

# Web Programming

Php





# References

- PHP 8 Basics, 2020, Springer
  - [https://link.springer.com/chapter/10.1007/978-1-4842-8082-9\\_2](https://link.springer.com/chapter/10.1007/978-1-4842-8082-9_2)
- The Absolute Beginner's Guide to HTML and CSS, 2023, Springer
  - [https://link.springer.com/chapter/10.1007/978-1-4842-9250-1\\_7](https://link.springer.com/chapter/10.1007/978-1-4842-9250-1_7)
- W3C Tutorial
  - <https://www.w3schools.com/php>
  - <https://www.w3schools.com/html>
  - <https://www.w3schools.com/js>
- Additional Topics
  - JQuery: <https://www.w3schools.com/jquery>
  - Bootstrap 5.0: <https://www.w3schools.com/bootstrap5>
  - Laravel/Blade Framework 11.0: <https://www.w3schools.in/laravel>





# PhP

## PHP Global Variables - Superglobals

Superglobals are built-in variables that are always available in all scopes.

- `$GLOBALS`
- `$_SERVER`
- `$_REQUEST`
- `$_POST`
- `$_GET`



- \$GLOBALS

Global variables are variables that can be accessed from any scope.

Variables of the outer most scope are automatically global variables, and can be used by any scope, e.g. inside a function.

To use a global variable inside a function you have to either define them as global with the global keyword, or refer to them by using the \$GLOBALS syntax.





# PHP

- \$GLOBALS

```
$x = 75;
```

```
function myfunction() {  
    echo $GLOBALS['x'];  
}
```

```
myfunction()
```

```
1  <?php  
2  $v=15;  
3  $x = 75;  
4  
5  function myfunction() {  
6  
7      global $v;  
8      echo $GLOBALS['x']+$v;  
9  }  
10  
11  $x=25;  
12  $v=30;  
13  myfunction();  
14  
15  ?>
```





# PhP

Create a global variable from inside a function, and use it outside of the function:

```
function myfunction() {  
    $GLOBALS["x"] = 100;  
}
```

```
myfunction();
```

```
echo $GLOBALS["x"];
```

```
echo $x;
```





# PHP

Create a global variable from inside a function, and use it outside of the function:

```
1  <?php
2  $v=15;
3  $x = 75;
4  function myfunction() {
5      global $v;
6      echo $GLOBALS['x']+$v ;
7  }
8  $x=25;
9  $v=30;
10 myfunction();
11 echo "\n" ;
12 function myfunction1() {
13     $GLOBALS["y"] = 100;
14 }
15 myfunction1();
16 echo $GLOBALS["y"]."\n";
17 echo $y
18 ?>
```





# PhP

## PHP - \$\_SERVER

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

```
/demo/demo_global_server.php
35.194.26.41
35.194.26.41
https://tryphp.w3schools.com/showphp.php?filename=demo_global_
server
Mozilla/5.0 (Linux; Android 10; TECNO KD7h)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.99
Mobile Safari/537.36
/demo/demo_global_server.php
```

```
<!DOCTYPE html>
<html>
<body>
```

```
<?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'];
?>
```

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







# Php

PHP - \$\_SERVER

/demo/demo\_global\_server.php

35.194.26.41

35.194.26.41

[https://tryphp.w3schools.com/showphp.php?filename=demo\\_global\\_server](https://tryphp.w3schools.com/showphp.php?filename=demo_global_server)

Mozilla/5.0 (Linux; Android 10; TECNO KD7h) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.99 Mobile Safari/537.36

/demo/demo\_global\_server.php

[https://www.w3schools.com/php/php\\_superglobals\\_server.asp](https://www.w3schools.com/php/php_superglobals_server.asp)





# PhP

- The other superglobals used in form handling for a better understanding.

- `$_REQUEST`
- `$_POST`
- `$_GET`

[https://www.youtube.com/playlist?list=PLrCBCI\\_Do84jHBv\\_mmGfjOwbAn7xgjKEA](https://www.youtube.com/playlist?list=PLrCBCI_Do84jHBv_mmGfjOwbAn7xgjKEA)





- **\$\_GET**      \$\_GET contains an array of variables received via the HTTP GET method. There are two main ways to send variables via the HTTP GET method:
  - Query strings in the URL
  - HTML Forms

## Query string in the URL

A query string is data added at the end of a URL. In the link below, everything after the **?** sign is part of the query string:

```
<a href="demo_phpfile.php?subject=PHP&web=W3schools.com">Test $GET</a>
```

The query string above contains two key/value pairs:

```
subject=PHP  
web=W3schools.com
```

```
<?php echo "Study " . $_GET['subject'] . " at " . $_GET['web']; ?>
```





# PhP

## \$\_GET in HTML Forms

A HTML form submits information via the HTTP GET method if the form's `method` attribute is set to `"GET"`.

```
<html>
<body>

<form action="welcome_get.php" method="GET">
  Name: <input type="text" name="name">
  E-mail: <input type="text" name="email">
  <input type="submit">
</form>

</body>
</html>
```



Name:

E-mail:





# PHP

When a user clicks the submit button, the form data is sent to a PHP file specified in the `action` attribute of the `<form>` tag.

