

Starter Patterns



- [Starter page patterns](#)
 - [How to create a page pattern](#)
 - [Starter template patterns](#)
 - [How to create template type patterns](#)
 - [Supported template types](#)
 - [Using template patterns in templates](#)
-

The WordPress block editor is powerful and can handle many different layouts and designs. However, when building a theme, you must remember that not all of your users will be skilled designers or even have much interest in piecing together their own layouts. This is where starter patterns can be beneficial to them.

WordPress supports two types of starter patterns for pages (or any post type) and templates. This feature lets you create starting points for your theme users to build out their pages and templates with minimal skills.

Starter page patterns

Page patterns let you create custom patterns your theme users can access when adding a new page via **Pages > Add New** in their WordPress admin. From there, a modal will pop up and show them a selection of patterns if any are registered.

Here is what the screen looks like when creating a new page with the default Twenty Twenty-Four theme installed:

[Modal overlaying the edit page screen, showing a grid of various page layouts.](#)

This is a powerful feature because it means that you can provide well-designed starting points for your theme's users or clients. And they don't need to build from scratch. They only need to select a pattern, and it will automatically be inserted into the content area. From there, they can make any customizations they want.

How to create a page pattern

Technically, any pattern in your theme can be converted to a page pattern. All you need to do is define the correct parameters to mark it as such. As described in [Registering Patterns](#), there are two available file header fields that you must set to make this happen:

- **Block Types**: Adding `core/post-content` as one of the block types for the pattern tells WordPress that the pattern should be used for the post content.
- **Post Types**: You can add one or more post types (separated by commas) to connect the pattern.

With these two fields combined, you create a starter page pattern. Here is what the file header looks like for a fictional `/patterns/example-page.php` pattern file:

```
<?php
/**
 * Title: Example Page
 * Slug: themeslug/example-page
 * Categories: page
 * Block Types: core/post-content
 * Post Types: page
 * Viewport width: 1376
 */
?>
<!-- Block code here. -->
```

And that's literally all you must do. Define the pattern's `Block Types` and `Post Types` parameters and you have a starter page pattern.

This will work with any post type that has opted into the block editor, including both the default `post` and `page` post types.

Because page patterns are actually tied to the content, you wouldn't typically include something like a site header or footer here. The pattern is output as post content.

Starter template patterns

Like page patterns, template patterns give users a starting point when building a new template. The difference is that template patterns work from the Site Editor.

If you visit **Appearance > Editor > Templates** in your WordPress admin and click the **+** icon button for creating a new template, you should see a new **Add template** modal:

[WordPress site editing screen with a modal popup asking the user to add a new template.](#)

From there, you can select the template that you want to create. If the template type has any patterns registered for it, a new **Choose a pattern** modal will appear, overlaying the template-editing interface.

For example, when choosing the **Front Page** option with the default Twenty Twenty-Four theme, you will see this:

[A modal overlay showing four template options for the homepage from within the WordPress site ed](#)

This gives your theme users a really nice onramp for building templates. It means they don't have to create everything from the ground up and can get started down the right path.

How to create template type patterns

Unlike page patterns, you will usually not make any of your patterns a template pattern. These are starting points for an entire template, so their use cases are much more limited and specific.

Because template patterns represent an entire template, you would typically include global elements like the header, footer, sidebar, and other sections that your theme's templates display.

As noted in [Registering Patterns](#), there is a single required header field needed for your template pattern (and one optional field):

- **Template Types**: One or more template types, separated by comma, that the pattern should be associated with.
- **Inserter**: (*Optional*) Often, you will not want template patterns to be available via the inserter. To disable this, set it to `no` or `false`.

Suppose that you created a pattern that would work well for the Front Page or Home templates. Here is what a fictional `/patterns/home-template.php` file would look like:

```
<?php
/**
 * Title: Home Template
 * Slug: themeslug/home-template
 * Template Types: front-page, home
 * Viewport width: 1376
 * Inserter: no
 */
```

```
?>
<!-- Block code here. -->
```

From that point, anytime a user attempted to create a new Front Page or Home template from the WordPress Site Editor, they would be presented with your template pattern as a starting point.

Supported template types

The following list includes the templates that you can define via the `Template Types` field, but you can always reference the [Template Hierarchy](#) documentation for a full overview of templates:

- `index`
- `home`
- `front-page`
- `singular`
- `single`
- `page`
- `archive`
- `author`
- `category`
- `taxonomy`
- `date`
- `tag`
- `attachment`
- `search`
- `privacy-policy`
- `404`

Using template patterns in templates

When creating custom template patterns, it also makes sense to reuse those patterns within your templates. *Why rewrite code?*

Imagine that you created a template pattern that would work as both a Home and Index template. It would look like this:

```
<?php
/**
 * Title: Index Template
 * Slug: themeslug/index-template
 * Template Types: home, index
 * Viewport width: 1376
```

```
* Inserter: no
*/
?>
<!-- Block code here. -->
```

Now suppose that you included a `/templates/index.html` template in your theme. Instead of adding the code in two places, you can simply call the pattern from the template file:

```
<!-- wp:pattern {"slug":"themeslug/index-template"} /-->
```

To learn more about including patterns in templates, check out the [Usage in Templates](#) documentation.

```
{} #2
{} {} {} 2026-08-29 15:36:32 UTC
{} {} {} 2026-08-29 15:36:32 UTC
```