

GETTING STARTED WITH ACF GUTEN-BLOCKS



KRISTIN FALKNER
WORDCAMP KANSAS CITY 2019
KRISTINFALKNER.COM
TWITTER: @KRISTINCODESWP



Documentation PRO Add-ons Blog Support

Download

Account

Edit content with

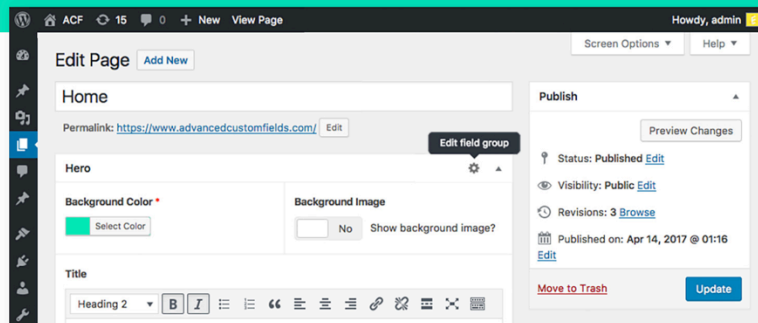
Advanced Custom Fields for WordPress Developers.

Content editing made easy

Use the Advanced Custom Fields plugin
to take full control of your WordPress
edit screens & custom field data.

Download for free

Get PRO



Learn more about ACF @
<https://www.advancedcustomfields.com>

What is ACF?