The form fields are sent to the PHP file, with your input, as query strings:

```
welcome_get.php?name=John&email=john@example.com
```

In the action file we can use the `$_GET` variable to collect the value of the input fields.

PHP code inside the `welcome_get.php` page:

```
<html>
```

```
<body>
```

```
Welcome <?php echo $_GET["name"]; ?><br>
```

```
Your email address is: <?php echo $_GET["email"]; ?>
```

```
</body>
```

```
</html>
```



# PhP

Name:

E-mail:

Welcome Elashry

Your email address is: elashry@gmail.com



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Super Globals</title>
  </head>
  <body>

    <form action='variable.php' method='get'>

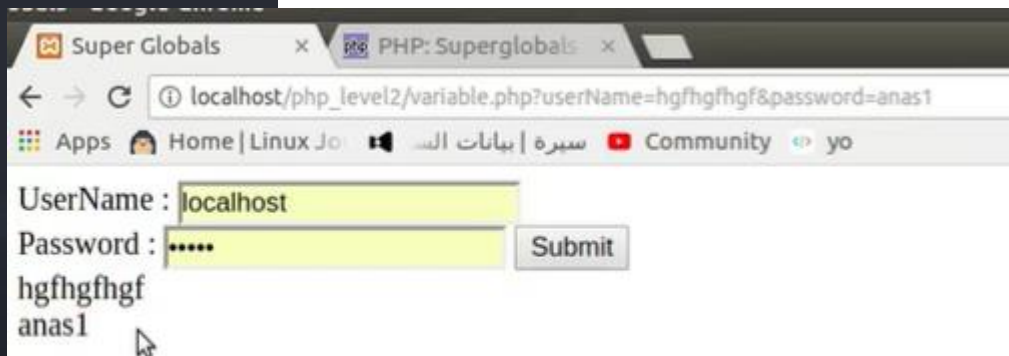
      UserName : <input type='text' name='userName' ><br/>
      Password : <input type='password' name="password" >
      <input type='submit'>

    </form>

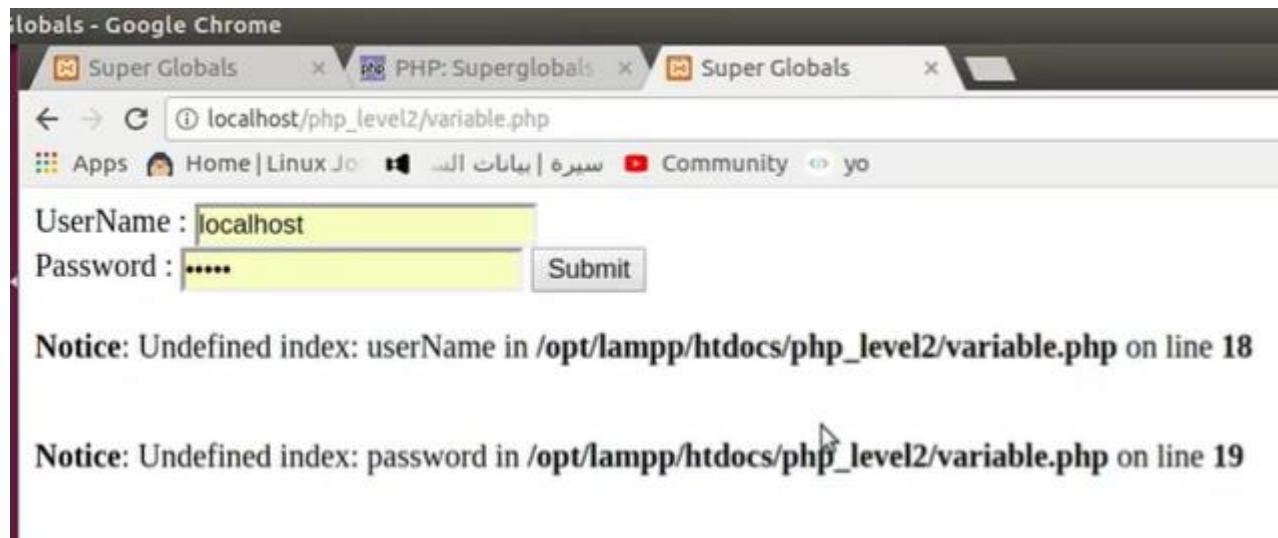
    <?php
    echo $_GET["userName"]. "<br/>";
    echo $_GET["password"];

    ?>

  </body>
</html>
```



# PHP



Why?





