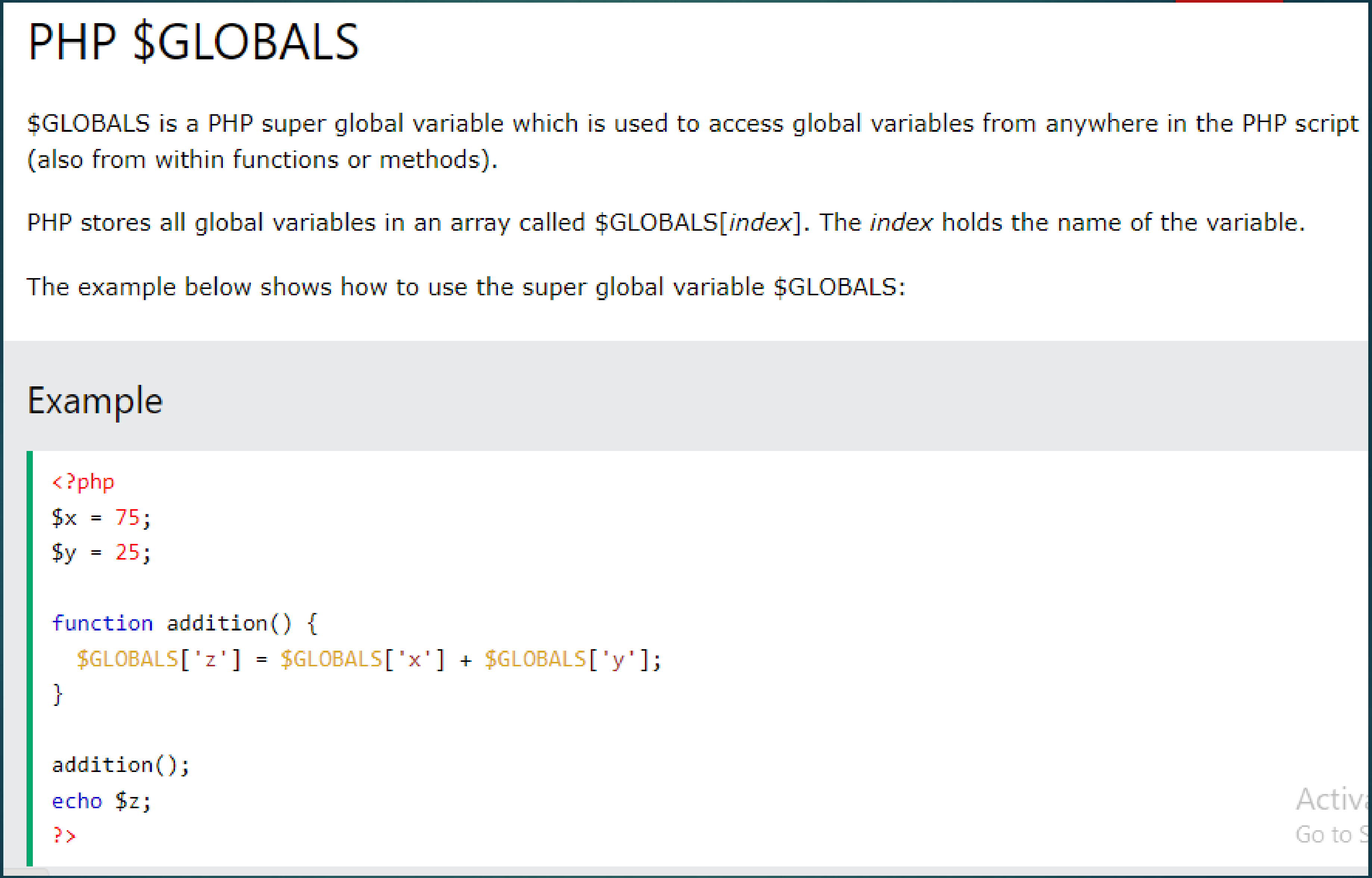
PHP- Superglobals

Super Global Variables

- $_GET
- $_POST
- $_REQUEST
- $_SERVER
- $_SESSION
- $_COOKIE
- $_FILES
- $GLOBALS



PHP $GLOBALS

$GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods).

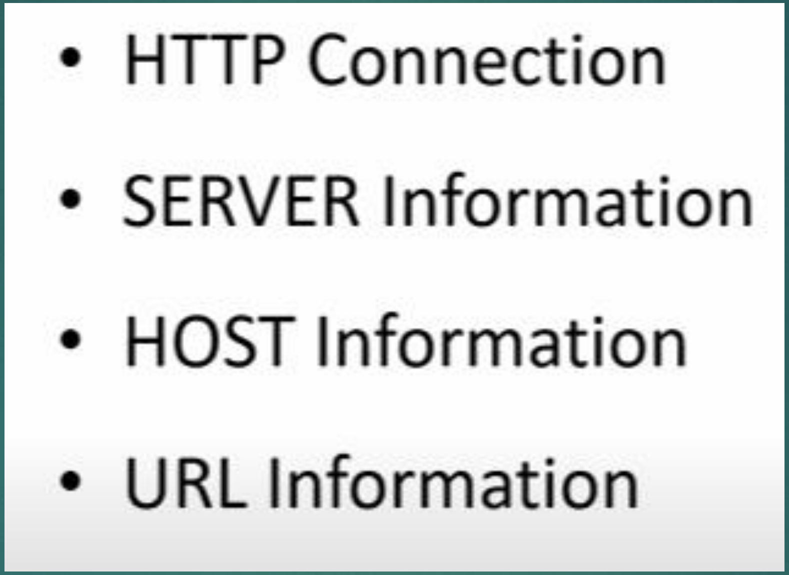
PHP stores all global variables in an array called $GLOBALS[*index*]. The *index* holds the name of the variable.

The example below shows how to use the super global variable $GLOBALS:

Example

```php
<?php
$x = 75;
$y = 25;

function addition() {
  $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}

addition();
echo $z;
?>
```

