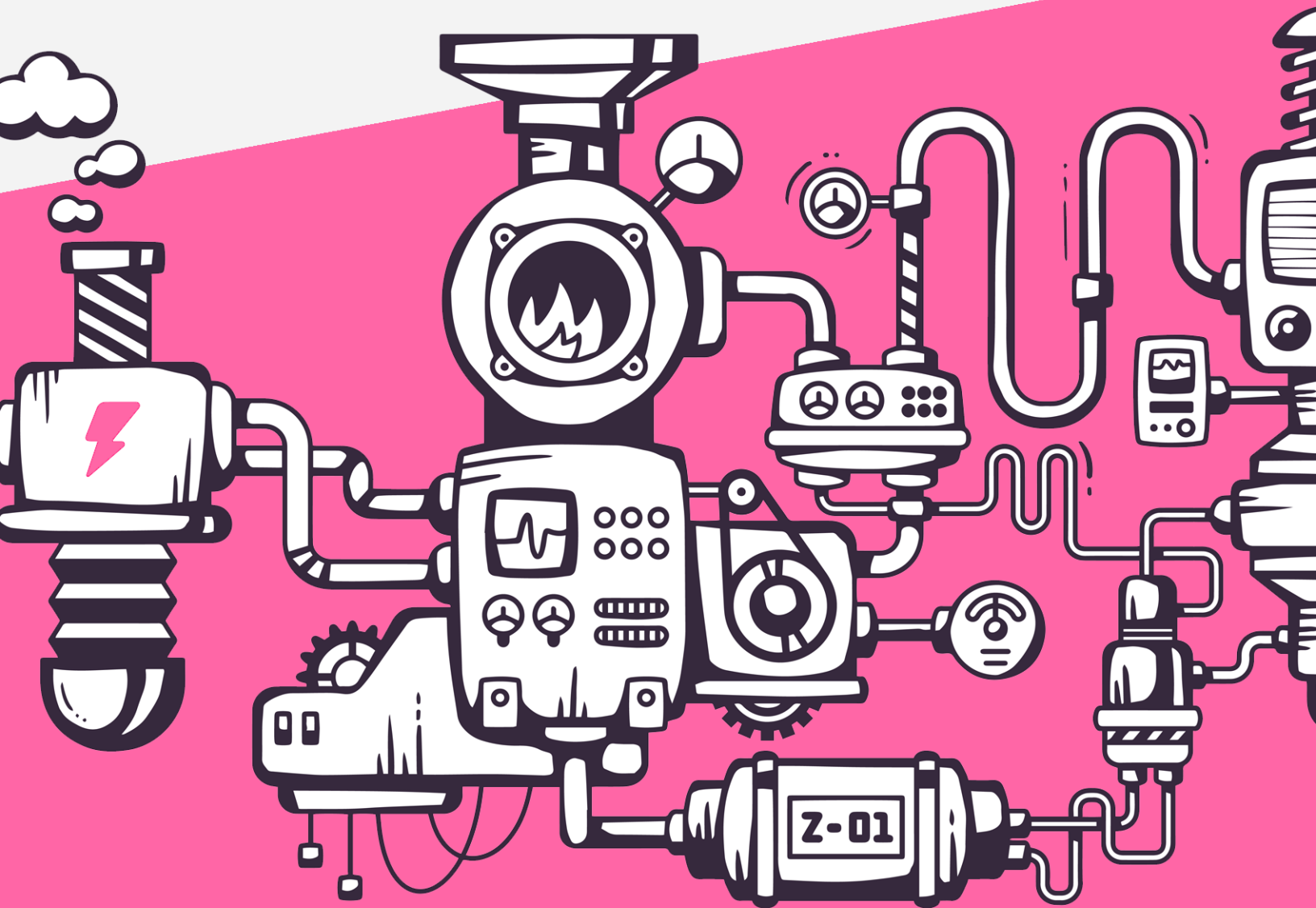


HOW TO IMPLEMENT:  
**CUSTOM POST TYPES &  
CUSTOM FIELDS**



WordPress isn't just limited to blogging anymore — it's a fully featured content management system capable of displaying and formatting any kind of content you give it. So let's talk about two great ways to extend WordPress: custom post types and custom fields.