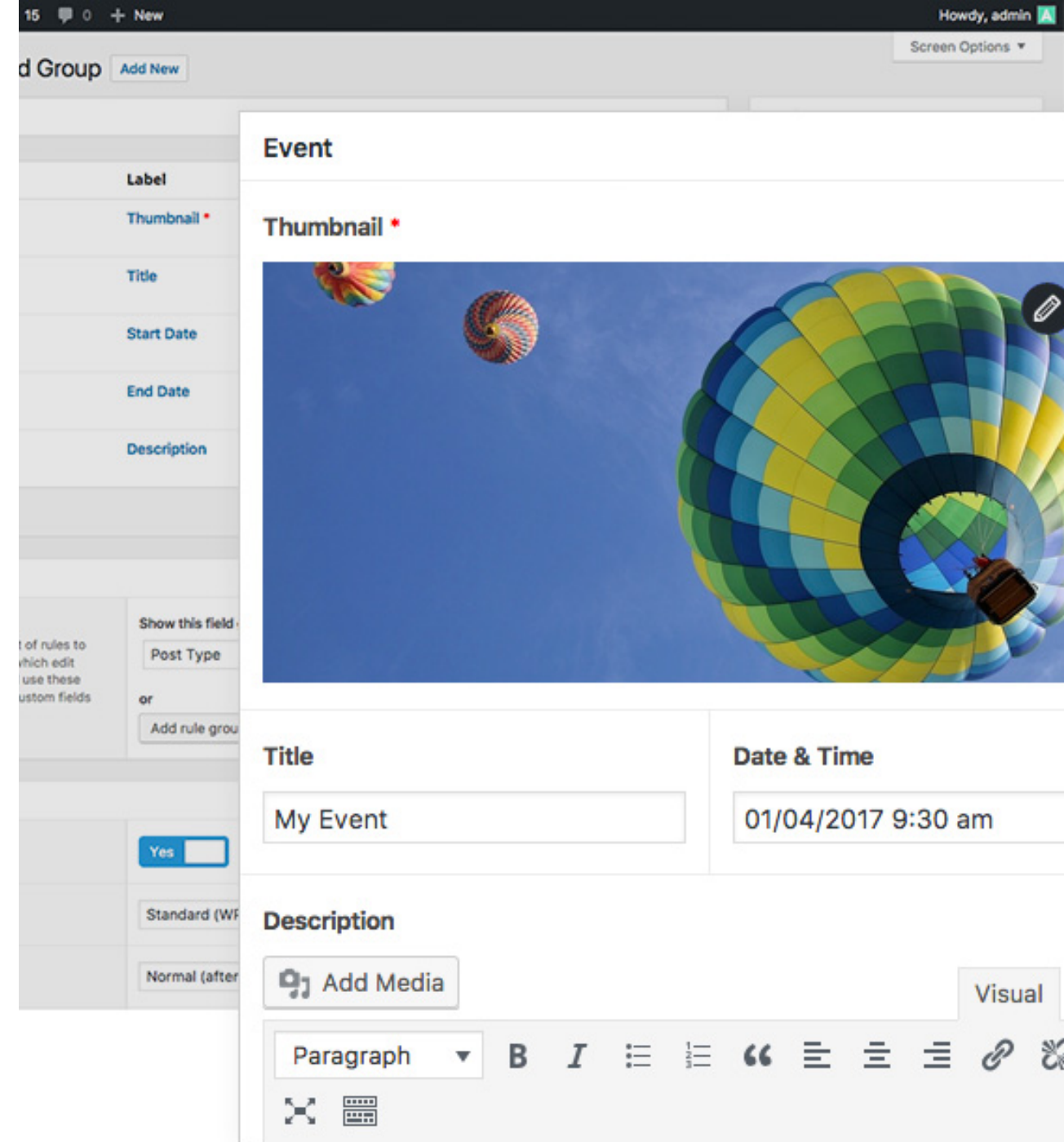
ACF = ADVANCED CUSTOM FIELDS

SIMPLIFIES CUSTOM FIELDS

EASY TO LEARN & USE

CREATE CLEAR ADMIN EXPERIENCES

FREE & PRO LICENSES
(PRO REQUIRED FOR ACF BLOCKS)



