PHP - File Inclusion

You can include the content of a PHP file into another PHP file before the server executes it. There are two PHP functions which can be used to included one PHP file into another PHP file.

- The include() Function
- The require() Function

This is a strong point of PHP which helps in creating functions, headers, footers, or elements that can be reused on multiple pages. This will help developers to make it easy to change the layout of complete website with minimal effort. If there is any change required then instead of changing thousand of files just change included file.

The include() Function

The include() function takes all the text in a specified file and copies it into the file that uses the include function. If there is any problem in loading a file then the **include()** function generates a warning but the script will continue execution.

Assume you want to create a common menu for your website. Then create a file menu.php with the following content.

```html
<a href="http://www.tutorialspoint.com/index.htm">Home</a>
<a href="http://www.tutorialspoint.com/ebxml">ebXML</a>
<a href="http://www.tutorialspoint.com/ajax">AJAX</a>
<a href="http://www.tutorialspoint.com/perl">PERL</a>
```

Now create as many pages as you like and include this file to create header. For example now your test.php file can have following content.

```php
<?php <b>include("menu.php");</b> ?>
<p>This is an example to show how to include PHP file!</p>
```

It will produce the following result −

Home -
ebXML -
AJAX -
PERL

This is an example to show how to include PHP file!

The require() Function

The require() function takes all the text in a specified file and copies it into the file that uses the include function. If there is any problem in loading a file then the **require()** function generates a fatal error and halt the execution of the script.

So there is no difference in require() and include() except they handle error conditions. It is recommended to use the require() function instead of include(), because scripts should not continue executing if files are missing or misnamed.

You can try using above example with require() function and it will generate same result. But if you will try following two examples where file does not exist then you will get different results.

```php
<?php include("xxmenu.php"); ?>
<p>This is an example to show how to include wrong PHP file!</p>
```

This will produce the following result −

This is an example to show how to include wrong PHP file!

Now lets try same example with require() function.

```php
<?php <b>require("xxmenu.php");</b> ?>
<p>This is an example to show how to include wrong PHP file!</p>
```

This time file execution halts and nothing is displayed.

NOTE − You may get plain warning messages or fatal error messages or nothing at all. This depends on your PHP Server configuration.