Key Concepts

- Blocks
  - Composability
  - Data and attributes
  - Block transforms
  - Block variations
- Reusable blocks
- Patterns
- Templates
- Styles

Blocks

Blocks are an abstract unit for structuring and interacting with content. When composed together they create the content for a webpage. Everything from a paragraph, to a video, to the site title is represented as a block.

Blocks come in many different forms but also provide a consistent interface. They can be inserted, moved, reordered, copied, duplicated, transformed, deleted, dragged, and combined. Blocks can also be reused, allowing them to be shared across posts and post types and/or used multiple times in the same post. If it helps, you can think of blocks as a more graceful shortcode, with rich formatting tools for users to compose content.

The settings and content of a block can be customized in three main places: the block canvas, the block toolbar, and the block inspector.

Composability

Blocks are meant to be combined in different ways. Blocks are hierarchical in that a block can be nested within another block. Nested blocks and its container are also called *children* and *parent* respectively. For example, a *Columns* block can be the parent block to multiple child blocks in each

of its columns. The API that governs child block usage is named `InnerBlocks`.

Data and attributes

Blocks understand content as attributes and are serializable to HTML. To this point, there is a new Block Grammar. Distilled, the block grammar is an HTML comment, either a self-closing tag or with a beginning tag and ending tag. In the main tag, depending on the block type and user customizations, there can be a JSON object. This raw form of the block is referred to as serialized.

```
<!-- wp:paragraph {"key": "value"} -->
<p>Welcome to the world of blocks.</p>
<!-- /wp:paragraph -->
```

Blocks can be static or dynamic. Static blocks contain rendered content and an object of Attributes used to re-render based on changes. Dynamic blocks require server-side data and rendering while the post content is being generated (rendering).

Each block contains Attributes or configuration settings, which can be sourced from raw HTML in the content via meta or other customizable origins.

More on Data format and data flow.

Block transforms

Blocks have the ability to be transformed into other block types. This allows basic operations like converting a paragraph into a heading, but also more intricate ones like multiple images becoming a gallery. Block transforms work for single blocks and for multi-block selections. Internal block variations are also possible transformation targets.

Block variations

Given a block type, a block variation is a predefined set of its initial attributes. This API allows creating a single block from which multiple configurations are possible. Variations provide different possible interfaces, including showing up as entirely new blocks in the library, or as presets when inserting a new block. Read the API documentation for more details.

More on blocks

- **Block API**
- **Tutorial: Building A Custom Block**

Reusable blocks

A reusable blocks is **an instance** of a block (or multiple blocks) that can be inserted and edited in multiples places, remaining in sync everywhere. If a reusable block is edited in one place, those changes are reflected across all posts and pages that block is used. Examples of reusable blocks include a block consisting of a heading whose content and a custom color that would be appear on multiple pages of the site and sidebar widgets that would appear on every page.

Any edits to a reusable block will appear on every other use of that block, saving time from having to make the same edit on different posts.

Internally, reusable blocks are stored as a hidden post type (`wp_block`) and are dynamic blocks that “ref” or reference the `post_id` and return the `post_content` for that block.

Patterns

A block pattern is a group of blocks that have been combined together creating a design pattern. These design patterns provide a starting point for building more advanced pages and layouts quickly, instead of inserting individual blocks. A block pattern can be as small as a single block or as large as a full page of content. Unlike reusable blocks, once a pattern is inserted it doesn’t remain in sync with the original content as the blocks contained are meant to be edited and customized by the user. Underneath the surface, patterns are just regular blocks composed together. Themes can register patterns to offer users quick starting points with a design language familiar to that theme’s aesthetics.

Templates

While the post editor concentrates on the content of a post, the template editor allows declaring and editing an entire site using blocks, from header to footer. Templates are broken down between templates (that describe a full page) and template parts (that describe reusable areas within a template, including semantic areas like header, sidebar, and footer).

These templates and template parts can be composed together and registered by a theme. They are also entirely editable by users using the block editor; a collection of blocks that interact with different properties and settings of the site (like the site title, description, logo, navigation, etc) are especially useful when editing templates and template parts. Customized templates are saved in a `wp_template` post type. Block templates include both static pages and dynamic ones, like archives, singular, home, 404, etc.

Note: custom post types can also be initialized with a starting `post_content` template that should not be confused with the theme template system described above.

More on Site editing templates.

Styles

Styles, formerly known as Global Styles and as such referenced in the code, is both an interface that users access through the editor and a configuration system done through a `theme.json` file. This file absorbs most of the configuration aspects usually scattered through various `add_theme_support` calls to simplify communicating with the editor. It thus aims to improve declaring what settings should be enabled, what specific tools a theme offers (like a custom color palette), the available design tools present, and an infrastructure that allows to coordinate the styles coming from WordPress, the active theme, and the user.

Learn more about Global Styles.

□□□ #2
□ □□□ □□□ 2026-10-31 18:24:03 UTC
□ □□□ □□□ 2026-10-31 18:24:04 UTC