

What's new for developers? (August 2023)

[WordPress 6.3](#) landed on August 8, 2023. If you're like me, you're like a kid in a toy shop, sprinting down the aisles and pulling out every new thing you can reach on the shelves.

Of course, the first thing you'll want to do is check the [WordPress 6.3 Field Guide](#) for new features and changes. Then, test your themes and plugins against the new release. And, finally, find your favorite new shiny objects!

But a new WordPress release doesn't mean the wheel stops spinning. Remember also that WordPress 6.4 has already been in alpha for a month and will land in November 2023. If you've got something on your wishlist, you still have a few days to [make it known on the Make Core blog](#).

This edition of the *What's new for developers?* monthly roundup is a mixed bag of goodies. Some features have landed in the 6.3 release; some are still only in the Gutenberg plugin.

To be fair, the plugin is already two versions ahead of what merged into 6.3. And when you're testing changes, it's always a good idea to run the latest version of everything.

Highlights

Phase 3 updates

Three recent posts on the Make Core blog expand on the [Phase 3: Collaboration roadmap](#):

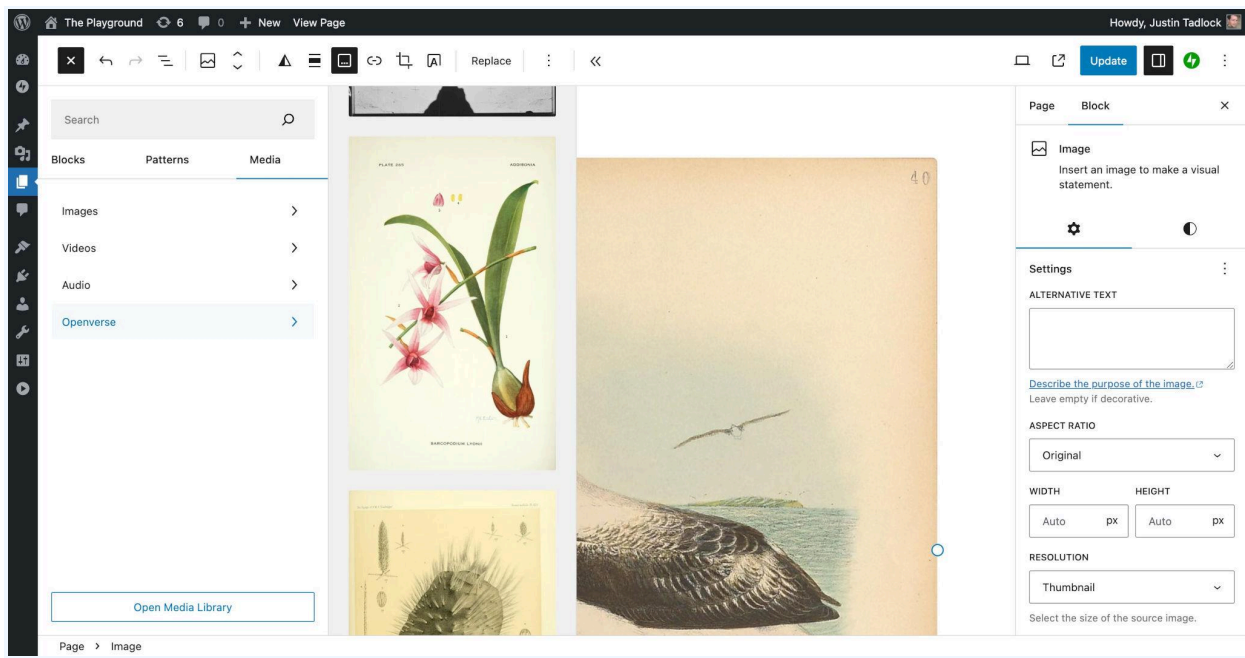
- [Block Library](#) envisions a single system for everything a user needs: blocks, patterns, post types, media, and more—and organizes it all, for a fluid experience in the editor and, especially, outside of it.
- [Admin Design](#) outlines several goals for a modern, flexible WordPress redesign that makes it a pleasure to build beautiful user experiences.
- [Real-time Collaboration: architecture](#) expands the previous discussion on collaborative features (below) with a broad-strokes outline of how the architecture would work in the post and site editors.