Gutenberg Editor Rundown

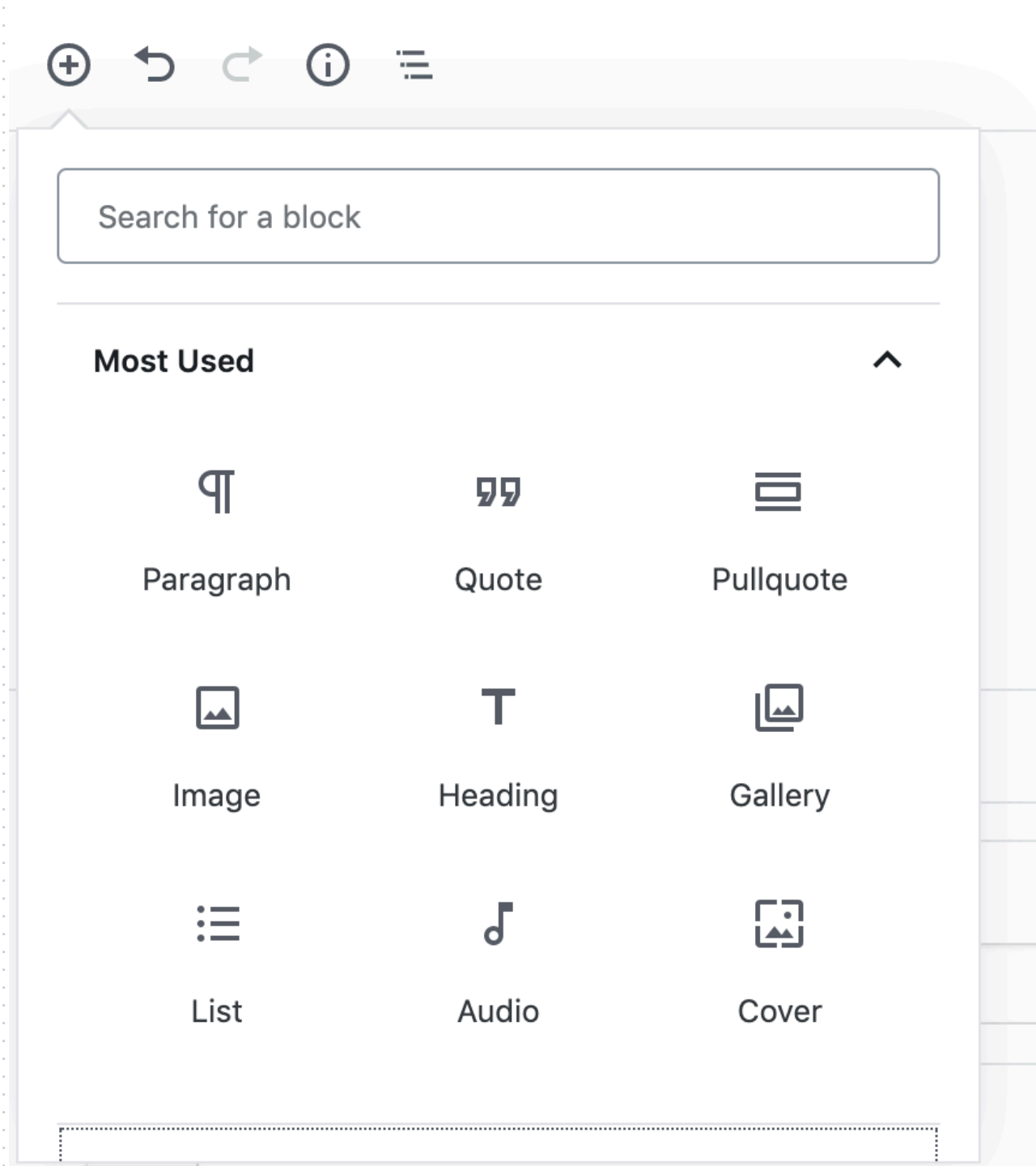
BUILT WITH REACT

EDITOR MADE UP OF BLOCKS

FLEXIBLE, PAGE BUILDER APPROACH

VISUAL STYLE IN CMS

CAN BE USED SELECTIVELY



When is **ACF** a good fit for blocks?

There are several options for building custom Gutenberg blocks out there (create-guten-block, Block Lab, Lazy Blocks, etc.) When does ACF make the most sense?



