




Block Editor Best Practices

For users, designers, and developers



Danish Shakeel

Senior Software Engineer @  Camp

<https://iamdani.sh> | <https://linkedin.com/in/danishshakeel> | <https://github.com/danish17>

“Gutenberg may be the *most hated feature* of
WordPress”



Ryan

July 22, 2024

On the team I work with

- Block styles
- Block variations
- No to block styles
- No to block variations
- Block patterns
- Now section styles

What makes this challenge



realchriswells · 3mo ago

I can see how it is useful, but it's not...

One major drawback for me (it's possible I haven't figured it out yet) is that you can't set different layout options for different devices.

For instance if I have a two column section I might want to have them stack 2/1, rather than force 1/2. I got around this by adding a custom css class but it would be great if you could set this in the editor.

2 ↓ Reply Award Share ...

1 more reply



r/WordPress · 3 mo. ago
Bitter-Pressure-67

I can't with Wordpress anymore... what is even the point of the block editor?

The block editor is a net negative. Most plugins aren't compatible with it and this isn't something anybody asked for. Gutenberg, being out for YEARS now, is still nowhere near ready to use. Especially not as a default builder.

Now, it seems Wordpress has entirely removed the Customizer panel. I used it mostly for the initial setup of the website and the custom CSS, with the excellent tool included, but it seems they've removed all of that and I have to add CSS through the block editor now, which has NO features. No syntax highlighting, very simple error-checking... and I have no idea where my previous custom CSS went. It's still applying. I just can't find it anymore. How am I



elchemy · 3mo ago

Gutenberg is crap, especially if you are a power user. Disable it.

<https://wordpress.org/plugins/disable-gutenberg/>

You're welcome.

2 ↓ Reply Award Share ...

able to.

variables name. It just feel wrong

create some kind of the partial

more natural than the wp class

responsiveness in the editor.

and really doubting.

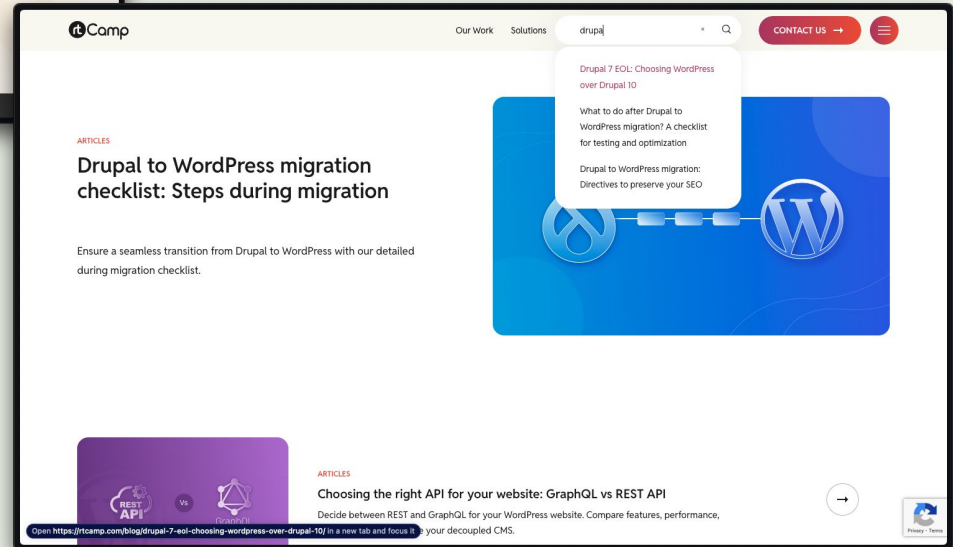
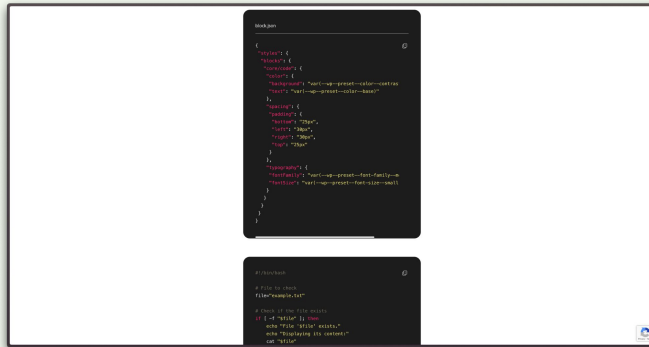
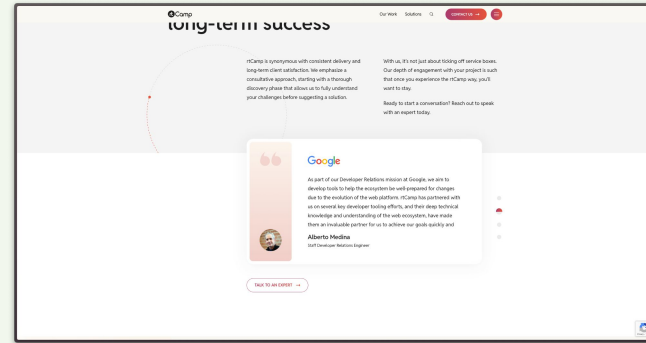
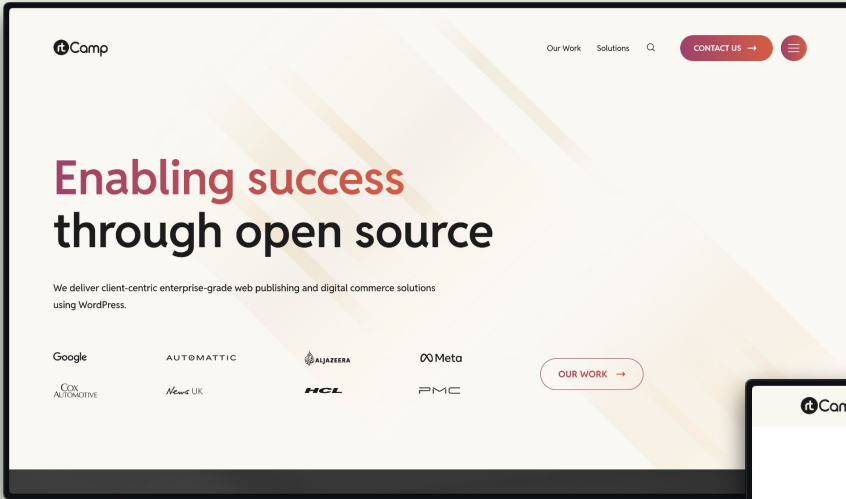
my site, not yours. Don't touch it