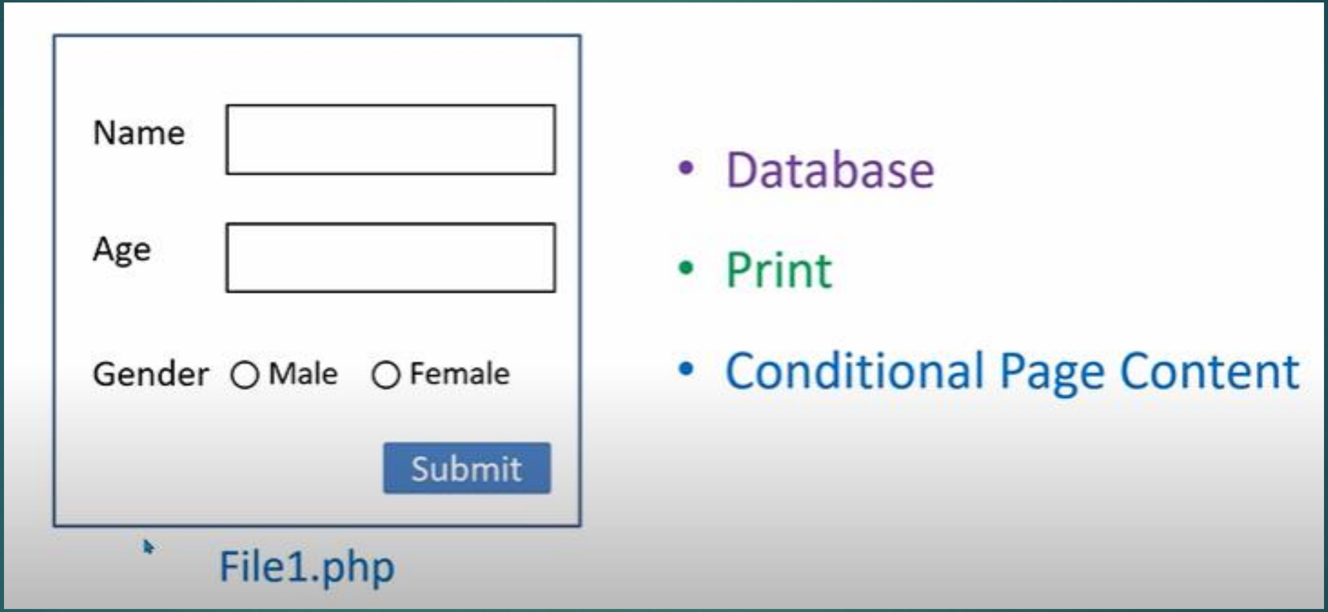
$_SERVER



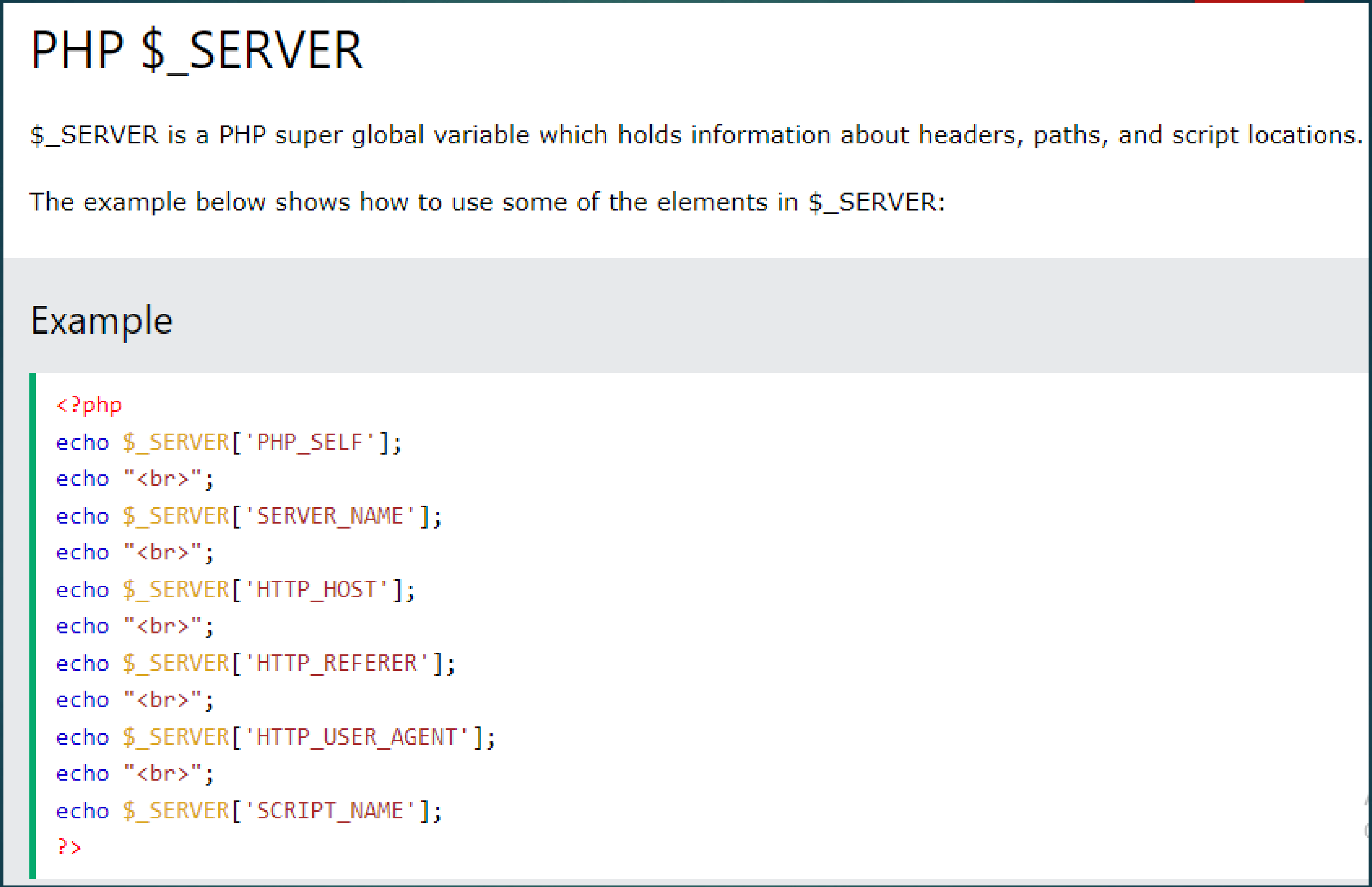
- HTTP Connection
- SERVER Information
- HOST Information
- URL Information

$_SERVER

Processing data on the same page



- Database
- Print
- Conditional Page Content



PHP $_SERVER

$_SERVER is a PHP super global variable which holds information about headers, paths, and script locations.

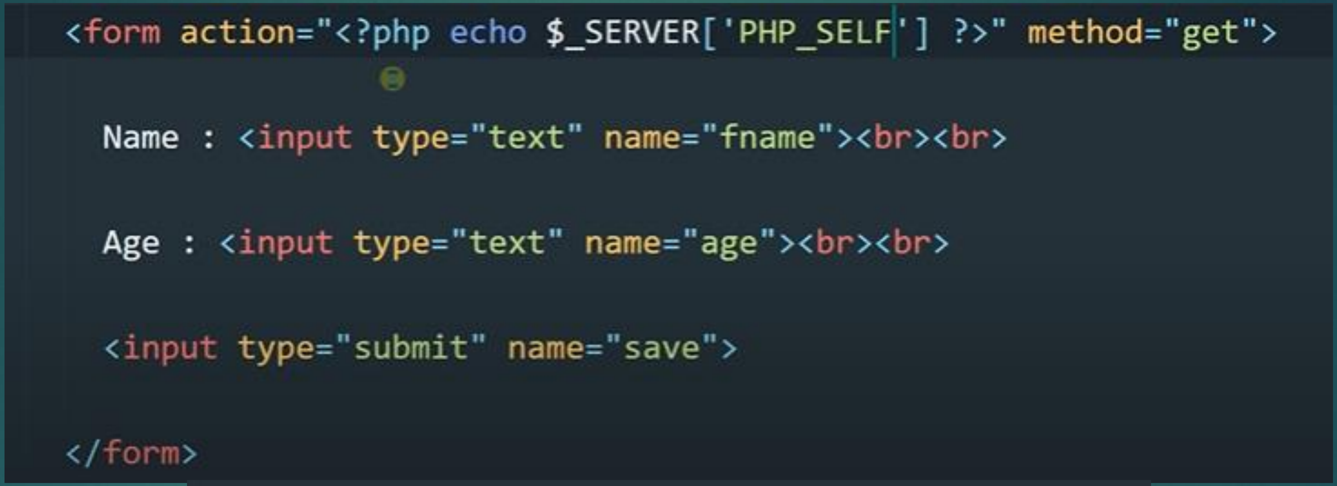
The example below shows how to use some of the elements in $_SERVER:

Example

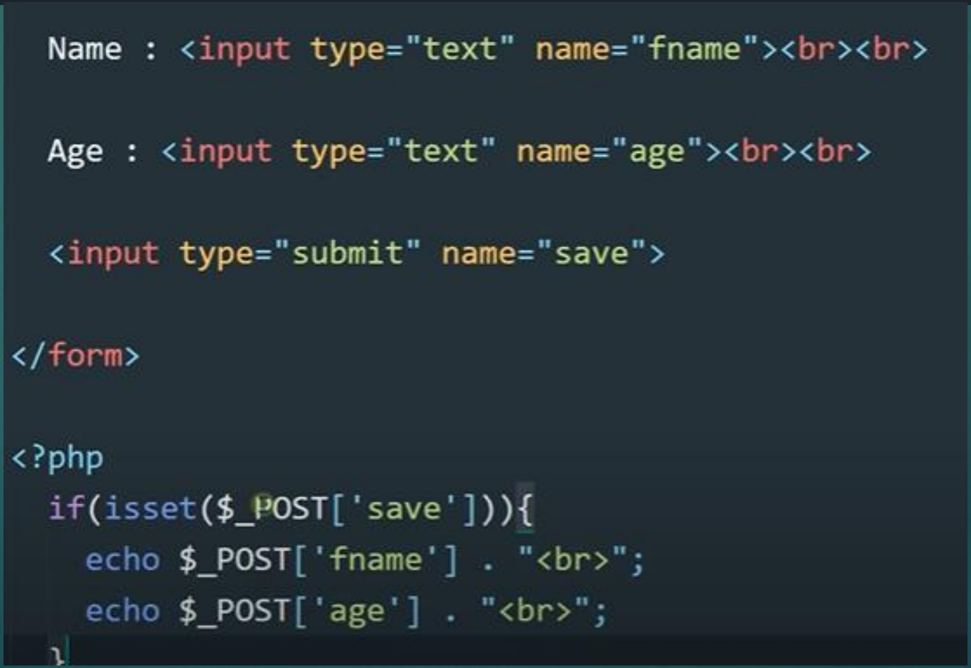
```php
<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>
```

$_SERVER

Processing data on the same page

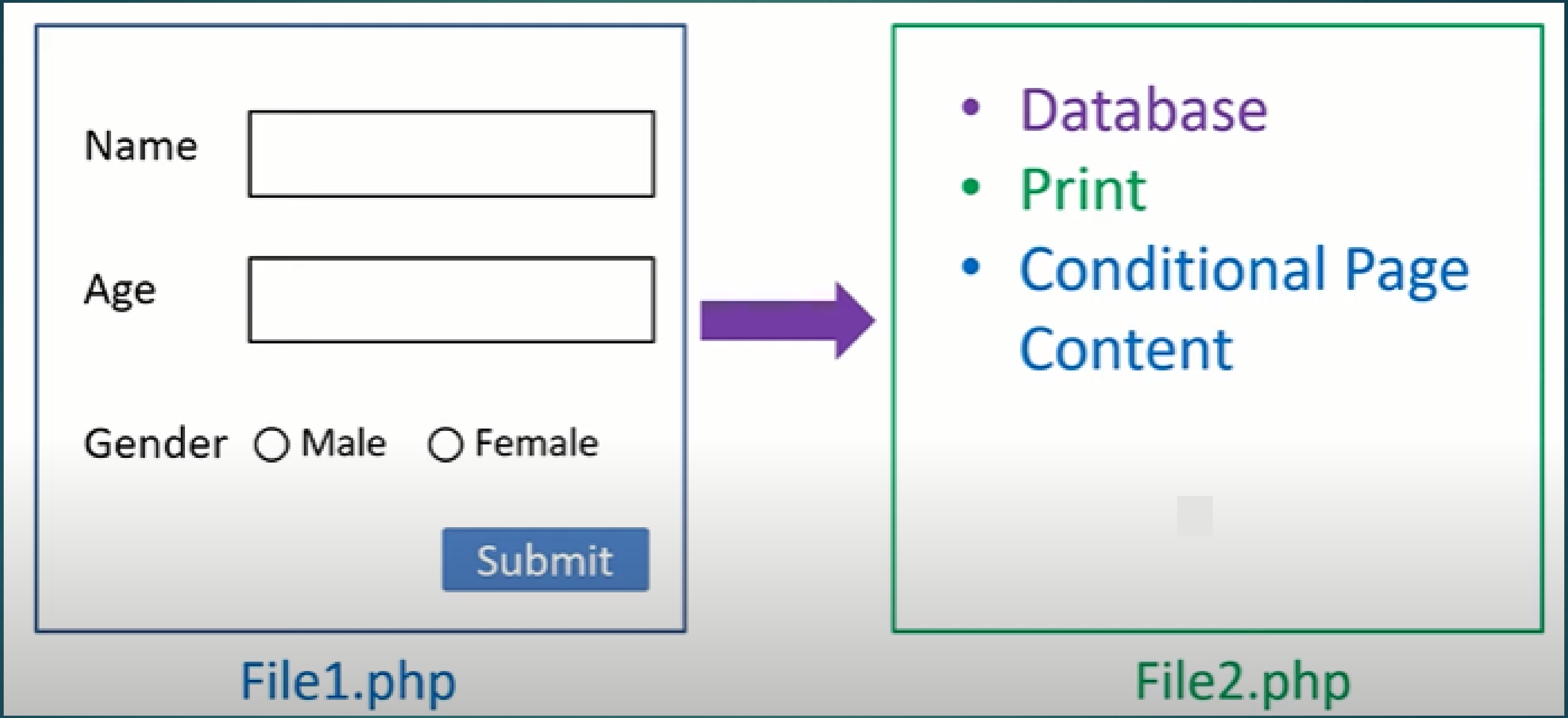


```
<form action="<?php echo $_SERVER['PHP_SELF'] ?>" method="get">

  Name : <input type="text" name="fname"><br><br>

  Age : <input type="text" name="age"><br><br>

  <input type="submit" name="save">

</form>
```



```
  Name : <input type="text" name="fname"><br><br>

  Age : <input type="text" name="age"><br><br>

  <input type="submit" name="save">

</form>

<?php
  if(isset($_POST['save'])){
    echo $_POST['fname'] . "<br>";
    echo $_POST['age'] . "<br>";
  }
```

