ACF Blocks

- Introduction
- Simplify Your Workflow
  - PHP Environment
  - Simple Templating
  - Custom Fields Compatible
  - Live Previews
- Total WordPress Compatibility
  - Native Compatibility
  - Anywhere and Everywhere
  - Block Asset Loading
  - block.json Support
- Flexible Design & Development
  - Block Versioning
  - Blocks Without Fields
  - InnerBlock Support
  - Block Context Changes
- Next Steps

Introduction

ACF Blocks are customizable, dynamic, and integrate deeply with custom fields. A premium feature found in ACF PRO, ACF Blocks uses a powerful PHP-based framework to develop custom block types without the thorough knowledge of JavaScript or React required to create WordPress blocks.

The PHP-based nature of ACF Blocks means PHP developers can easily create bespoke solutions in line with modern WordPress theme development. ACF Blocks maintain compatibility with WordPress core, providing access to many of the same features as native blocks.

A testimonial block created with ACF Blocks.

Simplify Your Workflow

PHP Environment

ACF Blocks is a PHP framework and does not require any JavaScript. This differentiates itself from the WordPress block API which relies heavily on modern JavaScript techniques, syntax and build tools.

Simple Templating

Similar to WordPress theme development, ACF Blocks are rendered using a PHP template file or callback function, allowing full control over the HTML output.

Custom Fields Compatible

ACF Blocks offer full compatibility with all field types including both the Repeater and Clone fields! Build your field group and attach it to your newly-registered block using ACF's familiar Location rules.

ACF Location rules configured to show this field group if the Block is equal to Testimonial.

It's a similar story for template functions. Whether you are loading a field value via `get_field()`, or looping over a Repeater field using `have_rows()`, the experience remains familiar and consistent with regular theme development.

Live Previews

Content changes, and so do block previews! When editing an ACF Block, the HTML will update in the backend giving you a real-time preview of your content. If you use an ACF Block in a Query Loop block or in a Full Site Editing Block Theme, your block preview will automatically update as your query changes.

Total WordPress Compatibility

Native Compatibility

ACF Blocks maintain native compatibility with WordPress core. This allows features such as “alignment”, “anchor” and even Reusable Blocks to work!

ACF Blocks are not compatible with the standalone Gutenberg plugin, as the plugin forces on new, experimental features and API versions which are not yet supported by ACF.

Anywhere and Everywhere

ACF Blocks are not tied to metadata, meaning they can be used anywhere in WordPress, and multiple times per post.

Block Asset Loading

ACF Blocks supported enqueueing block-specific assets before WordPress did! WordPress now supports this, requiring assets be loaded using specific WordPress asset configuration keys. Blocks created with earlier versions of ACF Blocks should be migrated to use the WordPress provided methods.

block.json Support

ACF Blocks are registered with the standard WordPress function, `register_block_type()`, which loads a `block.json` file containing your configuration.

```
add_action( 'init', 'register_acf_blocks' );
function register_acf_blocks() {
    register_block_type( __DIR__ . '/blocks/your-example-block' );
}
```

Flexible Design & Development

Block Versioning

Introduced in ACF 6.0, block versioning allows you to opt in to enable new features, or opt out to ensure blocks you’ve already created maintain backwards compatibility.

Blocks Without Fields

ACF Blocks can be created without fields, useful in cases where you want to show post templating elements such as titles or post content, or when you want to display fields which are added on the post being displayed, rather than from the block.

InnerBlock Support

ACF Blocks include full support for InnerBlocks, allowing you to nest blocks inside one another. This can be achieved by adding `<InnerBlocks />` to your PHP template file for the block.

Block Context Changes

ACF Blocks support block context. This means you can configure your blocks to pass data to their InnerBlocks, including data from the ACF fields on the parent block. When anything about this block context changes, it will automatically reload the template of the child blocks, while your template handles any changes the parent block's field values may cause.

Next Steps

ACF Blocks give you the functionality of native WordPress blocks, in a familiar PHP environment.

This feature of ACF PRO lets you give your clients perfectly tailored solutions that combine the power of Advanced Custom Fields with the ease of content editing and layout offered by the modern WordPress Site Editor.

Next, we'll look at some of the key concepts used in WordPress blocks, and how they apply when creating ACF Blocks.

Next: Key Concepts

Questions about ACF Blocks? Check out our FAQ.

□□□ #2
□ □□□ □□□ 2026-08-29 15:44:03 UTC
□ □□□ □□□ 2026-08-29 15:44:03 UTC