

PHP Coding Standards :-

Every company follows a different coding standard based on their best practices. Coding standard is required because there may be many developers working on different modules so if they will start inventing their own standards then source will become very un-manageable and it will become difficult to maintain that source code in future.

Here are several reasons why to use coding specifications –

- Your peer programmers have to understand the code you produce. A coding standard acts as the blueprint for all the team to decipher the code.
- Simplicity and clarity achieved by consistent coding saves you from common mistakes.
- If you revise your code after some time then it becomes easy to understand that code.
- Its industry standard to follow a particular standard to being more quality in software.

There are few guidelines which can be followed while coding in PHP.

- **Indenting and Line Length** – Use an indent of 4 spaces and don't use any tab because different computers use different setting for tab. It is recommended to keep lines at approximately 75-85 characters long for better code readability.
- **Control Structures** – These include if, for, while, switch, etc. Control statements should have one space between the control keyword and opening parenthesis, to distinguish them from function calls. You are strongly encouraged to always use curly braces even in situations where they are technically optional.

Examples

```
if ((condition1) || (condition2)) {  
    action1;  
}elseif ((condition3) && (condition4)) {  
    action2;  
}else {  
    default action;  
}
```

You can write switch statements as follows –

```
switch (condition) {  
    case 1:  
        action1;  
        break;
```

```

case 2:
    action2;
    break;

default:
    defaultaction;
    break;
}

```

- **Function Calls** – Functions should be called with no spaces between the function name, the opening parenthesis, and the first parameter; spaces between commas and each parameter, and no space between the last parameter, the closing parenthesis, and the semicolon. Here's an example –

```
$var = foo($bar, $baz, $quux);
```

- **Function Definitions** – Function declarations follow the "BSD/Allman style" –

```

function fooFunction($arg1, $arg2 = '') {
    if (condition) {
        statement;
    }
    return $val;
}

```

- **Comments** – C style comments (`/* */`) and standard C++ comments (`//`) are both fine. Use of Perl/shell style comments (`#`) is discouraged.
- **PHP Code Tags** – Always use `<?php ?>` to delimit PHP code, not the `<? ?>` shorthand. This is required for PHP compliance and is also the most portable way to include PHP code on differing operating systems and setups.
- **Variable Names** –
 - Use all lower case letters
 - Use `'_'` as the word separator.
 - Global variables should be prepended with a `'g'`.
 - Global constants should be all caps with `'_'` separators.
 - Static variables may be prepended with `'s'`.
- **Make Functions Reentrant** – Functions should not keep static variables that prevent a function from being reentrant.
- **Alignment of Declaration Blocks** – Block of declarations should be aligned.
- **One Statement Per Line** – There should be only one statement per line unless the statements are very closely related.
- **Short Methods or Functions** – Methods should limit themselves to a single page of code.

There could be many more points which should be considered while writing your PHP program. Over all intention should be to be consistent throughout of the code programming and it will be possible only when you will follow any coding standard. You can device your own standard if you like something different.