

WordPressowe Wojny. Czy to już czas by przejść na Reactową stronę mocy?

LENA SĘDKIEWICZ



Psycholog

Terapeuta TSR

Dziennikarka

Trener szermierki

Copywriterka

Od 2021 - web developer

F1 - Max!

szydełkowanie

WWO - icojarobietu.pl



Ale... czemu rozmawiamy o React w WordPress?

- Pracuję zarówno z React / Vue jak i z WordPressem
- Obserwuję rozwój ekosystemu WordPress
- Analizuję potrzeby klientów
- Analizuję oferty pracy
- Działam w różnych projektach

Chcę wiedzieć, w co inwestować swój czas!

Czy robisz coś w Vue?

WordPress

Co obecnie zawierają jego “trzewia”
i gdzie znajdziemy React?



Panel użytkownika

Gutenberg

Prerequisites

To complete this tutorial, you will need:

1. Code editor
2. Node.js development tools
3. Local WordPress environment

```
export default function Edit() {  
  const currentYear = new Date().getFullYear().toString();  
  
  return (  
    <>  
      <InspectorControls>  
        <PanelBody title={__( 'Settings', 'copyright-date-block' ) }  
          Testing  
        </PanelBody>  
      </InspectorControls>  
      <p { ...useBlockProps() }>© { currentYear }</p>  
    </>  
  );  
}
```

<https://developer.wordpress.org/block-editor/getting-started/tutorial/>

Gutenberg

This will work but can be improved by ensuring this code only runs once when the block is initialized. To do so, you can use the `useEffect` React hook. Refer to the React documentation for more information about how to use this hook.

First, import `useEffect` with the following code.

```
import { useEffect } from 'react';
```

Then wrap the `setAttribute()` code above in a `useEffect` and place this code after the `currentYear` definition in the `Edit()` function. The result should look like this.

```
export default function Edit( { attributes, setAttributes } ) {  
  const { fallbackCurrentYear, showStartingYear, startingYear } = attributes;  
  
  // Get the current year and make sure it's a string.  
  const currentYear = new Date().getFullYear().toString();  
  
  // When the block loads, set the fallbackCurrentYear attribute to the  
  // current year if it's not already set.  
  useEffect( () => {  
    if ( currentYear !== fallbackCurrentYear ) {  
      setAttributes( { fallbackCurrentYear: currentYear } );  
    }  
  } );  
}
```

<https://developer.wordpress.org/block-editor/getting-started/tutorial/>

Szablony



Budowanie strony na WP... z React?

Rozwiązania low code / no code

Low code / no code

- Umożliwiają edycję bez znajomości kodowania /przy minimalnej znajomości kodowania
- Osoby samodzielnie tworzące swoją stronę
- Osoby chcące móc często zmieniać wygląd strony
- Osoby chcące stworzyć szybko nowy projekt
- Page builders / FSE

Page buildery

<https://www.hostinger.com/tutorials/wordpress-page-builders>

Top 11 Free WordPress Page Builders >

1. Ultimate Blocks >
2. Spectra >
3. Beaver Builder >
4. SiteOrigin Page Builder >
5. Gutenberg by Kadence Blocks >
6. Themify Builder >
7. Live Composer >
8. Elementor >
9. Page Builder Sandwich >
10. Fusion Page Builder >
11. Colibri Page Builder >

Top 9 Premium WordPress Page Builders >

1. Divi >
2. Stackable >
3. Visual Composer >
4. Thrive Architect >
5. WPBakery >
6. Brizy >
7. SeedProd >
8. MotoPress >
9. Oxygen >

