



WEBDESIGN

OD PODSTAW 2

WORDPRESS, SQL, HTML, CSS, PHP



Wydanie 1

WEBDESIGN

OD PODSTAW 2

Autor, redakcja, ilustracje, skład:

Mateusz Wiliński

Wydanie 1

Kontakt:

PIKADEMIA Mateusz Wiliński

www.pikademia.pl

pikademia@gmail.com



Publikacja Webdesign od podstaw 2 została wydana w systemie self-publishing pod znakiem PIKADEMIA.

Wszystkie prawa zastrzeżone

Aktualna wersja ebooka z plikami do pobrania:

https://drive.google.com/drive/folders/1VF7eqE0XnyvO-E1_SUJnQBB0LABB2SPI?usp=sharing

Po więcej kursów i ebooków zapraszamy na <https://pikademia.pl/>

Legnica, 2021

| | |
|-------------------------------------------------------------------------------------------------|----|
| Spis treści | 3 |
| Tworzenie pluginu | 4 |
| Podstawy | 4 |
| Zmiana prefixu bazy danych | 4 |
| Filtr the_content | 5 |
| Zabezpieczenie przed bezpośrednim dostępem do pliku | 5 |
| Kompletny kod pluginu | 6 |
| Struktura motywu | 7 |
| Plik style.css | 8 |
| Plik functions.php, przypisanie plików css i js do motywu | 9 |
| Wyświetlenie artykułów i strony | 10 |
| Formatowanie daty i czasu w wordpress | 10 |
| Zmiana wyświetlenia ze względu na rodzaj posta | 10 |
| Menu - add_theme_support() | 11 |
| Przykład dodania opcji tła strony i miniatur artykułów: | 11 |
| CSS zmienne | 12 |
| Kolejność uruchamiania haków w wordpress: | 12 |
| Wyświetlenie i Zmiana długości zajawki (excerpt), której domyślna wartość wynosi 55 słów: | 12 |
| Rejestracja widżetów w pliku functions.php | 13 |
| Wyświetlanie tytułu jako link dla artykułów: | 14 |
| Dodanie bootstrap do strony wp | 14 |
| Ikony bootstrap | 14 |
| Pobieranie treści z bazy danych | 15 |
| Pobieranie postów w danej kategorii (t.term_id) i wyświetlenie ich tytułu i zawartości: | 15 |
| Tworzenie własnych funkcji | 16 |

Podstawy

Aby stworzyć plugin, niezbędny jest chociaż jeden plik. Jeśli umieszczamy ten plik wewnątrz folderu to jego nazwa powinna być taka sama, jak nazwa folderu.

Np. plik `mój_plugin.php` będzie wewnątrz folderu `mój_plugin`

Podczas tworzenia pluginów niezbędne jest używanie haków, czyli punktów zaczepnych, które wykonują się w określonym momencie. Będziemy przypisywać do tych haków różne akcje i filtry, tak, by mogły zostać one wywołane w odpowiednim momencie.

Na początku każdego pluginu należy dodać kilka informacji określających plugin. Robi się to w komentarzu blokowym, a podane informacje są zaczytywane przez wordpress i wyświetlane w panelu administracyjnym.

Np.

```
<?php
/*
Plugin Name: Add text after content
Description: Nasz najlepszy pierwszy plugin, który dodaje tekst do treści artykułu i strony
Author: pikademia
Version: 1.0.1
Author URI: https://pikademia.pl
*/
```

Przydatne funkcje:

`add_filter('nazwa_haka', 'nazwa funkcji dołączanej do haka')`

`add_action('nazwa_haka', 'nazwa funkcji dołączanej do haka')`

`is_front_page()` zwraca true, jeśli jesteśmy na stronie głównej

`is_home()` zwraca prawdę, jeśli jesteśmy na stronie z wpisami

`is_singular('post')` lub `is_single()` zwraca prawdę, jeśli jesteśmy wewnątrz artykułu

`is_page()` zwraca prawdę, jeśli jesteśmy na stronie statycznej

`in_category(array('b', 'a'))` zwraca prawdę, jeśli dany post jest w podanej kategorii