## □ Custom Post Types

Looks like our test site, Monday Morning Mug (which we used in our email integration blog post), has a new problem to deal with. Our blog has gotten so large that we've hired a full-time coffee reviewing staff. They've been busy brewing coffee and writing up reviews. They've been posting these reviews to the blog, but our readers told us that all of these reviews are getting in the way of our "real" blog content. We need to make a separate place for our coffee reviews, but normal WordPress pages just aren't going to cut it. The reviewers have also been suggesting that they'd love to have a place to easily store information, such as the country the coffee is from and tasting notes, on the website. This sounds like a perfect use for a custom post type, a WordPress feature that allows us to create new content sections on our site.

Custom post types are incredibly flexible and can be implemented for any kind of content: quotes, videos, and different shades of paint are all examples of what a custom post type could be.

We're going to want to add a new section of our website in the [/coffees/](#) subdirectory that is full of coffee reviews. This is going to require a bit of code wrangling, but it should be a relatively easy process.

*Note: To do this, you'll need to be able to edit the .php files in your WordPress install directory, which usually requires FTP/SFTP/SSH access. Make sure you have this before you proceed.*

Every WordPress theme can utilize a `functions.php` file, which is located at: `/wp-content/themes/your_current_theme/functions.php`

This file is generally used for theme-specific site changes as well as smaller custom site modifications, like the one we're about to do.

*Note: We will be adding code to the `functions.php` file, which will mean it is theme-specific. If you add this to your `functions.php` file and then change the theme on your WordPress installation, you will remove the custom post type from your website. You can maintain custom content types by utilizing a plugin, such as Custom Post Type UI.*

1.) Open up your `functions.php` file and add the following code at the bottom of the file:

```
function create_new_post_type() {  
    $labels = array(  
        'name' => __('Coffee Reviews'),  
        'singular_name' => __('Coffee Review'),  
    );  
    $args = array(  
        'labels' => $labels,  
        'public' => true,  
        'has_archive' => true,  
        'menu_position' => 5,  
        'description' => 'Reviews And Types of Coffee',  
        'rewrite' =>  
            array('slug' => 'reviews'),  
        'supports' =>  
            array('title',  
                'comments', 'editor',  
                'thumbnail', 'custom-fields', 'revisions'),  
    );  
    register_post_type('Reviews', $args);  
}  
add_action('init', 'create_new_post_type');
```

Because we don't particularly like utilizing code without an idea of what it does, let's walk through what this code does.

*Note: You may have noticed that `$labels` and `$args` are defined outside of the `register_post_type()` function even though they are just passed as arguments to that function later on. This method aids organization and readability, but you can condense both arrays into a single argument. It would just be more difficult to understand.*

The `$args` array contains all of the specifics we need to define our new content type. Here's what these do:

- `'labels'` – An array defining what we will call our content type. There are many more labels available that you can use, and a full list is available in the documentation here.
- `'public'` – A boolean that defines whether or not the post is viewable outside of the Admin panel.
- `'has_archive'` – If true, this boolean will create an archive of this type of custom posts, just like standard WordPress posts.
- `'menu_position'` – This integer changes where your custom post navigation button goes. The number 5 indicates that it will go directly below the posts tab. For a full list of numeric codes, check out the same documentation.
- `'description'` – Describes the post type.
- `'rewrite'` – This changes the slug to be “reviews,” which will help create a usable permalink structure. (Can also be boolean; check the documentation for more information.)

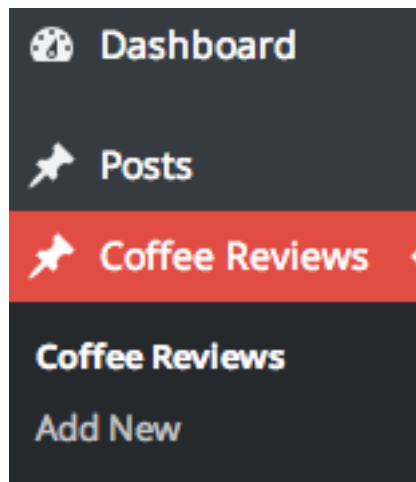
- **'supports'** – An array defining what native WordPress editing/post features are enabled. This one is pretty flexible, and there a lot more options, but we will be needing at least **custom-fields** and **editor** for this post.

The **register\_post\_type()** function is called next, which creates a new post type called Reviews using the array of arguments we just defined.

