



Genesis as an onramp for Gutenberg.

WHITE PAPER

Table of contents.

Welcome	4
A BRIEF HISTORY	4
BLOCKS	5
WHAT HAPPENS TO THE CLASSIC EDITOR.....	5
Gutenberg in context	5
GUTENBERG AND PAGE BUILDERS.....	6
GUTENBERG AND THE GENESIS FRAMEWORK.....	6
Using Gutenberg with Genesis.....	6
HOW TO KNOW IF YOU HAVE GUTENBERG.....	6
EDITING CONTENT WITH GUTENBERG	7
Developing with Gutenberg and Genesis	7
THE BLOCK EDITOR: A WORK IN PROGRESS	7
Gutenberg tips and quirks for Genesis child theme developers.....	7
Genesis sample theme	7
Twenty nineteen theme	7
GENESIS DEVELOPMENT: A FRAMEWORK OF HOOKS AND FILTERS	8
TECHNICAL OVERVIEW OF GUTENBERG	8
Block API	8
Meta boxes and Gutenberg.....	8
CREATING GENESIS CHILD THEMES WITH GUTENBERG SUPPORT	9
CHILD THEME ONBOARDING	9
What an onboarding configuration file does.....	9
BLOCK-BASED HOMEPAGE	10
GUTENBERG STYLES: FRONT-END AND BACK-END	10
Examples and tutorials of Gutenberg with Genesis.....	10
ADD THEME COLOR PALETTE TO THE GUTENBERG EDITOR FOR WORDPRESS	10
ADD BLOCK FONT SIZES TO THE GUTENBERG BLOCK EDITOR FOR WORDPRESS	11
CREATING A CUSTOM GUTENBERG BLOCK.....	11
HOW TO DEPRECATE CODE IN GUTENBERG EDITOR BLOCKS.....	11

- GENESIS + THE WP ENGINE PORTAL 11
- ADD THEME SUPPORT (AND STYLES) FOR WIDE IMAGES 11
- THE TWENTY NINETEEN THEME..... 11
- The future of Genesis with Gutenberg..... 12**
- ROADMAP FOR GENESIS 2.8 AND BEYOND 12
- WHAT TO EXPECT FROM CHILD THEMES 12
- Additional resources 12**
- GUTENBERG OVERVIEW AND NEWS..... 13
- PLUGIN AND THEME DEVELOPMENT WITH GUTENBERG..... 13
- BLOCK LIBRARIES AND FRAMEWORKS..... 13
- SETTING UP A SITE WITH A STUIDOPRESS THEME..... 13
- COMMUNITY GUTENBERG DEVELOPMENT 13
- About WP Engine..... 14**



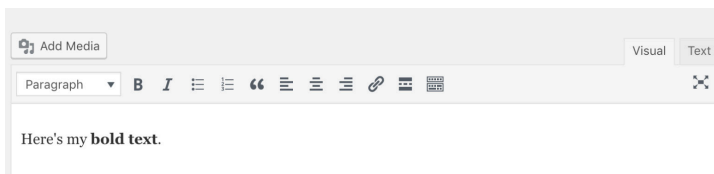
Welcome!

The purpose of this ebook is to explore Gutenberg, the new WordPress editing experience, and understand how you can use it with the Genesis Framework.

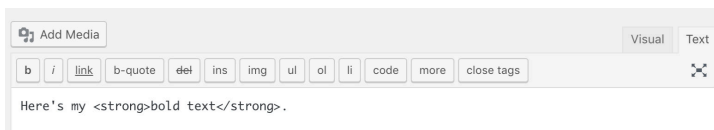
A brief history

Since the introduction of WordPress in 2003, the process of creating and editing pages and posts has not changed much. The original Classic Editor resembled a traditional word processing editor. It gave users two content editing options to choose from: the Visual editor (based on [TinyMCE](#)) or the Text editor which provided users a way to inject HTML into the content.

VISUAL EDITOR



TEXT EDITOR



While users could seamlessly toggle between the two editors, the process was not ideal for creating rich content (i.e. content with images, multimedia, social media embeds, and other elements).