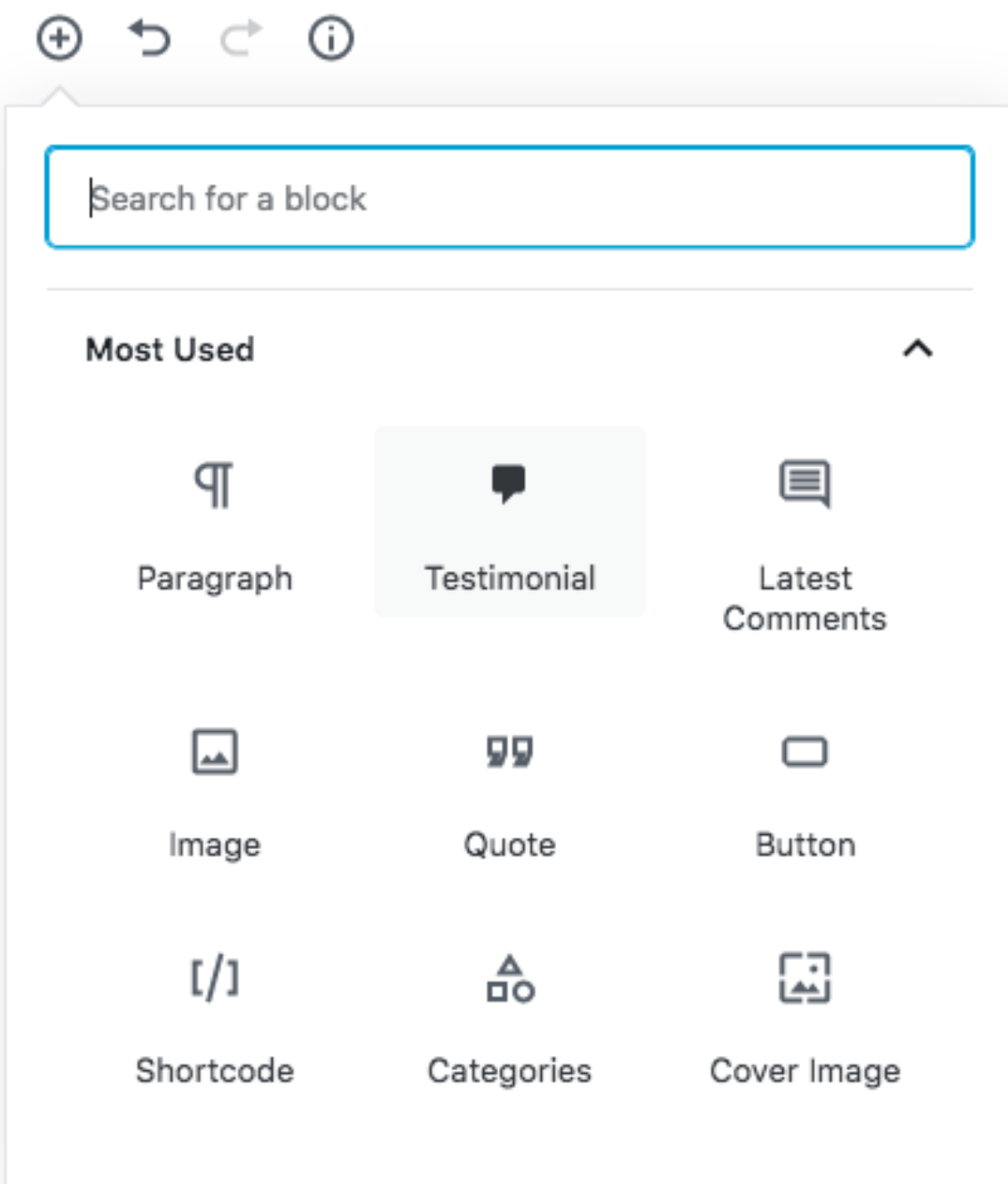
ALREADY USING ACF

If you use ACF for other parts of your project, a lot of code for blocks will be familiar. Plus, minimizes need for more plugin(s) just for block creation or libraries.



MORE COMFORTABLE W PHP/HTML/CSS THAN REACT/JS

While it's still recommended to “learn JavaScript deeply”, ACF Blocks creates a bridge for block creation for those not quite there yet.



Steps for Creating an **ACF Block**

1. Register block(s)
2. Assign fields to block(s) (OPTIONAL)
3. Set up block(s) display

```
add_action('acf/init', 'klf_custom_acf_blocks');
function klf_custom_acf_blocks() {

    // check function exists.
    if( function_exists('acf_register_block_type') ) {

        // Register a testimonial block.
        acf_register_block_type(array(
            'name' => 'testimonial',
            'title' => __('Testimonial'),
            'description' => __('A custom testimonial block.'),
            'render_template' => 'blocks/testimonial/testimonial.php',
            'category' => 'formatting',
        ));
    }
}
```

Registering ACF Block

ACF_REGISTER_BLOCK_TYPE BASICS

NAME: A unique string (think “slug”)

TITLE: Display title for block, shown in editor

DESCRIPTION: Short description

CATEGORY: Section where block will appear in editor. Options: common, formatting, layout, widgets, embed (custom categories can be created)

RENDER_TEMPLATE: Tells where the code for the block display can be found.

```
<?php

add_action('acf/init', 'klf_custom_acf_blocks');
function klf_custom_acf_blocks() {

    // check function exists.
    if( function_exists('acf_register_block_type') ) {

        // Register a testimonial block.
        acf_register_block_type(array(
            'name' => 'testimonial',
            'title' => __('Testimonial'),
            'description' => __('A custom testimonial block.'),
            'render_template' => 'blocks/testimonial/testimonial.php',
            'category' => 'formatting',
        ));
    }
}
```

```
// Specifying a relative path within the active theme  
'render_template' => 'blocks/testimonial/testimonial.php',
```

```
// Specifying a function  
'render_callback' => 'my_acf_block_render_callback',
```

×

RENDER_TEMPLATE REFERENCING SEPARATE FILE

If using this method, the code for the front-end display of the block exists in a PHP file, most likely in a custom area of the theme. I like to create a BLOCKS folder in my theme and place the templates there.

