

# Articles from Usman Transition Blog

## Share Each Others

### PHP Include File Server Side Includes (SSI)

2010-07-16 12:07:47 Usman Ependi

You can insert the content of one PHP file into another PHP file before the server executes it, with the `include()` or `require()` function. The two functions are identical in every way, except how they handle errors:

- `include()` generates a warning, but the script will continue execution
- `require()` generates a fatal error, and the script will stop

These two functions are used to create functions, headers, footers, or elements that will be reused on multiple pages.

Server side includes saves a lot of work. This means that you can create a standard header, footer, or menu file for all your web pages. When the header needs to be updated, you can only update the include file, or when you add a new page to your site, you can simply change the menu file (instead of updating the links on all your web pages).

---

## PHP `include()` Function

The `include()` function takes all the content in a specified file and includes it in the current file.

If an error occurs, the `include()` function generates a warning, but the script will continue execution.

### Example 1

Assume that you have a standard header file, called "header.php". To include the header file in a page, use the `include()` function:

```
<html>
<body>

<?php include("header.php"); ?>
<h1>Welcome to my home page!</h1>
<p>Some text.</p>

</body>
</html>
```

## Example 2

Assume we have a standard menu file, called “menu.php”, that should be used on all pages:

```
<a href="/default.php">Home</a>
<a href="/tutorials.php">Tutorials</a>
<a href="/references.php">References</a>
<a href="/examples.php">Examples</a>
<a href="/about.php">About Us</a>
<a href="/contact.php">Contact Us</a>
```

All pages in the Web site should include this menu file. Here is how it can be done:

```
<html>
<body>

<div>
<?php include("menu.php"); ?>
</div>

<h1>Welcome to my home page.</h1>
<p>Some text.</p>

</body>
</html>
```

If you look at the source code of the page above (in a browser), it will look like this:

```
<html>
<body>

<div>
<a href="/default.php">Home</a>
<a href="/tutorials.php">Tutorials</a>
<a href="/references.php">References</a>
<a href="/examples.php">Examples</a>
<a href="/about.php">About Us</a>
<a href="/contact.php">Contact Us</a>
</div>

<h1>Welcome to my home page!</h1>
<p>Some text.</p>

</body>
</html>
```

# PHP require() Function

The require() function is identical to include(), except that it handles errors differently.

If an error occurs, the include() function generates a warning, but the script will continue execution. The require() generates a fatal error, and the script will stop.

## Error Example include() Function

```
<html>
<body>

<?php
include("wrongFile.php");
echo "Hello World!";
?>

</body>
</html>
```

Error message:

```
Warning: include(wrongFile.php) [function.include]:
failed to open stream:
No such file or directory in C:\home\website\test.php on line 5

Warning: include() [function.include]:
Failed opening 'wrongFile.php' for inclusion
(include_path='.:C:\php5\pear')
in C:\home\website\test.php on line 5

Hello World!
```

Notice that the echo statement is executed! This is because a Warning does not stop the script execution.

## Error Example require() Function

Now, let's run the same example with the require() function.

```
<html>
<body>

<?php
require("wrongFile.php");
echo "Hello World!";
```

```
?>  
</body>  
</html>
```

Error message:

```
Warning: require(wrongFile.php) [function.require]:  
failed to open stream:  
No such file or directory in C:\home\website\test.php on line 5  
  
Fatal error: require() [function.require]:  
Failed opening required 'wrongFile.php'  
(include_path='.:C:\php5\pear')  
in C:\home\website\test.php on line 5
```

The echo statement is not executed, because the script execution stopped after the fatal error.

It is recommended to use the require() function instead of include(), because scripts should not continue after an error.

Related posts:

1. [PHP Functions \(14.8\)](#)
2. [PHP Forms and User Input \(12.7\)](#)
3. [PHP \\$\\_POST Function \(8\)](#)
4. [PHP Looping – For Loops \(7.9\)](#)
5. [PHP If...Else Statements \(7.5\)](#)
6. [PHP \\$\\_GET Function \(7.1\)](#)
7. [PHP Switch Statement \(7.1\)](#)
8. [PHP Looping – While Loops \(6.9\)](#)
9. [FSJ \(File Split and Join\) \(6.3\)](#)
10. [PHP Date\(\) Function \(5.5\)](#)