coding standards in php
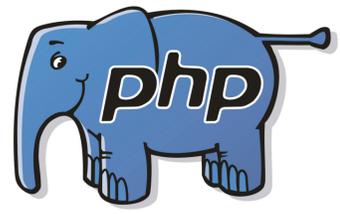
> writing clean code is a matter of professionalism

rule #1
Use an indent of 2 spaces, with no tabs.

```php
class ClassName {
__function __construct(argument) {
____// code...
__}
}
```

TWO SPACES
TWO TIMES TWO SPACES

rule #2
put a space before and after binary operators and concatenation symbols.
to improve the readability you may add a few more

```php
$short_____ =__ foo($bar);
$long_variable _ =__'text'__.__'more text';
$sum _____=__1__+__2;

$increment__._=__3;
```

rule #3
Control Structures

```php
if_($unhappy_||_$sad) {
__change_someting( );
}
elseif_($happy_&&_$satisfied) {
__keep_going( );
}
else_{
__find_out( );
}
```

1 one space between the control keyword and opening parenthesis

2 keep the opening curly brace on the same line as the control keyword

3 put the closing curly brace on a separate line

rule #4
function calls

```php
$var =_ foo($xkcd,_$is,_$cool);
```

1 no spaces between the function name, the opening parenthesis, and the first parameter

2 spaces between commas and each parameter

3 no space between the last parameter, the closing parenthesis, and the semicolon

rule #5 Arrays

```php
$pakstan = array(
  'Karachi'_____=> 'city of lights',
  'Lahore'_____=> 'city of gardens',
  'Quetta'_____=> 'the fruit basket',
  'Haiderabad'_=> 'city of perfumes',
);
```

```php
$var = foo($xkcd,_$is,_$cool);
```

1 comma at the end of the last array element

2 each element on it's own line

3 extra spaces in front of the arrows to increase readability

rule #6
PHP code tags

always start with `<?php`

only end with `?>`

in views, not in controllers

rule #7
Never omit semicolons

```php
<?php  print $php;  ?>
```
```php
<?php  print $php  ?>
```

I COULD RESTRUCTURE THE PROGRAM'S FLOW

OR USE ONE LITTLE 'GOTO' INSTEAD.

EH, SCREW GOOD PRACTICE. HOW BAD CAN IT BE?

goto main_sub3;

COMPILE