o know Wordpress really well, but
ther outdated or still for the "old"

lopers and users that nobody cares
to avoid loading the block editor.

it in the URL bar (/wp-
way from Wordpress as soon as I'm

“Gutenberg is trying to **appeal** to both ‘**simple**’ and ‘**advanced**’ users, and thus is doing **neither** one of them fairly well.”



Pain points during the project

- No support for **CSS Grid**
- No support for **viewport-based styles**
- Not possible to add anything but text in **list block**
- No support for **negative margins**
- No support for **custom width** for blocks
- No support for **simple animations** and **interactions**
- **Decisions - block or pattern?**

Outline

- **Using** Block Editor
- **Developing** Block Themes (and blocks)
- **Improving** Site Experience

A large, vibrant green ink splatter graphic is centered on the page. It features a central, irregular blotch with numerous fine, radiating lines extending outwards, resembling a starburst or a splash. Two smaller, more circular splatters are positioned above and to the right of the main central blotch.

Using Block Editor

Blocks

- **Atomic** elements
- **Consistent** design across posts and pages
- Can be developed using **JavaScript, React, and PHP**
- Core blocks:
 - Paragraph (core/paragraph)
 - Button (core/button)
 - ...and more
- Total core blocks (including experimental) as of Sep 5: 101

Patterns

- **Combination** of blocks
- **Holistically** serves a single purpose (generally)
- Examples:
 - Call-to-action banners
 - Gallery
 - Hero section

The wireframe shows a two-column layout. The left column has a title 'Platform migration' in a large, bold font. Below the title is a paragraph of text: 'We specialize in comprehensive, hassle-free transitions to WordPress, ensuring data integrity, operational continuity, and enhanced website performance. Our approach focuses on reduced costs, improved usability, and complete liberation from vendor lock-ins.' Underneath the text is a row of four logos: VIDEOJET, Dealertrack, VinSolutions, and Manheim. The right column has a title 'What do we offer?' in a bold font. Below the title is a list of six services: Network migration, Data & content re-architecture, SEO & URL redirections, DAM/ERP integrations, Upgrading usability & accessibility standards, Multi-geo & languages support, and Reducing TCO & removing vendor lock-in. At the bottom of the right column is a button labeled 'VIEW DETAILS' with a right-pointing arrow.

