

PHP – Fusion

Konwersja skórek do PHP – Fusion

Z v. 6 do v.7

Tłumaczenie: lomckw
Korekta i skład: jantom.

Spis treści:

Spis treści:	1
Konwersja skórek do PHP-Fusion v7	3
Idea skórek v7	3
Zmiany w skórkach v7	3
Plik theme.php	3
Nowe możliwości	3
Plik styles.css	4
Aktualizacja skórki PHP-Fusion 6 do PHP-Fusion 7	4
Aktualizacja theme.php	4
Zaczynamy	4
Funkcje render_header() i render_footer()	5
Funkcja render_news()	7
Funkcja render_article()	8
Funkcje openside() and opensidex()	8
Pierwszy test	9
Aktualizacja skórki PHP-Fusion 6 do PHP-Fusion 7	10
Aktualizacja pliku styles.css	10
Ustawianie szerokości	10
Styl dla nowych elementów	11
Ostatnie poprawki	15
Przegląd końcowy	16

Konwersja skórek do PHP-Fusion v7

Streszczenie:

Artykuł w całości poświęcony systemowi skórek, jaki jest wykorzystywany w PHP-Fusion v7. Znajdziesz w nim wszystkie informacje dotyczące tworzenia, modyfikowania oraz konwersji starszych skórek do najnowszej wersji PHP-Fusion. W tej części dowiesz się o idei, jaka przyświecała autorom systemu szablonów oraz jakie zmiany zostały wprowadzone w pliku *theme.php*, w którym zapisany jest szkielet każdej skórki dla PHP-Fusion.

Idea skórek v7

Jeśli porównamy sposób w jaki działają skórki v6 i v7, stwierdzimy różnice w pewnych aspektach. Autorzy PHP-Fusion 7 postanowili dać więcej możliwości twórcom skórek, a jednocześnie zachować system tak podobny jak to tylko możliwe do wcześniejszych wersji. Dzięki temu ich użytkownicy mogą łatwo przejść na nową wersję. W rezultacie podstawowa idea systemu skórek opartych na funkcjach została zachowana, ale dodane zostały niektóre korzyści skórek opartych na szablonach.

Zmiany w skórkach v7

Plik *theme.php*

Najistotniejsze zmiany zaszły w pliku *theme.php*. Funkcje *render_header()* i *render_footer()* zostały połączone w jedną *render_page()*. Daje to projektantom skórek większą kontrolę nad trzema kolumnami zawartości strony. W PHP-Fusion 6 nie można było wpływać na kolejność i strukturę tabeli głównej i jej trzech komórek. W PHP-Fusion 7 taka możliwość jest. Zostały również połączone funkcje *openside()* i *opensidex()*, powód był dość oczywisty – obie funkcje miały większość wyników identycznych, a różniły się tylko w zakresie przycisku zwijania/rozwijania. Aby umożliwić użytkownikom bardziej zaawansowane zarządzanie banerami (bez użycia modów), zmienna prosta *\$header_content*, przechowująca baner strony, została zastąpiona funkcją *showbanners()*, której wynikiem są parametry banera pobrane z ustawień głównych. Udostępniony został również nowy panel administracyjny banerów. Istniejąca w wersji 6 funkcja *showsublinks()* zwraca teraz listę nieuporządkowaną zamiast zwykłych hiperłączy. W *render_news()* i *render_article()* zostały usunięte funkcje *openform()* i *closeform()*. Opcje administracyjne dla newsa bądź artykułu zwraca teraz funkcja *itemoptions()*. Wyrzucono także funkcję *tablebreak()*, a margines między panelami jest teraz ustawiany z poziomu css.

Nowe możliwości

Oprócz wspomnianych zmian, pojawił się szereg nowych możliwości, które można wykorzystać w skórkach v7. Uniwersalna klasa do łatwej integracji zmieniarzki stylu, np. wielkości czcionki czy zmiany kolorów może być użyta do dodatkowego doprawienia skórek. Przy pomocy funkcji *get_head_tags* można także dodać znaczniki do sekcji *head* strony, a więc dodawać własny JavaScript lub arkusze stylów bez modyfikacji głównych plików.