Zmiana prefixu bazy danych

```
SELECT * FROM `xd_options` WHERE `option_name` LIKE '%naszwp_%'
```

```
SELECT * FROM `xd_usermeta` WHERE `meta_key` LIKE "%naszwp_%"
```

Filtr the_content

`add_filter('the_content','nazwa funkcji')` zmienia zawartość strony lub artykułu

```
function add_text_after_content_change_content($content){
    if( is_singular('post') && is_main_query() ){

        $before_content = '<p> A </p>';
        $after_content = '<p> Z </p>';
        $content = $before_content.$content.$after_content;
    }
    return $content;
}

add_filter('the_content','add_text_after_content_change_content');
```

Wyświetlanie dodatkowej treści w pojedynczych artykułach ze względu na kategorie

https://codex.wordpress.org/Conditional_Tags

<https://digwp.com/2010/03/add-plugin-to-wordpress-plugin-repository/>

<https://wordpress.org/plugins/developers/add/>

Zabezpieczenie przed bezpośrednim dostępem do pliku

```
if ( ! defined( 'ABSPATH' ) ) {
    exit; // Exit if accessed directly
}
```

lub

```
if ( !function_exists( 'add_action' ) ) {
    echo 'Hi there! I\'m just a plugin, not much I can do when called directly
    .';
    exit;
}
```

Kompletny kod pluginu

```
<?php
/*
Plugin Name: Add text after content
Description: Nasz najlepszy pierwszy plugin, który dodaje tekst do treści artykułu i strony
Author: pikademia
Version: 1.0.2
Author URI: https://pikademia.pl
*/
if ( ! defined( 'ABSPATH' ) ) {
    exit; // Exit if accessed directly
}
function add_text_after_content_change_content($content){
    if(is_single() && is_main_query()){
        $before_content = '<p class="tekst-dodany-przed"> Hello! </p>';
        $after_content = '<p class="tekst-dodany"> Do zobaczyska następnym razem! </p>';
        $content = $before_content.$content.$after_content;
        echo
        "<style>
            .tekst-dodany-przed{
                color:blue;
                font-size:1em;
            }
            .tekst-dodany{
                color:red;
                font-size:10px;
            }
        </style>";
    }
    return $content;
}
add_filter('the_content','add_text_after_content_change_content');
```

https://codex.wordpress.org/Theme_Development

Motyw w wordpresie jest stworzony z wielu plików i katalogów.
Foldery główne z motywami znajdują się w katalogu wp-content >> themes.

Przykładowa struktura motywu twntytwentyone wygląda następująco:

| | | | | | |
|-----------------------|-----------------------|-------------------|---------------|---------|--|
| .. | | | | | |
| assets | File folder | 15/04/2021 04:... | flcdmpe (0... | ftp ftp | |
| classes | File folder | 15/04/2021 04:... | flcdmpe (0... | ftp ftp | |
| inc | File folder | 15/04/2021 04:... | flcdmpe (0... | ftp ftp | |
| template-parts | File folder | 15/04/2021 04:... | flcdmpe (0... | ftp ftp | |
| .stylelintignore | 89 STYLELINTI... | 12/11/2020 19:... | adfrw (0644) | ftp ftp | |
| .stylelintrc-css.json | 688 JSON File | 18/02/2021 21:... | adfrw (0644) | ftp ftp | |
| .stylelintrc.json | 356 JSON File | 12/11/2020 19:... | adfrw (0644) | ftp ftp | |
| 404.php | 721 PHP Sourc... | 17/12/2020 15:... | adfrw (0644) | ftp ftp | |
| archive.php | 998 PHP Sourc... | 17/12/2020 15:... | adfrw (0644) | ftp ftp | |
| comments.php | 2,669 PHP Sourc... | 07/02/2021 13:... | adfrw (0644) | ftp ftp | |
| footer.php | 2,122 PHP Sourc... | 17/12/2020 15:... | adfrw (0644) | ftp ftp | |
| functions.php | 18,614 PHP Sourc... | 22/12/2020 16:... | adfrw (0644) | ftp ftp | |
| header.php | 982 PHP Sourc... | 17/12/2020 15:... | adfrw (0644) | ftp ftp | |
| image.php | 3,379 PHP Sourc... | 07/02/2021 13:... | adfrw (0644) | ftp ftp | |
| index.php | 946 PHP Sourc... | 17/12/2020 15:... | adfrw (0644) | ftp ftp | |
| package-lock.json | 172,893 JSON File | 14/04/2021 02:... | adfrw (0644) | ftp ftp | |
| package.json | 2,351 JSON File | 14/04/2021 02:... | adfrw (0644) | ftp ftp | |
| page.php | 590 PHP Sourc... | 17/12/2020 15:... | adfrw (0644) | ftp ftp | |
| postcss.config.js | 249 JavaScript ... | 18/01/2021 18:... | adfrw (0644) | ftp ftp | |
| readme.txt | 4,606 Text Docu... | 14/04/2021 02:... | adfrw (0644) | ftp ftp | |
| screenshot.png | 282,897 PNG File | 17/11/2020 20:... | adfrw (0644) | ftp ftp | |
| search.php | 1,647 PHP Sourc... | 07/02/2021 13:... | adfrw (0644) | ftp ftp | |
| searchform.php | 1,432 PHP Sourc... | 17/12/2020 15:... | adfrw (0644) | ftp ftp | |
| single.php | 1,698 PHP Sourc... | 07/02/2021 13:... | adfrw (0644) | ftp ftp | |
| style-rtl.css | 154,885 CSS Source... | 14/04/2021 02:... | adfrw (0644) | ftp ftp | |
| style.css | 155,185 CSS Source... | 14/04/2021 02:... | adfrw (0644) | ftp ftp | |

