



8 SPOSOBÓW NA TUNING WORDPRESS'A - CZ.1

Posted on 13 kwietnia 2014 by Maciej Cybulski

Kilka prostych snippetów do samodzielnego tuningu Twojej strony. Jak utrzymać porządek w koszu, podstawowe konfiguracje autozapisu...



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

Dzisiaj przyjrzymy się kilku prostym modyfikacjom plików motywu lub samego WordPress'a, które pozwolą nam na tuning funkcjonalności Twojej strony, jak też podnieść jej bezpieczeństwo, co w połączeniu z działaniami opisanymi w artykule "[Czy twardość ma znaczenie?](#)" pomoże znacznie polepszyć bezpieczeństwo Twojej strony.

Dla uproszczenia dalszych rozważań przyjmijmy że przez WordPress rozumiemy Twoją stronę, do której masz nieograniczony dostęp zarówno do kokpitu jak i do plików na FTP.

Najczęstszą formą zwiększenia funkcjonalności bądź modyfikacji WordPress'a jest instalacja wtyczek (pluginów). Jest to najprostsza metoda nie wymagająca znajomości kodowania w jakimkolwiek języku programowania.

Kolejna metoda już trochę bardziej wymagająca, to modyfikowanie plików motywu. I tutaj często możemy zastosować znów kilka rozwiązań. Możemy modyfikować bezpośrednio pliki motywu, a

możemy też by było bardziej bezpiecznie modyfikować pliki motywu potomnego.

Co to jest motyw potomny i jaka jest zależność pomiędzy motywem-rodzicem i motywem potomnym, oraz o plusach dodatnich i ujemnych tego rozwiązania będę pisał za jakiś czas. Do tego dochodzi jeszcze możliwość modyfikacji jednego pliku samego jądra WordPressa - wp-config.php. Ruszanie pozostałych plików jest zwiastunem potencjalnych problemów z bezpieczeństwem czy poprawnym funkcjonowaniem, teraz czy w przyszłości.

Dziś zatem nie będę mówić o wtyczkach, a skupię się na modyfikacjach wprowadzanych w plikach motywu i wp-config.php. Jądro WordPress'a zawiera podstawowy zestaw funkcjonalności wraz z interfejsem do ich poszerzania w plikach, które możesz Ty dodać. Stąd Twoja ingerencja w pliki dostarczane z instalacją WordPressa jest wręcz zbędna, wszystko można wykonać nie ruszając plików jądra.

Większość funkcji poszerzających funkcjonalność WordPress'a usytuowana jest w plikach motywu (wp-content/themes/nazwa_motywu/functions.php). Nie mówimy tutaj o modyfikacjach wyglądu. Za nie odpowiada w głównej mierze plik style.css, ale o jego modyfikacjach mowa będzie innym razem.

Fragmenty kodu o których dziś mowa, to małe fragmenty gotowego kodu, które mogą być dodawane bezpośrednio do plików motywu. Czasami istnieją już podobne funkcje w plikach motywu, więc trzeba będzie je tylko trochę zmodyfikować.

W artykule pokażę osiem użytecznych fragmentów kodu, które zwiększą możliwości i wygodę pracy z WordPress'em. Poprawność kodu sprawdzona jest na motywie Twenty Fourteen. jednak niektóre funkcje mogą nie działać poprawnie, jeśli motyw był już modyfikowany. Zatem przed wszelkimi modyfikacjami warto wykonać kopie plików źródłowych. No i ... modyfikacje wykonujesz na własną odpowiedzialność :)

Co nam będzie potrzebne:

1. Instalacja WordPress'a - czyli działająca strona
2. Edytor tekstu, tutaj polecam dla Windowsa Notepad++, a dla MacOS TextWrangler (oba sprawdzone i co najważniejsze bezpłatne)
3. Dostęp do Kokpitu oraz dostęp do plików przez FTP (jako klienta FTP polecam FileZillę, intuicyjny w obsłudze i bezpłatny program do przesyłania plików).