Projektanci skórek nie są już ograniczeni do używania domyślnej nazwy i formatów obrazów oraz ich lokalizacji, gdyż te ustawienia mogą być zmienione za pomocą odpowiednich funkcji.

Plik styles.css

Zarówno zmienione, jak i nowe cechy PHP-Fusion 7 wymagają kilku nowych klas, np.: identyfikator `#navigation` jest używany w nowym panelu nawigacyjnym (w wersji css), klasa `.news-category` jest używana w stosunku do obrazu kategorii newsów, klasa `.pagenav` jest używany do stronicowania (paginacji). Najczęściej konieczna jest ingerencja w więcej niż jeden z w/w elementów. Np. `#navigation li` dotyczy elementów listy (poszczególnych hiperłączy) w panelu nawigacyjnym, a `#navigation` pozwala określać jej ogólny wygląd.

Aktualizacja skórki PHP-Fusion 6 do PHP-Fusion 7

Aby skórka utworzone dla PHP-Fusion 6 stała się zgodna z PHP-Fusion 7, należy dokonać pewnych zmian w odpowiednich plikach. Przedstawimy to na przykładzie skórki Prime autorstwa Digitanium. Jednakże każdy, kto ma trochę doświadczenia ze skórkami PHP-Fusion 6 może po prostu porównać obie wersje skórki (v6 i v7) i stwierdzić samodzielnie jakie zmiany są konieczne. Poniższy opis jest skierowany do osób których doświadczenie w zakresie skórek PHP-Fusion jest niewielkie lub żadne, a mimo to są zdecydowane we własnym zakresie zaktualizować swoją ulubioną skórę.

Opis należy czytać ze zrozumieniem. W niektórych przypadkach należy coś zastąpić, w innych coś dodać, a jeszcze innym razem coś zmienić (zgodnie z podanym przykładem). Należy zrozumieć opisane zmiany i wtedy odpowiednio zmieniać swój kod. Skórki PHP-Fusion są zbyt różnorodne, aby możliwe było opracowanie precyzyjnego opisu “krok po kroku” dla każdego możliwego przypadku. Podane przykłady ułatwiają zrozumienie wymaganych poprawek i modyfikacji, ale nie należy nimi bezmyślnie zastępować kodu w aktualizowanej skórze.

Aktualizacja theme.php

Zaczynamy

Po pierwsze, dodajmy poniższy wiersz kodu na początku pliku theme.php:

```
if (!defined("IN_FUSION")) { die("Access Denied"); }
```

Poprawi to bezpieczeństwo przez zablokowanie możliwości indywidualnego uruchomienia pliku `theme.php`. Jeśli podobny kod już istnieje, należy go zastąpić powyższym, który jest bezpieczniejszy. W dalszej kolejności należy usunąć niepotrzebne zmienne z początku theme.php, gdyż nie są one używane w v7 (możliwe jest ich stosowanie, ale sugerujemy korzystanie bezpośrednio z możliwości CSS, zgodnie z ideą nowej wersji systemu). W skórze Prime usuwamy zmienne:

```
$body_text = "#444444";
```

```
$body_bg = "#ffffff";
$theme_width = "100%";
$theme_width_l = "170";
$theme_width_r = "170";
```

Jedyna definicja, jaką należy wykonać w PHP-Fusion 7 to punktator. W naszym przykładzie nie użyjemy obrazu, ale punktora html (kulki). Dodajemy zatem wiersz:

```
define("THEME_BULLET", "<span class='bullet'>&middot;</span>");
```

