

# PHP-Fusion 7 Theming

## From PHP-Fusion Wiki

### Contents

- 1 V7 Theme Concepts
- 2 V7 Theme Changes
  - 2.1 Theme.php
    - 2.1.1 New possibilities
  - 2.2 Styles.css
- 3 Updating PHP-Fusion 6 Themes to PHP-Fusion 7
  - 3.1 Updating theme.php
    - 3.1.1 Getting started
    - 3.1.2 `render_header()` and `render_footer()`
    - 3.1.3 `render_news()`
    - 3.1.4 `render_article()`
    - 3.1.5 `openseide()` and `openseidex()`
    - 3.1.6 First Test
  - 3.2 Updating Styles.css
    - 3.2.1 Setting widths
    - 3.2.2 Styling new elements
      - 3.2.2.1 Horizontal navigation
      - 3.2.2.2 Navigation Panel (CSS Version)
      - 3.2.2.3 Page Navigation (Pagination)
      - 3.2.2.4 Centered Elements
      - 3.2.2.5 Admin Message
      - 3.2.2.6 BBcodes
      - 3.2.2.7 Resembling the `tablebreak()` effect
  - 3.3 Cleanups and Fixes
  - 3.4 Final Preview

## V7 Theme Concepts

The way PHP-Fusion themes work changed in certain aspects if v6 and v7 themes are compared. In PHP-Fusion 7, the developers tried to give more power to theme authors and at the same time keep the system as similar as possible to the theme systems of PHP-Fusion 6 and earlier versions so that users can easily adopt the new system. Consequently the basic concept of a function-based theme system was kept, but some advantages of template-based systems were added.

## V7 Theme Changes

### Theme.php

The most evident changes were made in `theme.php`. `render_header()` and `render_footer()` were merged to `render_page()`. This gives theme designers more control of the three content columns. In PHP-Fusion 6 you could not influence the order and structure of the main layout table and its three cells, in PHP-Fusion 7, you

can do this.

Additionally, *openside()* and *opensidex()* were merged. The reason for this merge are obvious as these functions share most of their output and only differ in the expand/-collapse button.

To enable users to use a more sophisticated banner management system without mods, the simple `$header_content` var that had contained the site banner so far was replaced by the *showbanners()* function that outputs the banner settings made in main settings and the new banner administration panel. The *showsublinks()* function which already existed in PHP-Fusion 6 now returns a unordered list instead of the plain links.

In *render\_news()* and *render\_article()* the *openform()* and *closeform()* function calls were removed.

Instead, *itemoptions()* now outputs the administrative options for this news item or article.

*tablebreak()* was dropped and CSS is now used to set the margins between panels.

## New possibilities

Apart from the changes, a set of new possibilities can be used in PHP-Fusion 7 themes. A versatile class to easily integrate style switchers such as font-size or color-switchers can be used to spice up themes. Read more on the switcher class here.

You can also define head tags from within the theme now and thus add your own javascripts or style sheets without changing the core files. Read more about adding custom head tags.

Themes are not bound to use the default names and formats for images anymore as the locations of these can be changed using simple functions. Read more about setting custom image locations.

## Styles.css

The changed or newly added features of PHP-Fusion 7 need some new classes to be styled.

*#navigation* is used for the container of the new CSS navigation panel.

*.news-category* is used for the news category image in news.

*.pagenav* is used for the pagination or page navigation.

Usually it's necessary to style more than one element using these ids or classes. For example, *#navigation li* is used to style the list items of the CSS navigation panel and *#navigation a* is used to style the individual links.

## Updating PHP-Fusion 6 Themes to PHP-Fusion 7

Themes created for PHP-Fusion 6 are not compatible with PHP-Fusion 7 without some changes. I will explain the procedure using the Prime theme by Digitanium as an example.

However, anyone who has some experience with PHP-Fusion 6 themes will probably find it easier to just compare a older theme to one made for PHP-Fusion 7 to find out what changed. The following guide is mostly targeted at users who have little or no experience with PHP-Fusion themes, but still want to update their favourite theme themselves.

Feel free to request further detail on specific sections in the discussion page of this article.