These posts come on the heels of the original four, which came out earlier in the summer. If you missed those, be sure to give them a read:

- [Real-Time Collaboration](#) covers plans for concurrent collaboration, shared edits, visibility of peers, and lifting current editing restrictions.
- [Workflows](#) imagines seamless collaboration over the whole editorial process, from first draft through publication and beyond.

- [Revisions](#) proposes a complete overhaul of the current system in favor of one that works with blocks and tracks changes on every data layer, from the server stack to the user's latest edits.
- [Media Library](#) aims to expand media management capabilities, unify the interfaces and improve overall media workflows.

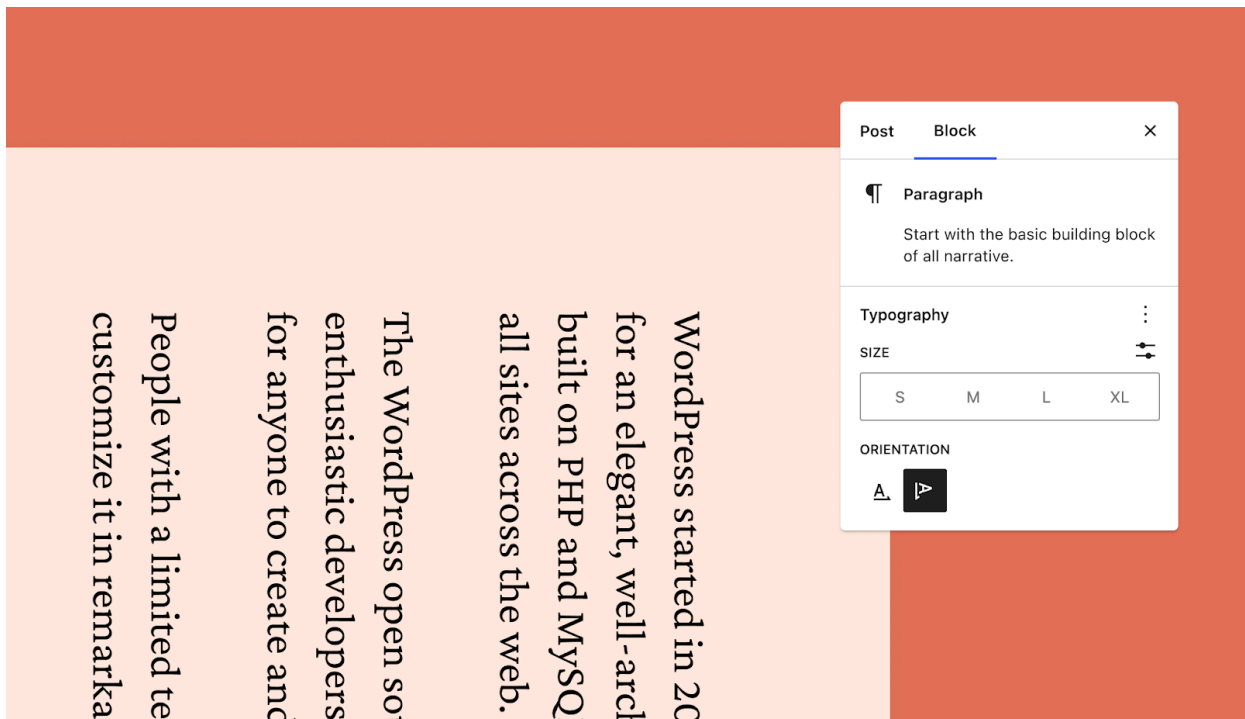
Extendable media inserter



Gutenberg 16.2 [introduced the `registerInserterMediaCategory` API](#) for registering media inserter categories. That lets you add a new sub-tab under **Media** in the inserter, which you can use for lots of things: to add more media types, perhaps, or integrate with third-party services—I know you'll think of more.

Check out the [API's documentation](#) in the Block Editor Handbook for more details.

Vertical text orientation support



New for your themes: support [vertical text orientation](#) in `theme.json`. Just set `settings.typography.writingMode` to `true`, and the Paragraph blocks in the UI will sprout a new **Text Orientation** design tool. You can set that to horizontal or vertical.

Want to make vertical orientation the default Paragraph block style? Set `styles[core/paragraph].typography.writingMode` to `vertical-rl`.

You probably won't set the default for every Paragraph block in most cases. But the feature should make it easier to build patterns that mix in some vertical text for effect.

