Keeping my PHP pretty

Asked 16 years, 3 months ago Modified 16 years, 1 month ago

Viewed 5k times Part of PHP Collective

11

I am creating a site in which different pages can look very different depending upon certain conditions (ie logged in or not, form filled out or not, etc). This makes it necessary to output diferent blocks of html at different times.

Doing that, however, makes my php code look horrific... it really messes with the formatting and "shape" of the code. How should I get around this? Including custom "html dump" functions at the bottom of my scripts? The same thing, but with includes? Heredocs (don't look too good)?

Thanks!

PHP php html formatting

Share

Improve this question

Follow

edited Sep 22, 2008 at 4:11

Factor Mystic

26.7k ● 16 ● 81 ● 96

asked Sep 22, 2008 at 3:46

stalepretzel

15.9k ● 23 ● 78 ● 92

14 Answers Sorted by: Highest score (default)

26

Don't panic, every fresh Web programmer face this problem.

You HAVE TO separate your program logic from your display. First, try to make your own solution using two files for each Web page :

- one with only PHP code (no HTML) that fills variables
- another with HTML and very few PHP : this is your page design

Then include where / when you need it. E.G :

myPageLogic.php

```php
<?php

// pure PHP code, no HTML

$name = htmlspecialchars($_GET['name']);
$age = date('Y') - htmlspecialchars($_GET['age']);

?>
```

myPageView.php

```php
// very few php code
// just enought to print variables
// and some if / else, or foreach to manage the data s

<h1>Hello, <?php $name ?> !</h1>
```

```
<p>So your are <?php $age?>, hu ?</p>
```

(You may want to use the alternative PHP syntax for this one. But don't try to hard to make it perfect the first time, really.)

myPage.php

```
<?php

require('myPageLogic.php');
require('myPageView.php');
?>
```

Don't bother about performance issues for now. This is not your priority as a newbie. This solution is imperfect, but will help you to solve the problem with your programming level and will teach you the basics.

Then, once your are comfortable with this concept, buy a book about the MVC pattern (or look for stack overflow entries about it). That what you want to do the *NEXT TIME*. Then you'll try some templating systems and frameworks, but *LATER*. For now, just code and learn from the beginning. You can perfectly code a project like that, as a rookie, it's fine.

edited Oct 29, 2008 at 18:52

answered Sep 22, 2008 at 8:47

Bite code
595k ● 116 ● 309 ● 334

That's what I do nowadays. Process the data in a script, the include the appropriate 'view' script. In view script, I only echo variables inside HTML and use if-else, loops when necessary. – Imran Sep 22, 2008 at 13:20

3 Yeah, sometimes I regret nobody just told me that during my first years. People should let noobs so just do something instead of asking them to make a perfect start for every fresh tech they come to. – Bite code Oct 29, 2008 at 18:43

8

Use a mvc approach.

http://www.phpmvc.net/

This is not something that you will pick up in a couple of hours. You really need to practice it. Main thing is the controller will access your model (the db layer), do stuff to your data and then send it to the view for rendering. This is oversimplified but you just need to read and practice it to understand it.

This is something I used to help me learn it.
http://www.onlamp.com/pub/a/php/2005/09/15/mvc_intro.html

Share Improve this answer

Follow

answered Sep 22, 2008 at 3:49

jdelator
4,241 ● 6 ● 40 ● 54

4

Try to separate your content and layout from your code as much as possible. Any time you write any HTML in a .php file, stop and think "Does this *really* belong here?"

One solution is to use templates. Look at the Smarty templating system for a pretty easy-to-use option.

Share Improve this answer

Follow

answered Sep 22, 2008 at 3:55

nickf

546k ● 198 ● 658 ● 725

Any template engine would be a solution to this without the overheads of a MVC framework. Smarty is a most excellent choice. – Ian Sep 22, 2008 at 8:54

1 A templating system (Smarty) on top of another templating system (PHP itself) is overhead in principle. I'm all for using techniques like XSL Transformations (w3schools.com/xsl) Twan – Twan Sep 22, 2008 at 9:40

1 man, seriously... XSLT is waaaaay too complicated and definitely unneeded from what the OP was talking about. For starters, you need some XML to transform in the first place. – nickf Sep 22, 2008 at 12:34

2

> Doing that, however, makes my php code look horrific... it really messes with the formatting and "shape" of the code. How should I get around this?