**Important Note:** Read the guide carefully. Sometimes you have to replace something, sometimes you have to add something and sometimes you have to change some code yourself, using my code as an example. Try to understand the changes I describe and then change your code accordingly. PHP-Fusion themes are too different to provide a step by step guide for every change. Use the examples to understand what you have to change, don't use them as a replacement for your code.

## Updating theme.php

### Getting started

First, we add this line at the top of theme.php:

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

This might increase security by preventing that the theme.php script is executed individually. If a similar line already exist, replace it with this version which is more secure.

Next, we delete all the old vars at the top of theme.php which are not used in v7 anymore (you could use them, but I'd suggest you to style everything via CSS directly). In Prime, I remove these vars:

```
$body_text = "#444444";
$body_bg = "#ffffff";
$theme_width = "100%";
$theme_width_l = "170";
$theme_width_r = "170";
```

The only setting we have to set in PHP-Fusion 7 is the bullet image. I decide to not use an image but plain text for the bullets and thus add this line:

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

If you want to use an image, use an image tag here. The bullet class will help us later when we style the navigation.

Now make sure that this or a comparable line exists:

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

This is the case in some newer PHP-Fusion 6 themes, but not in Prime. This include provides us with helper functions for the theme that make our work easier.

The top of theme.php looks like this for Prime now:

```
<?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";
```

We can also remove the tablebreak() function as it is not used in PHP-Fusion 7 anymore (css is used now to define margins between panels).

## render\_header() and render\_footer()

Let's move on to the functions. As you already know, *render\_header()* and *render\_footer()* have been merged to *render\_page()* and so we do that as well now. The straight forward way to do this is to **rename** *render\_header()* to *render\_page()*. We also change the parameters. *render\_page()* only needs the \$license parameter of the former *render\_footer()* function as the *\$header\_content* parameter of *render\_header()* has been succeeded by a function and thus we do not need this one anymore. The changes in a nutshell:

### Before:

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

### After:

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

Now we copy the contents of `render_footer()` to the end of our new `render_page()` function (except the global statement which we already got once in `render_page()` ).

In Prime this looks like this:

### Before:

```
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'>P
                }
                echo $settings['counter']." ".($settings['counter'] == 1 ? $locale['140'])."<br><
            </td>
        </tr>
    </table>
    </td>
</tr>
</table>\n";
}
```

### After:

```
function render_page($license=false) {
global $settings;
[...]

echo "
<table cellpadding='0' cellspacing='0' width='100%'>
  <tr>\n";
//Next step goes here
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'>P
        }
        echo $settings['counter']. " " . ($settings['counter'] == 1 ? $locale['140']). "<br><
      </td>
    </tr>
  </table>
</td>
</tr>
</table>\n";
}
```

You can **delete** the remains of `render_footer()` now. You might have noticed the `//Next step goes here` comment in the last code snippet and so we will go on there. In PHP-Fusion 6 the columns html code was defined outside theme.php in side-border-right and -left.php. In PHP\_Fusion 7 we can define them ourselves and we do that at the marked spot.

All PHP-Fusion 6 themes did use table cells here as they can't choose whether to use tables or divs, but in PHP-Fusion 7 you could also use divs here. We will start with the good ol' table way though. These are the table cells we need:

```
<td class='side-border-left'></td>
<td class='main-bg'></td>
<td class='side-border-right'></td>
```

We fill them with the constants that hold the panels. Their names should be self-explanatory:

```
<td class='side-border-left'>".LEFT."</td>
<td class='main-bg'>".U_CENTER.CONTENT.L_CENTER."</td>
<td class='side-border-right'></td>".RIGHT."
```

We need some more code to make it work right and at the end it looks like this in Prime:

```
if (LEFT) { echo "<td class='side-border-left' valign='top'>".LEFT."</td>"; }
echo "<td class='main-bg' valign='top'>".U_CENTER.CONTENT.L_CENTER."</td>";
if (RIGHT) { echo "<td class='side-border-right' valign='top'>".RIGHT."</td>"; }
```

What does it do? The if statements will make sure that the side columns are only shown if there are panels in them. The `valign='top'` is just some html to make the table contents start at the top and not the middle of the table as of default.

