there are no Dumb Questions

Q: Does it matter whether I put PHP commands in uppercase or lowercase?

A: Yes and no. For the most part, PHP isn't case-sensitive, so you can get away with mixing the case of most commands. That means you can use `echo`, `ECHO`, or `EchC` when echoing content. However, as a matter of convention, it's a very good idea to be consistent with case in your scripts. Most PHP coders prefer lowercase for the vast majority of PHP code, which is why you'll see `echo` used throughout the example code in the book.

Q: So even if it's a bad coding convention, I can mix and match the case of PHP code?

A: No, not entirely. The huge exception to the case insensitivity of PHP is variable names, which apply to data storage locations that you create. So let's take the `$email` variable used in the Report an Abduction script as an example. This variable name is case-sensitive, so you can't refer to it as `$EMAIL` or `$eMail`. All variable names in PHP are case-sensitive like this, so it's important to name variables carefully and then reference them consistently in your code. More on variable names in just a moment.

Q: Is it really OK to put both PHP and HTML code in the same file?

A: Absolutely. In fact, in many cases it's absolutely necessary to do so.

Q: Why would I want to do that?

A: Because the whole idea behind a web server is to serve up HTML web pages to browsers. PHP doesn't change that fact. What PHP allows you to do is change the HTML content on the fly with things like today's date, data pulled from a database, or even calculated values such as the order total in a shopping cart. So PHP allows you to manipulate the HTML that goes into web pages, as opposed to them just being created statically at design time. It's very common to have HTML code for a page with PHP code sprinkled throughout to plug in important data or otherwise alter the HTML programmatically.

Q: Does PHP code embedded in an HTML file have to be on its own line, or can I embed it in an HTML line, like as part of an HTML tag attribute?

A: Other than needing to place your PHP code within the `<?php` and `?>` tags, there are no restrictions in how you embed it in HTML code. In fact, it's often necessary to wedge a piece of PHP code into the middle of HTML code, like when you're setting the attribute of an HTML tag. This is a perfectly legitimate usage of PHP.

Q: I've seen PHP code that's enclosed by <? as the start tag instead of `<?php`. Is that right?

A: Not really. Technically speaking, it's legal, but it isn't recommended. A server setting must be enabled for the short open tag (`<?`) to work. The usual `<?php` tag always works, so it's better to use that and know that your code will just work.

Q: If a web server always returns pure HTML code to a client browser, why do URLs show the PHP script name, like webpage.php?

A: Remember that every web page is the result of a two-sided communication involving a request from the client browser and a response from the web server. The URL is the basis of the request, while the content returned from the server is the response. PHP scripts are requested just like normal HTML web pages through URLs entered into the browser or linked from other pages, or as form actions. That explains why the URL for a PHP "page" shows the name of the PHP script.

The other half of the equation is the response from the server, which is the resulting code that's generated by the PHP script. Since most PHP scripts generate HTML code, it makes sense that the code is HTML and not PHP. So it's no accident that a URL references a .php file on a server, which causes PHP code to be executed on the server, ultimately resulting in pure HTML content being returned to the browser.

Q: Can PHP variables store any other kinds of data?

A: Absolutely. You can use variables to store Boolean (true/false) data. And numeric data can be either integer or floating-point (decimal). There are also arrays, which store a collection of data, as well as objects, which associate a collection of data with code that is used to manipulate the data. Arrays are covered a little later in this chapter, while objects are tackled in Chapter 12. There is also a special data type called `NULL`, which represents no value. For example, a variable that hasn't been assigned a value is considered `NULL`.