PHP - File Include

The **include statement** in PHP is similar to the **import statement** in Java or Python, and **#include directive** in C/C++. However, there is a slight difference in the way the include statement works in PHP.

The Java/Python import or #include in C/C++ only loads one or more language constructs such as the functions or classes defined in one file into the current file. In contrast, the include statement in PHP brings in everything in another file into the existing PHP script. It may be a PHP code, a text file, HTML markup, etc.

The "include" Statement in PHP

Here is a typical example of how the include statement works in PHP −

myfile.php

```
<?php
   # some PHP code
?>
```

test.php

```
<?php
   include 'myfile.php';
   # PHP script in test.php
?>
```

The include keyword in PHP is very handy, especially when you need to use the same PHP code (function or class) or HTML markup across multiple PHP scripts in a project. A case in point is the creation of a menu that should appear across all pages of a web application.

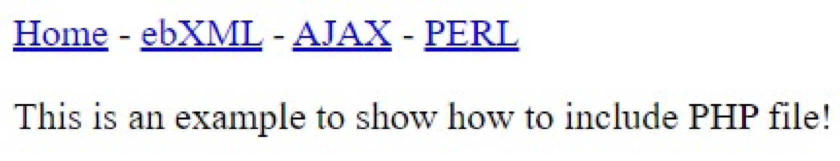
Suppose you want to create a common menu for your website. Then, create a file "menu.php" with the following content.

```
<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> <br />
```

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

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

Both the files are assumed to be present in the document root folder of the XAMPP server. Visit **http://localhost/test.php** URL. It will produce the following **output** –



```
Home - ebXML - AJAX - PERL

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

When PHP parser encounters the **include** keyword, it tries to find the specified file in the same directory from which the current script is being executed. If not found, the directories in the "include_path" setting of "php.ini" are searched.

When a file is included, the code it contains inherits the variable scope of the line on which the include occurs. Any variables available at that line in the calling file will be available within the called file, from that point forward. However, all functions and classes defined in the included file have the global scope.

Example

In the following example, we have a "myname.php" script with two variables declared in it. It is included in another script test.php. The variables are loaded in the global scope.

myname.php

```
<?php
   $color = 'green';
   $fruit = 'apple';
?>
```

test.php

```
<?php
   include "myname.php";
   echo "<h2>$fname $lname</h2>";
?>
```

When the browser visits **http://localhost/test.php**, it shows −

```
Ravi Teja
```

However, if the file is included inside a function, the variables are a part of the local scope of the function only.

myname.php

```
<?php
   $color = 'green';
   $fruit = 'apple';
?>
```

test.php

```
<?php
   function showname() {
      include "myname.php";
   }
   echo "<h2>$fname $lname</h2>";
?>
```

Now when the browser visits **http://localhost/test.php**, it shows undefined variable warnings −

```
Warning: Undefined variable $fname in C:\xampp\htdocs\test.php on line 7
Warning: Undefined variable $lname in C:\xampp\htdocs\test.php on line 7
```

include_once statement

Just like **include**, PHP also has the "**include_once**" keyword. The only difference is that if the code from a file has already been included, it will not be included again, and "include_once" returns true. As the name suggests, the file will be included just once.

"include_once" may be used in cases where the same file might be included and evaluated more than once during a particular execution of a script, so it can help avoid problems such as function redefinitions, variable value reassignments, etc.

PHP – Include vs Require

The **require** keyword in PHP is quite similar to the **include** keyword. The difference between the two is that, upon failure **require** will produce a fatal E_COMPILE_ERROR level error.

In other words, **require** will halt the script, whereas include only emits a warning (E_WARNING) which allows the script to continue.

require_once keyword

The "require_once" keyword is similar to **require** with a subtle difference. If you are using "require_once", then PHP will check if the file has already been included, and if so, then the same file it will not be included again.