If users wanted to create advanced layouts, it required [a mix of different approaches](#) that weren't always obvious for users:

- Media library/HTML for images, multimedia and approved files.
- Pasted links for embeds.
- Shortcodes for specialized assets from plugins.
- Featured images for the image at the top of a post or page.
- Excerpts for subheadings.
- Widgets for content on the side of a page.

Even with these options, some layouts were only possible through the development of custom meta boxes and templates.

Meet the solution: **The Gutenberg Block Editor.**

Gutenberg is a new spin on the classic WordPress editing experience that makes publishing easier for people who aren't coders or designers.

"Gutenberg" is the name of the project to create a new editor experience for WordPress. The goal is to create a new post and page editing experience that makes it easy for anyone to create rich post layouts."

<https://wordpress.org/gutenberg/handbook/contributors/faq/>

Gutenberg is a “block-based” editor designed to replace the Classic Editor and the goal is to make it easier to create robust layouts, regardless the user’s level of experience with WordPress.

Blocks

In Gutenberg, blocks are the term used to describe pieces of content on a page or post.

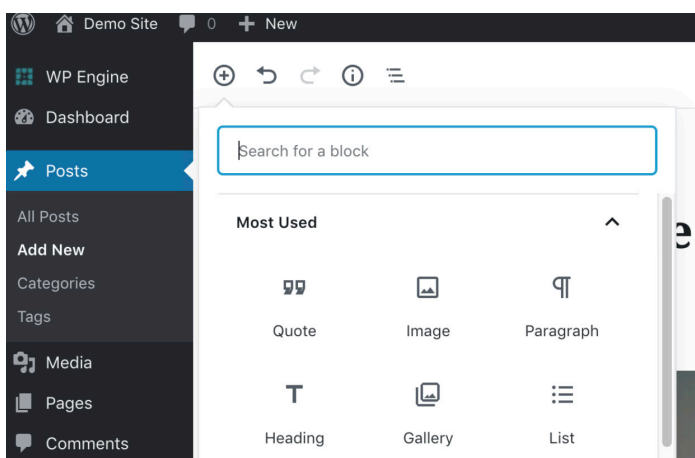
“Blocks are an abstract unit for organizing and composing content, strung together to create content for a webpage.”

[WordPress Developer Handbook](#)

You can think of blocks as individual types of content you might want to add to a page or post, things such as:

- Heading
- Image or Gallery
- Cover image with text overlay
- Quote
- Paragraph
- Button
- Video

[WordPress core includes over 30+ blocks](#) you can use to write rich content. Developers can also add custom blocks to themes and plugins via the [Block API](#). Both WordPress core blocks and custom blocks are accessible via the same interface, creating a unified editing experience for users.



What happens to the Classic Editor

During its development, Gutenberg was available to users as a plugin, while the Classic Editor remained the default editor for WordPress.

With the release of WordPress 5.0, this reversed. Users who've updated to 5.0 automatically have the Gutenberg editor. Users who are not ready to adopt Gutenberg can install the Classic Editor plugin. The [Classic Editor plugin](#) additionally gives users the option to switch to the block editor on a per-user or per-post basis.

Although Gutenberg is part of WordPress core, it's still a work in progress. There are various reasons someone might choose to use the Classic Editor plugin with WordPress 5.0+, such as:

- Site has accessibility requirements not yet met by Gutenberg (Read here for more about [Gutenberg and accessibility](#))
- Site has a theme, plugins, or other dependencies that conflict with Gutenberg (Use a staging or development environment to test your site with Gutenberg. If you're hosting with WP Engine, here's how you can [set up a staging environment](#))
- You're a developer or agency supporting multiple websites and need additional time to test Gutenberg with your sites and/or train your clients on usage.

The Classic Editor plugin will be officially [supported through December 31, 2021](#).

Gutenberg in context.

While Gutenberg brings a major change to the editing experience, ultimately it's just one aspect of developing sites with WordPress.

Gutenberg is intended to work in conjunction with plugins and themes.*

** It's up to individual plugin and theme authors to ensure their products are Gutenberg ready.*

We'll take a look at how Gutenberg works with popular products such as page builders and the Genesis Framework.