Platform migration

We specialize in comprehensive, hassle-free transitions to WordPress, ensuring data integrity, operational continuity, and enhanced website performance. Our approach focuses on reduced costs, improved usability, and complete liberation from vendor lock-ins.

VIDEOJET Dealertrack VinSolutions Manheim

What do we offer?

- Network migration
- Data & content re-architecture
- SEO & URL redirections
- DAM/ERP integrations
- Upgrading usability & accessibility standards
- Multi-geo & languages support
- Reducing TCO & removing vendor lock-in

[VIEW DETAILS →](#)

Synced vs Unsynced Patterns

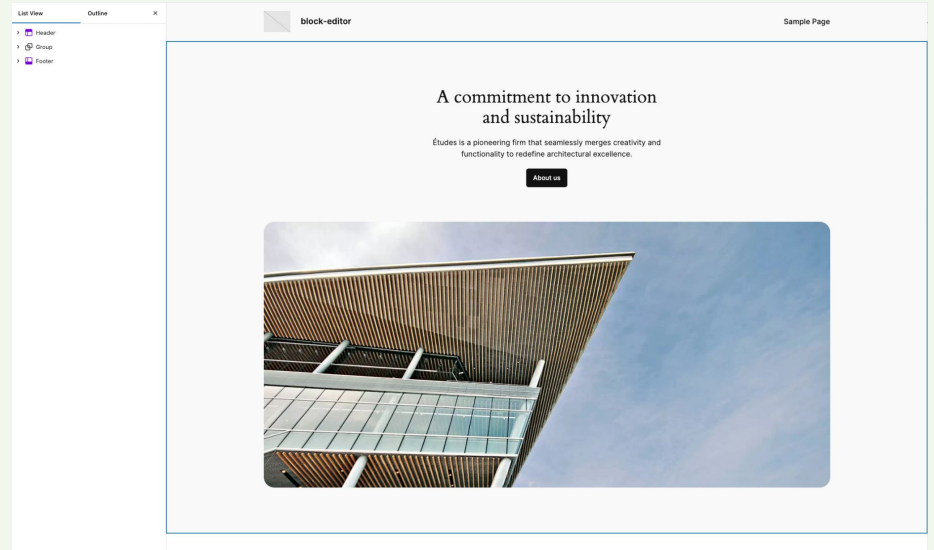
- Synced:
 - Define once
 - Use anywhere
 - Change once
- Unsynced:
 - Define once
 - Use anywhere
 - Change anywhere

Demo

How would you create a section showing
business hours?

Templates and Template Parts

- **Section** of a page (template part)
- Or, an **entire** page (template)
- Defines **site layout**
- Does **not** serve a single purpose
- Examples:
 - Header (template part)
 - Single Post



Patterns vs Templates

	Template Part	Synced Pattern (Reusable block)	Regular Pattern
Type	Site structure	User content	User content
Syncing Ability	Synced	Synced	Un-synced
Examples	Header Footer Sidebar	Business hours Banner Call to action button	Gallery Hero section

The idea of components

- Requirements of components:
 - control the **markup/structure** (HTML)
 - control the **styling** (CSS)
 - add **dynamic/editable** data
 - add **interactivity** (JS)
 - load **only** when used
- Blocks ***almost*** qualify BUT they are atomic
- Examples:
 - Carousel
 - Card

So, what can be used as a component?

- Custom blocks
 - *very* customizable
 - flexible
 - Time-consuming
- In future, a combination of:
 - Synced patterns
 - Pattern Overrides/Content Locking (structure)
 - Section Styles (styling)
 - Block Bindings API (data)
 - Interactivity API (scripts)

Stylebook

- Site Editor (formerly Full-Site Editing/FSE) feature
- **Visual** representation of theme.json (we'll learn about it later)
- Can be used to **modify global** styles
 - Typography
 - Layouts
 - Colors
- Can be used to **modify core block styles**
- Can be used to add **additional CSS**



Developing block themes

Custom blocks: theme or plugin?