Aby użyć obrazka, należy wprowadzić w odpowiednim miejscu znacznik img (zamiast ·). Klasa bullet ułatwi w dalszej kolejności ustawienie wyglądu elementów nawigacyjnych. Teraz należy upewnić się, że kod zawiera poniższy wiersz (lub jego odpowiednik):

```
require_once INCLUDES."theme_functions_include.php";
```

Dołączmy to szereg funkcji pomocniczych, niezbędnych w dalszych pracach. Niektóre nowsze skórki dla PHP-Fusion 6 zawierają go, ale skórka Prime niestety nie, więc trzeba go dodać. Początek pliku theme.php dla skórki Prime powinien teraz wyglądać tak:

```
<?php
if (!defined("IN_FUSION")) { die("Access Denied"); }
// theme settings
define("THEME_BULLET", "<span class='bullet'>&middot;</span>");
require_once INCLUDES."theme_functions_include.php";
```

Możemy także usunąć funkcję tablebreak(), ponieważ nie jest ona używana w PHP-Fusion 7 (odległość między panelami jest określana za pomocą CSS).

Funkcje render_header() i render_footer()

Jak wspomniano wcześniej, *render_header()* i *render_footer()* zostały połączone w jedną funkcję *render_page()*. Prostym sposobem jest zmiana nazwy funkcji *render_header()* na *render_page()* oraz zmiana jej argumentów. *render_page()* wymaga tylko argumentu *\$license* funkcji *render_footer()*, gdyż argument *\$header_content* funkcji *render_header()* jest odziedziczony i w związku z tym zbędny na liście parametrów funkcji. Zmiany w skrócie wyglądają tak:

Przed:

```
function render_header($header_content) {
```

Po:

```
function render_page($license=false) {
```

Teraz należy skopiować treść funkcji *render_footer()* i wkleić na końcu nowej funkcji *render_page()* (oprócz deklaracji zmiennych globalnych, które już występują w *render_page()*). W przypadku skórki Prime wygląda to następująco:

Przed:

```

function render_page($license=false) {
global $settings;

[...]

echo "
    <table cellpadding='0' cellspacing='0' width='100%'>
        <tr>\n";

}
function render_footer($license=false) {
global $theme_width,$settings,$locale;

echo "
        </tr>
    </table>\n";
    tablebreak();
    echo "
        <table cellpadding='0' cellspacing='0' width='100%'>
            <tr>
                <td
class='footer'><br>".stripslashes($settings['footer'])."</td>
                </tr>
                <tr>
                    <td align='center' class='footer'>\n";
                        if ($license == false) {
                            echo "Powered by <a href='http://www.php-
fusion.co.uk' target='_blank'>PHP-Fusion</a> v".$settings['version']."
&copy; 2003-2005<br><br>\n";
                        }
                        echo $settings['counter']." ".($settings['counter']
== 1 ? $locale['140']."<br><br>\n" : $locale['141']."<br><br>\n")."
                    </td>
                </tr>
            </table>
        </td>
    </tr>
</table>\n";

}

```

Po:

```

function render_page($license=false) {
global $settings;

[...]

echo "
    <table cellpadding='0' cellspacing='0' width='100%'>
        <tr>\n";

}

```

```

function render_footer($license=false) {

global $theme_width,$settings,$locale;

echo "
        </tr>
    </table>\n";
        tablebreak();
        echo "
        <table cellpadding='0' cellspacing='0' width='100%'>
            <tr>
                <td
class='footer'><br>".stripslashes($settings['footer'])."</td>
            </tr>
            <tr>
                <td align='center' class='footer'>\n";
                    if ($license == false) {
                        echo "Powered by <a href='http://www.php-
fusion.co.uk' target='_blank'>PHP-Fusion</a> v".$settings['version']."
&copy; 2003-2005<br><br>\n";
                    }
                    echo $settings['counter']." " .($settings['counter']
== 1 ? $locale['140']."<br><br>\n" : $locale['141']."<br><br>\n")."
                </td>
            </tr>
        </table>
    </td>
</tr>
</table>\n";

}

```

Funkcja render_news()

Funkcja render_news() jest łatwa do aktualizacji. W nowszych skórkach PHP-Fusion 6 występuje kod:

```

".openform("N",$info['news_id']).newsposter($info,"&middot;").newsopts($info,"&middot;").closeform("N",$info['news_id'])."