Gutenberg and page builders

Page builders are a popular tool for creating highly customized site designs via a visual interface. Many include a WYSIWYG (What You See is What You Get) editor on the front-end of the website. By contrast, creating pages with the Gutenberg editor is done on the back-end.

Page builders enable non-technical users to build complex layouts for their websites with little to no code. The downside is that they're typically not performant and can become problematic if a site owner decides to migrate to a different theme.

An average web page layout might include:

- Header
- Sidebar
- Content Area
- Footer

Page builders can control the design of an entire layout. Gutenberg, by contrast, only applies to the content area. While the Gutenberg block concept will likely incorporate other [page layout elements in the future](#), **Gutenberg should not be characterized as a page builder.**

Beaver Builder, one of the most popular page builders for WordPress, published some [interesting thoughts](#) about how they anticipate Gutenberg will impact their business. Robby McCullough, co-founder of Beaver Builder, said this:

In an interview... [Matt Mullenweg] describes Gutenberg as the 20%, baseline experience that creates an opportunity for page builders and a 3rd party ecosystem to extend. He even mentions he is bullish on Page Builders.

While all page builders may not share Beaver Builder's optimism, it's fair to say that Gutenberg is meant to work with page builders, not replace them.

Gutenberg and the Genesis Framework

Genesis is a theme framework that serves as the foundation for 600,000+ WordPress sites and combines all of the security, design, layout, SEO, and performance features you need to power a professional website.

The Genesis Framework is a tool for do-it-yourself (DIY) site builders, site assemblers, and developers alike. [StudioPress themes](#) that run on Genesis allow DIYers and site assemblers to create attractive, SEO-friendly websites with little to no code. Genesis also provides a "developer's playground." It's a robust platform developers can use to build fully customized web experiences for clients.

Later in the article, we'll cover additional information about using Genesis with Gutenberg, but the bottom line is this:

The Genesis Framework is fully compatible with Gutenberg.

Using Gutenberg with Genesis.

From a user perspective there's nothing "extra" required to use Gutenberg with the Genesis Framework. Gutenberg is simply a new editing experience.

That said, Genesis does include some Gutenberg-specific features that developers can leverage in child theme development. We'll discuss these features in detail in the [Developing with Gutenberg and Genesis](#) section.

How to know if you have Gutenberg

Prior to Gutenberg's inclusion in WordPress core, it was available as a plugin. If you are using WordPress 5.0 or later, you automatically have Gutenberg. If you are running an older version of WordPress and would like to use Gutenberg, you would need to [install and activate the plugin](#).

That said, we recommend staying up-to-date with the latest version of WordPress in order to keep your site secure and take advantage of new features. If your site is hosted with WP Engine, we offer [development and staging environments](#) you can use to test updates with your site prior to updating your live site.

You can access the Gutenberg editor by adding or editing any post or page from your WordPress admin area.

Editing content with Gutenberg

The new Gutenberg editing experience is a vast departure from the Classic Editor veteran WordPress users are familiar with.

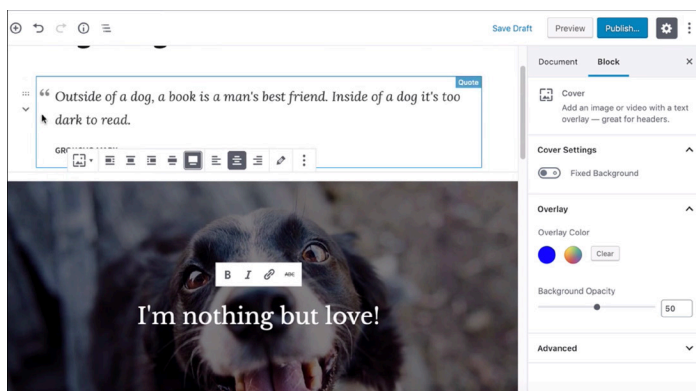
From the block library (indicated by the + sign), users can select the type of content to add to a post or page.

As mentioned earlier, [WordPress core includes over 30+ blocks](#) related to text, media, quotes, code, and layout. Any custom blocks added by plugins will also be available in this same block library.