Elementor

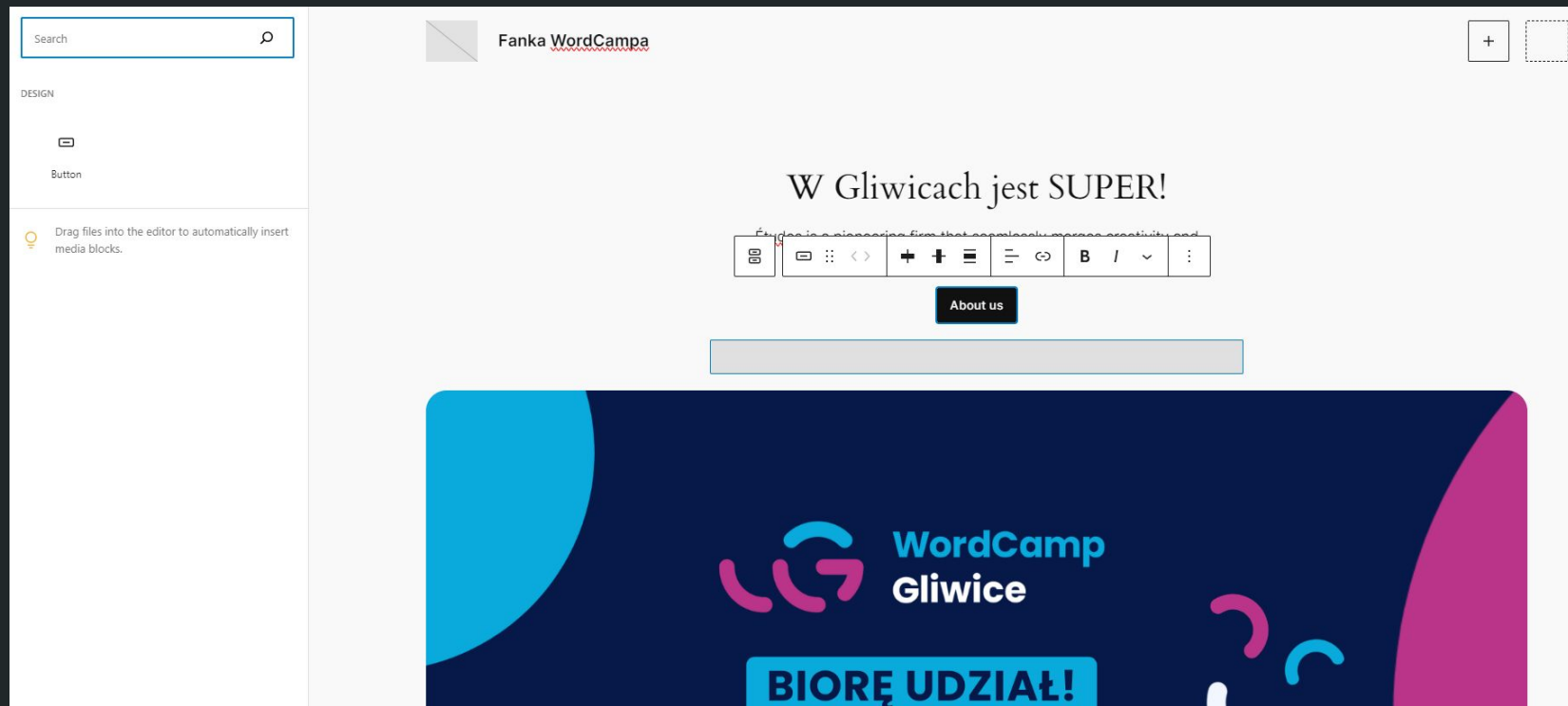
```
class Module extends BaseModule {  
  
    const PAGE_ID = 'elementor-apps';  
  
    public function get_name() {  
        return 'apps';  
    }  
  
    public function __construct() {  
        parent::__construct();  
  
        Admin_Pointer::add_hooks();  
  
        add_action( 'elementor/admin/menu/register', function( Admin_Menu_Manager $admin_menu ) {  
            $admin_menu->register( static::PAGE_ID, new Admin_Menu_Apps() );  
        }, 115 );  
    }  
}
```

Fragment kodu Elementora

```
<?php  
/**  
 * Define MyClass  
 */  
class MyClass  
{  
    public $public = 'Public';  
    protected $protected = 'Protected';  
    private $private = 'Private';  
  
    function printHello()  
    {  
        echo $this->public;  
        echo $this->protected;  
        echo $this->private;  
    }  
}  
  
$obj = new MyClass();  
echo $obj->public; // Works  
echo $obj->protected; // Fatal Error  
echo $obj->private; // Fatal Error  
$obj->printHello(); // Shows Public, Protected and Private
```