Treat your PHP and HTML as a single hierarchy, with a single, consistent indentation structure. So a PHP

enclosing-structure such as an 'if' or 'for' introduces a new level of indentation, and its contents are always a balanced set of start and end-tags. Essentially you are making your PHP 'well-formed' in the XML sense of the term, whether or not you are actually using XHTML.

Example:

```
<div class="prettybox">
    Hello <?php echo(htmlspecialchars($name)) ?>!
    Your food:
    <?php foreach($foods as $food) { ?>
        <a href="/food.php?food=<?php echo(urlencode($
            <?php echo(htmlspecialchars($food)) ?>
        </a>
    <?php } ?>
    <?php if (count($foods)==0) { ?>
        (no food today)
    <?php } ?>
</div>
```

Be wary of the religious dogma around separating logic and markup rearing its head in the answers here again. Whilst certainly you want to keep your business-logic out of your page output code, this doesn't necessarily mean a load of overhead from using separate files, classes, templates and frameworks is really necessary for what you're doing. For a simple application, it is likely to be enough just to put the action/logic stuff at the top of the file and the page output below.

(For example from one of the comments above, doing htmlspecialchars() is page-output functionality you definitely *don't* want to put in the action bit of your PHP,

mixed up in all the business logic. Always keep text as plain, unescaped strings until the point where it leaves your application logic. If typing 'echo(htmlspecialchars(...))' all the time is too wordy, you can always make a function with a short name like 'h' that does the same.)

Share Improve this answer

Follow

answered Sep 22, 2008 at 16:25

bobince

536k ● 110 ● 671 ● 843

This is an interesting way to use `urlencode` and `htmlspecialchars` inside the template/view. Is this truly a better method than relying on the controller/framework view object to escape the characters and just pass the data onto the view for echoing? I personally use the latter. The controller is responsible for getting and manipulating the data. This is not the views responsibility. It simply echos data, not manipulating it. – Gary Green Aug 13, 2010 at 14:27

How does the framework (much less the controller) know what escaping the output view stage is going to need to do? It may include not only HTML-escaping, but also templating values into URLs (`urlencode` / `rawurlencode` depending on which bit), or JavaScript string literals (`json_encode`), or a CSS string literal, or a pasteable HTML snippet (double HTML-encode), or maybe it's spitting the content out into a plain text e-mail (no encoding). If the controller's pre-encoded everything to HTML, this can't work. Escaping for context is absolutely a core concern of the view. – bobince Aug 13, 2010 at 23:07

1

From the sound of your problem, it seems you might not have much separation between logic and presentation in your code. When designing an application this is a very important consideration, for reasons exactly demonstrated by the situation your currently facing.

If you haven't already I'd take a look at some PHP templating engines such as Smarty.

Share Improve this answer

Follow

answered Sep 22, 2008 at 3:53

Chris Cherry

28.5k ● 6 ● 72 ● 71

0

In cases like that I write everything incrementally into a variable or sometimes an array and then echo the variable/array. It has the added advantage that the page always renders in one go, rather than progressively.

Share Improve this answer

Follow

answered Sep 22, 2008 at 3:49

da5id

9,136 ● 9 ● 41 ● 53

0

What I end up doing in this situation is building 'template' files of HTML data that I Include, and then parse through with regular expressions. It keeps the code clean, and makes it easier to hand pieces off to a designer, or other HTML person.

Check out the ideals behind MVC programming practices at Wikipedia

Share Improve this answer

Follow

answered Sep 22, 2008 at 3:49

Issac Kelly

6,359 ● 6 ● 44 ● 50

0

You need a framework. This will not only make your code pretty (because of the already-mentioned model-view-controller pattern, MVC) - it will save you time because of the components that are already written for it.

I recommend QCodo, it's amazing; there are others - CakePHP, Symfony.

Share Improve this answer

Follow

answered Sep 22, 2008 at 3:57

Alex Weinstein

9,871 ● 9 ● 45 ● 59

0

If it's a big project a framework might be in order. If not, create some template files and at the top of the page decide which template file to include.

for example

```
if ($_SESSION['logged_in'])
     include(TPL_DIR . 'main_logged_in.tpl');
else
     include(tPL_DIR . 'main.tpl');
```

just a simple example

Share Improve this answer

answered Sep 22, 2008 at 4:18

Follow

Galen
30.2k ● 9 ● 73 ● 89

0

as mentioned above.. you are fixing the symptom, not the problem..