Next up, we replace `$header_content` by `showbanners()`, the PHP-Fusion 7 way of handling the site banners. In Prime I do it like this:

**Before:**

```
<td>$header_content</td>
```

**After:**

```
<td>".showbanners()."</td>
```

We now clean up the code a bit by utilizing the helper functions we included at the top. You can skip this if your theme already utilizes *showsublinks()* and directly go to the next section. Older PHP-Fusion 6 themes like Prime use a lengthy code construct to display the horizontal navigation/subheader/white-header links. In prime, it's this code:

```
<td class='white-header'>;
    $result = dbquery("SELECT * FROM ".DB_PREFIX."site_links WHERE link_position>='2' OR");
    if (dbrows($result) != 0) {
        $i = 0;
        while($data = dbarray($result)) {
            if (checkgroup($data['link_visibility'])) {
                if ($data['link_url']!="---") {
                    if ($i != 0) { echo " &middot;\n"; } else { echo "\n"; }
                    $link_target = ($data['link_window'] == "1" ? " target=" : "");
                    if (strstr($data['link_url'], "http://") || strstr($data['link_url'], "https://")) {
                        echo "
                        <a href='".$data['link_url']."'".$link_target." class='white'>".
                    } else {
                        echo "
                        <a href='".BASEDIR.$data['link_url']."'".$link_target." class='white'>".
                    }
                }
                $i++;
            }
        }
        echo ($i == 0 ? "&nbsp;" : "");
    }
</td>
```

We can replace that by a much shorter:

```
<td class='white-header'>".showsublinks(" ".THEME_BULLET." ", "white")."</td>
```

The first parameter of this function is the separator and the second one is the CSS class applied to the individual items.

Some v6 themes use this for the date:

```
ucwords(showdate($settings['subheaderdate'], time()))
```

you can strip that down to this in PHP-Fusion 7:

```
showsubdate()
```

The PHP-Fusion credit string can be **replaced** by a small function as well:

**Before:**

```
echo "Powered by <a href='http://www.php-fusion.co.uk' target='_blank'>PHP-Fusion</a> v".$settings['vers
```

### After:

```
echo showcopyright();
```

In analogy, the counter code can also be **replaced** by a function:

### Before:

```
echo $settings['counter']. " ".($settings['counter'] == 1 ? $locale['140']. "<br><br>\n" : $locale['141']).
```

### After:

```
echo showcounter()."
```

Now we only have to drop some *tablebreak()* instances as this function was dropped in PHP-Fusion 7. We will later correct the margins using CSS.

The Prime *render\_page()* function looks like this after cleaning up the code and merging some echo statements:

```

function render_page($license=false) {
global $settings;
echo "<table align='center' cellspacing='0' cellpadding='0'>
  <tr>
    <td>
      <table cellpadding='0' cellspacing='0' width='100%'>
        <tr>
          <td class='full-header'>
            <table cellpadding='0' cellspacing='0' width='100%'>
              <tr>
                <td>".showbanners()."</td>
              </tr>
            </table>
          </td>
        </tr>
      </table>
      <table cellpadding='0' cellspacing='0' width='100%'>
        <tr>
          <td class='white-header'>".showsublinks(" ".THEME_BULLET." ", "white")."</td>
          <td align='right' class='white-header'>".showsubdate()."</td>
        </tr>
      </table>
      <table cellpadding='0' cellspacing='0' width='100%'>
        <tr>";
        if (LEFT) { echo "<td class='side-border-left' valign='top'>".LEFT."</td>"; }
        echo "<td class='main-bg' valign='top'>".U_CENTER.CONTENT.L_CENTER."</td>";
        if (RIGHT) { echo "<td class='side-border-right' valign='top'>".RIGHT."<
        echo "
      </tr>
    </table>
  <table cellpadding='0' cellspacing='0' width='100%'>
    <tr>
      <td class='footer'><br>".stripslashes($settings['footer'])."</td>
    </tr>
    <tr>
      <td align='center' class='footer'>\n";
      if ($license == false) {
        echo showcopyright();
      }
      echo showcounter()."
    </td>
  </tr>
</table>
</td>
</tr>
</table>\n";
}

```

## render\_news()

The *render\_news()* function is easily updated. In newer PHP-Fusion 6 themes where you can find something like this:

```

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

```

You only have to **delete** *openform()* and **replace** *closeform()* by *itemoptions()*, e.g.:

```

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

```

In themes like Prime that do not use these functions, you replace the whole code by the new version, e.g.:

**Before:**

```
<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_
  echo $info['news_reads']. $locale['044']. "
  <a href='print.php?type=N&item_id=".$info['news_id']."'><img src='".THEME."/images/printer.gif' alt='
</td>
```

**After:**

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

Remember to drop *tablebreak()* function calls here as well.