Zatem zaczynamy ;)

1. Opróżniamy kosz... czyli śmieci w WordPress'ie

WordPress ma wbudowaną "odporność" na uszkodzenia - przechowuje kopie wszystkich postów, stron i komentarzy. Można je usuwać, ale trzeba to zrobić w sposób jednoznaczny wrzucając do kosza. Kosz działa w taki sam sposób jak Kosz w systemie Windows. Zatem co jakiś czas powinien być opróżniany...

WordPress domyślnie opróżnia kosz co 30 dni. Można ten okres zmienić poprzez dodanie poniższego kodu do pliku wp-config.php. Znajdziemy go po zalogowaniu się na FTP, w katalogu głównym naszej strony.

Dla każdego modyfikowanego pliku warto zachować sobie jego kopię na wypadek gdyby pojawiły się jakieś problemy. Co do problemów na które warto zwracać uwagę to przede wszystkim puste znaki na końcu linii. Problem występuje szczególnie podczas edycji w systemie Windows. Efektem jego działania jest m.in. biały ekran podczas wchodzenia na stronę - strona nie wyświetla się.

No ale wracając do tematu.

Poniższy fragment kodu umieszczony w pliku wp-config.php spowoduje, że kosz będzie opróżniany automatycznie co 7 dni.

```
define ( 'EMPTY_TRASH_DAYS' , 7);
```

Zmiana wartości z 7 na 0 spowoduje całkowite wyłączenie kosza.

2. WordPress - wersje wpisu i autozapis.

Mechanizm edycji wpisów i stron w WordPressie został tak skonstruowany że przechowuje sporo wersji wpisu, oraz ma zdefiniowany czas autozapisu. Zatem jeśli nad artykułem długo pracujemy to wykona się nam wielokrotnie autozapis treści i każda poprzednia wersja (etap pracy) naszego artykułu będzie przechowywana w pamięci. Z czasem może się tego sporo uzbierać, co znacząco obciąża bazę danych i może spowalniać działanie samej strony.

By zapanować nad tym problemem wystarczy znów w pliku wp-config.php dodać linijkę:

```
define ( 'WP_POST_REVISIONS' , 3);
```

Spowoduje to, że tylko 3 ostatnie wersje wpisu/strony będą przechowywane w bazie. Starsze będą

automatycznie usuwane.

By całkowicie wyłączyć wersje wpisu wystarczy wpisać:

```
define ( 'WP_POST_REVISIONS', false );
```

Możemy też zdefiniować interwał w jakim ma następować autozapis.

```
define ( 'AUTOSAVE_INTERVAL', 160 );
```

Co definiuje nam autozapis co 160 sekund.

3. Przenosimy zawartość folderu WordPress wp-content

Jednym z niewielu folderów w co do którego w miarę bezpiecznie możemy sobie pozwolić na ingerencję jest wp-content. Do niego trafiają wszystkie pliki wtyczek, motywów i przesyłane pliki mediów. Z tego też powodu folder ten bywa często celem ataków hakerów. Szczególnie tych ataków które kończą się dodaniem złośliwego kodu do motywu czy wtyczki. Zmieniając domyślną lokalizację folderu możemy znacznie utrudnić takie działania.

Jeśli chcemy po prostu tylko przenieść folder wp-content do innej lokalizacji, to w pliku wp-config.php należy dodać następujący wiersz (*należy zauważyć że po wp-content nie ma ukośnika*)

```
defined ( 'WP_CONTENT_DIR', dirname(__FILE__) . '/newlocation/wp-content' );
```

Można też podać nową lokalizację jako adres URL

```
define( 'WP_CONTENT_URL', 'http://www.yourwebsite.com/newlocation/wp-content' );
```

Ale można też po prostu zmienić jego nazwę za pomocą poniższego kodu:

```
define ( 'WP_CONTENT_FOLDERNAME', 'newfoldername' );
```