- Almost always, **plugins**
- If the blocks are highly coupled with the theme; maybe in theme
- Blocks **should not** rely on theme functionality

Directory Structure

your-blocks-plugin

```
├─ assets/ // non-php code
|   ├─ block-extensions/ // extensions for core blocks
|   ├─ block-variations/ // variations for core blocks
|   ├─ blocks/ // your custom blocks javascript
|   ├─ js/ // other scripts
|   ├─ styles/ // css
|   └─ lib/ // utilities
├─ inc/ // usually php code
|   ├─ classes/ // plugin-related functionality
|   ├─ helpers/ // helper functions
|   └─ traits/ // we use singleton trait
├─ tests/
└─ your-blocks-plugin.php
```

theme.json

- theme.json is **powerful!**
- Understanding the **capability** is important
- Helps achieve
 - Consistency
 - Configurability
 - Variability (Style Variations)
 - Templatzation
- Define as many styles to **theme.json** as possible
- See [WordPress Docs](#)

Use Fluid Typography

- The old approach: **media queries**
 - hard to account for **all screen sizes**
 - hard to **maintain** multiple media queries
- Fluid Typography
 - Improved **UX**
 - **Easy** to maintain
 - Greater design **freedom**
 - Greater **flexibility**
 - Improved **A11Y**

Use Fluid Typography [cont'd...]

```
.text {  
  font-size: clamp(min-value, pref-value, max-value);  
}
```

```
"settings": {  
  "typography": {  
    "fluid": true,  
    "fontSizes": [  
      {  
        "name": "Medium",  
        "slug": "medium",  
        "size": "2.375rem", // preferred value  
        "fluidSize": {  
          "min": "1.875rem", // used below 768px  
          "max": "2.375rem" // used above 1600px  
        }  
      }  
    ]  
  }  
}
```


Use block.json

- **Standard** way to register block types (PHP or JS)
- Allows **code sharing** between PHP and JS
- Simplifies server-side registration of blocks (useful for [Block Type Endpoint](#))
- Asset enqueueing **optimization**
 - only be enqueued when the block is present on the page
 - reduced page sizes
- See [WordPress Docs](#)

Extending Core Blocks vs Custom Blocks

Core Blocks	Custom Blocks
Avoid recreating the wheel by making the most of existing features; saves time	Developers have more control over the final product.
Streamlines the development process for faster production and delivery	Poses less risk for disruption from breaking changes in WordPress core.
Can benefit from future enhancements to the WordPress core block	Developers typically have a lot of experience in developing new blocks.

Case Study

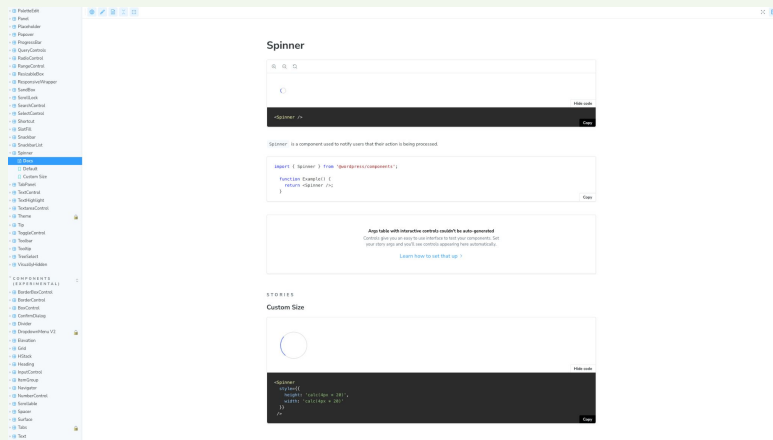
Adding Video Controls to Cover Video

Use Internationalization (i18n)

- Everyone knows why it is important.
- Jeder weiß, warum es wichtig ist.
- 每個人都知道為什麼它很重要。
- ہر کوئی جانتا ہے کہ یہ کیوں ضروری ہے۔
- हर कोई जानता है कि यह क्यों महत्वपूर्ण है।

Use WordPress Components

- Always try to use @wordpress/components for editor UI
- Improves Editorial Experience (EX):
 - **Consistent** with the rest of the admin pages
 - **Familiar** functionality
 - **Saves** a ton of **time**
- See [Storybook](#)