```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Super Globals</title>
  </head>
  <body>

    <form action='variable.php' method='get'>

      UserName : <input type='text' name='userName' ><br/>
      Password : <input type='password' name="password" >
      <input type='submit'>

    </form>

    <?php
    if(isset($_GET["userName"]) && isset($_GET["password"])){
      echo  $_GET["userName"]."<br/>";
      echo  $_GET["password"];

    }

  ?>
```





# PHP

## PHP - \$\_POST

\$\_POST contains an array of variables received via the HTTP POST method.

- HTML forms

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Super Globals</title>
  </head>
  <body>

    <form action='variable.php' method='post'>

      UserName : <input type='text' name='userName' ><br/>
      Password : <input type='password' name="password" >
      <input type='submit' name="btn">
        I
    </form>

    <?php
    if(isset($_POST["btn"])){
      echo  $_POST["userName"]. "<br/>";
      echo  $_POST["password"];

    }
```





# PhP

Send to another page

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Super Globals</title>
  </head>
  <body>

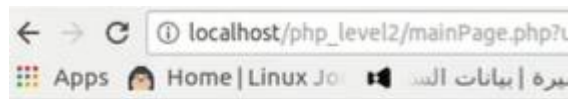
    <form action='mainPage.php' method='post'>

      UserName : <input type='text' name='userName' ><br/>
      Password : <input type='password' name="password" >
      <input type='submit' name="btn">

    </form>

  </body>
</html>
```

```
<?php
echo "My Name is " . $_POST["userName"]."<br/>";
echo "My Pass is " . $_POST["password"];
```



My Name is anas  
My Pass is 123





# PHP

`$_REQUEST` is a PHP super global variable which contains submitted form data, and all cookie data. In other words, `$_REQUEST` is an array containing data from `$_GET`, `$_POST`

```
<!DOCTYPE html>
<html>
<body>
```

```
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>
```

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // collect value of input field
    $name = htmlspecialchars($_REQUEST['fname']);
    if (empty($name)) {
        echo "Name is empty";
    } else {
        echo $name;
    }
}
?>
```

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

Name:

Name is empty

Name:

hany

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Super Globals</title>
  </head>
  <body>
    <a href="mainPage.php?myname=anas&myage=22" >Go To main page </a>

    <form action='variable.php' method='post'>

      UserName : <input type='text' name='userName' ><br/>
      Password : <input type='password' name="password" >
      <input type='submit' name="btn">

    </form>
    <?php
    echo $_SERVER["REQUEST_METHOD"]."<br/>";
    if(isset($_POST["btn"])){
    echo $_REQUEST['userName'];
    |
    }
    ?>

  </body>
```





# PHP

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Super Globals</title>
  </head>
  <body>

    <form action='mainPage.php' method='post'>

      UserName : <input type='text' name='userName' ><br/>
      Password : <input type='password' name="password"><br/>
      Languages : <br/>
      <input type="checkbox" name='lang[]' value='c++'>C++
      <input type="checkbox" name='lang[]' value='java'>java
      <input type="checkbox" name='lang[]' value='php'>PHP
      <input type="checkbox" name='lang[]' value='swift'>Swift
    <br/><br/>
      Gender : <br>
      <input type="radio" name='gender' value='female'>Female
      <input type="radio" name='gender' value='male'>male

      <input type='submit' name="btn">

    </form>
```



```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Super Globals</title>
  </head>
  <body>

    <form action='mainPage.php' method='post'>

      UserName : <input type='text' name='userName' ><br/>
      Password : <input type='password' name="password"><br/>
      Languages : <br/>
      <input type="checkbox" name='lang[]' value='c++'>C++
      <input type="checkbox" name='lang[]' value='java'>java
      <input type="checkbox" name='lang[]' value='php'>PHP
      <input type="checkbox" name='lang[]' value='swift'>Swift
    <br/><br/>
    Gender : <br>
    <input type="radio" name='gender' value='female'>Female
    <input type="radio" name='gender' value='male'>male

    <input type='submit' name="btn">

  </form>

```

Super Globals x PHP: Superglobals x

localhost/php\_level2/variable.php

Apps Home | Linux Jo سيرة | بيانات الش Community yo

UserName : localhost

Password : .....