*Finally, we have to call one last function to complete the process. **register\_post\_type()** should always be initialized by the action 'init'. We can hook our new function to init by calling*

```
add_action('init', 'create_new_post_type');
```

Awesome! Now if we log into our WordPress admin panel, we should see a new tab on the left called “Reviews.”



## 2.) Update the permalink structure.

Now that we have Reviews set up as a custom post type, we need to update the permalink structure so our readers can easily find our posts. Plus it makes our URLs much easier to read and SEO friendly.

Under the Settings -> Permalinks menu bar, you'll find a bunch of common options for permalink structures. We're going to change the custom setting to match the field `/%category%//%postname%/`

This means WordPress will generate URLs to match the category and title of our posts.

*Note: This is a site-wide change and will affect your posts in other categories.*

### Permalink Settings

By default WordPress uses web URLs which have question marks and lots of numbers in them; however, WordPress offers you the ability to create a custom

#### Common Settings

<input type="radio"/> Default	<code>http://mondaymorningmug.flywheelsites.com/?p=123</code>
<input type="radio"/> Day and name	<code>http://mondaymorningmug.flywheelsites.com/2014/05/29/sample-post/</code>
<input type="radio"/> Month and name	<code>http://mondaymorningmug.flywheelsites.com/2014/05/sample-post/</code>
<input type="radio"/> Numeric	<code>http://mondaymorningmug.flywheelsites.com/archives/123</code>
<input type="radio"/> Post name	<code>http://mondaymorningmug.flywheelsites.com/sample-post/</code>
<input checked="" type="radio"/> Custom Structure	<code>http://mondaymorningmug.flywheelsites.com</code> <input type="text" value="/%category%//%postname%/"/>

Now let's take a look at the URL of a post our reviewers wrote for the Sumatra Ketiar coffee:

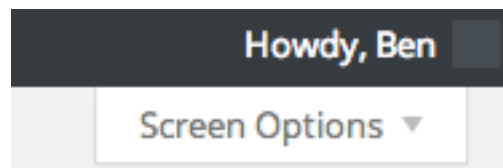


Fantastic! Check out that URL. We have reviews separated into their own directory and the URL is written in plain English. Now our readers know exactly what they're going to get when they click on a link.

## □ Custom Fields

Although our reviewers love the new Review section, they did ask us to incorporate something on the back end that allows them to enter common info that they write about every coffee, such as country of origin and tasting notes. This sounds like the perfect use of custom fields, which, luckily, are very easy to implement.

1.) On an individual review editor, you will see the screen options button at the top right corner of the editor:



which expands to this when clicked:

### Show on screen

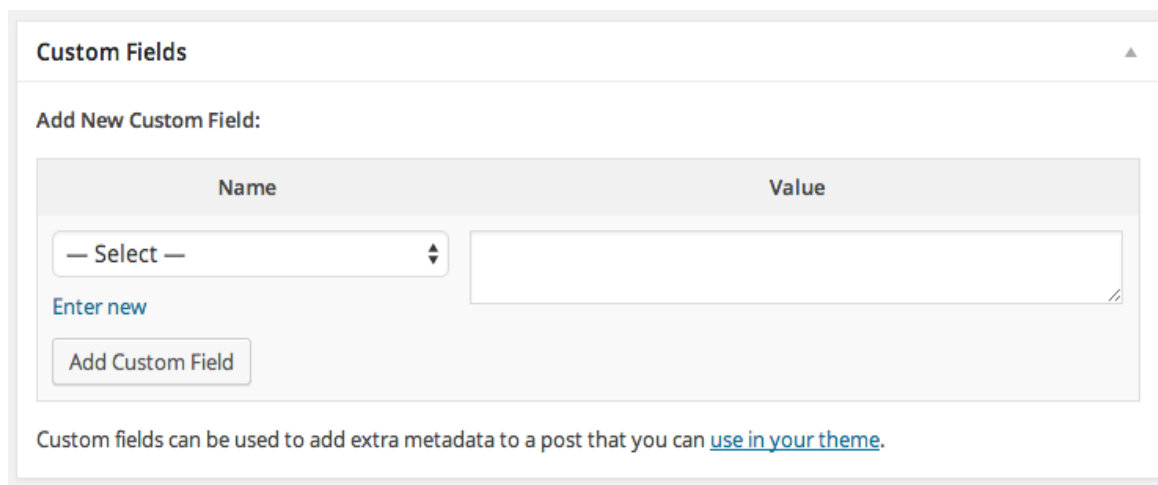
☒ Featured Image ☒ Custom Fields ☒ Discussion ☒ Comments ☐ Slug

