

CHAPTER 3



Modular Programming

“Yes, I am a terrible coder, but I am probably still better than you :)”

—Rasmus Lerdorf

Chapter Objectives/Student Learning Outcomes

After completing this chapter, the student will be able to:

- Create an error-free simple objected-oriented (OO) modular PHP program
- Create a PHP class and make an instance of the class (object)
- Create an OO PHP encapsulated program, including GET and SET methods
- Create PHP methods (functions) that accept parameters and return information
- Create PHP public and private properties (variables)
- Import existing PHP code from another file or library into a program
- Validate information received using ternary (conditional) operators

PHP Libraries, Extensions, Classes, and Objects

One of the strengths of PHP is the ability to easily store modules of code in libraries. Once code has been installed in a library it can easily be reused in other programs. The reuse of code that has already been well tested and used in “live” environments greatly reduces program errors and increases productivity because you don’t have to reinvent the wheel. There is no need to recreate code that is already successfully working. It’s a waste of time and energy and likely to cause unnecessary program errors. The programmer actually may not know what actual code exists in a module of code (class). However, the programmer knows what parameters (such as numbers) can be passed into the “black box” and what is returned (sum of the numbers) from the black box.

It may concern you that the programmer blindly passes information into the black box and blindly receives information back. However, this is an advantage, not a disadvantage. This allows the creator of the module to update the code without affecting the way the module is used. As long as the module accepts the same input and returns the same output, the programmer who is using the module notices no changes. Updates can occur to the module for better efficiency, better security, or to correct any program code problems without causing the users to change how they handle the module in their code.

Electronic supplementary material The online version of this chapter (doi:[10.1007/978-1-4842-1730-6_3](https://doi.org/10.1007/978-1-4842-1730-6_3)) contains supplementary material, which is available to authorized users.