

Cebu Technological University - Main Campus

PSR-12 Extended Coding Style

The purpose of following this guideline is to avoid conflict from different authors. It does so by enumerating a shared set of rules and expectations about how to format PHP code. This specification extends, expands and replaces PSR-2, the coding style guide and requires adherence to PSR-1, the basic coding standard.

In General, we must follow the coding guidelines in PSR-1. As for the files, All PHP files **MUST** use the Unix LF (linefeed) line ending only. All PHP files **MUST** end with a non-blank line, terminated with a single LF. The closing `?>` tag **MUST** be omitted from files containing only PHP. As for the lines, The soft limit on line length **MUST** be 120 characters. Lines **SHOULD NOT** be longer than 80 characters; lines longer than that **SHOULD** be split into multiple subsequent lines of no more than 80 characters each. As for indenting, Code **MUST** use an indent of 4 spaces for each indent level, and **MUST NOT** use tabs for indenting. For keywords and types, all PHP reserved keywords and types **MUST** be in lower case. Any new types and keywords added to future PHP versions **MUST** be in lower case.

In declaring statements, namespace, and import statements, the header of a PHP file may consist of a number of different blocks. If present, each of the blocks below **MUST** be separated by a single blank line, and **MUST NOT** contain a blank line. The term "class" refers to all classes, interfaces, and traits. Any closing brace **MUST NOT** be followed by any comment or statement on the same line. When instantiating a new class, parentheses **MUST** always be present even when there are no arguments passed to the constructor. The `extends` and `implements` keywords **MUST** be declared on the same line as the class name. The `use` keyword used inside the classes to implement traits **MUST** be declared on the next line after the opening brace. Visibility **MUST** be declared on all constants if your project PHP minimum version supports constant visibilities (PHP 7.1 or later). The `var` keyword **MUST NOT** be used to declare a property. There **MUST NOT** be more than one property declared per statement. Visibility **MUST** be declared on all methods.

Method names **MUST NOT** be prefixed with a single underscore to indicate protected or private visibility. That is, an underscore prefix explicitly has no meaning. When present, the abstract and final declarations **MUST** precede the visibility declaration. When making a method or function call, there **MUST NOT** be a space between the method or function name and the opening parenthesis, there **MUST NOT** be a space after the opening parenthesis, and there **MUST NOT** be a space before the closing parenthesis. In if else statements, expressions in parentheses **MAY** be split across multiple lines, where each subsequent line is indented at least once. When doing so, the first condition **MUST** be on the next line.

As for operators, style rules for operators are grouped by arity (the number of operands they take). When space is permitted around an operator, multiple spaces **MAY** be used for readability purposes. Closures **MUST** be declared with a space after the function keyword, and a space before and after the use keyword. The opening brace **MUST** go on the same line, and the closing brace **MUST** go on the next line following the body. In Anonymous classes, they **MUST** follow the same guidelines and principles as closures in the above section. The opening brace **MAY** be on the same line as the class keyword so long as the list of implemented interfaces does not wrap. If the list of interfaces wraps, the brace **MUST** be placed on the line immediately following the last interface.