A couple of notable features about working with blocks:

- Block placement can be changed after insertion by drag/drop or up/down arrows
- [Reusable blocks](#) enable users to quickly duplicate identical content across multiple pages
- The [multi-select feature](#) can be used for creating reusable templates (a group of reusable blocks)

The easiest way to get familiar with editing content with Gutenberg is to [try it out for yourself with this demo](#).



Developing with Gutenberg and Genesis.

With the release of Genesis 2.8, support is available for [Gutenberg opt-in features](#) (i.e. wide alignment, block color palettes) and editor styles. This enables child themes to leverage the Gutenberg API via a config file while Genesis does the heavy lifting.

In addition to simplifying the way developers interact with the Gutenberg API, Genesis 2.8 gives theme developers the ability to add two important features for users:

- One-click demo install
- Block style guide

While Genesis makes these features available to developers, it's up to developers to decide which features they will implement. In other words, feature implementation will vary by child theme.

In the sections below, we'll explore these features in more detail.

The block editor: A work in progress

Even though Gutenberg is in WordPress core, it's still a work in progress. As such, it's important to keep track of changes. Here are a few resources we recommend:

GUTENBERG TIPS AND QUIRKS FOR GENESIS CHILD THEME DEVELOPERS

This [wiki page](#) outlines issues we've encountered so far making Genesis child themes Gutenberg-ready.

GENESIS SAMPLE THEME

The Genesis Sample theme is a free, bare-bones theme for the Genesis Framework that shows how to leverage Genesis for implementing features such as a [block-based homepage](#), [onboarding config](#), [Gutenberg front-end and editor styles](#), and more. This is a great theme to deconstruct and understand for working with Genesis and Gutenberg.

<https://github.com/studiopress/genesis-sample/>

TWENTY NINETEEN THEME

The Twenty Nineteen theme by WordPress has in-development support for Gutenberg. While not Genesis-specific, it's worth taking a look at how this theme leverages Gutenberg features.

<https://github.com/WordPress/twenty十九teen>

In addition to these resources, you can chat about all things Gutenberg via the #gutenberg channel in the [GenesisWP Slack](#) group.

If you're new to Genesis, continue reading for a better understanding of how Genesis works.

Genesis development: a framework of hooks and filters

Genesis provides a layer of rich features on top of WordPress and serves as the foundation for child themes.

A child theme is a layer of code that sits on top of the Genesis Framework and is comprised mainly of the design elements for a site, but can also extend and modify the default functionality of the Genesis Framework.

In the same way you can extend WordPress by using action hooks and filters inside plugins, you can extend Genesis by using its extensive library of [hooks](#) and [filters](#) inside a child theme. Rather than build a theme from scratch, you can build a child theme with unique specifications while leaning on the framework for core site functionality.

If you've written WordPress themes from scratch, created child themes as a way to customize a parent theme, or used starter themes like `_s` as a base, then you're familiar with the WordPress theme development principles outlined in the [WordPress Theme Handbook](#).

What makes Genesis different from typical theme development is how you customize the features and markup on the page. Rather than directly editing the core theme files—like you would typically do in a custom theme—you use hooks and filters to remove or modify core Genesis features, and add your new features. This enables developers to create custom themes with minimal code, which can significantly save you time and money over the lifecycle of the theme.

Technical overview of Gutenberg

WordPress documentation includes a [Designer & Developer Handbook for Gutenberg](#) that's well worth a look to understand the specifics of developing with Gutenberg.

That said, we'll summarize a few key concepts here.

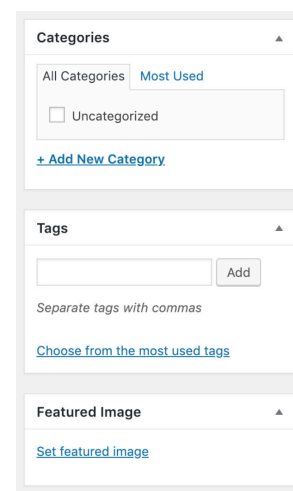
BLOCK API

As mentioned earlier, the Gutenberg editing experience is based on blocks. In addition to the blocks included in WordPress core, developers can use the [Block API](#) to extend Gutenberg.