```

Wystarczy tylko usunąć *openform()* i zamiast *closeform()* wpisać *itemoptions()* :

```

".newsposter($info,"&middot;").newsopts($info,"&middot;").itemoptions("N",$info['news_id'])."

```

W skórkach, które nie wykorzystują nowych funkcji (np. Prime), należy zastąpić cały kod nową wersją, jak poniżej:

Przed:

```

<td align='right' class='news-footer'>
    <img src='".THEME."images/bullet.gif' alt='&middot;';>
    ".$locale['040'].<a
href='profile.php?lookup=".$info['user_id']."'>".$info['user_name'].</a>
    ".$locale['041'].showdate("longdate", $info['news_date'])."
    ".($info['news_ext'] == "y" ? "<a
href='news.php?readmore=".$info['news_id']."'>".$locale['042'].</a> " :
    "");

```

```

        if ($info['news_allow_comments']) echo "<a
href='news.php?readmore="'. $info['news_id']. "'>". $info['news_comments']. $loc
ale['043']. "</a> \n";
        echo $info['news_reads']. $locale['044']. "
        <a href='print.php?type=N&item_id="'. $info['news_id']. "'><img
src='".THEME."/images/printer.gif' alt='". $locale['045']. "' style='vertical-
align:middle;border:0px;'></a>
    </td>

```

Po:

```

<td align='right' class='news-footer'>".
        newsposter($info, THEME_BULLET).newsopts($info,
THEME_BULLET).itemoptions("N", $info['news_id'])."
    </td>

```

Nie zapomnij o usunięciu wywołań funkcji *tablebreak()* jeśli występują.

Funkcja render_article()

Funkcję aktualizujemy na identycznych zasadach jak *render_news()*. Wystarczy zastąpić "news" przez "article" oraz "N" przez "A":

```

<td align='right' class='news-footer'>".
        articleposter($info, THEME_BULLET).articleopts($info,
THEME_BULLET).itemoptions("A", $info['article_id'])."
    </td>

```

Tu również należy usunąć wywołania funkcji *tablebreak()*.

Funkcje openside() and opensidex()

Jak już wiadomo, funkcje te również zostały połączone. Akurat w tym przypadku jest to trochę skomplikowane, gdyż autor każdej skórki robił je po swojemu. Dlatego niezłym sposobem jest porównanie funkcji *opensidex()/openside()* w już przekonwertowanych skórkach. Oczywiście opiszemy teraz kolejne kroki, które pozwolą to zrealizować. Na początek należy usunąć funkcje *openside()* i *closeside()*, a funkcji *closesidex()* zmienić nazwę na *closeside()* i zmienić nagłówek funkcji *opensidex()* na nową *openside()*:

Przed:

```
function opensidex($title, $open="on") {
```

Po:

```
function openside($title, $collapse = false, $state = "on") {
```

Następnie wykonaj poniższe kroki: Usuń poniższy wiersz kodu:

```
if($open=="on"){ $box_img="off"; }else{ $box_img="on"; }
```

Dodaj poniższe wiersze na samym początku nowej funkcji *openside()*:

```
global $panel_collapse; $panel_collapse = $collapse;
$boxname = str_replace(" ", "", $title);
```

Dodaj poniższy wiersz na samym początku nowej funkcji *closeside()*:

```
global $panel_collapse;
```


Zastąp starą wersję przycisku zwijania nową, opartą na funkcji (niektóre skórki mogą już korzystać z nowej wersji) i umieść ją w instrukcji if, co spowoduje, że przycisk będzie widoczny tylko w panelach, które można zwiijać:

Przed:

```
<td class='scapmain' align='right'><img  
src='".THEME."/images/panel_{$box_img}.gif' name='b_{$title}' alt=''  
onclick="\`javascript:flipBox('{$title}')\"`></td>
```

Po:

```
".($collapse ? "<td class='scapmain' align='right'>".panelbutton($state,  
$boxname)."</td>" : "")."
```

Zastąp także poprzednie znaczniki blokowe (div) nową funkcją odpowiedzialną za ukrywanie zawartości:

```
<td colspan='2' class='side-body'>  
  <div id='box_{$title}'; if($open=="off"){ echo  
"style='display:none'"; } echo ">\n";  
}
```

```
function closeside() {
```

```
global $panel_collapse;
```

```
echo "
```

```
  </div>
```

```
</td>
```

```
</tr>
```

```
</table>\n";
```

Po:

```
<td colspan='2' class='side-body'>";
```

```
  if ($collapse == true) { echo panelstate($state, $boxname); }
```

```
}
```

```
function closeside() {
```

```
global $panel_collapse;
```

```
if ($panel_collapse == true) { echo "</div>\n"; }
```

```
echo "
```

```
  </td>
```

```
</tr>
```

```
</table>\n";
```

Pierwszy test

Możemy już spróbować włączyć naszą aktualizowaną skórke. Strona powinna się wyświetlić, chociaż na pewno nie wszystkie element wyglądają tak, jak powinny. Jeśli nie wyświetla się zupełnie lub zwraca błędy, należy jeszcze raz sprawdzić wszystkie poprzednie kroki tego przewodnika i ogólną poprawność kodu. Jeśli nie jesteś w stanie zlokalizować błędu, szukaj pomocy na stronach PHP-Fusion Themes (<http://themes.php-fusion.co.uk>) lub na stronach polskiego supportu (<http://www.php-fusion.pl>).

Aktualizacja skórki PHP-Fusion 6 do PHP-Fusion 7

W poprzedniej części artykułu można było zapoznać się ze zmianami, jakie zostały wprowadzone do pliku *theme.php*. Ta będzie w całości poświęcona plikowi opisującemu wygląd skórki - *styles.css*. Zapraszam wszystkich do dalszej lektury.

Aktualizacja pliku *styles.css*

Po obejrzeniu swojej strony po zmianach dokonanych w pliku *themes.php*, prawdopodobnie stwierdzisz, że nie wszystko wygląda tak, jak to było w wersji dla PHP-Fusion 6. Nie jest to nic niezwykłego, gdyż nie zaktualizowany jest jeszcze plik *styles.css*, który w PHP-Fusion 7 steruje wyglądem skórki.

Ustawianie szerokości

Oglądając stronę po dotychczasowych zmianach, na pewno stwierdzisz błędy w szerokościach elementów. Przyczyna jest prosta – dla łatwiejszego sterowania wyglądem strony te ustawienia nie są już zawarte w pliku *theme.php*, a w pliku *style.css*. Aby ustawić całkowitą szerokość skórki musimy mieć możliwość odwołania się do największej (zewnętrznej) tabeli/bloku skórki z poziomu *css*. W niektórych skórkach ta tabela już posiada klasę lub identyfikator, które możemy wykorzystać. W innych, jak np. Prime, trzeba ją dodać. Wracamy do pliku *theme.php* i upewniamy się, że zewnętrzna tabela posiada klasę/identyfikator. Jeśli nie – dodajemy ją wg poniższego przykładu:

Przed:

```
function render_page($license=false) {  
  
    global $settings;  
  
    echo "<table align='center' cellspacing='0' cellpadding='0'>
```

Po:

```
function render_page($license=false) {  
  
    global $settings;  
  
    echo "<table align='center' cellspacing='0' cellpadding='0' id='main'>  
Skoro możemy już identyfikować główną tabelę, ustawmy jej szerokość z poziomu css. Wystarczy w pliku style.css w dowolnym miejscu dodać :
```

```
#main {width: 100%;}
```

