

In the previous examples, we have separated the initial interface (HTML and JavaScript) from the PHP program. You could actually place the HTML (to display the initial web page) and PHP code in the same file.

You can create a PHP file that will determine whether the user has ever requested the page before. If they have not, or the browser has timed them out, then the PHP program could display an initial HTML page for the user to interact (such as clicking a button). The same program could then call itself again to determine if the button has been clicked and return a response.

Let's look at a different version of the Hello World program to do just that.

Example 2-6. The `callmyself.php` file

```
<?php
if (isset($_POST['submitbutton']))
{
print "<h1> Hello World </h1>";
}
else
{
print "<html><head><title>PHP Example</title></head>";
print "<form method='post' action='callmyself.php'>";
print "<input type='submit' id='submitbutton' name='submitbutton' value='Find Hello
World!' />";
print "</form>";
print "</body></html>";
}
?>
```

For a more in-depth demonstration of PHP If statements, visit the following free videos from “The New Boston” (thenewboston) at:

<https://www.thenewboston.com/videos.php?cat=11&video=17004>.

This program includes a simple if conditional statement to determine if the user has clicked the submit button.

The format of the if statement is as follows:

```
if (conditional statement)
{
// code to execute if the conditional statement is true
}
else
{
// code to execute if the conditional statement is false
}
```

A conditional statement usually compares two values to determine if they are the same or different or calls a method which returns a true or false value. We will look at functions returning values in a moment. Let's look at a quick example of the first type.

Conditional statements use a comparison operator (`==`, `<`, `>`, `<=`, and `>=`) to determine if the statement is true or false.