Use Root Padding Aware Alignments

- What is root padding?
- Enabling root padding aware alignments
  - How does this work?

The `settings.useRootPaddingAwareAlignments` property can be one of the most confusing settings in `theme.json`. It is not tied to an interface option. Nor is it used for registering presets. Instead, it's for configuring where WordPress places your theme's horizontal padding styles.

This means it works in conjunction with `styles.spacing.padding` in `theme.json`. You can find out more about general styling in the Styles documentation, but you'll learn how this specific style works alongside this setting here.

What is root padding?

Before understanding how the `settings.useRootPaddingAwareAlignments` property works, you must first understand what root padding is.

Root padding is the padding that is applied to a web page's "root" element. In the case of WordPress themes, this is the `<body>` element. To customize the spacing for the root element, you must target the `styles.spacing.padding` element in `theme.json`.

Take the following `theme.json` snippet, for example. It adds `0` for top and bottom padding and `2rem` for left and right padding:

```
{
    "version": 2,
    "styles": {
        "spacing": {
            "padding": {
```

```
                "top": "0",
                "bottom": "0",
                "left": "2rem",
                "right": "2rem"
            }
        }
    }
}
```

By default, this will add `2rem` of padding on the left and right sides of the `<body>` (root) element.

As shown in this screenshot, there is horizontal padding at the root, and a full-width Cover block stretches until it hits that padding:

> WordPress site header with a Cover block background. It has padding to the left and right of it

This is the CSS that WordPress outputs based on the above `theme.json` code:

```
body {
    padding-top: 0;
    padding-right: 2rem;
    padding-bottom: 0;
    padding-left: 2rem;
}
```

But—and this is where things can look odd to seasoned designers—when `settings.useRootPaddingAwareAlignments` is enabled, the “root” padding is no longer applied to the root element. It is applied to container blocks like Group.

You’ll learn more about why this happens in the next section. The main goal right now is to understand that root padding is traditionally applied to the `<body>` element, and that is what most theme authors would expect.

For root padding aware alignments, WordPress is only concerned with the horizontal (left and right) padding. So the vertical (top and bottom) padding is not relevant to this documentation.

Enabling root padding aware alignments

By default, WordPress will apply root padding to the `<body>` element. If this makes sense for your theme's design, you don't need to do anything else.

*But what if you want to let full-width items stretch to the edges of the screen **and** *have padding on the root element? *

When compared to the first screenshot, notice how the Cover block stretches to edges of the screen but there is padding still applied to the nested blocks here:

> WordPress site header with a Cover block background that stretches the full width of the scree

This is a common design pattern, and there are several ways to stretch elements beyond their containers in CSS. But WordPress has a standard approach that would work with any theme.

That's where `settings.useRootPaddingAwareAlignments` comes in. When this property is set to `true`, it puts the root padding on container elements instead of `<body>`. This must also be combined with a `styles.spacing.padding` object, particularly with horizontal padding applied.

Try this code in your `theme.json`:

```
{
    "version": 2,
    "settings": {
        "useRootPaddingAwareAlignments": true
    },
    "styles": {
        "spacing": {
            "padding": {
                "top": "0",
                "bottom": "0",
                "left": "2rem",
                "right": "2rem"
            }
        }
    }
}
```

There is no right or wrong way to handle root padding. It is a situational setting where you must decide which option is best for your theme's design.

How does this work?

It’s not particularly vital to know how WordPress handles all of this under the hood, but sometimes you might just want a deeper understanding of what’s going on.

When you enable `settings.useRootPaddingAwareAlignments`, as shown in the last `theme.json` example, WordPress will generate two new bits of CSS. The first is that it defines some CSS custom properties for the root padding:

```
body {
    --wp--style--root--padding-top: 0;
    --wp--style--root--padding-right: 2rem;
    --wp--style--root--padding-bottom: 0;
    --wp--style--root--padding-left: 2rem;
}
```

The second is that it adds a `.has-global-padding` class:

```
.has-global-padding {
    padding-right: var(--wp--style--root--padding-right);
    padding-left: var(--wp--style--root--padding-left);
}
```

This class is then given to container blocks with constrained layouts (for example, Group blocks with the **Layout > Inner blocks use content width** option enabled):

```
<div class="wp-block-group has-global-padding is-layout-constrained wp-block-group-is-layout-constrained">
    <!-- nested blocks... -->
</div>
```

Beyond that, WordPress adds inline CSS to stretch nested wide and full-width blocks beyond their parent block’s width (including the additional padding that’s added to the width).

□□□ #2
□ □□□ □□□ 2026-10-31 18:39:42 UTC
□ □□□ □□□ 2026-10-31 18:39:44 UTC