**render\_article()**

Works exactly the same as updating news.php. Just replace news by article and N by A, e.g:

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

Remember to drop *tablebreak()* function calls here as well.

**openside() and opensidex()**

As you know, these functions were merged as well. This is pretty complicated as this was done differently by every author in each theme, so it's probably the best way to compare your *opensidex()/openside()* function with the finished one at the end of this section. I'll try to explain the steps though.

Start with **deleting** the *openside()* and *closeside()* functions. Now **rename** *closesidex()* to *closeside()* and **replace** the function opening of *opensidex()* with the new *openside()*:

**Before:**

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

**After:**

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

**Delete** stuff like this:

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

And **add** these lines at the top of our new *openside()*:

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

**Add** this line at the top of our new *closeside()*:

```
global $panel_collapse;
```

**Replace** the old collapse button by the new function based one (some themes might do this already) and wrap it into an if-statement so that it is only shown if the panel is supposed to be collapsible:

**Before:**

```
<td class='scapmain' align='right'><img src='".THEME."/images/panel_{$box_img}.gif' name='b_{$title}' alt=''
```

**After:**

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

Also **replace** the old div tags with the new function responsible for hiding the contents, also including collapse checks:

**Before:**

```

        <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";

```

**After:**

```

        <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";

```

## First Test

At this point, you can try to switch to the theme you are currently updating. It should display although not necessarily like it should. If it does not show at all or returns errors, browse through the guide again to see if you did everything and check for general issues in the code. If you can't find the problem, seek help on the PHP-Fusion Themes site [1] (<http://themes.php-fusion.co.uk>) .

## Updating Styles.css

If you took a look at your site after updating theme.php, you probably noticed that not everything looks like it did in v6. This is nothing unusual as we still did not update styles.css where a lot of the theme properties are handled in PHP-Fusion 7.

## Setting widths

When previewing your theme you probably noticed the wrong widths setting. This is because widths are not defined in theme.php anymore but in styles.css to make the output cleaner.

To set the width of the whole theme, we need to identify the outmost table/div of the theme via css. In some themes the outer table already has a class or id we could use for this purpose, in other themes, like Prime, we don't have such a class/id yet and add it now. Go back to theme.php and make sure that the outer table has a class/id or **add** one like me:

### Before:

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

### After:

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

Now that we can identify the main table, we can set its width via css. Just add this to *styles.css* somewhere:

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

Make sure to use your id and width here. If you use a class the code looks like this:

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

The theme should look better already, but the side columns are likely to be too wide now. We already added some classes to them when we updated theme.php, so we can add their widths right away. Most *styles.css* should already contain properties for *.side-border-left* and *.side-border-right*. Find them and **add** the width property or add the classes to *styles.css*. In Prime both classes had already been there and I just added the width:

### Before:

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

### After:

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

The widths should now be okay but there is still some stuff to do.

## Styling new elements

Next, we will style the elements that worked differently in PHP-Fusion 6 or were newly added.

### Horizontal navigation

One of these new items are the lists utilized for the horizontal navigation as well as the CSS navigation panel. This snippet works fine to fix the horizontal navigation. Just **add** it to *styles.css*:

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

If your horizontal navigation is in a cell with a different class name, just **replace** *.white-header* with your class name.

What does it do? It removes the default margin and padding of the list container (ul) so that it does not indent the items or take too much space. It also makes the list items (li) align next to each other and not below each other. Finally the bullet image of the first item is removed. This should make the horizontal navigation look like it did in PHP-Fusion 6 in most themes.

The position and layout of the links should be fine, but it is possible that their colour is wrong. This is because the class of the horizontal links was added to the link in PHP-fusion 6 and is added to the list item in PHP-Fusion 7. We can fix it like this in Prime where *.white* is the horizontal navigation links class:

#### Before:

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

#### After:

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

If your links use a different class (refer to the *showsublinks()* function parameter in *render\_page()*), **apply** this changes to the class you are using. We keep the old way as well as there might be other links, for example in the footer, which are also styled using the *.white* class.

