



8 SPOSOBÓW NA TUNING WORDPRESS'A - CZ.2

Posted on 16 kwietnia 2014 by Maciej Cybulski



Categories: [aktualności](#), [inne](#), [instalacja](#),
[motywy](#), [porady](#), [programowanie](#), [tuning](#),
[wtyczki](#)

W pierwszej części artykułu o [tuningu WordPress'a](#) skupiliśmy się na edycji pliku wp-config.php, by pod koniec rozpocząć przygodę z modyfikacjami motywu (functions.php)

Dziś dalszy ciąg ciekawych i prostych modyfikacji tego pliku.

5. Tuning WordPress'a - Przekierowanie na treść posta, gdy wynik wyszukiwania zwraca jedną pozycję.

Wyobraźmy sobie, że w wyszukiwarce na stronie wpisujemy jakąś frazę, w wyniku czego otrzymujemy informację, że wyszukiwanie zakończyło się znalezieniem jednego wpisu w tym temacie.

Standardowo efekt będzie taki, że wyświetli się nam w formacie blogowym (miniatura, tytuł i wypis) ten wpis. Słabo to wygląda... Co możemy z tym zrobić?

Ano... poniżej podam przykład kodu, który w przypadku zwracania przez wyszukiwarkę jednego wpisu, przeniesie nas od razu do tego wpisu.

Zatem znów dodajemy do pliku *functions.php* tym razem funkcję i akcję ją wywołującą.

```
add_action('template_redirect', 'redirect_single_post');
function redirect_single_post() {
    if (is_search()) {
        global $wp_query;
        if ($wp_query->post_count == 1 && $wp_query->max_num_pages
== 1) {
            wp_redirect( get_permalink( $wp_query->posts->ID ) );
            exit;
        }
    }
}
```

6. Tuning WordPress'a - Wyłączenie stron z wyników wyszukiwania

Typowy blog ma setki lub dziesiątki wpisów i tylko kilka stron. Te strony są ważnymi elementami bo zawierają informacje choćby "O nas" czy "Kontakt" i zawsze są wyświetlane w menu na eksponowanym miejscu. Dlatego nie ma też logicznego powodu by strony te były zwracane w wynikach wyszukiwania. Efektem ubocznym choć równie ważnym jest to że wyłączenie stron z wyników wyszukiwania pozwoli nam na proste ukrycie przed odwiedzającymi stron prywatnych np. strony z subskrypcją newslettera czy stroną na której można pobrać eBooka.

Zatem jak powinien wyglądać kod który nam to zapewni?

```
function remove_comment_fields($fields) {
    unset($fields);
    return $fields;
}
add_filter('comment_form_default_fields', 'remove_comment_fields');
```

7. Tuning WordPress'a - Usunięcie pola URL z formularza komentarzy.

Temat ograniczenia ilości spamowych komentarzy na stronie jest zawsze aktualny i wprost "niewyczerpany". Tutaj pokażę jeden z trików, który pozwoli zredukować ręczne dodawanie spamowych komentarzy na stronie.

Brak możliwości pozostawienia swojego URLa powoduje, że komentarze będą zostawiać tylko osoby którym faktycznie zależy na skomentowaniu merytorycznym treści a nie na linku do swojej strony. Ograniczymy w ten sposób komentarze o niskiej wartości. Zatem usuńmy z domyślnej instalacji WordPress'a z formularza komentarzy pole URL.

I znów edytujemy plik functions.php

```
add_filter( 'preprocess_comment', 'minimal_comment_length' );
function minimal_comment_length( $commentdata ) {
    $minimalCommentLength = 20;
    if ( strlen( trim( $commentdata ) ) < $minimalCommentLength ){
        wp_die( 'Wszystkie komentarze muszą być dłuższe niż '
        . $minimalCommentLength . ' znaków.' );
    }
    return $commentdata;
}
```

8. Tuning WordPress'a - Wymuszanie minimalnej długości komentarza.

Jak już jesteśmy przy komentarzach, to kolejnym sposobem, by odsiać komentarze niskiej wartości typu "Fajny wpis" jest wymuszenie by komentarz miał np. minimum 20 znaków. W sumie możemy przyjąć dwie strategie związane z komentarzami. Jeśli nie chcemy usuwać pola URL z formularza komentarzy, bo może akurat pozostawienie adresu autora komentarza będzie dla nas wartościowe, to możemy wymusić żeby autor jednak bardziej postarał się w komentarzu i napisał więcej niż kilka znaków. Tutaj już zostawiam Wam pomysł na zastosowanie rozwiązania.

W poniższym kodzie wartość zmiennej **`$minimalCommentLength`** definiuje nam minimalną ilość znaków, które powinien wpisać komentujący by jego komentarz mógł zostać przesłany do moderatora. Krótsze komentarze spowodują wyświetlenie komunikatu o błędzie.

```
function filter_search($query) {  
    if ($query->is_search) {  
        $query->set('post_type', 'post');  
    }  
    return $query;  
}  
add_filter('pre_get_posts', 'filter_search');
```

Mam nadzieję, że przydadzą Ci się zaprezentowane przeze mnie fragmenty kodu. Jeśli chcesz być na bieżąco z publikacjami na stronie zapisz się na newslettera, bo praktycznych informacji w tym temacie będzie przybywało.

Nie zapomnij też o udostępnieniu informacji na FB czy G+, co pozwoli też innym na zapoznanie się z możliwościami WordPressa i prostej samodzielnej jego modyfikacji. Mam nadzieję też, że zainspiruje Cię to do własnych testów i poszukiwań dotyczących dostosowywania WordPressa do własnych potrzeb. A jeśli masz pytania czy własne doświadczenia w tym temacie to podziel się nimi w komentarzach.

There are no comments yet.