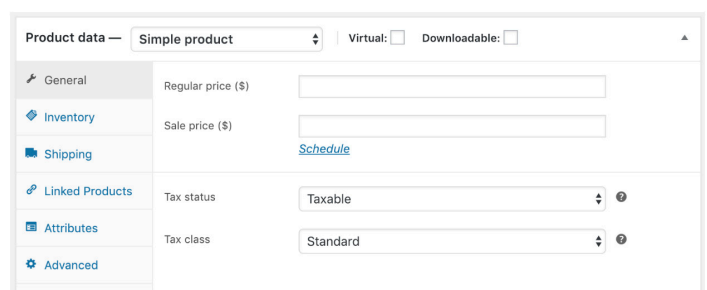
It's worth noting that Gutenberg is built with [React.js](#). That said, [you don't have to learn React](#) in order to build custom blocks with Gutenberg, you only need to understand how to work with the Block API. Of course, you'll have an easier time settling into custom block development with a fundamental understanding of JavaScript.

META BOXES AND GUTENBERG

Meta boxes provide an interface for interacting with meta data. By default, the Classic Editor screen included meta boxes for things like categories, tags, and a featured image.



Developers could use the [Meta Box API](#) to add custom meta boxes, enabling users to enter related information to a post. For example, here are the custom meta boxes used by WooCommerce (pre-Gutenberg).



The power of meta boxes is two-fold: they give users a nice interface for entering specific post details and they give developers a way to create specific templates for laying out the content.

If the power of Gutenberg is to use blocks to create and lay out content, then what happens to meta boxes? This has been a much debated topic throughout the development of Gutenberg.

Here are the most important details to understand about meta boxes and Gutenberg for now:

- Meta boxes are written in PHP and Gutenberg blocks are written in JavaScript (React)
- The Gutenberg editor has the ability to load meta boxes (although this should be tested on a per-site basis)
- Ultimately, **converting meta boxes to blocks is highly encouraged**

The bottom line? While the Meta Box API isn't being deprecated and there are tools in place for maintaining existing meta boxes, use Gutenberg blocks for future development.

Creating Genesis child themes with Gutenberg support

Earlier we mentioned the [Genesis Sample theme](#). We'll reference this theme as we take a look at how to:

- Create an onboarding config for new theme installs
- Create a block-based homepage

- Add Gutenberg styles

Read on to learn more.

Child theme onboarding

When users install a new theme, they expect it to look like the theme demo. As developers, we know that new theme installs require some level of configuration before a site starts to resemble a demo.

As of Genesis 2.8, there are new features developers can use to provide a better onboarding experience for users.

WHAT AN ONBOARDING CONFIGURATION FILE DOES

The configuration file loads:

- Required plugins on theme activation
- Homepage content on theme activation
- Desired theme-specific configurations, such as site layout, breadcrumbs, or footer widgets

The end result of including a configuration file with a child theme is that a new theme install can closely resemble a demo. This both reduces the time it takes a user to setup and become familiar with a new theme and decreases support requests for theme setup assistance.

With Genesis 2.8, the configuration file can be used to setup the homepage. In future versions of Genesis, this feature will extend to the entire site.



HOW TO IMPLEMENT AN ONBOARDING CONFIGURATION FILE

In the past, the `functions.php` in a Genesis child theme was the primary place to include theme configuration options, such as theme support for menus or accessibility. In the [Genesis Sample theme](#), Those configuration options are abstracted into individual configuration files located in a `/config` folder.

An [onboarding configuration file](#) can also be included in this location and can specify any plugin dependencies as well as a template for setting up the home page.

Please visit these developer wikis for additional instructions on working with onboarding configurations..

- [Onboarding](#)
- [Configuration](#)

Block-based homepage

With both WordPress and Genesis, complex home page layouts were traditionally accomplished via widgets or custom fields. Both of these options enabled developers to pre-style a home page based on known widget or custom field IDs.

These methods weren't necessarily intuitive for theme users, but aside from using a page builder, those were the primary methods available to theme developers.