Podstawowymi plikami każdego motywu są pliki index.php i style.css.

Plik style.css

W pliku style.css podajemy wszelkie informacje związane z motywem, np.;

```
@charset "UTF-8";

/*
Theme Name: Pika theme
Theme URI: https://pikademia.pl
Author: Pikateam
Author URI: https://pikademia.pl
Description: Pika theme is a simple theme ready for adjustments
Requires at least: 5.3
Tested up to: 5.7
Requires PHP: 5.6
Version: 0.1
License: GNU General Public License v2 or later
License URI: http://www.gnu.org/licenses/gpl-2.0.html
Tags: one-column, accessibility-ready, custom-colors, custom-menu, custom-
logo, editor-style, featured-images, footer-widgets, block-patterns, rtl-
language-support, sticky-post, threaded-comments, translation-ready

Pika theme is distributed under the terms of the GNU GPL.
*/
```


Plik functions.php, przypisanie plików css i js do motywu

Kolejnym krokiem jest dodanie pliku functions.php do folderu głównego naszego motywu, tuż obok pozostałych plików. Dodam jeszcze, że te nazwy powinny zostać niezmienione, ponieważ wordpress oczekuje właśnie takich nazw i ich zmiana spowoduje błędy.

Plik functions.php jest bardzo ważny, będziemy tutaj umieszczać wiele powiązań i funkcjonalności dotyczącej naszej strony. Na początek dodamy odniesienia do wcześniej utworzonego pliku CSS.

```
function pikademia_scripts() {  
    wp_enqueue_style( 'pikademia_style', get_template_directory_uri() . '/style.css', array(),  
wp_get_theme()->get( 'Version' ) );  
}  
add_action( 'wp_enqueue_scripts', 'pikademia_scripts' );
```

Aby dodać skrypt js o nazwie pikademia_js.js, który znajduje się w folderze js, można umieścić wewnątrz funkcji linię:

```
wp_enqueue_script('pikademia_js',get_template_directory_uri().'/js/pikademia_js.js', array(  
) , '1.0.0', true);
```

W pliku header.php należy umieścić kod `<?php wp_head(); ?>`, który wyznaczy miejsce w którym należy dodać plik css.

```
<!doctype html>  
<html>  
    <head>  
        <meta charset="utf-8">  
        <title>Pika theme</title>  
        <?php wp_head(); ?>  
    </head>  
    <body>
```

W pliku footer.php należy umieścić kod `<?php wp_footer(); ?>`, który wyznaczy miejsce w którym należy dodać plik js.

```
<footer></footer>  
<?php wp_footer(); ?>  
</body>  
</html>
```