Zmiana nazwy folderu wp-content z pewnością podniesie nam poziom bezpieczeństwa naszej strony, jednak może się to wiązać z pewnymi niedogodnościami. Wielu developerów WP zamiast używać zmiennych systemowych do opisanie położenia plików w folderze wp-content, używa jawnej ścieżki domyślnej, co powoduje że gdy zmienimy nazwę folderu, dana ścieżka domyślna nie zostanie

znaleziona. Zatem zanim wprowadzisz na stałe zmianę nazwy folderu wp-content sprawdź czy nie wpłynie to na działanie pluginów z których korzystasz. Jeśli przestają poprawnie działać, to zostaje Ci ręczne poprawianie w kodzie pluginu wszystkich odniesień do wp-content i to niestety za każdym razem jak zaktualizujesz plugin. No ale taki jest niestety koszt podniesienia bezpieczeństwa strony poprzez zastosowanie niestandardowych (nie domyślnych) rozwiązań. Oczywiście mając jakieś doświadczenie w programowaniu w PHP można sobie napisać skrypt który będzie odszukiwał i zamieniał wystąpienia wp-content na wskazaną przez nas nazwę. Ale to już pozostawiam jako temat do własnych ćwiczeń.

4. Przekierowanie linku archiwum autora do strony "o mnie"

Często spotykamy się na stronach że np. tuż poniżej tytułu artykułu znajduje się informacja-link o autorze artykułu. Taki link najczęściej prowadzi nas do listingu (archiwum) wpisów danego autora. Jednak gdy strona ma jednego tylko autora, taka funkcjonalność nie ma większego sensu, bo przecież dana osoba jest autorem każdego wpisu na stronie :) Możemy wtenczas to domyślne rozwiązanie zamienić na link do strony biograficznej autora.

W jaki sposób to osiągnąć? Tę funkcjonalność osiągniemy edytując plik functions.php w motywie. Jak go znajdziemy? Możemy znaleźć go na dwa sposoby. Pierwszy to odszukać w zapleczu WP.

Wygląd => Edytor .

W edytorze odszukujemy plik functions.php i edytujemy, po edycji zapisujemy. Oczywiście zanim będziemy go edytować, warto wykonać kopię jego zawartości.

Drugim sposobem znalezienia pliku jest odszukanie go przez dostęp FTP i to jest bardziej polecana metoda jeśli chodzi o bezpieczeństwo. Bo zawsze możemy wykonać kopię całego pliku w bezpiecznym miejscu i w razie problemów przywrócić jego pierwotną wersję z archiwum.

Na FTP znajdziemy plik pod następującą ścieżką:

wp-content/themes/nazwa_naszego_motywu_/functions.php

W pliku functions.php musimy dodać tzw. filtr author_link który wywoła funkcję my_author_link. Kod całości wygląda następująco.

```
add_filter( 'author_link', 'my_author_link' );
```

```
function my_author_link() {  
    return home_url( 'about' );  
}  
  
add_filter( 'author_link', 'my_author_link' );  
  
function my_author_link() {  
    return home_url( 'about' );  
}
```

Gdzie 'about' jest uproszczoną nazwą (slug) strony, która ma się wyświetlić zamiast archiwum wpisów. Ok to byłaby pierwsza część prostych sposobów na tuning naszej strony. Druga już niebawem, a w niej skupimy się na dalszej modyfikacji i edycji pliku functions.php. Powiem w niej m.in. o kilku logiczno-funkcjonalnych rozwiązaniach jak np. przekierowanie do konkretnego wpisu jeśli wynik wyszukiwania na stronie zwraca nam tylko jeden wpis, czy np. usuwanie pola URL z formularza komentarzy i jeszcze kilka innych ciekawych modyfikacji kodu, które podniosą bezpieczeństwo naszej strony i funkcjonalność.

Póki co zapraszam do pytań i komentarzy, no i oczywiście do udostępnienia wpisu na FB czy G+ za co będę wdzięczny i co z pewnością zmotywuje mnie do dalszego przybliżania tematu :)

There are no comments yet.