### Navigation Panel (CSS Version)

Next, we style the CSS Navigation Panel list. First, we **add** this to take care of the list container display once again:

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

Next, we style the category labels. Refer to the side-label class to find out which colours you should use

here:

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

Now we style the links and their hover state. Use colours that fit your theme here:

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

This last bit is necessary to fix the display in IE6:

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

**Add** all this to *styles.css*.

### Page Navigation (Pagination)

In PHP-Fusion 7, the "pagenav", the small list of pages you'll find here and there on PHP-Fusion websites does not share the style of the forum tables but can be styled individually.

To do this, we **add** this to *styles.css* (change the colours to fit your theme. Refer to *tbl1* and *tbl2* if necessary)

```
.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;
}
```

The first bit will style the active page box, the second will style the other page boxes which link to the corresponding pages and the third bit styles the hover state of these links.

## Centered Elements

In PHP-Fusion 6 many items on the site were centered using the obsolete *align* tag (e.g. `<table align='center' [...]>`). In PHP-fusion 7 this is only done in some places and done by using a new center CSS class in others. To make both evrsions woking in both, div and table themes, we can **add** this line to *styles.css*:

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

## Admin Message

PHP-Fusion 7 uses a `admin_message` class to style important messages. This is the class added in Prime:

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

Adapt the style to fit your theme.

## BBCodes

The bbcode Buttons can be styled individually in PHP-Fusion 7. In Prime, I merely added a light border:

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

## Resembling the `tablebreak()` effect

If you previewed the theme at some point while editing *styles.css*, you probably noticed that it's already pretty close to what we want or at least it should be.

One thing we still have to do is to fix the margins between the panels which were removed when we dropped *tablebreak()*.

To do this we have to apply a CSS property to the parent element of the panels or to the element after which the panel ends if there is no parent element. If there is not suitable element we can already identify using a class (many v6 themes use a *border* class for the parent panel table), we have to add one. In Prime there is a *border* class, but it is only applied to center panels. I decided to create new class of the name *tablebreak* and add it wherever I need to add the margin *tablebreak()* created before. I **add** it in five spots:

## Horizontal Navigation/Date Bar:

```
<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>
```

## News:

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

## Articles:

```
<table cellpadding='0' cellspacing='0' width='100%' class='border tablebreak'>
  <tr>
    <td class='main-body'>
      " .($info['article_breaks'] == "y" ? nl2br($article) : $article)."
    </td>
  </tr>
```

## Side Panels:

```
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='
  </tr>
```

## Center Panels:

```
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'>
```

Now that the tables are identified using classes, we can **add** this to *styles.css* to reset the margins:

```
.tablebreak { margin-bottom: 8px; }
```

## Cleanups and Fixes

At the end we can make sure that we have a clean and working theme by

1. checking for any remaining *tablebreak()* calls,
2. setting the opening php tag (`<?` or `<?php`) to the long variant (`<?php`) to avoid problems on some PHP 5 configurations,
3. closing `<br>` tags (`<br/>`),
4. as well as closing `<img [...]>` tags (`<img [...]/>`) and any other tags that don't use a closing tag like this: `</tag>`,
5. adding *alt* attributes to images,
6. adding *img { border: 0; }* to *styles.css* in order to drop borders around linking images.

7. remove *filter* properties from style.css. Many older themes use these invalid properties for a gallery image hover effect.

## Final Preview

Now that we finished updating styles.css, we can view our changes again. This time most things should more or less look like they did in PHP-Fusion 6. If they don't and you did follow all the steps, seek help in the Updating Help Forum on PHP-Fusion Themes ([http://themes.php-fusion.co.uk/forum/viewforum.php?forum\\_id=16|V7](http://themes.php-fusion.co.uk/forum/viewforum.php?forum_id=16|V7))

The finished v7 Version of Prime can be downloaded here (<http://themes.php-fusion.co.uk/downloads/Prime.zip>) .

Retrieved from "[http://wiki.php-fusion.co.uk/index.php?title=PHP-Fusion\\_7\\_Theming](http://wiki.php-fusion.co.uk/index.php?title=PHP-Fusion_7_Theming)"

Category: Development

---

- This page was last modified 09:11, 20 August 2008.