functions.php to enable the function:

```
/**
 * Helper function to get escaped field from ACF
 * and also normalize values.
 *
 * @param $field_key
 * @param bool $post_id
 * @param bool $format_value
 * @param string $escape_method esc_html / esc_attr or NULL for none
 * @return array|bool|string
 */
function get_field_escaped($field_key, $post_id = false, $format_value = true, $escape_method = 'esc_html')
{
    $field = get_field($field_key, $post_id, $format_value);
```

```

/* Check for null and falsy values and always return space */
if($field === NULL || $field === FALSE)
    $field = ' ';

/* Handle arrays */
if(is_array($field))
{
    $field_escaped = array();
    foreach($field as $key => $value)
    {
        $field_escaped[$key] = ($escape_method === NULL) ? $value :
$escape_method($value);
    }
    return $field_escaped;
}
else
    return ($escape_method === NULL) ? $field : $escape_method($field);
}

```

Source: <https://snippets.khromov.se/sanitizing-and-securing-advanced-custom-fields-output/>

More about the different sanitization options in the WordPress Codex:

https://codex.wordpress.org/Data_Validation#Output_Sanitization

Get Field With Radio Buttons

Here's an example on how to use ACF to output differently based on options (color selections in this case). While you can use `<?php echo get_field('color_options'); ?>` to output the value directly, you can also change the markup depending on the selection.

```

<?php $option = get_field('color_options'); ?>

<?php if( $option == 'red' ){ ?>

<?php } else if( $option == 'blue' ){ ?>

<?php } else if( $option == 'green' ){ ?>

<?php } else if( $option == 'yellow' ){ ?>

<?php } ?>

```

Read `get_field()` online: <https://riptutorial.com/advanced-custom-fields/topic/7515/get-field-->

Chapter 5: the_field()

Syntax

- `the_field($field_name, $post_id);`

Parameters

Parameter	Details
<code>\$field_name</code>	the name of the field to be retrieved. eg "page_content" (required)
<code>\$post_id</code>	Specific post ID where your value was entered. Defaults to current post ID (not required). This can also be options / taxonomies / users / etc

Remarks

Use `get_field()` when assigning field values to variables, or when manipulating the returned content in your code. `the_field()` is the same as `echo get_field($field_name);`

Examples

Check if field exists

```
<?php if( get_field('text_field') ): ?>
    <?php the_field('text_field'); ?>
<?php endif; ?>
```

Here we use `get_field()` to determine if a value exists, and `the_field()` to echo it.

Read `the_field()` online: <https://riptutorial.com/advanced-custom-fields/topic/7525/the-field-->

Credits

S. No	Chapters	Contributors
1	Getting started with advanced-custom-fields	Aibrean , Community
2	Custom Options page	danjah
3	Date Time Picker	Aibrean
4	<code>get_field()</code>	Aibrean , danjah , Joe
5	<code>the_field()</code>	danjah