×

RENDER_CALLBACK REFERENCING FUNCTION

Instead of calling out a separate file that contains the block's code, this references a function that will output the frontend display for the block.

Registering ACF Block

THE EXTRAS

ICON: Dashicons or custom icons

KEYWORDS: Search terms for block

POST_TYPES: Limit display to certain types

MODE: Auto, preview, edit

ALIGN: Default alignment: left, center, right, wide, full

SUPPORTS: Block API supports elements like “multiple”. Reference Block API or ACF documentation for full available options here.

```
// Check if function exists and hook into setup.
if( function_exists('acf_register_block_type') ) {
    add_action('acf/init', 'register_acf_block_types');
}

function register_acf_block_types() {

    // register a testimonial block.
    acf_register_block_type(array(
        'name' => 'testimonial',
        'title' => __('Testimonial'),
        'description' => __('A custom testimonial block.'),
        'render_template' => 'blocks/testimonial/testimonial.php',
        'category' => 'formatting',
        'icon' => 'testimonial',
        'keywords' => array( 'testimonial', 'quote' ),
        'post_types' => array('post','page'),
        'mode' => 'auto',
        'align' => 'full',
        'enqueue_style' => get_template_directory_uri() . '/blocks/testimonial/
        testimonial.css',
        'enqueue_script' => get_template_directory_uri() . '/blocks/testimonial/
        testimonial.js',
    ));
}
```

```
// Specifying a custom HTML svg for the block
'icon' => '<svg viewBox="0 0 24 24" xmlns="http://www.w3.org/2000/
svg"><path fill="none" d="M0 0h24v24H0V0z" /><path d="M19
13H5v-2h14v2z" /></svg>',|
```

```
// Specifying colors
'icon' => array(
    // Specifying a background color to appear with the icon e.g.: in
    the inserter.
    'background' => '#7e70af',
    // Specifying a color for the icon (optional: if not set, a
    readable color will be automatically defined)
    'foreground' => '#fff',
    // Specifying a dashicon for the block
    'src' => 'book-alt',
),
```

×

SVG ICONS

INCLUDE SVG CODE IN ARRAY

If using this method, you place the SVG code for your custom icon in the ACF block declaration array