Plugins and tools

Loading assets and the post editor iframe

WordPress 6.3 now places the post editor content in an iframe—if all registered blocks are on API version 3. If your themes and plugins are using the recommended methods to load scripts and styles, you should be fine with this.

Of course there's an exception (there always is, somehow ...) and that's if you are using the ``enqueue_block_editor_assets`` hook to enqueue a script/style that specifically targets the content canvas. You can find out what hooks and functions you need to use, and why, in the [Dev Note](#) and this [ticket on GitHub](#).

Development mode conditional function renamed

The ``wp_in_development_mode()`` function, added during the WordPress 6.3 development cycle, has a new name: ``wp_is_development_mode\(\)`` to make it consistent with similar functions. If you were testing this before the 6.3 release, make sure you update all the instances accordingly.

New ProgressBar component

In Gutenberg 16.4, a new [`ProgressBar` component](#) lets you show progress to a user in a variety of scenarios. Eventually, for instance, the bar will replace the `Spinner` component when the [Site Editor is loading](#), but that implementation is still to come.

Component updates

Now the `RangeControl` component supports the [large `40px` number input size](#). That size shows up [across multiple core instances](#).

You can also [pass an `id` prop](#) to the `TextControl` component. That's a good way to reference it somewhere else with an `aria-*` attribute.

Stabilized APIs

Three [experimental APIs were stabilized](#) in Gutenberg 16.4. That means they should be safe from backward-compatibility issues when you use them in production:

- `defaultBlock`
- `directInsert`
- `getDirectInsertBlock`

A minimal HTML Processor

As part of the ongoing work for a more complete HTML API, a minimal version of the [HTML Processor landed in trunk](#). This feature is targeted for a WordPress 6.4 release, but there is [some pushback](#) in the ticket on whether it should be included at all.

The HTML Processor currently only supports a subset of all HTML elements. This will give contributors time to test, review, and garner feedback on the initial architecture.

Interactivity API updates and Create Block script support

Gutenberg 16.3 [adds runtime support](#) for the ``wp-style`` core directive. Plus, a new ``gutenberg_should_block_use_interactivity_api`` filter hook lets you disable Interactivity API support on a per-block basis.

Contributors also [added a template](#) for the [@wordpress/create-block](#) package to help you start experimenting with custom interactive blocks.

Rest API updates

The REST API has a new [endpoint for global styles revisions](#), a feature that shipped with WordPress 6.3. Get to it with ``/wp/v2/global-styles/revisions``.

The `WP_REST_Templates_Controller` has a new [`modified` field](#) for template and template part objects. The field returns the post modified datetime.

Themes

Fixed width for wide/full Image blocks

The `` element nested inside the Image block now has a [fixed width of `100%`](#) when it is set to wide or full-width alignment. Test this with your themes to make sure this change does not break your design.

Avatar border selector

The global styles selector for the Avatar block now [targets the wrapper](#) instead of the inner `` element. That's how borders get applied at the block level, but it is a potential breaking design change in some cases, depending on how you were styling the block.

Pattern updates

WordPress 6.3 has shipped an overhauled Site Editor navigation interface that mostly helps your users manage patterns and template parts. Still, you should see what the changes are and how they work, so you can answer support questions. And especially if you build themes from the Site Editor.

The biggest change: the **Library** nav item [is now called Patterns](#). This panel houses synced patterns ([formerly reusable blocks](#)), unsynced patterns (can be registered by the user, WordPress, themes, and plugins), and template parts ([grouped by area](#)).

Some related tickets worth checking out are:

- [The Custom patterns label was renamed to My Patterns.](#)
- [Sync status added to the pattern details screen.](#)
- [Updated template part icons in the library mosaic.](#)
- [Reinstated Manage all template parts page.](#)
- [Lock icon added for theme patterns.](#)
- [Display client-side pagination for the patterns list.](#)
- [Show further details for the template part panel.](#)

Helper function for getting patterns from the Pattern Directory

If you've been grabbing remote patterns via the `'WP_Theme_Json_Resolver'` class methods, you will want to switch to the ['`wp_get_remote_theme_patterns\(\)`' function](#) now that WordPress 6.3 is out. The class is private and marked for internal WordPress use only. By contrast, the helper function is there for you to use. It will make sure you don't run into breaking changes in the future.

