

Drupal tpl.phps are not templates

By Bevan Rudge

<http://Drupal.geek.nz>

<http://Twitter.com/BevanR>

Front End Software Engineer

NowPublic.com

(Examiner.com, Clarity Digital Group)

Drupal tpl.phps;

- contain significant amounts of PHP (instead of markup)
- have logic blocks, iterations, complex conditionals, and function calls
- are not easily edited by non-programmers (e.g. designers)

Templates;

- have markup
- do not have logic
- are readable and editable by non-programmers

CCK: content-admin-field-overview-form.tpl.php

```
18 <?php  
19 $count = 0;  
20 foreach ($rows as $row): ?>  
21 <tr class="<?php print $count % 2 == 0 ? 'odd' : 'even'; ?> <?php print $row->class ?>">  
22 <?php  
23 switch ($row->row_type):  
24 case 'field': ?>  
25 <td>  
26 <?php print $row->indentation; ?>
```

- Iteration; foreach ()
- Modulo '%' operator; \$count % 2
- If-else shorthand; \$boolean ? 'odd' : 'even'
- Incrementer; \$count++
- Complex conditional; switch-case

CCK: content-field.tpl.php.png

```
33 <div class="field-items">
34   <?php $count = 1;
35   foreach ($items as $delta => $item) :
36     if (!$item['empty']) : ?>
37       <div class="field-item <?php print ($count % 2 ? 'odd' : 'even') ?>">
38         <?php if ($label_display == 'inline') { ?>
39           <div class="field-label-inline<?php print($delta ? '' : '-first')?>">
40             <?php print t($label) ?>:&nbsp;</div>
41           <?php } ?>
42           <?php print $item['view'] ?>
43         </div>
44       <?php $count++;
45     endif;
46   endforeach;?>
```

- Arrays; \$item['empty']
- t()

Core block module; block-admin-display-form.tpl.php.png

```
29 <?php-
30 // Add table javascript.-
31 drupal_add_js('misc/tableheader.js');-
32 drupal_add_js(drupal_get_path('module', 'block') .'/block.js');-
33 foreach ($block_regions as $region => $title) {-
34     drupal_add_tabledrag('blocks', 'match', 'sibling', 'block-region');-
35     drupal_add_tabledrag('blocks', 'order', 'sibling', 'block-weight');-
36 }-
37 ?>-
38 <table id="blocks" class="sticky-enabled">-
39     <thead>-
40         <tr>-
41             <th><?php print t('Block'); ?></th>-
42             <th><?php print t('Region'); ?></th>-
43             <th><?php print t('Weight'); ?></th>
```

- Drupal API calls; drupal_add_js() and drupal_add_tabledrag()

Garland; page.tpl.php

```
23 <div id="header">
24   <div id="logo-floater">
25     <?php
26       // Prepare header
27       $site_fields = array();
28       if ($site_name) {
29         $site_fields[] = check_plain($site_name);
30       }
31       if ($site_slogan) {
32         $site_fields[] = check_plain($site_slogan);
33       }
34       $site_title = implode(' ', $site_fields);
35       if ($site_fields) {
36         $site_fields[0] = '<span>'. $site_fields[0] . '</span>';
37       }
38       $site_html = implode(' ', $site_fields);
39
40       if ($logo || $site_title) {
41         print '<h1><a href="'. check_url($front_page) .'" title="'. $site_title .' ">';
42         if ($logo) {
43           print '';
44         }
45         print $site_html . '</a></h1>';
46       }
47     ?>
48   </div>
49
50   <?php if (isset($primary_links)) : ?>
51     <?php print theme('links', $primary_links, array('class' => 'links primary-links')) ?>
52   <?php endif; ?>
```

Garland; page.tpl.php

- Building an array() based on variables;
`$site_fields = array()`
- PHP array/object manipulation; `implode()`
- Drupal API functions; `theme()`, `check_plain()`,
`check_url()`
- Markup in PHP strings; `print '<h1><a href=...`

tpl.phps with markup in PHP strings!

- WTF!!??
- Treat templates as .HTML files with PHP snippets.
- **Not** PHP files with HTML snippets
- (That's what theme and preprocess functions are for.)

Webform; webform-mail.tpl.php

```
26 <?php print t('Submitted on @date', array('@date' => format_date(time(), 'small'))) ?>
27 -
28 <?php if ($user->uid): ?>
29 <?php print t('Submitted by user: @username [@ip_address]', array('@username' => $user->name, '@ip_address' => $ip_address)) ?>
30 <?php else: ?>
31 <?php print t('Submitted by anonymous user: [@ip_address]', array('@ip_address' => $ip_address)) ?>
32 <?php endif; ?>
33 -
34 -
35 <?php print t('Submitted values are') ?>:
36 -
37 <?php
38 // Print out all the Webform fields. This is purposely a theme function call
39 // so that you may remove items from the submitted tree if you so choose.
40 // unset($form_values['submitted_tree']['element_key']);
41 print theme('webform_mail_fields', 0, $form_values['submitted_tree'], $node);
42 ?>
43 -
44 <?php print t('The results of this submission may be viewed at:') ?>
45 -
46 <?php print url('node/'. $node->nid .'/submission/'. $sid, array('absolute' => TRUE)) ?>
```

Where is the markup!!??

hook_preprocess()

- Most of these things should be moved to hook_preprocess() functions

But the real problem is...

tpl.phps are not templates

- tpl.phps are executable PHP scripts
- Developers forget that tpl.phps are for markup
- Drupal allows executable PHP in tpl.phps

Drupal allows executable PHP in tpl.phps

Drupal allows executable PHP in tpl.phps

- Theming Drupal is hard for non-programmers
 - (They don't think in terms of function calls and code execution order)
- Templates are not portable across systems/languages;
 - javascript/jQuery templates for client-side theming
js/ajax data objects
 - Other PHP CMS
 - CMS in other languages

jQuery core templating (proposal)

```
<script id="list-item" type="text/html">  
  <li id="%identifier%" class="%classes%">  
    <a href="%url%" title="%title%">  
      %text%  
    </a>  
  </li>  
</script>
```

Templating systems;

- are independent of the programming language
- force developers to create markup-based templates (not PHP-logic blocks)
- may allow for some very basic logic, e.g. if (isset(\$title)):

Possible solution: **Smarty**

`<h2>[title]</h2>`

Smarty has disadvantages (by reputation) such as performance

Possible solution:

Roll our own templating system

```
<h2>{$title}</h2>
```

- Create our own (restrictive) templating system?
- Treat tpl.php's as PHP strings?
- What about logic like if () statements?

Possible solution: **PHP is a templating language**

- Templating systems are not programming languages
- This is how PHPtemplate currently works
- It doesn't solve the problem

Possible solution: **Education**

- Document, promote and/or enforce tpl.php standards
- Educate Drupal core and contrib developers
- We already do this to a large extent
- Coder module can help

Acceptable tpl.php logic;

- `print $var;`
- `if (isset($var)): ... endif;`
- `if (!empty($var)): ... endif;`

And, arguably;

- if-elseif
- if-else
- t('Hello world');
- l(t('Hello world'), 'node/1');
- drupal_render()?
- theme()?

Where do you draw the line?

Thank you!

Discuss.