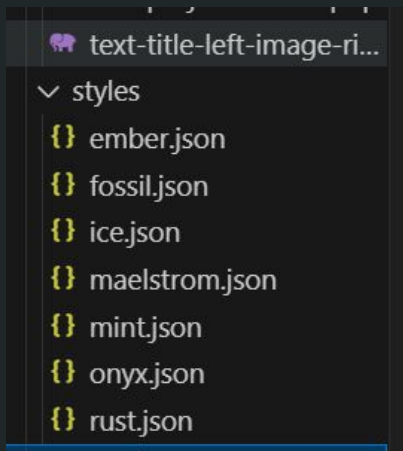
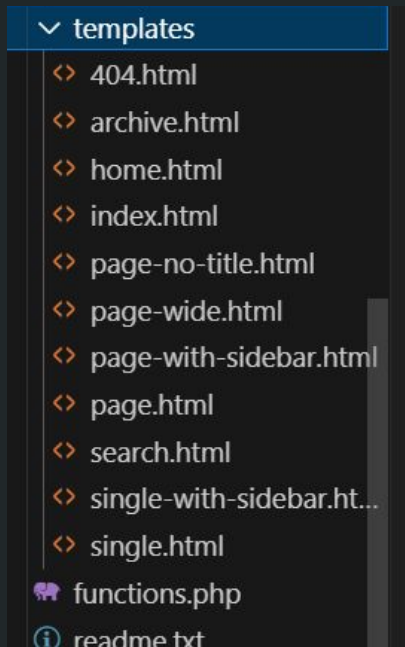
<https://www.php.net/manual/en/language.oop5.visibility.php>

Full Site Editing



Full Site Editing

Block Grammar & Markup



<https://fullsiteediting.com/blocks/calendar/>

```
<!-- wp:calendar /-->
```

```
{
  "align": true,
  "color": {
    "link": true,
    "__experimentalSkipSerialization": [
      "text",
      "background"
    ],
    "__experimentalDefaultControls": {
      "background": true,
      "text": true
    },
    "__experimentalSelector": "table, th"
  },
  "typography": {
    "fontSize": true,
    "lineHeight": true,
    "__experimentalFontFamily": true,
    "__experimentalFontWeight": true,
    "__experimentalFontStyle": true,
    "__experimentalTextTransform": true,
    "__experimentalLetterSpacing": true,
    "__experimentalDefaultControls": {
      "fontSize": true
    }
  }
}
```

Full Site Editing

Block grammar basics

A basic block uses an HTML comment that WordPress parses as a block front:

```
<!-- -->
```

Format

Theme.json uses JSON format. One of the reasons the Gutenberg developers selected JSON was its ease of use. If you are familiar with arrays or JavaScript Objects, you will have no problems working with theme.json.

New to JSON and want to read more about it? Then, I recommend these resources:

<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>

<https://www.json.org/json-en.html>

<https://fullsiteediting.com/lessons/>

Full Site Editing

Using the development version of Gutenberg

If you want to try the latest site editing features, you can do a git clone from the GitHub repository.

The node and npm versions needed for Gutenberg were updated in January 2024. You will now need node greater than or equal to 20.10.0, and npm version 10.2.3 or higher.

Using the terminal inside the plugin folder of your WordPress installation:

```
git clone https://github.com/WordPress/gutenberg
```

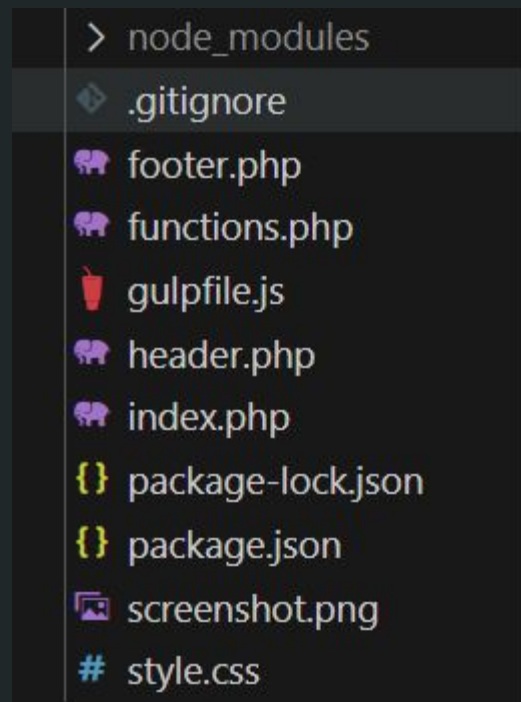
Copy

<https://fullsiteediting.com/lessons/preparations-quick-start-guide/>

Motywy “klasyczne”