Better Query Title block support for author archives

The Query Title block has always shown a generic title in the Site Editor—but not anymore. Now, in Gutenberg 16.3, it will [contextually show the archive type](#) in the placeholder text. For example, it will output “Category: Name” for a generic category template but “Category: Recipes” for the `category-recipes.html` template.

Block library updates

Several blocks have newly updated design-tool and feature support:

- Footnotes: added [support for the background, link, and color](#) design tools.
- Preformatted: added [support for spacing](#).
- Social Links: includes a new [Threads social icon](#).
- Verse: hitting the `Enter` key now [creates a line break](#).

Technically, captions are an element, not a block. Either way, now you can control their [color and typography](#) in the **Styles** interface.

Bug fixes

Several fixes have shipped with Gutenberg and WordPress that should make your life easier.

You can style nested elements for blocks in `theme.json`. You could before, but a bug [kept the styles from being generated](#) on the front end.

When you're on the site's home page, the Home link block [shows the `.current-menu-item` class](#). That's how the other links work.

[New and improved logic](#) for caching the results in ``wp_theme_has_theme_json()`` takes into account that a site can switch themes.

Events and resources

Developer Hours

[July's Developers Hours](#) showed attendees how to style blocks with CSS custom properties and then give theme authors access to the results.

Learn WordPress

Learn WordPress hosted a workshop in July on WP-CLI for PHP developers. If you missed it, or you were there but would like to see it again, [it's on WordPress.tv](#).

Learn also two new developer-oriented tutorials:

- [Improving website performance with caching](#)
- [Installing Node.js and npm for local WordPress development](#)

Developer Blog

Several articles landed on the Developer Blog in the past month:

- [Make your site's typography make a statement](#)
 - [Understand and use WordPress nonces properly](#)
 - [Styling blocks: empowering users with CSS custom properties](#)
 - [Beyond Block Styles series:](#)
 - [Part 1: using the WordPress scripts package with themes](#)
 - [Part 2: building a custom style for the Separator block](#)
 - [Part 3: building custom design tools](#)
-

!!! Notes !!!

Reviewers, this document section is not a part of the final post. Please ignore.

Resources:

- [Keeping up with Gutenberg: Index 2022](#)
- [A Week in Core](#)
- [Gutenberg plugin release posts](#)
- [Core editor chat](#), [dev chat](#), [performance team](#) meeting summaries
- [Dev Notes](#)

Key:



- ★ Highlights
- ! Deprecations and warnings
- 👍 Intro mention
- 🐛 Bugs

Plugins and tools

- ★ Add new registerInserterMediaCategory API to make media categories extensible. ([51542](#))
- [A template](#) for [@wordpress/create-block](#) to create interactive blocks
- RangeControl: Add support for large 40px number input size. ([49105](#))
- Use next40pxDefaultSize on RangeControl components. ([52257](#))
- ~~Experimental: Disable TinyMCE compatibility with classic editor~~ [#52325](#)
 - Removed since this is just a compat layer for plugins that disable the block editor.
- Adding support for defined IDs in TextControl component [#52028](#)
- Stabilize defaultBlock, directInsert API's and getDirectInsertBlock selector. ([52083](#))
- Rename wp_in_development_mode() to wp_is_development_mode() – [#57487](#)
- Introduce minimal HTML Processor – [#58517](#)
 - This is targeted at WP 6.4 and has been reopened.

- Pattern: Add `getBlockRootClientId` call [53206](#)
- ~~Updated “position” default value. ([52148](#))~~
 - Minor documentation change.
- Interactivity API
 - Add filter to turn off Interactivity API for a block. ([52579](#))
 - Add runtime support for the `wp-style` directive. ([52645](#))
- **!** Note issue with iframed post content editor (devs using `enqueue_block_editor_assets` incorrectly).
 - I dropped a note in the [July edition](#), but we should highlight this in August.
- Introduce a basic `ProgressBar` component. ([53030](#))
- ~~The Themes REST API can now indicate whether a theme supports the Site Editor by adding an `is_block_theme` property to each theme in the `wp/v2/themes` API response. ([#58123](#))~~
- A revisions endpoint for global styles, `/wp/v2/global-styles/revisions`, has been added to the REST API. ([#58524](#))
- `WP_REST_Templates_Controller` has a new `modified` field for `template` and `template part` objects that returns post modified datetime for `Templates`. ([#58540](#))
-

