PHP Commenting Style

Any programmer can tell you that good commenting in your source code is an integral part of programming. Whether the language you're dealing with is an interpreted language like Javascript or compiled like C++, good comments lead to better readability and better flow of logic.

Developers of the Java programming language have come up with a strict commenting standard for Java to interface with a program called Javadoc. Not only does this standard remind the programmer to make appropriate comments, it also allows Javadoc to parse the strict comments into HTML documentation. The success of Javadoc carried over PHP with the creation of PHPDoc. The commenting syntax of PHPDoc is widely supported in the PHP community. Parsers for this standard include phpxref and phpdocumentor.

Basic Syntax

Here's an example of PHPDoc style comments:

```
/*** This is a short description** This is a longer description of the element that* I will comm
```

All PHPDoc comments begin with /** on its own line at the top and ends with a */ at the bottom. Each line of commenting is denoted by the single asterik at the beginning of each line. By convetion, PHPDoc comments begin with a short description followed by a longer description. Let's look at a more realistic example where we wish to comment on a dispenseIceCream() function:

```
/*** Dispenses desired flavored ice cream** Called upon when user has entered their* favorite ic
```

I've thrown a lot of new things in the above code, but the syntax should be somewhat self-explanatory. First, you should have noticed the @ tags. Tags in PHPDoc tell the parser exactly what you are going to be describing in your comments. Notice that each tag has its own parameters, separated by a space as in the @param tag. Tags are, in essence, the strength of PHPDoc.

@access tells the parser that the following element we are commenting on (the dispenseIceCream function) is a private function. @param gives the parser information about the parameters that the function takes in. Finally, @return provides information about what kind of output to expect out of the function.

Each tag in PHPDoc has its own syntax. For more information, look at the documentation

Useful Tags

Using **every single tag** in PHPDoc leads to the issue of overcommenting and obscurity of the actual code! Here, I outline what I deem to be useful tags to place in your document (tag info from phpdocumentor):

@access (private | protected | public)
The access tag will allow the documentation to clearly state how an element should be accessed. This is important for programming design purposes.

@param datatype $paramname description
I consider this tag to be one the most important tags in PHPDoc. It allows a quick look at how the input of a function should be formatted in order to use it.

@return datatype description
Similar to the @param tag, when dealing with functions, one of the things a programmer cares to look for is what this function will output.

@uses (please refer to documentation for proper syntax)
This tag allows actual linking and backlinking to another element! Its a neat feature that allows someone browsing the documentation to quickly see related functions or variables the element uses. The syntax is a bit tricky, so please study it.

@var datatype description
A nice tag to comment on important variables.

Other Tags to Note

The following tags might be useful or nice in your commenting and documentation, but I wouldn't consider them to be the bread and butter of PHPDoc.

@global datatype $globalvariablename | description
Global variables should be given more attention to. This tag should be used with @var.

@ignore
For whatever reason, you may not want PHPDoc to parse the following element. the @ignore tag (with no parameters) does just that. You can go ahead and give it a description if you feel its necessary.

@static

Similar to @access, program design is one of the things programmers may want to reference in your code. The static tag whether a a certain class of method should be treated as static.

@staticvar
Same as @static, except for variables.

inline {@link URL description}
This in-line tag is very useful whenever you wish to reference something outside within your descriptions. It creates a link to the URL you specify within the documentation. There are other inline tags, but this one is the most versatile (but possibly not the easiest to use).

Revision #1
Created Thu, Apr 19, 2007 6:17 PM by Wong, Duncan
Updated Fri, Apr 20, 2007 9:23 AM by Wong, Duncan