$_GET AND $_POST

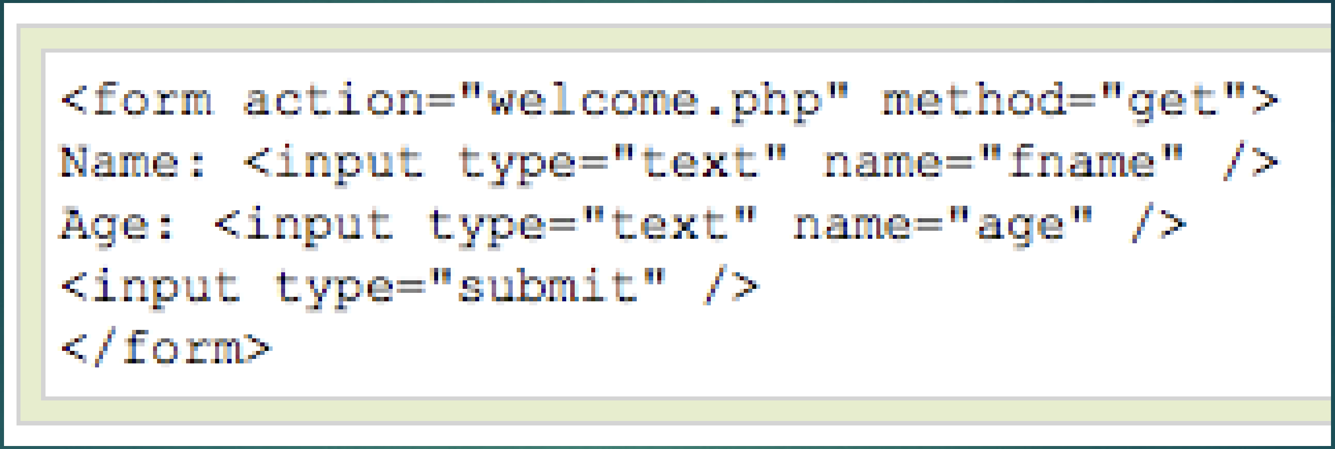


- Database
- Print
- Conditional Page Content

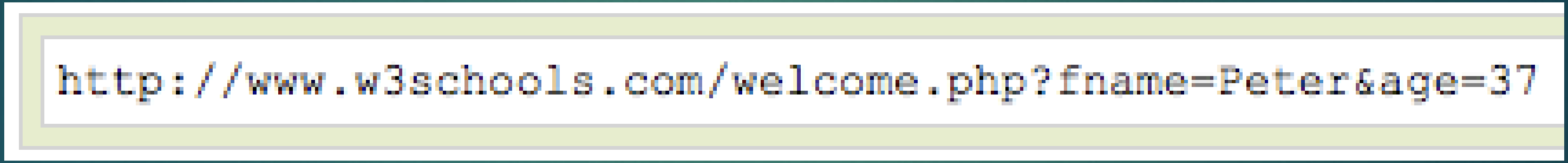
PHP Forms - $_GET Function

- The built-in $_GET function is used to collect values from a form sent with method="get".
- Information sent from a form with the GET method is visible to everyone (it will be displayed in the browser's address bar) and has limits on the amount of information to send (max. 100 characters).

PHP Forms - $_GET Function

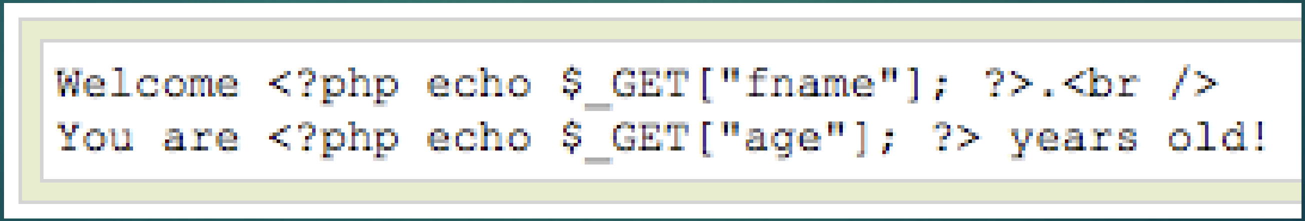


```
<form action="welcome.php" method="get">
Name: <input type="text" name="fname" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>
```



```
http://www.w3schools.com/welcome.php?fname=Peter&age=37
```

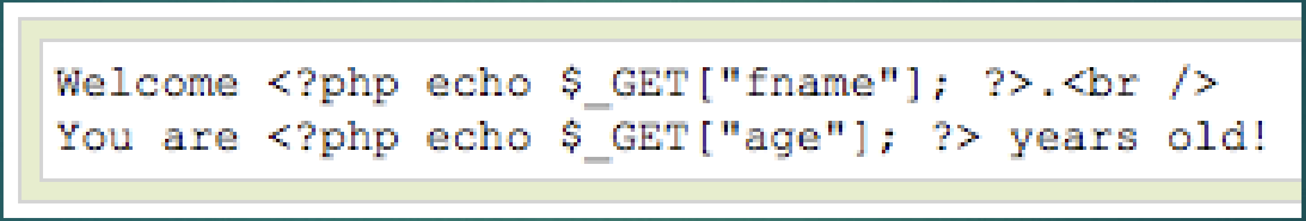
Notice how the URL carries the information after the file name.



```
Welcome <?php echo $_GET["fname"]; ?>.<br />
You are <?php echo $_GET["age"]; ?> years old!
```

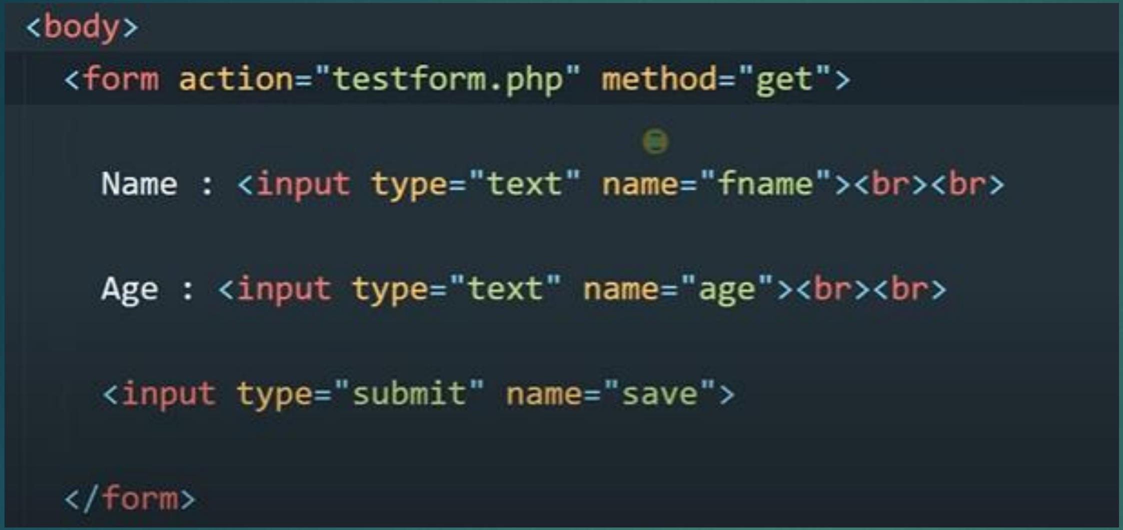
PHP Forms - $_GET Function

- The "welcome.php" file can now use the $_GET function to collect form data (the names of the form fields will automatically be the keys in the $_GET array)



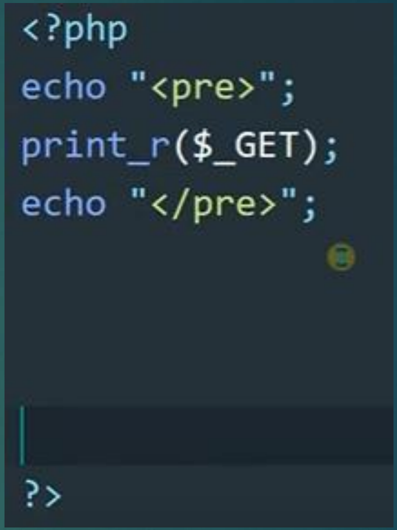
```
Welcome <?php echo $_GET["fname"]; ?>.<br />
You are <?php echo $_GET["age"]; ?> years old!
```

PHP Forms - $_GET Function



```
<body>
  <form action="testform.php" method="get">

    Name : <input type="text" name="fname"><br><br>

    Age : <input type="text" name="age"><br><br>

    <input type="submit" name="save">

  </form>
```