Themes

- Improved caching in `wp_theme_has_theme_json()` – [#58758](#)
-  Add Typography: Text orientation (writing mode). ([50822](#))
-  Add: Patterns to the template start modal. ([47322](#))

- Everything patterns:
 - *Note: consolidate all of this into just what themers need to know.*
 - Rename Reusable blocks to Patterns. ([51704](#))
 - Add sync status to pattern details screen. ([51954](#))
 - Rename Library to Patterns. ([52102](#))
 - Update custom patterns label to 'My patterns'. ([51949](#))
 - Update pattern creation modal in library. ([51946](#))
 - Update template part icons in the library mosaic (grid items). ([51963](#))
 - Reinstate manage all template parts page. ([51961](#))
 - Add lock icon for theme patterns. ([51990](#))
 - Update section heading levels. ([52273](#))
 - Patterns: Add client side pagination to patterns list [#52538](#)
 - Try adding further details to template part panel. ([52476](#))
- Interactivity API:
 - [Add runtime support for the wp-style directive](#) #52645
 - [Add filter to turn off Interactivity API for a block](#) #52579
- Update: Show more intuitive archive titles on Query Title block. [#52521](#)
- Font Face: to generate and print font-face styles for theme.json fonts [#51770](#)
- A new public function `wp_get_remote_theme_patterns()` has been added to query the patterns datum from theme.json and substitutes current usage of private APIs. ([#58460](#))
- Show more intuitive archive titles on Query Title block. ([52521](#))

- Note: dig into this a bit to look at potential future changes.
- Display all custom template part areas in sidebar nav. ([52355](#))
- Add back old Navigation and File blocks JavaScript implementation when Gutenberg is not installed. ([52553](#))
- Home link block: Add 'current-menu-item'. ([51478](#))
- Update: Improve titles of author templates in query title block. ([52732](#))
- Block design tools updates:
 - Footnotes: Add link, background, and text color support. ([52897](#))
 - Preformatted: Add spacing support. ([45196](#))
 - Social Links: Add Threads Icon. ([52685](#))
 - Verse: Enable the line breaks. ([52928](#))
 - Caption element UI controls for color and typography. ([49141](#))
 - An element rather than a block
- 🐛 Avatar: Fix global border styles generation. ([53007](#))
- 🐛 Image block: Fix image size at wide and full width. ([53184](#))
 - Note: could be a breaking change
- Fix a PHP notice appearing when adding a new template part – [#57851](#)
- Fix bug where it was not possible to style custom block elements in theme.json – [#57868](#)
-

Resources

- 👍 [Block Library](#)
- 👍 [Admin Design](#)
- 👍 [Real-time Collaboration: architecture](#)
- 👍 Also: [WordPress 6.4: What's on your wishlist?](#)

Learn WordPress

- [Improving website performance with caching](#)
- [Installing Node.js and npm for local WordPress development](#)

WordPressTV

- [WP-CLI for PHP developers](#)

Developer Blog

- [Make your site's typography make a statement](#)
- [Understand and use WordPress nonces properly](#)
- [Beyond block styles series](#)
 - [Using the WordPress scripts package with themes](#)
 - [Building a custom style for the Separator block](#)

WordPress 6.3 Dev Notes

- [WordPress 6.3 Field Guide](#)

- [Layout updates in the editor for WordPress 6.3](#)
Learn more about the changes to layout support, CSS specificity, the new Grid type, to post template and spacer block.
- [WP_Query used internally in get_pages\(\)](#)
- [New in 6.3: Rollback for failed manual plugin and theme updates](#)
- [Improved Caching for Database Queries in WP_User_Query](#)
- [Improvements to the Cache API in WordPress 6.3](#)
- [Image performance enhancements in WordPress 6.3](#)
- [Improvements to the metadata API in WordPress 6.3](#)
- [I18N Improvements in 6.3](#)
- [Registering scripts with `async` and `defer` attributes in WordPress 6.3](#)
- [Social Icons block: Applied colors now dynamically update based on theme.json and Global Styles](#)
- [Introducing the WordPress Command Palette API](#)
WordPress 6.3 ships with a new command palette. Initially included in the post and site editors, users can open the command palette using the ctrl + k or command + k keyboard shortcuts.