Wyświetlenie artykułów i strony

Wyświetlanie postów w połączeniu z html

```
if ( have_posts() ) {  
    while ( have_posts() ) {  
        the_post();  
    }  
    <h3><?php the_title(); ?></h3>  
    <h4>Kategoria: <?php the_category(); ?></h4>  
    <h4>Data publikacji: <?php the_time('j M Y'); ?></h4>  
    <p><?php the_content(); ?></p>  
    <hr>  
<?php  
}
```

Formatowanie daty i czasu w wordpress

<https://wordpress.org/support/article/formatting-date-and-time/>

Zmiana wyświetlenia ze względu na rodzaj posta

```
if ( have_posts() ) {  
    while ( have_posts() ) {  
        the_post();  
        if(is_page()){  
            the_content();  
        }else{  
            ?>  
            <h3><?php the_title(); ?></h3>  
            <h4>Kategoria: <?php the_category(); ?></h4>  
            <h4>Data publikacji: <?php the_time('j F Y'); ?></h4>  
            <p><?php the_content(); ?></p>  
            <hr>  
            <?php  
        }  
    }  
}
```

<https://blog.teamtreehouse.com/wordpress-loop-beginners-guide>

Menu - add_theme_support()

Aby dodać funkcjonalność menu należy w pliku functions.php dodać funkcję `add_theme_support('menus')`. Możemy dodać kilka miejsc w których może pokazać się menu na stronie za pomocą metody `register_nav_menu('id_menu', 'nazwa menu')`

```
function pikademia_menu_setup(){
    add_theme_support('menus');
    register_nav_menu('primary', 'Main Header Menu');
    register_nav_menu('footer', 'Footer Menu');
}

add_action('after_setup_theme', 'pikademia_menu_setup');
```

Aby wyświetlić menu, należy w odpowiednim miejscu na stronie dodać kod z id menu, np.:

```
<?php wp_nav_menu(array('theme_location'=>'primary')); ?>
<?php wp_nav_menu(array('theme_location'=>'footer')); ?>
```

`add_theme_support` opcje: https://developer.wordpress.org/reference/functions/add_theme_support/

Przykład dodania opcji tła strony i miniatur artykułów:

```
add_theme_support( 'custom-background' );
add_theme_support( 'post-thumbnails' );
<div class="thumbnail-img"><?php the_post_thumbnail('thumbnail'); ?></div>
```

`add_theme_support()` powinny być wywoływane w pliku functions.php lub przypisane do haka `after_setup_theme`

Dodanie klasy wraz z obrazkiem:

```
<?php if ( has_post_thumbnail() ) {
    the_post_thumbnail('large', array('class' => 'img-
fluid'));
}
```

CSS zmienne

```
:root{
  --theme-color: #44aaaa
}
background-color: var(--theme-color);
```

Kolejność uruchamiania haków w wordpress:

https://codex.wordpress.org/Plugin_API/Action_Reference

https://codex.wordpress.org/Post_Thumbnails

Wyświetlenie i Zmiana długości zajawki (excerpt), której domyślna wartość wynosi 55 słów:

```
<p><?php the_excerpt(); ?></p>
```

```
function new_excerpt_length($length) {
    return 15;
}
add_filter('excerpt_length', 'new_excerpt_length');
```

Rejestracja widgetów w pliku functions.php

```
add_theme_support('widgets');
```

```
function pikademia_register_widgets() {

    $shared_args = array(
        'before_title'  => '<h2 class="widget-title">',
        'after_title'   => '</h2>',
        'before_widget' => '<div class="widget %2$s"><div class="widget-content">',
        'after_widget'  => '</div></div>',
    );

    register_sidebar(
        array_merge(
            $shared_args,
            array(
                'name'          => __( 'Widget #1', 'pikademia' ),
                'id'            => 'sidebar-1',
                'description'   => __( 'Desc', 'pikademia' ),
            )
        )
    );

    register_sidebar(
        array_merge(
            $shared_args,
            array(
                'name'          => __( 'Widget #2', 'pikademia' ),
                'id'            => 'sidebar-2',
                'description'   => __( 'Desc', 'pikademia' ),
            )
        )
    );
}

add_action( 'widgets_init', 'pikademia_register_widgets' );
```

Umieszczenie widgetów na stronie

```
<div class="pi-widgets">
    <?php dynamic_sidebar( 'sidebar-1' ); ?>
</div>

<div class="pi-widgets">
    <?php dynamic_sidebar( 'sidebar-2' ); ?>
</div>
```

