Customizin Gutenberg Features from functions.php

Rather than defining settings in each individual components, you can define what core features the theme supports in the theme’s `functions.php`, using add_theme_support.

`add_theme_support` takes one or two arguments. The first argument is the name of the feature you are adding support for and the second argument, if present, is a list of argument for the feature you’re adding support for.

```php
<?php
add_theme_support( 'cust'custom-header'?>
```

`custom-header` takes an optional array of parameters. As with many functions in WordPress, you can assign the array of parameters to a variable and use that variable when you declare the `add_theme_support` function.

```php
<?php
$defaults = array(
    'default-image'          => '',
    'random-default'         => false,
    'width'                  => 0,
    'height'                 => 0,
    'flex-height'            => false,
    'flex-width'             => false,
    'default-text-color'     => '',
    'header-text'            => true,
    'uploads'                => true,
    'wp-head-callback'       => '',
    'admin-head-callback'    => '',
    'admin-preview-callback' => '',
    'video'                  => false,
    'video-active-callback'  => 'is_front_page',
);
add_theme_support( 'custom-header', $defaults );
```

This function allows you to customize your theme or child theme with just the feature set that you want or need.

The more information section of the `add_theme_support` function has more information on the arguments you can pass to individual features.

remove_theme_support()

remove_theme_support() is the opposite function. It allows you to remove feature support from your theme or child theme.

An example of why this is useful is when your parent theme enabled support for a feature you don’t want. You could change it in the parent, ignore it, or you could explicitly remove support for the feature.

add_theme_support() equivalency in theme.json

theme.json (new in WordPress 5.8) aims to take over and consolidate all the various `add_theme_support` calls that were previously required for controlling the editor. Rather than cluttering `functions`.php with calls to add_theme_support, you can get the same functionality by performing the corresponding calls in `theme.json`.

add_theme_support	theme.json setting
custom-line-height	Set `typography.customLineHeight` to false
custom-spacing	Set `spacing.customPadding` to true
custom-units	Provide the list of units via `spacing.units`
disable-custom-colors	Set `color.custom` to false
disable-custom-font-sizes	Set `typography.customFontSize`

add_theme_support	**theme.json setting**
disable-custom-gradients	Set `color.customGradient` to false
editor-color-palette	Provide the list of colors via `color.palette`
editor-font-sizes	Provide the list of font size via `typography.fontSizes`
editor-gradient-presets	Provide the list of gradients via `color.gradients`
experimental-link-color	Set `color.link` to true