



[Home](#) / PHP programming shortcuts: save time and lines of useless code.

[Previous](#) [Next](#)

## PHP programming shortcuts: save time and lines of useless code.

Before I even started writing this, I am sure that the comments will be filled about how consolidating code makes it hard to understand, more braces makes it easier to read, etc. This is for those who think it's better to have less code, such as yours truly. I like seeing less code; I find it more efficient and much better.

### Short-hand mathematical operators

Those are known by most programmers however missed by a lot, the following can be shortened by it's alternative next to it.

```
<?php $int1 = $int1 + $int2; // or alternatively: $int += $int2;
$int1 = $int1 - $int2; // or alternatively: $int -= $int2; $int1 =
$int1 * $int2; // or alternatively: $int *= $int2; $int1 = $int1 /
$int2; // or alternatively: $int /= $int2; ?>
```

### Counters

Here is another very simple one, but good to know for those who don't know about it.

```
<?php $int = $int + 1; // or alternatively: $int++; $int = $int - 1;
// or alternatively: $int--; ?>
```

### Short-hand print statements

Ever find yourself have to write so much just to be able to echo a variable in HTML? No more!

```
<?php echo $string; ?>
```

```
<?=$string?>
```

### Braces: you don't always need them.

You don't always need to use the braces in any if or while statement if it only contains one

directive.

```
<?php if ($x) { echo "is x"; } ?>
```

Could just be

```
<?php if ($x) echo "is x"; ?>
```

## Ternary operator

My personal favorite, saves a lot of time and I still think it makes readable code

```
<?php if ($type == 'human') { echo 'human'; } else { echo 'robot'; } ?>
```

Or why not..

```
<?php echo ($type == 'human') ? 'human' : 'robot'; ?>
```

You can even set variables using that method

```
<?php $type_of_person = ($type == 'human') ? 'human' : 'robot'; ?>
```

## Verifying zero values not being false

Sometimes we deal with a function that can return a number and we want to make sure it returns a number, but if it returns 0, PHP will consider it as false.

```
<?php if (getNumber()) { doThings(); } ?>
```

The number can return 0 and be as false, where as the following will not consider the integer 0 false

```
<?php if (getNumber() !== false) { doThings(); } ?>
```

## Alternative control structure syntax

This is a great feature to do when you're trying to do a lot of PHP work with HTML, it would be hard to list all of them but you can refer to it here: <http://www.php.net/manual/en/control-structures.alternative-syntax.php>.

Feel free to post any other ones in the comments section.. we all have that one trick we always use!