Wyświetlanie tytułu jako link dla artykułów:

```
<header class="entry-header">
    <?php
        if (is_singular()){
            the_title('<h1 class="entry-title">', '</h1>' );
        }else{
            the_title( sprintf('<h2 class="entry-
title"><a href="%s">', esc_url( get_permalink() ) ), '</a></h2>' );
        }
    ?>
</header>
```

lub

```
<a href="<?php the_permalink(); ?>">
    <?php the_title(); ?>
</a>
```

Dodanie bootstrap do strony wp

W sekcji <head>:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-+0n0xVW2eSR50omGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOO17+AMvyTG2x" crossorigin="anonymous">
```

W footerze, przed sekcją zamykającą </body>:

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-gtEjrD/SeCtmISkJKNUaakMoLD0//ElJ19smozuHV6z3Iehds+3U1b9Bn9Plx0x4" crossorigin="anonymous"></script>
```

Ikony bootstrap

<https://icons.getbootstrap.com/>

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">
```

```
<i class="bi-alarm" style="font-size: 2rem; color: cornflowerblue;"></i>
```

Pobieranie treści z bazy danych

<https://www.smashingmagazine.com/2011/09/interacting-with-the-wordpress-database/?fbclid=IwAR0JrefnpkbLKeiU-x8W9GMla9SslSHGk5i9gzqKUlyTY6bVsQq9YLeZflQ>

Pobieranie danych dla rekordu(strony) o tytule 'baner', a następnie wyświetlenie post_content za pomocą funkcji

```
function pi_get_page($arg){
    global $wpdb;
    $posts = $wpdb->get_row("
        SELECT *
        FROM ` $wpdb->posts ` p
        WHERE p.post_title='$arg' AND p.post_type='page' AND p.post_status = 'publish' "
    );

    if($posts != 0){
        echo $posts->post_content;
    }
}
pi_get_page("baner");
```

W bazie danych:

```
SELECT * FROM `xd_posts` p
WHERE p.post_title='baner' AND p.post_type='page' AND p.post_status = 'publish'
```

Pobieranie postów w danej kategorii (t.term_id) i wyświetlenie ich tytułu i zawartości:

```
$postsInCategory = $wpdb->get_results("
select p.post_title, p.post_content, t.name
from `xd_posts` p, `xd_term_relationships` tr, `xd_terms` t, `xd_term_taxonomy` tt
where p.ID=tr.object_id AND tr.term_taxonomy_id=t.term_id AND tt.term_id = t.term_id and
p.post_type='post' AND p.post_status='publish'
AND t.term_id=6");

foreach($postsInCategory as $post){
    echo $post->post_title;
    echo $post->post_content;
    echo "<hr>";
}
```

W bazie danych:

```
select p.post_title, p.post_content, t.name
from `xd_posts` p, `xd_term_relationships` tr, `xd_terms` t, `xd_term_taxonomy` tt
where p.ID=tr.object_id AND tr.term_taxonomy_id=t.term_id AND tt.term_id = t.term_id and
p.post_type='post' AND p.post_status='publish' AND t.term_id=6
```

Tworzenie własnych funkcji

Własne funkcje możemy tworzyć w pliku functions.php. W tym przykładzie stworzymy funkcję, która zwraca ikonę bootstrap w zależności od kategorii. Pamiętaj o dołączeniu linku do ikon bootstrapa w sekcji <head>

```
function get_category_icon($catList){
    $icons = "";
    foreach($catList as $c){
        if($c == 5){
            $icons.='<i class="bi bi-back"></i>'. " ";
        }elseif($c == 6){
            $icons.='<i class="bi bi-alarm-fill"></i>'. " ";
        }else{
            $icons.='<i class="bi bi-emoji-heart-eyes"></i>'. " ";
        }
    }
    return $icons;
}
```

W miejscu na stronie, wewnątrz loopa umieść kod:

```
$id= get_the_ID();
$catList = wp_get_post_categories( $id);
echo get_category_icon($catList);
```

lub za pomocą jednej linii:

```
echo get_category_icon(wp_get_post_categories(get_the_ID()));
```