Form.php



```
<?php
echo "<pre>";
print_r($_GET);
echo "</pre>";



?>
```

Testform.php

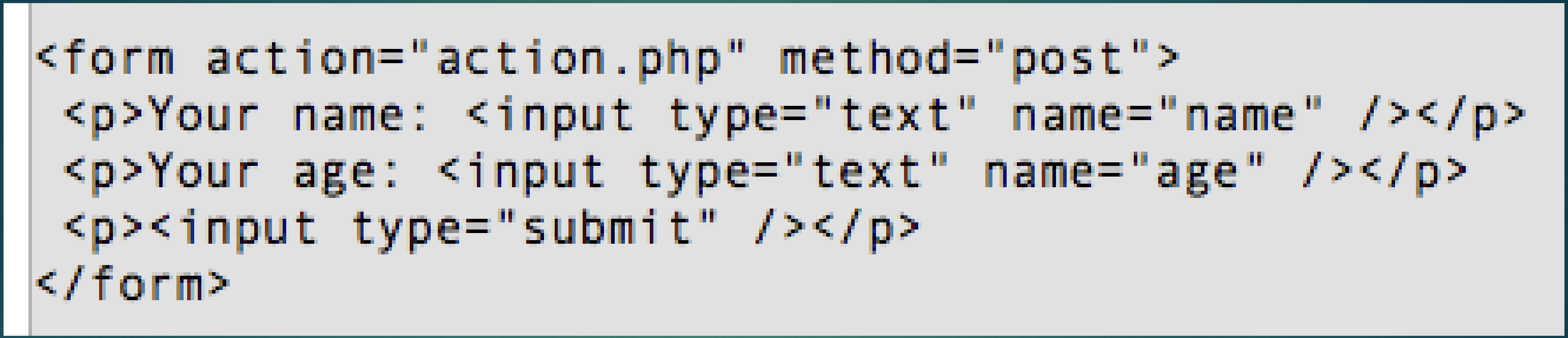
PHP Forms - $_GET Function

- When using method="get" in HTML forms, all variable names and values are displayed in the URL.
- This method should not be used when sending passwords or other sensitive information!
- However, because the variables are displayed in the URL, it is possible to bookmark the page. This can be useful in some cases.
- The get method is not suitable for large variable values; the value cannot exceed 100 chars.

PHP Forms - $_POST Function

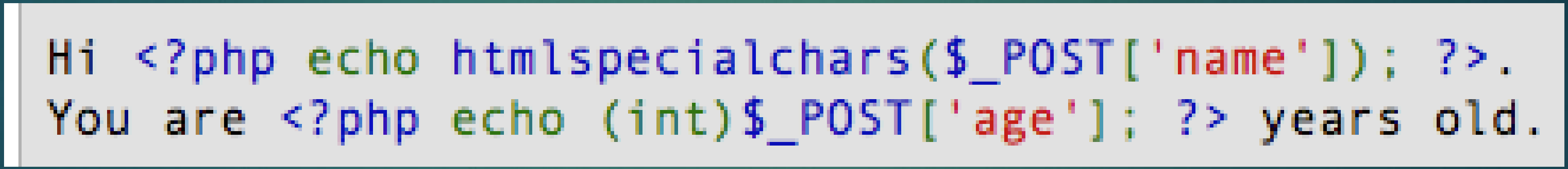
- The built-in $_POST function is used to collect values from a form sent with method="post".
- Information sent from a form with the POST method is invisible to others and has no limits on the amount of information to send.
- Note: However, there is an 8 Mb max size for the POST method, by default (can be changed by setting the post_max_size in the php.ini file).

PHP Forms - $_POST Function



```
<form action="action.php" method="post">
 <p>Your name: <input type="text" name="name" /></p>
 <p>Your age: <input type="text" name="age" /></p>
 <p><input type="submit" /></p>
</form>
```

And here is what the code of action.php might look like:

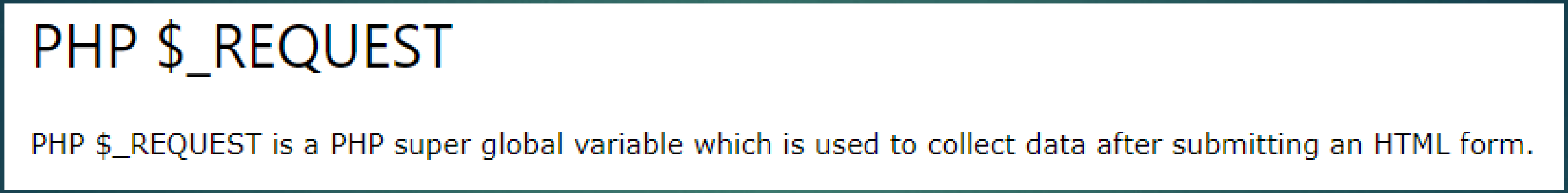


```
Hi <?php echo htmlspecialchars($_POST['name']); ?>.
You are <?php echo (int)$_POST['age']; ?> years old.
```

PHP Forms - $_POST Function

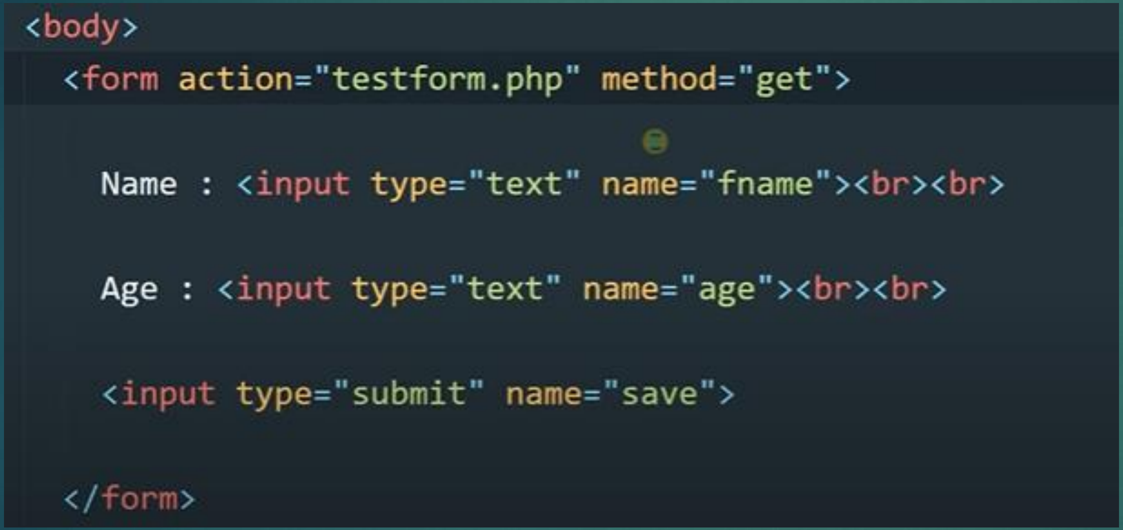
- When to use **method="post"**?
- Information sent from a form with the **POST** method is invisible to others and has no limits on the amount of information to send.
- However, because the variables are not displayed in the URL, it is not possible to bookmark the page.

PHP Forms - $_REQUEST



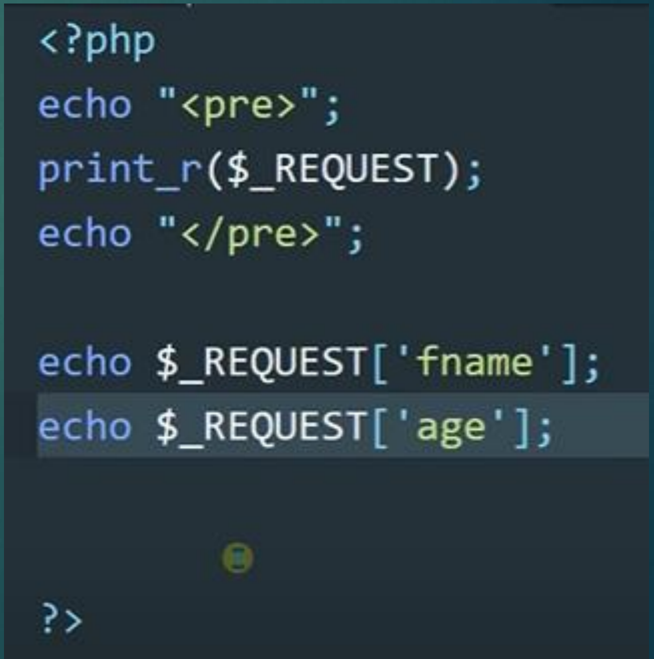
PHP $_REQUEST

PHP $_REQUEST is a PHP super global variable which is used to collect data after submitting an HTML form.



```
<body>
  <form action="testform.php" method="get">

    Name : <input type="text" name="fname"><br><br>

    Age  : <input type="text" name="age"><br><br>

    <input type="submit" name="save">

  </form>
```

