

01/11/09 - PHP Has An Identity Crisis



Hello there! If you are new here, you might want to **subscribe to the RSS feed** for updates on this topic. I have been using PHP for quite a while. I first learnt PHP 4 when I was 18-19, building a dynamic art gallery type site for my drawing attempts. I thought it was wonderful that I could programmatically display new images without having to update the HTML file. I also used PHP for my final year project in university, and by the time I graduated, I was quite proficient in it. However, as I slowly got more and more interested in programming, and the principles of programming, I learn more and more languages, and learnt OOP practices that I never applied to PHP, simply because I was self taught, and at the time PHP wasn't pushing itself as an OOP language.

However, times change, and the PHP community tends to push the fact that PHP is an OOP language. Firstly, this isn't entirely accurate. PHP is a language that supports OOP, however it isn't an OOP language. There is no string object? I cannot take a substring of a string by calling a method on the string variable. Instead I must use a global function `substr()` to do it. That's not OOP, it's procedural wrapped in OOP.

However, PHP does support OOP. You can define custom classes, you can create objects and you can call methods on that object. In a hostile world full of more OOP languages like Ruby and Python that are gaining traction in the web development world, PHP must push more and more it's OOP support. If you talk with serious developers in the world of PHP, they will insist that although PHP is not inherently an OOP language, it's close enough.

Today I found out that from PHP has introduced in PHP 5.3 the GOTO statement.

For those of you who are unaware of the specifics of the GOTO function, [this post](#), also the post that I learnt this fact from, outlines it. While experienced programmers know enough to avoid the GOTO statement, new programmers may not, and PHP can be abused enough already without the GOTO statement.

So, what is it PHP? Are you an OOP language, or are you procedural? The PHP community pushes it's support for OOP, and then it introduces a control statement that characterises the worst of procedural code. PHP needs to clarify it's future as a

language, and decide whether to push itself as an OOP language or as a procedural language.