```
If ( a > b)
{
print "it's A!";
}
else
{
print "it's B!";
}
```

In this example, the values in properties a and b are compared. If a is greater than b, then “it’s A!” is displayed. Otherwise, “it’s B!” is displayed. PHP does some type conversion when it determines that it might be necessary. For example, assume `a = "5"`, and `b = 6`. PHP would convert the value in a from a string 5 to a number 5 so it can do the comparison. Many languages do not do this and will display an error if you try to compare strings and numbers. If you don’t want this conversion to occur, you can use some special comparison operators. For example, you can use three = signs to see if values are exactly the same (`a === b`) instead of two.

Security and performance—Whenever possible use === instead of ==. This will assure that you get exactly what you expect.

In Example 2-6, the `if` statement calls a method (`isset`). `$_GET` tries to retrieve the property (`'submitbutton'`) and its value (“`Find Hello World!`”) from the HTML form, which used HTTP GET to pass the information (you could have used `$_POST` and HTTP POST). `isset` will return a `false` or `true` back to the `if` statement, depending on whether or not `$_GET` could retrieve the property (and its contents). The `true` or `false` will cause the `if` statement to determine which block of code to execute.

As mentioned in Chapter 1, an object (such as a button) within an HTML form will produce a property and value combination based on the object’s name and whatever is contained in the value statement of that object. This is even true for submit buttons that have been given a name (`id`) and value (as in Example 2-6).

The first time the program is called from the browser, the button has not been clicked. So there is no `submitbutton` variable created. A `false` is returned by the `isset` method. The code jumps to the `else` section and executes the `print` statements that display the HTML form and submit button (as in Figure 2-10).

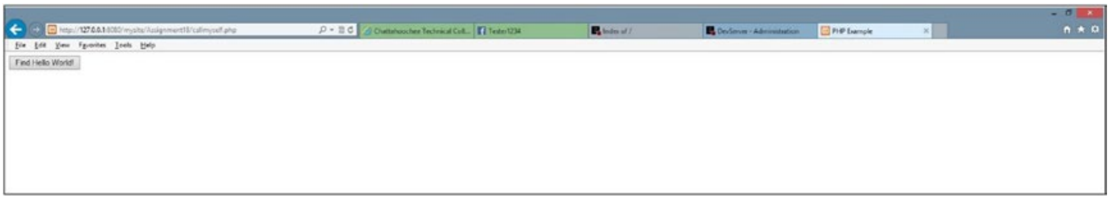
http://127.0.0.1:8080/mysite/Assignment18/callmyself.php
Chattahoochee Technical Coll...
Tester1234
Index of /
DevServer - Administration
PHP Example
File Edit View Favorites Tools Help
Find Hello World!

Figure 2-10. *callmyself.php before the button is clicked*

When the user clicks the button, the program calls itself (look at the `action` parameter of the `form` tag). This time, since the user has clicked the button, there is a submitbutton property and a value (`'Find Hello World!'`) for that variable. The program determines that the variable is “set” (has a value in it) and returns a `true`. The `if` statement then executes the one line of code between the `if` and `else` statements. `Hello World` is then displayed. The PHP program handled all the functionality of the application without using any existing static HTML page.