

## Re: PHP scripts and IE

**Source:** <http://coding.derkeiler.com/Archive/PHP/alt.php/2005-01/0825.html>

---

**From:** Ashmodai (*ashmodai\_at\_mushroom-cloud.com*)

**Date:** 01/30/05

Date: Sun, 30 Jan 2005 03:24:49 +0100

Geoff Berrow scribbled something along the lines of:

> *I noticed that Message-ID: <cthahu\$7gm\$1@sparta.btinternet.com> from  
> Tony Wainwright contained the following:*  
>  
>  
>>*Sorry Geoff, just learning – if you could point out what is wrong with the  
>>code I would appreciate it. And perhaps suggest a better book (I am using  
>>PHP, MySQL and Apache All in One, published by Sams)*  
>  
>  
> *Actually it's not bad enough to cause the error but I think it could  
> lead you into bad habits. I find it easier to be consistent in my use  
> of superglobals and use them like so :- \$\_POST['user']*  
>  
> *If you tried to do*  
> *echo \$\_POST[user];*  
> *You would get a warning notice. It only works in your case because it  
> is contained within double quotes. I find it simpler to use just one  
> way. Unfortunately this means you have to write the lines as:-*  
>  
> *echo "<p>Welcome <b>".\$\_POST['user'].</b>!</p>";*  
> *echo "<p>Your message is:<br><b>".\$\_POST['message'].</b></p>";*  
> *But it saves you having to think of whether to use single quotes or not.*  
>  
> *Also, it's a good idea to check if variables exist before using them. I  
> know this script won't get called unless the POST variables exist, but  
> again it's all about good habits.*

Actually the most sensible code would be:

```
echo '<p>Welcome <b>', $_POST['user'], '</b>!</p>';  
echo '<p>Your message is:<br>  
<b>', $_POST['message'], '</b></p>';
```

Single quotes mean that the content will be printed as-is. Double quotes mean the content might contain magic (variables, etc) which should be parsed. Also whitespace behaves differently from what you'd expect when you're using double quotes.

Thus single quoted strings are processed faster.

Using the dot operator leads to the string being combined before being echoed, the comma will lead to every single value being echoed on its own (as both things lead to the same output the latter should be preferred because it's faster).

You should always check whether the array key exists before you use it (array\_key\_exists). Then you should usually perform a value check if you're going to do anything with it.

Of course the best solution if you have lots of text to output is not to run the static part through PHP in the first place, so:

```
<p>Welcome <b><?php echo $_POST['user']; ?></b>!</p>  
<p>Your message is:<br>  
<b><?php echo $_POST['message']; ?></b></p>
```

would be the best solution IMO (although you might want to make it multilingual later on, but I won't go into that).

--  
Ashmo