Oczywiście należy użyć identyfikatora takiego jak powyżej, w pliku *theme.php* (u nas "main") oraz pożądaną szerokości. Jeśli zastosujemy klasę, trzeba użyć nieco innego kodu:

```
.main {width: 100%;}
```

Skórka powinna już wyglądać lepiej, ale kolumny boczne mogą nadal być zbyt szerokie. Aktualizując plik *theme.php* użyliśmy już odpowiednich klas, a więc możemy teraz określić ich szerokości. Większość plików *styles.css* zawiera definicje właściwości dla klas *.side-border-left* i *.side-border-right*. Wystarczy je odnaleźć i dodać właściwość *width* (lub całą klasę, jeśli nie występuje), jak pokazano w poniższym przykładzie. W skórcie Prime obydwie klasy już są na miejscu, więc wystarczyło dodać definicję szerokości:

Przed:

```
.side-border-left { padding:0px 10px 0px 10px; }
.side-border-right { padding:0px 10px 0px 10px; }
```

Po:

```
.side-border-left {
    padding:0px 10px 0px 10px;
    width: 170px;
}

.side-border-right {
    padding:0px 10px 0px 10px;
    width: 170px;
}
```

Szerokości powinny już być poprawne, ale ciągle jeszcze mamy do wykonania kilka rzeczy.

Styl dla nowych elementów

Określimy teraz wygląd elementów, które działają inaczej niż w PHP-Fusion 6 lub po prostu są całkiem nowe.

Nawigacja pozioma

Jedną ze wspomnianych nowych dodatków są listy używane w nawigacji poziomej i w panelach nawigacyjnych opartych na css. Poniższy kod umożliwia poprawne działanie nawigacji poziomej. Wystarczy dodać go do *styles.css*.

```
.white-header ul{
    margin: 0;
    padding: 0;
}
.white-header ul li{ display: inline; }
.white-header ul li.first-link .bullet{ display: none; }
```

Jeśli w nawigacji poziomej używasz innej klasy niż *white-header* – użyj jej w tym miejscu. Co robi ten fragment? Po pierwsze usuwa domyślne marginesy i padding dla całej listy (*ul*), dzięki czemu poszczególne pozycje nie mają wcięć a cała lista nie zajmuje za dużo miejsca. Po drugie ustawia elementy listy (*li*) jeden za drugim (a nie pod). I wreszcie, po trzecie, usuwa punktory dla pierwszego elementu listy. Wszystko to sprawia, że nawigacja pozioma powinna wyglądać tak, jak w większości skórek PHP-Fusion 6. Pozycja i rozmieszczenie hiperłączy powinno już być prawidłowe, ale możliwe, że ich kolory są niewłaściwe. Przyczyną jest fakt, że w PHP-Fusion 6 klasa dla nawigacji poziomej odnosiła się do

hiperłączy, a w PHP-Fusion 7 dotyczy elementów listy. W skórcie Prime odpowiednia klasa to `.white`, a poprawki wyglądają następująco:

Przed:

```
a.white { color:#fff; text-decoration:none; }
a:hover.white { color:#fff; text-decoration:underline; }
```

Po:

```
li.white a, a.white{ color:#fff; text-decoration:none; }
li.white a:hover, a:hover.white { color:#fff; text-decoration:underline; }
```

Jeśli twoje hiperłącza korzystają z innej klasy (zwróć uwagę na argumenty funkcji `showsublinks()` , której wywołanie znajdziesz wewnątrz funkcji `render_page()` - zastosuj zmiany do tej klasy. W naszym przykładzie pozostawimy klasę `.white` , gdyż niektóre inne hiperłącza (np. w stopce strony) również z niej korzystają.