×

STYLED ICONS

ADD COLORS

If you want to add custom colors, you can do so via this route

Registering ACF Block

THE EXTRAS

ICON: Dashicons or custom icons

KEYWORDS: Search terms for block

POST TYPES: Limit display to certain types

MODE: Auto, preview, edit

ALIGN: Default alignment: left, center, right, wide, full

SUPPORTS: Block API supports elements like “multiple”. Reference Block API or ACF documentation for full available options here.

```
// Check if function exists and hook into setup.
if( function_exists('acf_register_block_type') ) {
    add_action('acf/init', 'register_acf_block_types');
}

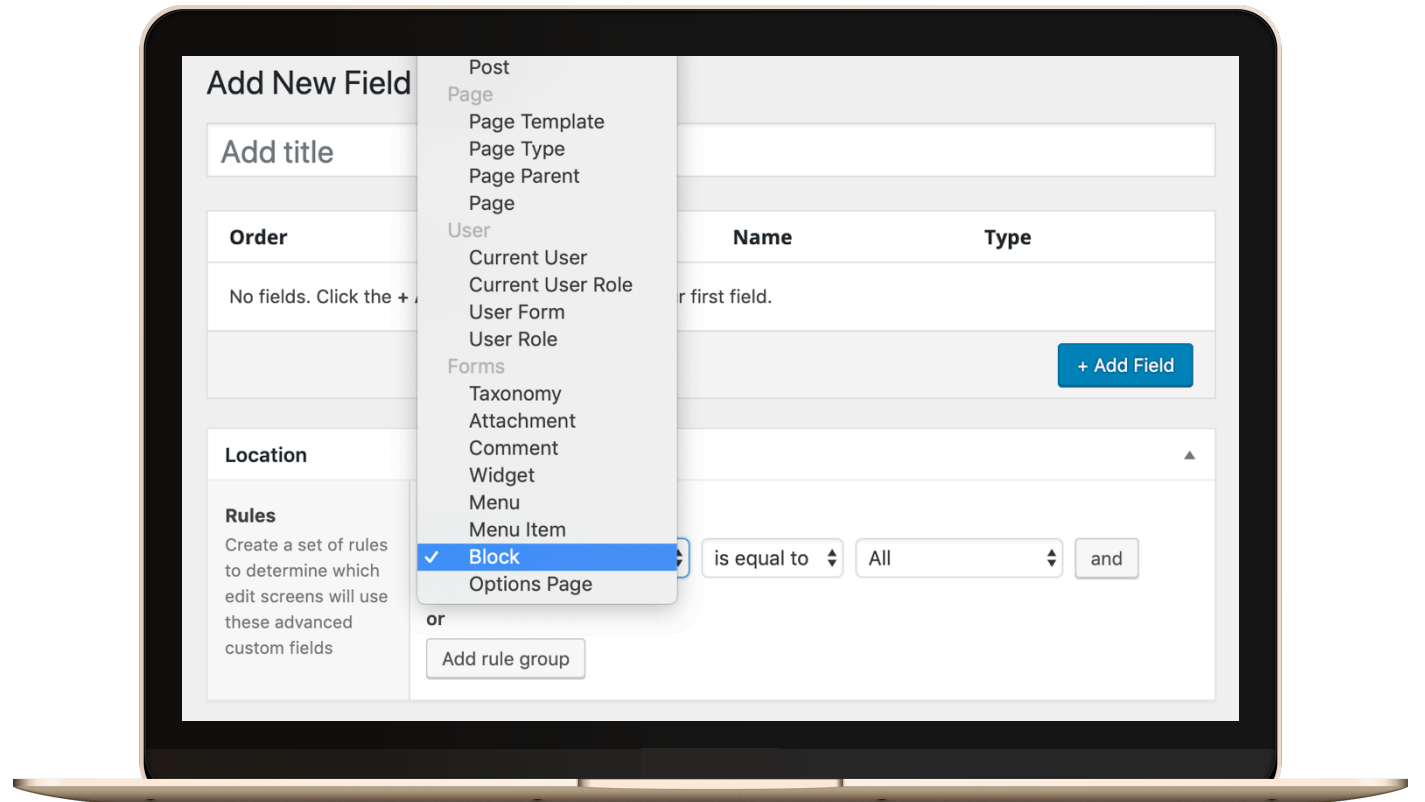
function register_acf_block_types() {

    // register a testimonial block.
    acf_register_block_type(array(
        'name' => 'testimonial',
        'title' => __('Testimonial'),
        'description' => __('A custom testimonial block.'),
        'render_template' => 'blocks/testimonial/testimonial.php',
        'category' => 'formatting',
        'icon' => 'testimonial',
        'keywords' => array( 'testimonial', 'quote' ),
        'post_types' => array('post','page'),
        'mode' => 'auto',
        'align' => 'full',
        'enqueue_style' => get_template_directory_uri() . '/blocks/testimonial/
        testimonial.css',
        'enqueue_script' => get_template_directory_uri() . '/blocks/testimonial/
        testimonial.js',
    ));
}
```