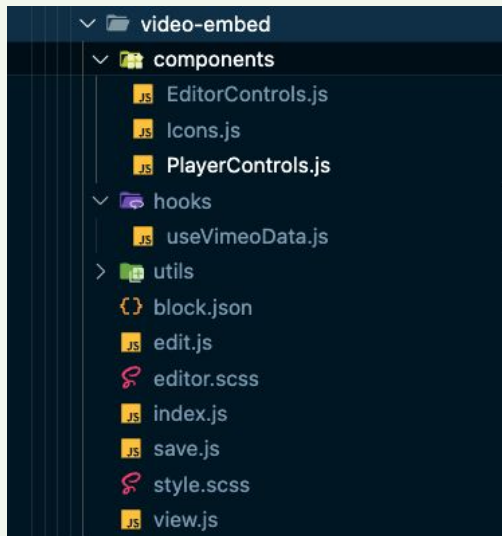


Never compromise on code quality

- For JavaScript:
 - ESLint (@wordpress/eslint-plugin/recommended-with-formatting)
- For PHP:
 - PHPCS (**PHP Coding Standards**)
 - PHPCBF (**PHP Code Beautifier and Fixer**)
- For CSS or other flavors:
 - Stylelint (@wordpress/stylelint-config/scss)

Structure your block code properly

- Blocks are usually **JS/React-heavy**
- Use **structure** and **naming** standards in JS ecosystem



Documentation is key (and the lock)

- Developers usually **struggle** with documentation
- Use **doc-comments** and **inline-comments**
- **Don't beat around the bush!**
- Good comment:
 - Explain intent
 - Warn of consequences
 - Use identifiers: @todo etc

A large, vibrant green ink splatter graphic is centered on the page. It features a central, irregular blotch with numerous fine, radiating lines extending outwards, resembling a starburst or a splash of paint. Two smaller, more circular splatters are positioned above and below the main central blotch.

Improving Site Experience

Site = **Editor** + **Frontend**

Keeping Responsiveness in Mind

- Block editor lacks **out-of-box** responsive controls
- It is a **double-edged sword**:
 - Nice-to-have for editors
 - Nightmare for developers and designers
- Designers:
 - Keep the layouts intuitive and follow natural flow
 - Define breakpoints (and stick to them)
 - Recommended breakpoints: 1920px, 1470px, 1280px, 768px, 360px*
- Developers:
 - Hope that the designers do a good job 🙌
 - Always use relative units (rem, %, vh, vw, etc.)
 - Be ready to write media queries
 - May need to use plugins (like [Block Visibility](#))

Keeping Responsiveness in Mind [cont'd...]

Enabling success through open source

We deliver client-centric enterprise-grade web publishing and digital commerce solutions using WordPress.

Google

AUTOMATTIC

ALJAZEERA

Meta

COX
AUTOMOTIVE

News UK

HCL

PMC

OUR WORK →

>=1280px

Enabling success through open source

We deliver client-centric enterprise-grade web publishing and digital commerce solutions using WordPress.

OUR WORK →

Google

AUTOMATTIC

ALJAZEERA

Meta

COX
AUTOMOTIVE

News UK

HCL

PMC

<1280px

Accessibility

- Cross-check the **order of headings** (use Outline in the editor)
- Do not miss out on the **alt** attribute
- Make sure that your content is **keyboard-accessible** (for devs: *tabindex*)
- Make use of **ARIA** attributes
- Ensure that the contrast is **WCAG-compliant**
- If you're overriding element behavior, use **role** attribute

Final Words

- **Plan** your website properly
- Gutenberg is **powerful**, it is just **misunderstood**
- Honest opinion: for **certain** use-cases, Gutenberg makes **no sense**
- **Learn** about the core functionality; there is usually a **WordPress way**
- Follow **D.R.Y** and **K.I.S.S**
- Keep **accessibility** and **performance** in mind
- **Contribute to the community!**



A large, vibrant green ink splatter graphic is centered on the page. It features a central, irregular blotch with several thin, radiating lines extending outwards, resembling a paint splatter or a stylized starburst. The color is a bright, slightly translucent green.

Thank you!
See you on 23.01.2025