### Screen Layout

Number of Columns: ☐ 1 ☒ 2

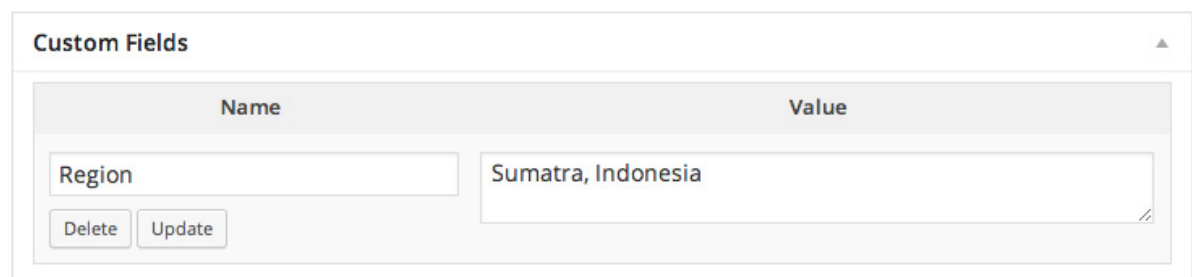
## 2.) Check the Custom Fields box.

Below the editor, you'll see the Custom Fields box that we just enabled.



The screenshot shows the 'Custom Fields' box with the title 'Custom Fields' and a collapse icon. Below the title is the section 'Add New Custom Field:'. It contains a table with two columns: 'Name' and 'Value'. Under the 'Name' column is a dropdown menu with the text '— Select —' and a double-headed arrow. Below the dropdown is a link that says 'Enter new'. Under the 'Value' column is a large text input field. Below the input field is a button that says 'Add Custom Field'. At the bottom of the box is a note: 'Custom fields can be used to add extra metadata to a post that you can [use in your theme](#).'

3.) You can type in anything in both boxes; anything you type and add in the Name field will be saved for later use. Here, we list the area and country that our coffee is from.



The screenshot shows the 'Custom Fields' box with the title 'Custom Fields' and a collapse icon. Below the title is a table with two columns: 'Name' and 'Value'. Under the 'Name' column is a text input field containing the word 'Region'. Below the input field are two buttons: 'Delete' and 'Update'. Under the 'Value' column is a large text input field containing the text 'Sumatra, Indonesia'.

Custom fields make it incredibly easy to standardize data that matters for a category across every post in a separate place. Now we just have to get that information to display on our review.incredibly easy to standardize data that matters for a category across every post in a separate place. Now we just have to get that information to display on our review fields box that we just enabled.



Custom Fields

Add New Custom Field:

Name	Value
<div><div>— Select —</div><div>Enter new</div><div>Add Custom Field</div></div>	<div></div>

Custom fields can be used to add extra metadata to a post that you can [use in your theme](#).

Custom fields make it incredibly easy to standardize data that matters for a category across every post in a separate place. Now we just have to get that information to display on our review.

4.) We're going to add this code tag to one of our theme documents:

```
<br />&lt;?php the_meta(); ?&gt;<br />
```

Note: In this example, we're going to put this tag in our content.php file, but you will need to find a suitable location for your table of values to go. You should also style this tag to match the rest of your theme, which you can read more about [here](#).

Checking back to our blog, we can see that our tag has inserted a list into our review.

## Sumatra Ketiara

 May 29, 2014  Ben  Edit

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam a bibendum lacus. Suspendisse commodo diam quis purus ultricies lobortis. Integer quis pharetra arcu. Integer ac blandit risus. Cras et dignissim turpis. Vestibulum at ligula erat. Donec hendrerit tellus ac mattis aliquam. Donec leo purus, dignissim sit amet faucibus at, aliquam in massa. Proin cursus convallis pretium.

- Region: Sumatra, Indonesia
- Tasting Notes: Plum, caramel sweet, bourbon
- Method: Wet Hulled

Swanky! Now our readers and our reviewers are both happy. We were able to build a custom post type that allowed us to save reviews about coffee, and we were able to give our reviewers a place to store all of their important information for every different review.



# FLYWHEEL

For more content you can put to work today, visit  
<http://getflywheel.com>.  
Or follow us on Twitter at @HeyFlywheel.