Assigning Fields to Block

FOLLOWS NORMAL ACF PATTERN

Create fields like you would a post or page for your block. You'll see block as an option in Location Rules. You can create fields to show on *all* blocks or just a *specific* block (most common)



Dashboard

Posts

Media

Forms

Pages

Comments

Appearance

Plugins

Users

Tools

Settings

Custom Fields

Field Groups

Add New

Tools

Updates

Options

Appearance

Plugins

Users

Tools

Add New Field Group

Testimonial Block

Order	Label	Name	Type
1	Testimonial *	testimonial	Text Area
2	Author *	author	Text
3	Role *	role	Text
4	Image *	image	Image
5	Background Color *	background_color	Color Picker
6	Text Color *	text_color	Color Picker
			+ Add Field

Location

Rules

Create a set of rules to determine which edit screens will use these advanced custom fields

Show this field group if

Block

is equal to

Testimonial

and

or

[Add rule group](#)

Screen Options

Publish

Status: Active

[Publish immediately](#)

[Publish](#)



Testimonial *

This is a testimonial.



Author *

John Doe

Role *

CEO of This Company

Background Color *



Select Color

Text Color *



Select Color

```

/**
 * Testimonial Block Template.
 *
 * @param array $block The block settings and attributes.
 * @param string $content The block inner HTML (empty).
 * @param bool $is_preview True during AJAX preview.
 * @param (int|string) $post_id The post ID this block is saved to.
 */

// Create id attribute allowing for custom "anchor" value.
$id = 'testimonial-' . $block['id'];
if( !empty($block['anchor']) ) {
    $id = $block['anchor'];
}

// Load values and defaults.
$text = get_field('testimonial');
$author = get_field('author');
$role = get_field('role');
$background_color = get_field('background_color');
$text_color = get_field('text_color');

?>
<div id="<?php echo esc_attr($id); ?>" class="testimonial">
    <blockquote class="testimonial__blockquote">
        <span class="testimonial__text"><?php echo $text; ?></span>
        <span class="testimonial__author"><?php echo $author; ?></span>
        <span class="testimonial__role"><?php echo $role; ?></span>
    </blockquote>
    <style type="text/css">
        #<?php echo $id; ?> {
            background: <?php echo $background_color; ?>;
            color: <?php echo $text_color; ?>;
        }
    </style>
</div>

```

Example

This is a testimonial.

John Doe

CEO OF THIS COMPANY

Getting Backend Block Styling

Because Gutenberg is a visual editor, it differs from traditional ACF fields where layout styling is usually not brought into editor. You can achieve styling in the admin multiple ways.



INDIVIDUAL DECLARATIONS

You can use `enqueue_style` and `enqueue_script` directly in `acf_register_block_type` for individual CSS or JS files.



GUTENBERG-SPECIFIC STYLESHEET W EVERYTHING

You can also create one CSS &/or JS file with the code for any custom blocks and query that in.

×

ENQUEUE_BLOCK_ASSETS FRONT & BACKEND

This can be used to bring CSS & JS into both the front & back end of your site instead of doing individual block CSS & JS.

×

ENQUEUE_BLOCK_EDITOR_ASSETS BACKEND ONLY

This can be used to bring CSS & JS into the backend editor of the site.

×

DON'T WANT TO MESS WITH IT?