With the introduction of Gutenberg and features introduced with Genesis 2.8+, developers can now leverage a much more intuitive block-based configuration for laying out a home page.

We'll provide a link to an example of this in the "Examples and tutorials" section below.

Gutenberg styles: front-end and back-end

Theme developers are likely familiar with the concept of adding front-end styles to a theme. These are the styles that dictate how a site looks.

Developers typically include styles for many of common elements (blockquotes, buttons, etc.), but be aware that Gutenberg loads its own styles. You may want to include additional styles to ensure consistency with your theme design.

In this tutorial, Bill Erickson explains how you can [override front-end Gutenberg styles with your own](#).

So that's the front-end, but what about incorporating styles on the back-end so that the Gutenberg editor resembles what users see on the front-end?

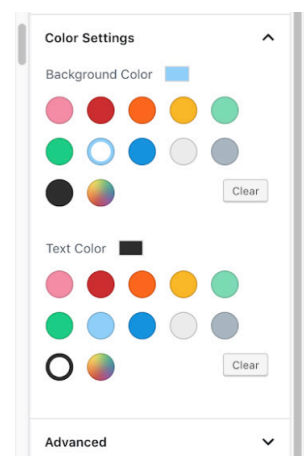
With the Classic Editor, developers could use the `add_editor_style` function to load custom styles into the editor. This same function can be used to add styles for the Gutenberg editor. There's also an alternative option.

Developers can also enqueue scripts and styles using `enqueue_block_editor_assets`. That loads styles and scripts for the back-end in the same way that `wp_enqueue_scripts` loads styles and scripts for the front-end.

Examples and tutorials of Genesis with Gutenberg.

Below are some real-world examples of implementing Gutenberg features in Genesis child themes.

Add theme color palette to the Gutenberg Editor for WordPress



The Gutenberg editor ships with 11 default colors, including pink, red, orange, yellow, green, blue, and gray. In addition, it will always include a color picker with the palette, for further customization.

Gutenberg gives the user full control over what color is used for various elements, such as text, background, and buttons. But there is also a way to [create custom color palettes](#), too.

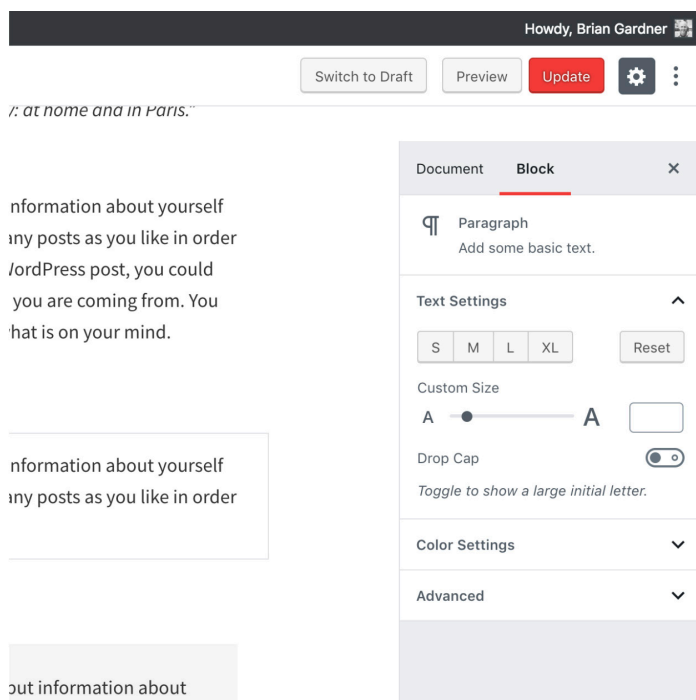
Visit this tutorial to learn how to add theme support for a custom color palette and/or disable the default color palette: <https://studiopress.blog/theme-color-palette/>

Note: A color palette can alternatively be specified as part of a theme config file.

Add block font sizes to the Gutenberg Block Editor For Wordpress

According to the Gutenberg Handbook, blocks may allow the user to configure the font sizes they use, e.g., the paragraph block.