Motywy oparte o PHP

- “Stara szkoła” tworzenia motywów
- JDG / Firmy, które mają budżet na programistę
- Różne podejścia
 - czysty PHP
 - PHP + Laravel, Blade PHP etc.
 - ACF
 - Bedrock, Sage (<https://roots.io/>)





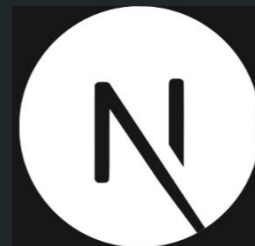
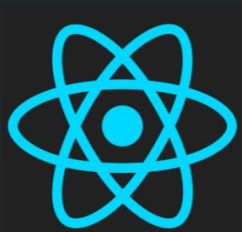
Headless WP

Headless WP, czyli?

- Wordpress jako backend (tzw. Decoupled CMS od wersji 4.7)
- Frontend - co podpowiada wyobraźnia!
 - SSR

Headless WP, czyli?

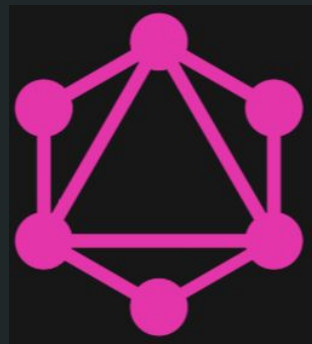
- Wordpress jako backend (tzw. Decoupled CMS od wersji 4.7)
- Frontend - co podpowiada wyobraźnia!
 - SSR, albo....



Łącznik?

REST API

WP Graph QL



Headless - zastosowania, wady, zalety

- Rozbudowane portale i aplikacje
- Zalety
 - skalowalny
 - marketing wciąż korzysta z CMS
 - strona wygląda tak, jak w marzeniach prezesa
 - (security?)
- Wady
 - utrzymanie
 - potrzebny dev z wiedzą poza WP
 - nie wszystkie funkcjonalności / pluginy są dostępne

Hybrydy

“Hybrydy”

- Strona na WP
 - ułatwienie publikacji / szybkie zmiany treści niezależne od devów
 - mniejszy próg wejścia w obsłudze strony (m.in. development, SEO)
- Aplikacja - React / Vue / Angular etc.
 - więcej możliwości rozwoju
 - większe możliwości zaopiekowania tematu bezpieczeństwa

Podsumowanie, czyli kto wygra WordPressowe Wojny?

Małe projekty - WP!

- mniej pracy
 - przedsiębiorca wiele może zrobić sam
 - mniej kwestii związanych z utrzymaniem do zaopiekowania
-
- React? Gdy ktoś będzie chciał rozwijać Gutenberga na stronie na WP

Większe projekty?

- Headless WP z różnym stackiem
- szybkość (SEO)
- bezpieczeństwo
- zasoby do zakodowania szalonych snów osób nietechnicznych / mniej technicznych

WordPress w większych projektach komercyjnych to coraz częściej Headless!



Headless WordPress



Lisan al-Gaib

Headless is the future of the web.

The demand for powerful, omnichannel digital experiences is driving enterprise headless adoption.

64%

64% of enterprise organizations are currently using a headless approach—up nearly 25% from 2019.

92%

92% of respondents see the importance of digital experiences in terms of their organization's success.

92%

92% of respondents say headless technologies make it easier to deliver a consistent content experience across channels.

90%

Among respondents not currently using headless, more than 90% plan to evaluate headless solutions over the next 12 months.

Tak, robię w Vue ;)

- <https://www.linkedin.com/in/lena-se dkiewicz/>
- <https://msedkiewicz.pl/>
- <https://www.instagram.com/msedkie wicz/> - szydełkowanie



Źródła

- <https://raidboxes.io/en/blog/webdesign-development/headless-cms-wordpress/>
- <https://nextjs.org/>
- <https://vercel.com/guides/wordpress-with-vercel>
- <https://www.clarity-ventures.com/ecommerce/integrations/content-management-systems/wordpress-cms/how-to-integrate-angular-in-a-wordpress-website>
- <https://wpengine.com/headless-cms-research/>
- <https://www.experro.com/blog/headless-commerce-statistics/>
- <https://www.evrenbal.com/en/graphql-vs-rest-api-which-is-the-best-choice-for-headless-wordpress/>
- <https://kinsta.com/blog/wpgraphql-vs-wp-rest-api/>