What you need is the separation of logic and presentation. Which can be done with some sort of mvc framework. Even if you don't want to go all the way with a mvc framework.. A templating engine is a must at least if your logic is complicated enough that you have trouble with what you are explaining

Share Improve this answer

Follow

answered Sep 22, 2008 at 7:24

paan
7,182 ● 8 ● 40 ● 44

0

I strongly suggest savant instead of smarty,

- you don't have to learn a new templating "language" in order to create templates, savant templates are written in php
- if you don't want to use php you can define your own template "language" with a compiler
- it's easily extendable by your own php objects

Separating logic from presentation does not mean that all you business logic has to be in php and your presentation logic in something else, the separation is a conceptual thing, you hae to separate the logic that prepares data

form the one that shows it. Obviously the business logic does not have to contain presentation elements.

Share Improve this answer

Follow

answered Sep 22, 2008 at 9:02

yann.kmm

837 ● 7 ● 21

0

In stead of implementing templating systems on top of a templating system (PHP itself), creating overhead per default, you can go for a more robust solution like XSL Transformations, which also complies with the MVC princples (provided you seperate your data-retrieval; plus, I personally split the logic from displaying the XML with different files).

Imagine having the following information in an array, which you want to display in a table.

```
Array
{
  [car] => green
  [bike] => red
}
```

You easily create a script that outputs this information in XML:

```
echo "<VEHICLES>\n";
foreach(array_keys($aVehicles) as $sVehicle)
  echo "\t<VEHICLE>".$sVehicle."</NAME><COLOR>".$aVehi
</VEHICLE>\n";
echo "</VEHICLES>\n";
```

Resulting in the following XML:

```
<VEHICLES>
  <VEHICLE>
    <NAME>car</NAME>
    <COLOR>green</COLOR>
  </VEHICLE>
  <VEHICLE>
    <NAME>bike</NAME>
    <COLOR>red</COLOR>
  </VEHICLE>
</VEHICLES>
```

Now this is all excellent, but that won't display in a nice format. This is where XSLT comes in. With some simple code, you can transform this into a table:

```
<xsl:template match="VEHICLES">
  <TABLE>
    <xsl:apply-templates select="VEHICLE">
  </TABLE>
</xsl:template>

<xsl:template match="VEHICLE">
  <TR>
    <TD><xsl:value-of select="NAME"></TD>
    <TD><xsl:value-of select="COLOR"></TD>
  </TR>
</xsl:template>
```

Et voila, you have:

```
<TABLE>
  <TR>
    <TD>car</TD>
    <TD>green</TD>
  </TR>
  <TR>
```

```
    <TD>bike</TD>
    <TD>red</TD>
  </TR>
</TABLE>
```

Now for this simple example, this is a bit of overkill; but for complex structures in big projects, this is an absolute way to keep your scripting logic away from your markup.

Share Improve this answer

Follow

edited Sep 22, 2008 at 10:27

answered Sep 22, 2008 at 10:21

Twan

438 ● 3 ● 8

Use htmlspecialchars when creating xml/html with string concatenation. – troelskn Sep 22, 2008 at 10:46

You are absolutely right. And a CDATA tag is always good too. But I left those out, since they'd only obfuscate the goal of this post :) – Twan Sep 22, 2008 at 10:52

0

Check this question about separating PHP and HTML, there are various ways to do it, including self written templating systems, templating systems such as Smarty, PHP as a templating system on it's own, etc etc etc...

Share Improve this answer

Follow

edited May 23, 2017 at 10:33

Community Bot

1 ● 1

answered Sep 22, 2008 at 13:03

Jrgns

25.2k ● 19 ● 73 ● 77

0

I think you've two options here, either use a MVC Framework or use the lazy templating way which would put significant overhead on your code. Obviously I'm with the former, I think learning MVC is one of the best web development tricks in the book.

Share Improve this answer

Follow

answered Sep 22, 2008 at 15:57

Adam Gibbins

363 ● 3 ● 9