IS_ADMIN() CONDITIONAL

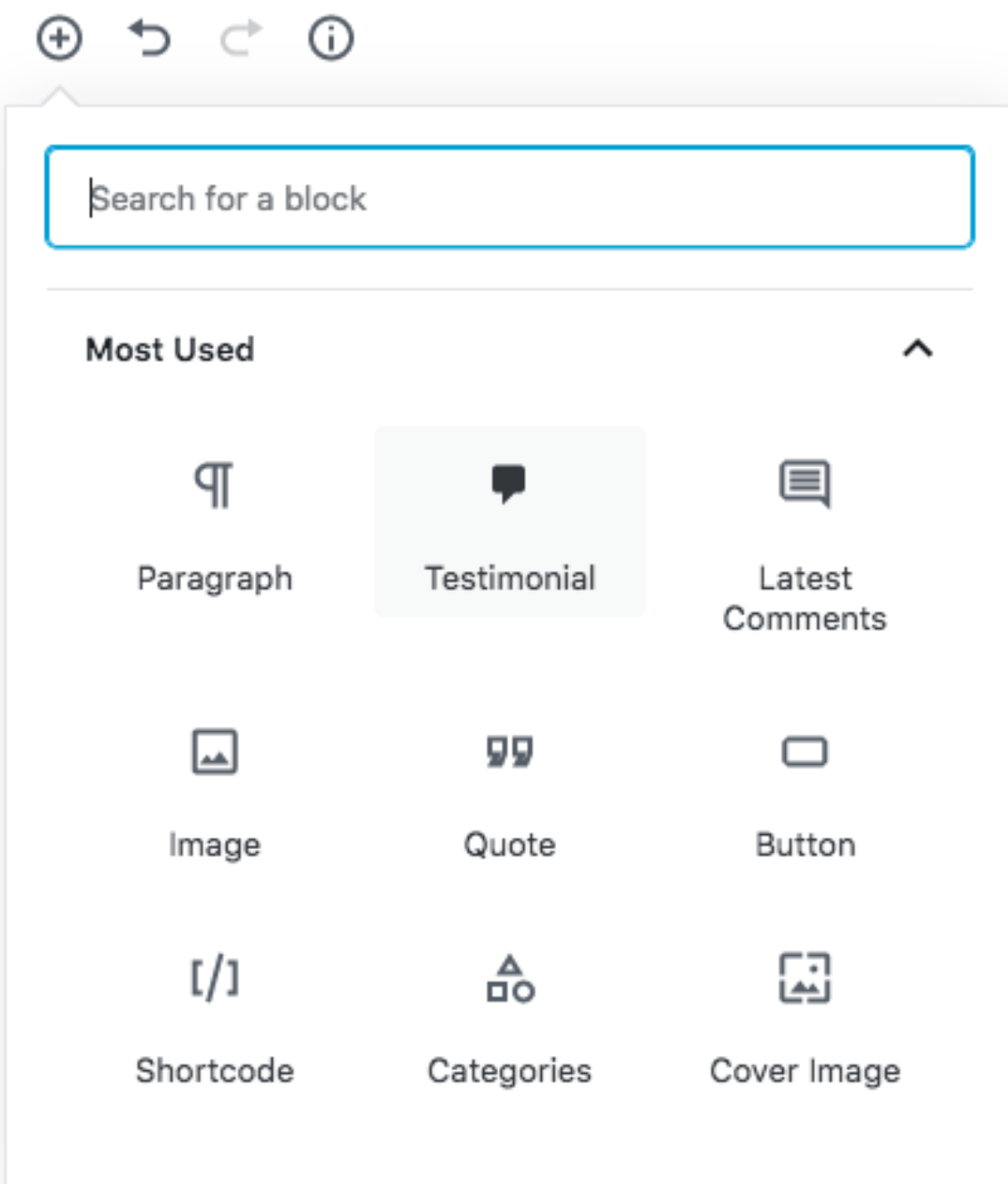
Put placeholder text to display in good ol' Guty and just style front-end like any other module.

Example

This is a testimonial.

John Doe

CEO OF THIS COMPANY



Example Uses for Custom ACF Blocks

1. Testimonials
2. Email Opt In Box
3. Photo Carousel/Slider
4. Sponsor/Ad Blocks
5. Guest Author Box
6. ...and much more!

ACF Blocks Resources

ACF Blocks Documentation -

<https://www.advancedcustomfields.com/resources/blocks/>

ACF Blocks - <https://acfblocks.com/>

Deeper into Gutenberg / JS Skills

Gutenberg Courses - <https://gutenberg.courses/>

Wes Bos Courses - <https://wesbos.com/courses/>