Form.php



```
<?php
echo "<pre>";
print_r($_REQUEST);
echo "</pre>";

echo $_REQUEST['fname'];
echo $_REQUEST['age'];


?>
```

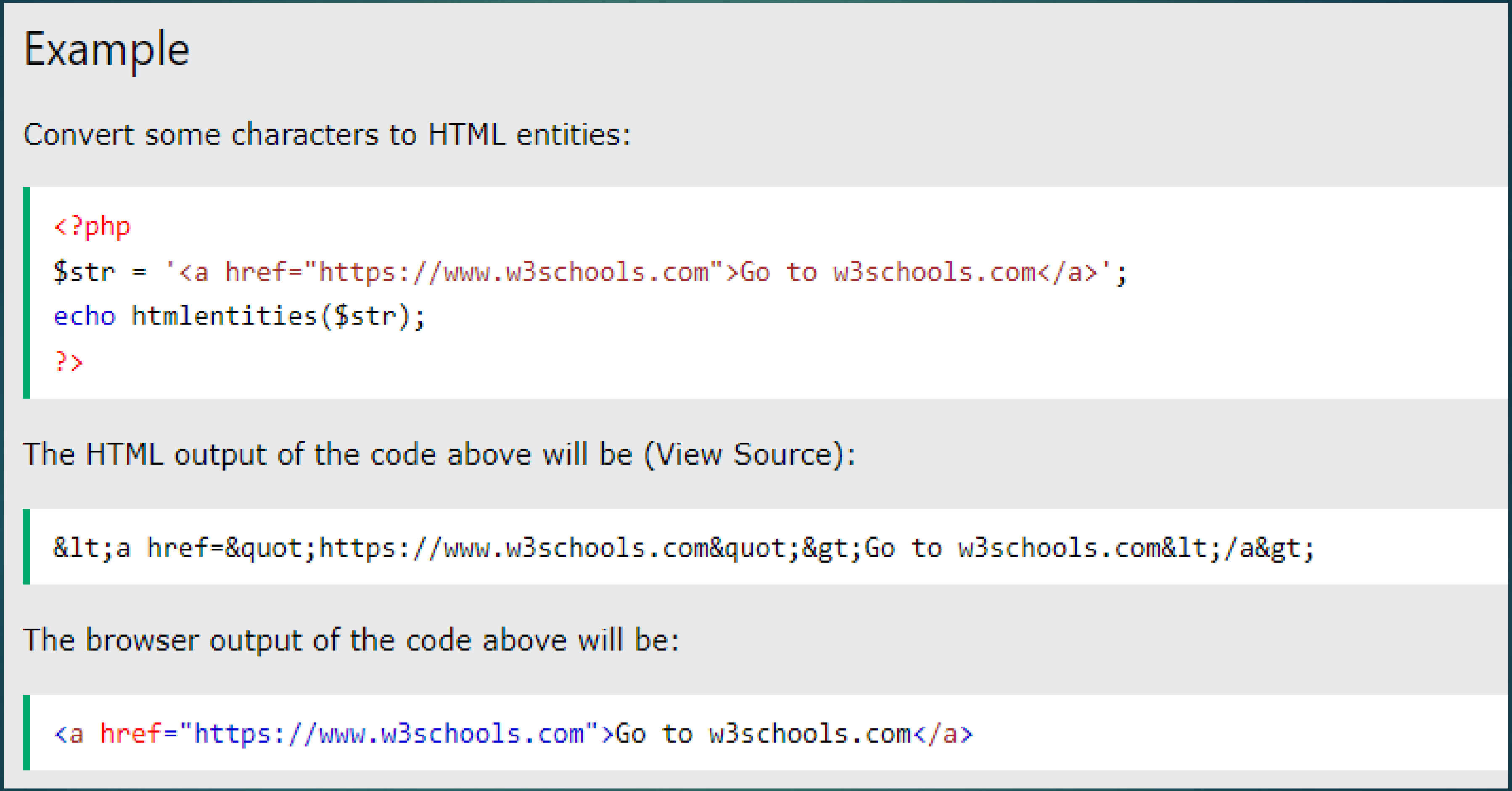
$_GET and $_POST Example

- https://www.javatpoint.com/get-and-post-methods-in-php

PHP- Special functions

- Htmlentities()
- Htmlspecialchars()

htmlentities()



Example

Convert some characters to HTML entities:

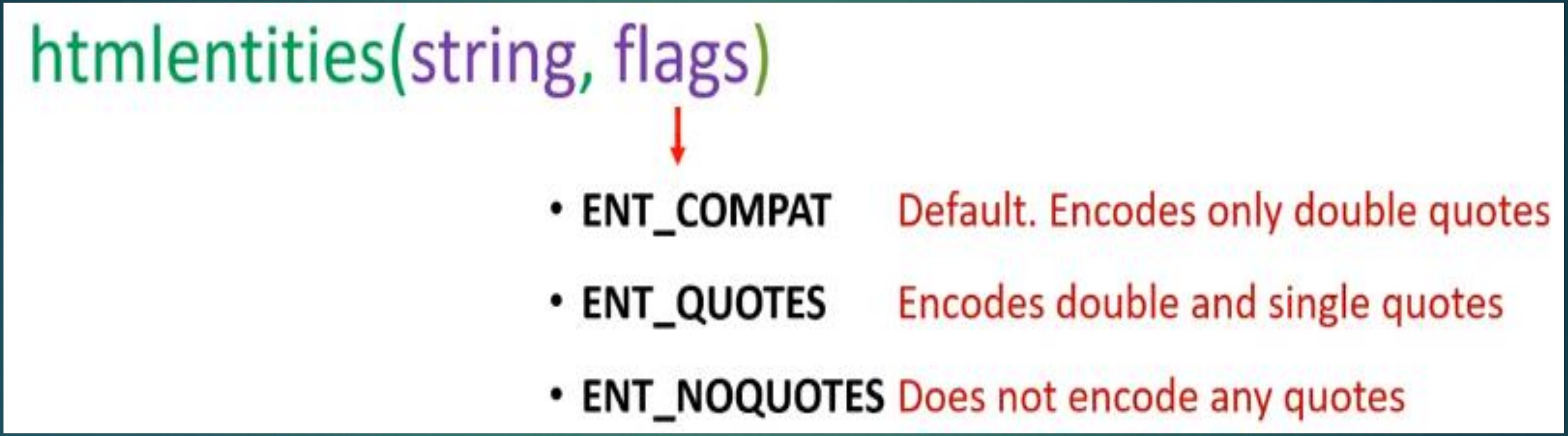
```php
<?php
$str = '<a href="https://www.w3schools.com">Go to w3schools.com</a>';
echo htmlentities($str);
?>
```

The HTML output of the code above will be (View Source):

```
&lt;a href=&quot;https://www.w3schools.com&quot;&gt;Go to w3schools.com&lt;/a&gt;
```

The browser output of the code above will be:

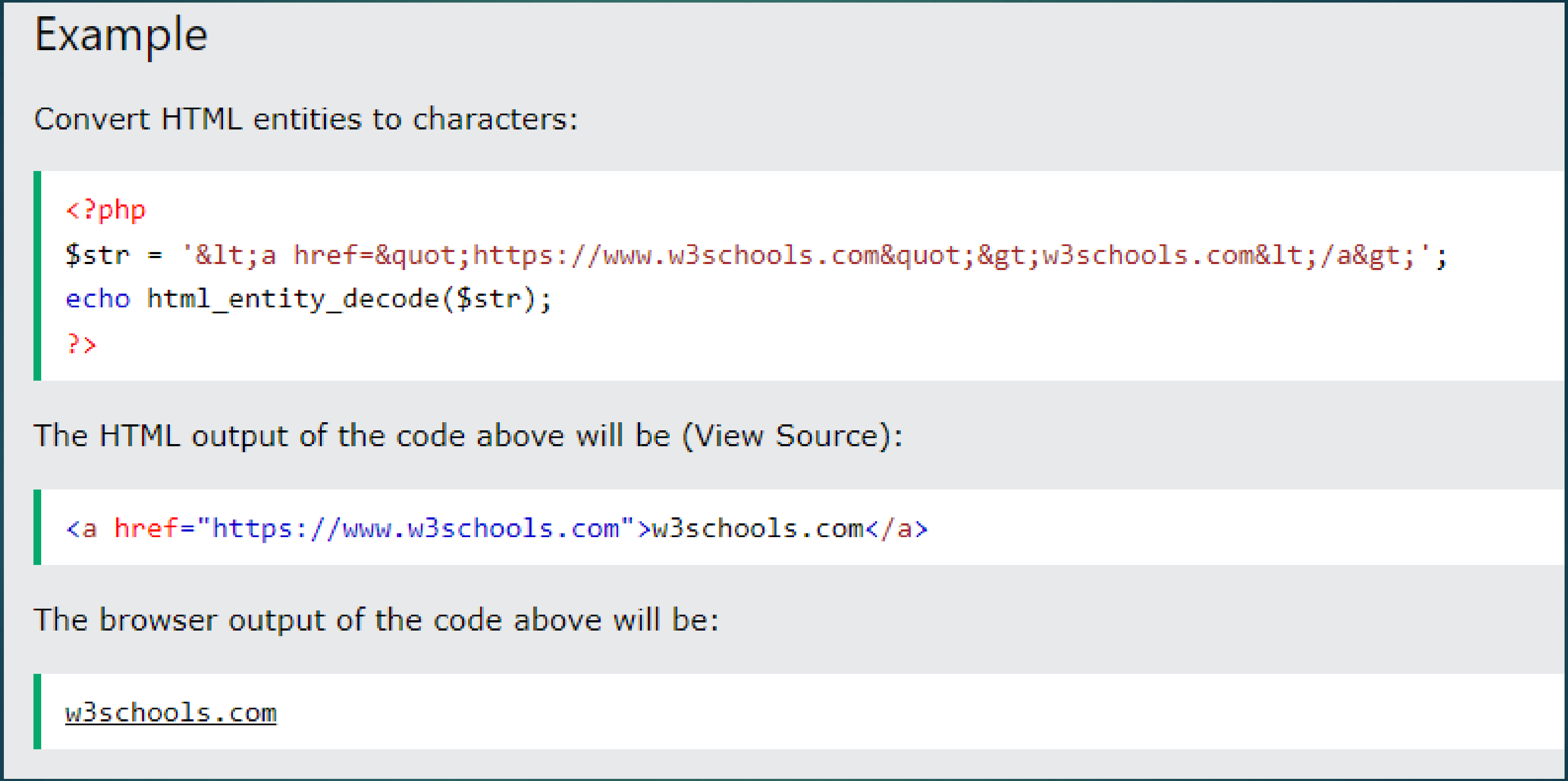
```
<a href="https://www.w3schools.com">Go to w3schools.com</a>
```



htmlentities(string, flags)

• ENT_COMPAT	Default. Encodes only double quotes
• ENT_QUOTES	Encodes double and single quotes
• ENT_NOQUOTES	Does not encode any quotes

html_entity_decode()



Example

Convert HTML entities to characters:

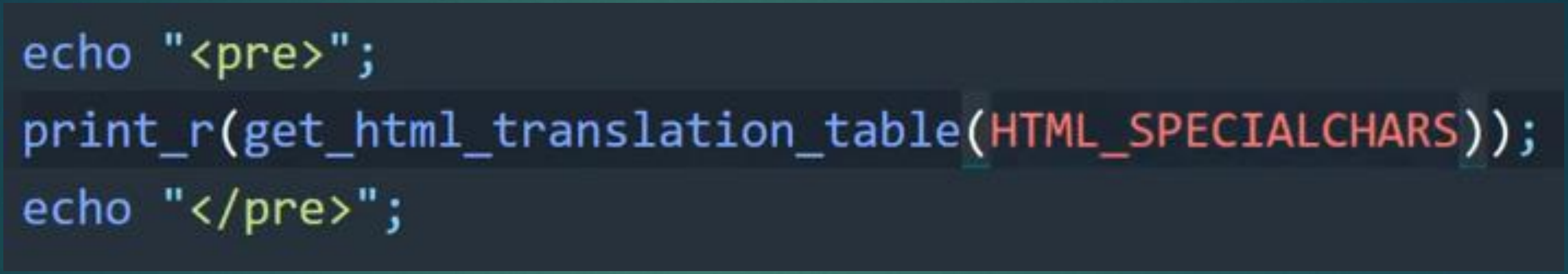
```
<?php
$str = '&lt;a href=&quot;https://www.w3schools.com&quot;&gt;w3schools.com&lt;/a&gt;';
echo html_entity_decode($str);
?>
```

The HTML output of the code above will be (View Source):

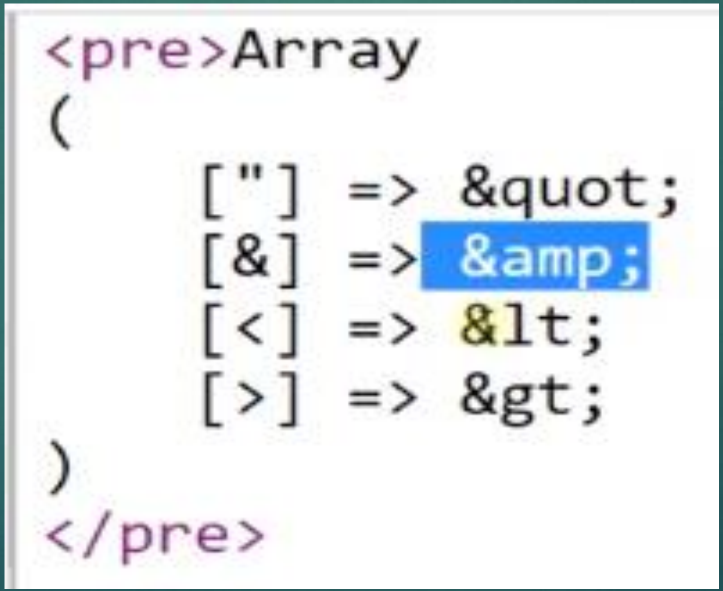
```
<a href="https://www.w3schools.com">w3schools.com</a>
```

The browser output of the code above will be:

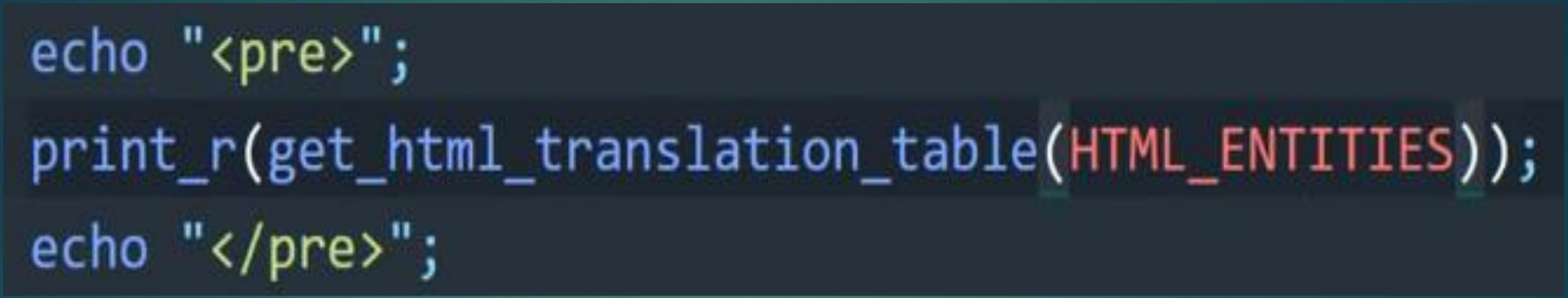
```
w3schools.com
```



```
echo "<pre>";
print_r(get_html_translation_table(HTML_SPECIALCHARS));
echo "</pre>";
```



```
<pre>Array
(
    ["] => &quot;
    [&] => &amp;
    [<] => &lt;
    [>] => &gt;
)
</pre>
```



