> *Programming note—`TRUE` and `FALSE` are constants that are included as part of the PHP language. Constants cannot be changed and are in all caps. `TRUE` is actually represented internally as a `1` and `FALSE` is represented internally as a `0`.*

■ **Note** In PHP 7, Scalar Type Hints can be used to enforce the data type being passed and returned.

```
declare(strict_types = 1);
function set_dog_name(string $value) : string
{
$error_message = TRUE;
(ctype_alpha($value) && strlen($value) < 21) ? $this->dog_name = $value : $error_message = FALSE;
return $error_message;
}
```

If the declare line is not included or `strict_types = 0`, the data type will not be enforced. To allow for backward compatability, the examples in this textbook will not show the use of Scalar Type Hints.

> *Programming note—`&&` is an `AND` operator. In order for the `(ctype_alpha($value) && strlen($value) < 21)` statement to be `TRUE`, `$value` must include only alphabetic characters and must be fewer than 21 characters.*

A ternary operator looks at the two possible statuses of the `$value` property (which contains whatever was passed into the method).

1. The `ctype` method is used to determine if the characters in `$value` are alphabetic (`ctype_alpha($value)`).
2. The `strlen` method is used to determine if the length of the string in `$value` is less than 21 characters (`strlen($value) < 21`).

To learn about additional `ctype` functions, visit:

http://php.net/manual/en/book.ctype.php.

If the `$value` property contains alphabetic characters only and is less than 21 characters, the `$dog_name` property is updated with the value that has been passed. If there are non-alphabetic characters or the length of the string is more than 20 characters, the `$error_message` is updated with a `FALSE` value (indicating the update did not occur). Finally, the value in `$error_message` (either `TRUE` or `FALSE`) is returned to the calling program.

> *Security and performance—This process may be a bit confusing now. However, it is important to create secure programs. Whenever an application or object accepts information from an outside source (such as another program or user) the information must be validated. This validation should include limitations on the size of the information accepted, along with other restrictions. Data that has been passed across the Internet (such as from the user's browser to a web server) can be intercepted and changed. It is vital that the information be verified within the application on the server before it is used. Validation may be done in the browser (via JavaScript) to ensure the user has entered correct information. However, as stated, packet stiffing programs can intercept that information and change it before it is received by an application on a web server.*

www.it-ebooks.info