19. PHP Include Files

19.1. PHP include and require Statements

In PHP, you can insert the content of one PHP file into another PHP file before the server executes it.

The include and require statements are used to insert useful codes written in other files, in the flow of execution.

Include and require are identical, except upon failure:

- require will produce a fatal error (E_COMPILE_ERROR) and stop the script
- include will only produce a warning (E_WARNING) and the script will continue

So, if you want the execution to go on and show users the output, even if the include file is missing, use include. Otherwise, in case of FrameWork, CMS or a complex PHP application coding, always use require to include a key file to the flow of execution. This will help avoid compromising your application's security and integrity, just in-case one key file is accidentally missing.

Including files saves a lot of work. This means that you can create a standard header, footer, or menu file for all your web pages. Then, when the header needs to be updated, you can only update the header include file.

Syntax

```
include 'filename';

or

require 'filename';
```

19.2. PHP include and require Statement

Basic Example

Assume that you have a standard header file, called "header.php". To include the header file in a page, use include/require:

```
<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 that should be used on all pages.

"menu.php":

```
echo '<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 class="leftmenu">
<?php include 'menu.php'; ?>
</div>

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

</body>
</html>
```

Example 3

Assume we have an include file with some variables defined ("vars.php"):

```
<?php
$color='red';
$car='BMW';
?>
```

Then the variables can be used in the calling file:

```
<html>
<body>

<h1>Welcome to my home page.</h1>
<?php include 'vars.php';
echo "I have a $color $car"; // I have a red BMW
?>

</body>
</html>
```