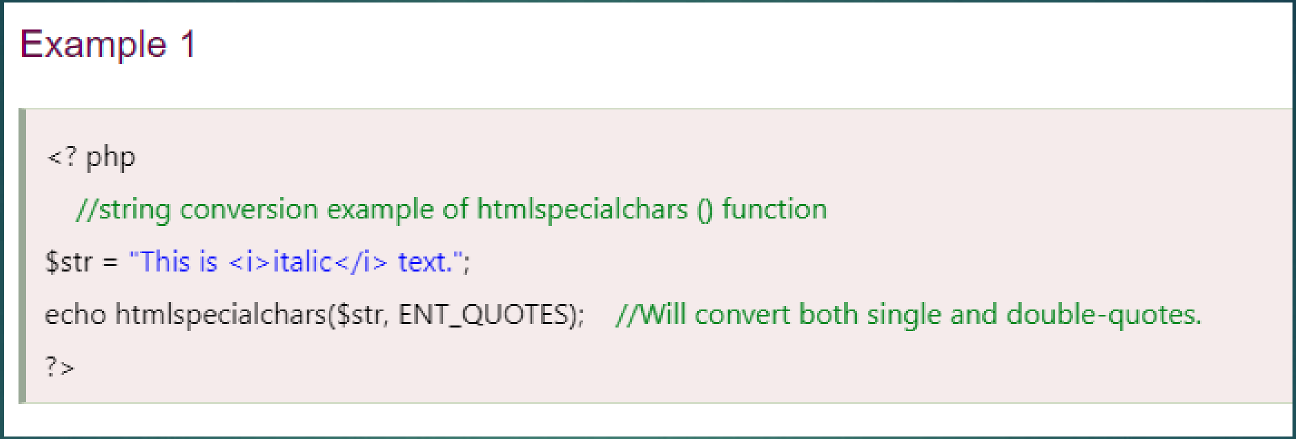
```
echo "<pre>";
print_r(get_html_translation_table(HTML_ENTITIES));
echo "</pre>";
```

Considers all special characters.

htmlspecialchars()

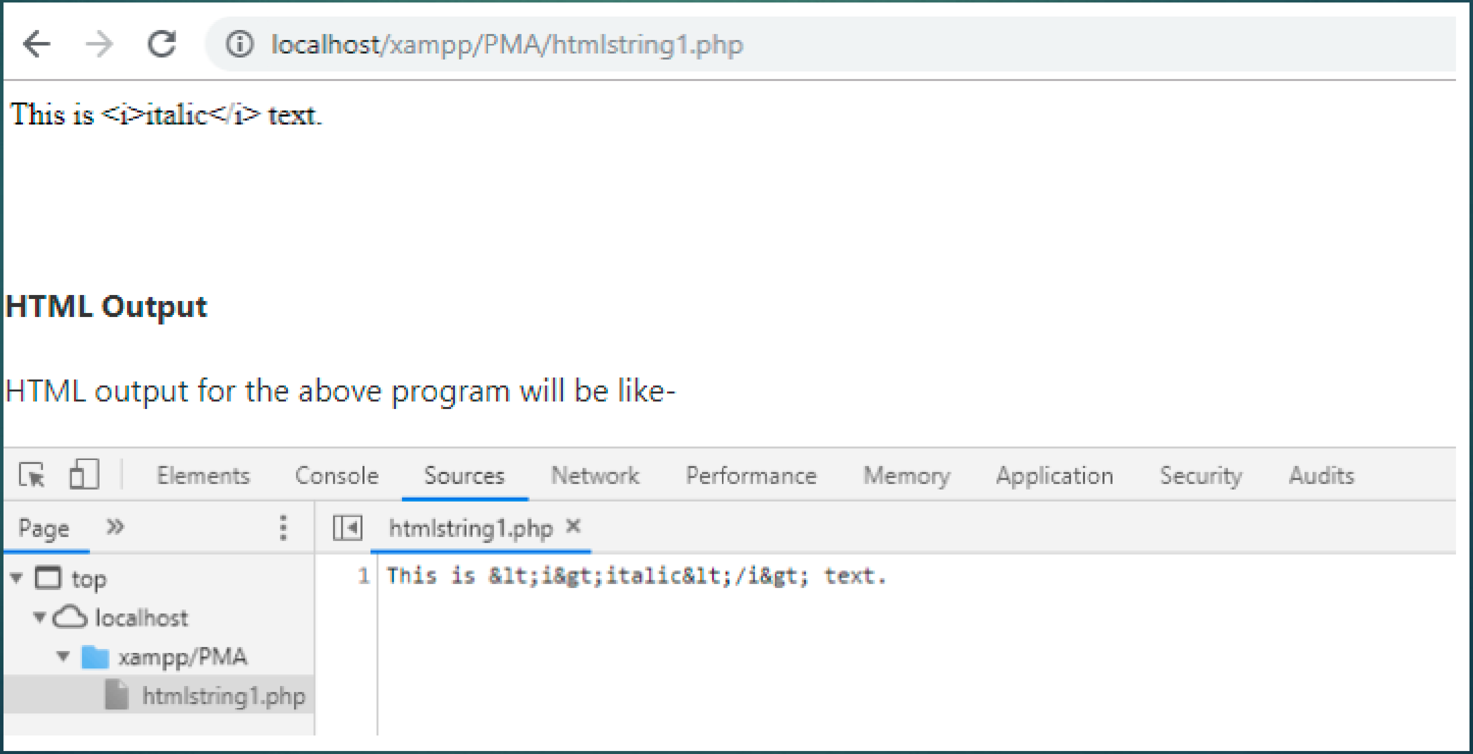
- makes sure any characters that are special in html are properly encoded so people can't inject HTML tags or Javascript into your page.
- It is the in-built function of PHP, which converts all pre-defined characters to the HTML entities. The pre-defined characters are:
  - & (ampersand) converted as &amp;
  - " (double quote) converted as &quot;
  - ' (single quote) converted as &#039;
  - < (less than) converted as &lt;
  - > (greater than) converted as &gt;

htmlspecialchars()



Example 1

```
<? php
   //string conversion example of htmlspecialchars () function
$str = "This is <i>italic</i> text.";
echo htmlspecialchars($str, ENT_QUOTES);    //Will convert both single and double-quotes.
?>
```



```
This is <i>italic</i> text.
```

HTML Output

HTML output for the above program will be like-

```
This is &lt;i&gt;italic&lt;/i&gt; text.
```

Functions for form validation

trim()

- Removal of spaces

Stripslashes()

- Remove backslashes (\) from the user input data

preg_match()

- check if the name field only contains letters, dashes, apostrophes and whitespaces.

PHP Forms – special Functions

filter_var()-

to check whether an email address is well-formed

Form validation example

https://www.w3schools.com/php/php_form_validation.asp

https://www.w3schools.com/php/php_form_complete.asp