Panel nawigacyjny w wersji css

Dostosujemy teraz wygląd listy wyświetlanej w panelu nawigacyjnym. Po pierwsze dodamy poniższy kod, aby móc sterować wyglądem listy:

```
#navigation ul {
    margin: 0;
    padding: 0;
    list-style-type: none;
}
```

Następnie zajmiemy się etykietami kategorii w menu. Kolory jakich należy użyć, zależą od klasy `side-label` i dla każdej skórki mogą być inne:

```
#navigation h2 {
    font-size: 12px;
    color: #555;
    background-color: #fff;
    margin: 0;
    padding: 2px;
}
```

Określimy teraz wygląd hiperłączy, używając kolorów, które pasują do skórki:

```
#navigation a {
    display: block;
    color: #777;
    padding: 2px;
    text-decoration : none;
}
#navigation a:hover {
    color: #f6f6f6;
    background-color: #ccc;
}
```

I jeszcze poniższy fragment - konieczny, aby zapewnić poprawne wyświetlanie pod IE6:

```
* html #navigation ul li{
    height: 1%;
}
```

```
}
```

Wszystkie powyższe definicje należy dodać w pliku styles.css.

Podział na strony (paginacja)

W PHP-Fusion 7 nawigacja “między stronami”, czyli lista stron, jaką często spotykamy na witrynach PHP-Fusion, nie korzysta już z tego samego stylu co tabele używane w forum. Do tego celu w PHP-Fusion 7 służy dedykowana klasa .pagenav. Dodajmy zatem odpowiedni fragment do pliku styles.css. Kolory należy dobrać zgodnie z kolorystyką skórki (w razie potrzeby można zerknąć na klasy tbl1 i tbl2):

```
.pagenav span {
    color: #777;
    background-color: #f0f0f0;
    border: 1px solid #e1e1e1;
    padding: 2px 4px;
    margin: 2px;
}

.pagenav a {
    color: #000;
    background-color: #fff;
    border: 1px solid #e1e1e1;
    padding: 2px 4px;
    margin: 2px;
    text-decoration: none;
}

.pagenav a:hover {
    color: #ccc;
    background-color: #444;
    text-decoration: none;
}
```

Pierwszy fragment ustala parametry dla strony bieżącej, drugi dla pozostałych, a trzeci wygląd hiperłączy w stanie wskazania myszą.

Centrowanie elementów

W PHP-Fusion 6 wiele elementów strony było centrowanych z użyciem atrybutu align (np. <table align='center' [...]>). W PHP-Fusion 7 takie przypadki są nieliczne, a w większości centrowanie jest realizowane z użyciem nowej klasy css, przewidzianej do tego celu. Jeśli do pliku styles.css dodamy poniższy wiersz, zapewnimy poprawne działanie obu możliwości, i to w zarówno w skórkach opartych na blokach (div) jak i tabelach (table):

```
.center, *[align="center"] {margin: 0 auto;}
```

Klasa admin_message

W PHP-Fusion 7 jest ona używana do określenia wyglądu ważnych komunikatów. Dla skórki Prime wygląda następująco:

```
.admin-message {
    border: 1px solid #666;
    border-left: 3px solid #999;
    padding: 8px;
    margin-bottom: 8px;
}
```

Oczywiście powyższy styl należy dopasować kolorystycznie do swojej skórki.

Kody BBCode

Wygląd przycisków BBCode można w PHP-Fusion 7 dostosowywać indywidualnie. W Prime dodamy tylko cienką ramkę:

```
.bbcode { border: 1px solid #ccc; }
```

Przywrócenie efektu funkcji tablebreak()