Gutenberg provides a default set of font sizes, but a theme can overwrite it and provide its own. Here is a screenshot that shows the Gutenberg editor with default settings:



Visit this tutorial for code snippets and instructions on adding theme support for custom block font sizes: <https://studiopress.blog/block-font-sizes>

Note: block font sizes can alternatively be specified as part of a theme config file.

Creating a custom Gutenberg Block

Bill Erickson uses the [example code](#) from Zac Gordon's [Gutenberg Development Course](#) as the starting point for the "List Subpages" block he builds in [this tutorial](#).

Peter Tasker demonstrates [how to build a basic "callout" block](#) using Ahmad Awais' [create-guten-block](#) tool as a boilerplate.

Josh Pollock shows how to [create Gutenberg blocks with WP CLI](#).

How to deprecate code in Gutenberg editor blocks

Changing the markup of a custom block after it's in use will cause the block to break in the editor.

Mike McAlister of [Atomic Blocks](#) shows how you can safely [deprecate code in a custom block](#).

Add theme support (and styles) for wide images

Gutenberg provides the ability to include images that extend outside of the content area or even span to the full width of a page.

To take advantage of this feature, theme developers need to declare theme support for wide images as well as provide the supporting CSS.

To learn how, check out Rich Tabor's [The Ultimate Guide to Gutenberg Image Alignments in WordPress Themes](#).

The Twenty Nineteen Theme

Each year WordPress introduces a new default theme that highlights the latest and greatest features of WordPress. The new Twenty Nineteen theme is designed to show off Gutenberg's capabilities and serve as an example of best practices when developing themes for Gutenberg.

- [Twenty Nineteen specs and overview](#)
- [Twenty Nineteen theme on WordPress.org](#)
- [Twenty Nineteen theme on Github](#)

Gutenberg block support has also been added to [previous years' default themes](#).

The future of Genesis with Gutenberg.

We're committed to following the development and maturity of Gutenberg in order to make sure Genesis evolves with it to deliver every advantage of this new editing experience to both developers and users.

Genesis 2.8 is a big step forward as Genesis fully embraces the new editing experience by providing a way for theme developers to more easily interact with Gutenberg features.

Roadmap for Genesis 2.8 and beyond

At WordCamp US 2018, the [Genesis Shapers team](#) met to discuss the future of Genesis. Below is an excerpt of that discussion:

- Nathan Rice—the lead developer and co-creator of Genesis—took some time to review the existing Genesis roadmap with the Shapers group so everyone would be aware of what has already been planned. As published in many places including the StudioPress blog, GitHub, and Slack, Nathan shared the following:
- **2.8** – With Genesis 2.8 we will be extending the power of Gutenberg to help you spin up sites faster allowing you to autoload demo content and we've included other optimizations for Gutenberg. This will include phase 1 of the demo content feature with further improvements to come.
- **2.9** – Will include much needed refactoring including a bump for what minimum versions we support in an effort to make Genesis faster and to simplify the code base in order to help the Genesis community take advantage of the advances in the underlying technologies of Genesis. This is in alignment with similar efforts and timing within WordPress core.
- **2.10** – This release will continue on the refactoring efforts of 2.9 including moving settings and layout APIs to classes among other improvements.
- **3.0** – In addition to further refactoring, we plan on adding in a few more capabilities to help with mobile including a special AMP integration the Genesis R&D team is working on in partnership with Google.

As always, we'll be releasing beta versions for each release for testing and direct feedback from everyone in the community.

What to expect from child themes

While the Genesis Framework provides the foundation for working with Gutenberg, it's up to theme developers to decide whether or not to leverage specific features in a particular child theme.

For the sake of clarity, we'll define how our child themes work with Gutenberg.

- **Gutenberg-ready** - All StudioPress child themes are *Gutenberg-ready*, meaning there are no breaking changes and users will have access to the new editing experience regardless of which StudioPress theme is used.
- **Gutenberg-optimized** - StudioPress is in the process of retrofitting it's existing theme collection to take advantage of the new Gutenberg features made possible with Genesis 2.8.
- **Gutenberg-first** - If a *Gutenberg-optimized* theme leverages the configuration and onboarding features introduced in Genesis 2.8 in an existing theme, then a *Gutenberg-first theme* describes a new theme created from the ground up to take advantage of all Gutenberg offers.

