PHP Coding Standards

**PSR-1 Basic Coding Standard**

The more basic parts of PHP coding standards are defined in PSR-1. For example:

* Only <?php or <?= are allowed for PHP tags
* Files must be in UTF-8 without BOM(Byte Order Mark)
* Namespaces and class names must follow the standards in PSR-0 and PSR-4
* Class names must be defined in UpperCamelCase
* Class variables must be defined in UPPER\_SNAKE\_CASE
* Method names must be defined in camelCase

Standard functions in PHP are defined in snake\_case, but in PSR-1, method names must be defined in camelCase. There are no explicit rules for variable and property names, so you can use whichever style you like, but it is noted that they should be consistent. For example, defining normal properties in camelCase and static properties in UpperCamelCase like below:

class Something

{

public $normalPropterty;

public static $StaticProperty;

}

**Defining Classes**

You must put a newline before { in class definitions. Also, extends and implements must be written on the same line as the class name.

class ClassName extends ParentClassName implements Interface1, Interface2

{

// Class definition

}

**Defining Properties**

In PSR-2, you must not omit public/protected/private modifiers. In PHP, properties become public if these are omitted, but because it is hard to tell if one purposely omitted these modifiers or they just forgot, you should always explicitly write public. The static keyword comes next. You must not use var when defining properties because you can’t add any modifiers to var.

class ClassName

{

public $property1;

private $property2;

public static $staticProperty;

}

**Methods**

Like properties, you must have either one of public/protected/private and abstract/final comes after them if used. static is the last modifier. You must not put any spaces before and after braces, and you must put a newline before curly braces. Also, you must not put any whitespaces before commas in arguments, and you must put one whitespace after them.

class ClassName

{

abstract protected function abstractDoSomething();

final public static function doSomething($arg1, $arg2, $arg3)

{

// ...

}

}

**Conditional Statements**

For conditional statements,

* You must put one whitespace before (
* You must not put any whitespaces after (
* You must not put any whitespaces before )
* You must put one whitespace after )
* Also, use elseif rather than else if.

if ($condition1) {

// ...

} elseif ($condition2) {

// ...

} else {

// ... }

Be careful, else if and elseif are not the complete same things. elseif is one statement by itself, but else if on the other hand is interpreted as an if statement in the else of the first if.

if ($condition1) {

// ...

} else if ($condition2) {

// ...

} else {

// ...

}

The syntax above is actually interpreted like below:

if ($condition1) {

// ...

} else {

if ($condition2) {

// ...

} else {

// ...

}

}

For switch statements, case statements must be indented once from switch, and bodies for the cases must be indented once from case. When not breaking

after any kind of operations in case, you must write a comment.

switch ($condition) {

case 0:

echo 'First case, with a break';

break;

case 1:

echo 'Second case, which falls through';

// no break

case 2:

case 3:

case 4:

echo 'Third case, return instead of break';

return;

default:

echo 'Default case';

break;

}