Jeśli podejrzmy stronę na dotychczasowym etapie naszych prac, możemy stwierdzić, że jej wygląd jest już bliski oczekiwanemu (przynajmniej powinien być). Do poprawienia ciągle są marginesy między panelami, które zostały usunięte wraz z funkcją tablebreak(). Aby to osiągnąć musimy zastosować własność css do elementu nadrzędnego (rodzica) paneli (lub elementu, po którym panel następuje, jeśli brak elementu nadrzędnego). Jeśli brak odpowiedniego elementu, możemy zawsze użyć własnej klasy (np. wiele skórek v6 stosuje klasę border dla tabeli będącej rodzicem paneli), oczywiście w tym celu należy ją zdefiniować. W skórcie Prime istnieje klasa border, ale jest ona używana tylko w stosunku do środkowych paneli skórki. Dodamy zatem nową klasę o nazwie tablebreak i zastosujemy ją wszędzie, gdzie potrzebny jest margines generowany poprzednio przy użyciu funkcji tablebreak(). Oto pięć miejsc, gdzie użyjemy wspomnianej klasy w skórcie Prime:

Nawigacja pozioma/Data:

```
<table cellpadding='0' cellspacing='0' width='100%' class='tablebreak'>
  <tr>
    <td class='white-header'>".showsublinks(" ".THEME_BULLET." ",
"white")."</td>
    <td align='right' class='white-header'>".showsubdate()."</td>
  </tr>
</table>
```

Newsy

```
<table cellpadding='0' cellspacing='0' width='100%' class='border
tablebreak'>
  <tr>
    <td class='main-body'>
      $news
    </td>
  </tr>
```

Artykuły

```
<table cellpadding='0' cellspacing='0' width='100%' class='border
tablebreak'>
  <tr>
```

```

        <td class='main-body'>
            " .($info['article_breaks'] == "y" ? nl2br($article) :
$article)."
        </td>
    </tr>

```

Panele boczne

```

function openside($title, $collapse = false, $state = "on") {

global $panel_collapse; $panel_collapse = $collapse;
$boxname = str_replace(" ", "", $title);

echo "<table cellpadding='0' cellspacing='0' width='100%'
class='tablebreak'>
    <tr>
        <td colspan='2' class='side-body'><img
src='\".THEME.\"images/blank.gif' alt='\"' height='2' width='1'></td>
    </tr>

```

Panele środkowe:

```

function opentable($title) {

echo "<table cellpadding='0' cellspacing='0' width='100%' style='margin-
bottom:2px;'>
    <tr>
        <td class='capmain'>$title</td>
    </tr>
</table>
<table width='100%' cellpadding='0' cellspacing='0' class='border
tablebreak'>

```

Skoro w powyższych tabelach odwołujemy się do naszej nowej klasy, możemy dodać poniższy wiersz do pliku styles.css co ustawi odpowiednią wielkość marginesów:

```

.tablebreak { margin-bottom: 8px; }

```

Ostatnie poprawki

Na zakończenie powinniśmy upewnić się, że nasza skórka ma czysty i poprawny kod. Kolejno realizujemy:

1. sprawdzenie i usunięcie zbędnych wywołań funkcji tablebreak(),
2. poprawienie znacznika otwierającego php <? do wersji pełnej <?php, aby uniknąć problemów z niektórymi konfiguracjami php5,
3. zamknięcie znaczników
 (
),
4. oraz znaczników () i innych znaczników nieparzystych (które nie używają znacznika zamykającego w postaci </znacznik>),
5. uzupełnienie atrybutu alt dla obrazów,
6. dodanie wpisu img {border: 0;} w pliku styles.css, co usunie ramkę wokół hiperłączy obrazkowych,
7. usunięcie wpisów filter z pliku styles.css (wiele starszych skórek używa tej techniki do uzyskania efektu hover dla obrazów w galerii).

Przegląd końcowy

Po zaktualizowaniu pliku styles.css możemy przyjrzeć się efektom naszej pracy. Tym razem większość elementów powinno wyglądać i zachowywać się podobnie jak w PHP-Fusion 6. Jeśli tak nie jest, powinieneś poszukać pomocy na forum http://themes.php-fusion.co.uk/forum/viewforum.php?forum_id=14 .