At the time of this writing, StudioPress has re-released [six Gutenberg-optimized themes](#). StudioPress is also planning the release of a new, Gutenberg-first theme. This theme, code-named "R3volution," will include a home page layout controlled by the Gutenberg editor.

In addition to StudioPress themes, you can likely expect third-party theme developers to offer both Gutenberg-optimized and Gutenberg-first themes in the future. If you are currently using a third-party theme with Genesis, we recommend you check with the developer (or their website) to confirm whether the theme is Gutenberg compatible.

We look forward to the future of WordPress and Genesis with Gutenberg, and we can't wait to see what you use it for too.

Additional resources.

Below are some additional resources for understanding what Gutenberg is, how you can develop for it, and what the WordPress ecosystem is creating in response to Gutenberg's release.

Gutenberg overview & news

- [Gutenberg Demo](#) - Test out Gutenberg with this interactive demo
- [Gutenberg News](#) - A collection of Gutenberg tutorials
- [WordPress 5.0 Field Guide](#) - An overview of what's new in 5.0

Plugin and theme development with Gutenberg

- [Gutenberg Developer & Designer Handbook](#) - Official WordPress documentation
- [Gutenberg Theme Support](#) - New concepts to consider when theming for Gutenberg
- [Gutenberghub.com](#) - Collection of Gutenberg WordPress Editor blocks, resources & tutorials.

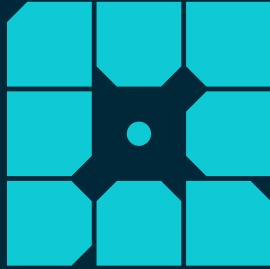
Block libraries & frameworks

- [Atomic Blocks](#) - Collection of page building blocks
- [Editorblockswp.com](#) - A collection of blocks for building a business website
- [Gutenbergcloud.org](#) - Package manager (NPM) for a library of CMS-agnostic blocks
- [Getblocklab.com](#) - A WordPress Admin interface and a simple templating system for building custom Gutenberg blocks.
- [Lazy Blocks](#) - Gutenberg blocks visual constructor for WordPress developers

Community Gutenberg development

- [360 Image](#) - Add interactive 360° images to your WordPress site
- [BigCommerce](#) - Gutenberg-ready e-commerce system
- [Caxton Shape Divider Block](#) - A page layout plugin for Gutenberg
- [Drop It](#) - Allows one-click inserts of Unsplash images with the Gutenberg editor
- [Ecwid Ecommerce Shopping Cart](#) - Gutenberg-ready shopping cart
- [Ghost Writing](#) - A block that applies the "typing effect" to headings
- [Google AMP](#) - The official AMP plugin for WordPress with Gutenberg integration
- [Jetpack Form Block](#) - Add a form to a post or page via the Gutenberg editor
- [Recipe Card Blocks](#) - Adds blocks to Gutenberg for creating recipe cards in posts
- [Spoiler Alert](#) - A block that reveals "spoiler text" after the user clicks a button
- [Tarot](#) - Generate and display Tarot threads in posts
- [WooCommerce Products Block](#) - Flexible way to display products on posts and pages
- [Yoast SEO](#) - Introducing the Yoast SEO Gutenberg sidebar & a revamped meta box

As Gutenberg adoption increases and developers continue innovating their themes and plugins to take advantage of Gutenberg features, you can expect additional tutorials and information to be available online.



About WP Engine.

WP Engine is the world's leading WordPress digital experience platform that gives companies of all sizes the agility, performance, intelligence, and integrations they need to drive their business forward faster. WP Engine's combination of tech innovation and an award-winning team of WordPress experts are trusted by over 70,000 companies across 130 countries to provide counsel and support, helping brands create world-class digital experiences. Founded in 2010, WP Engine is headquartered in Austin, Texas, and has offices in San Francisco, California; San Antonio, Texas; London, England; Limerick, Ireland; and Brisbane, Australia.

www.wpengine.com