Languages :

☐ C++ ☐ java ☐ PHP ☐ Swift

Gender :

☐ Female ☐ male



```

<form action='mainPage.php' method='post'>

  UserName : <input type='text' name='userName' ><br/>
  Password : <input type='password' name="password"><br/>
  Languages : <br/>
<input type="checkbox" name='lang[]' value='c++'>C++
<input type="checkbox" name='lang[]' value='java'>java
<input type="checkbox" name='lang[]' value='php'>PHP
<input type="checkbox" name='lang[]' value='swift'>Swift
<br/><br/>
Gender : <br>
<input type="radio" name='gender' value='female'>Female
<input type="radio" name='gender' value='male'>male
: <br/>Years :
<select name='years'>
  <option value='1999'>1999</option>
  <option value='1888'>1888</option>
  <option value='1777'>1777</option>

</select>
<br/>
color : <input type="color" name='color'>

```

lobals - Google Chrome

Super Globals

PHP: Superglobals

localhost/php\_level2/variable.php

Apps Home | Linux Jo سيرة | بيانات الب Community yo

UserName : localhost

Password : .....

Languages :

☐ C++ ☐ java ☐ PHP ☐ Swift

Gender :

☐ Female ☐ male :

Years : 1999 ▼

color :

Submit







# PhP

```
variable.php  mainPage.php

<?php

$urName = $_POST["userName"];
$pass = $_POST["password"];
$gender = $_POST["gender"];
$year = $_POST["years"];
$color = $_POST["color"];

echo "Ur Name is ". $urName."<br/>";
echo "Ur pass is ". $pass."<br/>";
echo "Ur gender is ". $gender."<br/>";
echo " year is ". $year."<br/>";
echo "Ur color is ". $color."<br/>";

?>
```

localhost/php\_level2/mainPage.php

Apps Home | Linux Jo سيرة | بيانات الش Community yo

Ur Name is anas  
Ur pass is 123  
Ur gender is male  
year is 1888  
Ur color is #db5151



variable.php

mainPage.php

```
<?php
```

```
$urName = $_GET["userName"];  
$pass = $_GET["password"];  
$gender = $_GET["gender"];  
$year = $_GET["years"];  
$color = $_GET["color"];  
$lang = $_GET["lang"];  
echo "Ur Name is ". $urName."<br/>";  
echo "Ur pass is ". $pass."<br/>";  
echo "Ur gender is ". $gender."<br/>";  
echo " year is ". $year."<br/>";  
echo "Ur color is ". $color."<br/>";
```

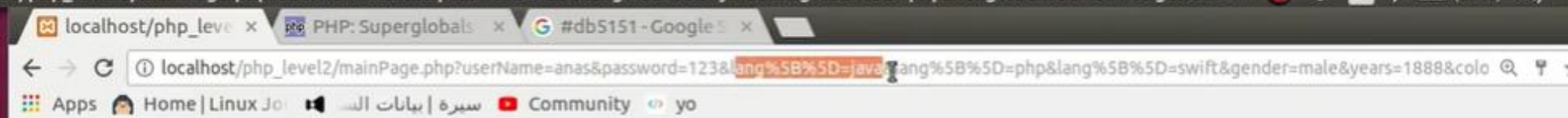
```
foreach($lang as $myLang){  
    echo $myLang I " ";  
}  
?>
```



```
variable.php | mainPage.php |
<?php

$urName = $_GET["userName"];
$pass = $_GET["password"];
$gender = $_GET["gender"];
$year = $_GET["years"];
$color = $_GET["color"];
$lang = $_GET["lang"];
```

st/php\_level2/mainPage.php?userName=anas&password=123&lang%5B%5D=java&lang%5B%5D=php&lang%5B%5D=swift&gender=ma



Ur Name is anas  
Ur pass is 123  
Ur gender is male  
year is 1888  
Ur color is #c83b3b  
java php swift



# PhP

`$_GET` is an array of variables passed to the current script via the URL parameters.

`$_POST` is an array of variables passed to the current script via the HTTP POST method.

## When to use GET?

Information sent from a form with the GET method is **visible to everyone** (all variable names and values are displayed in the URL).

GET also has limits on the amount of information to send. The limitation is **about 2000 characters**. However, because the variables are displayed in the URL, it is possible to **bookmark the page**.

GET may be used for sending non-sensitive data.

**Note:** GET should NEVER be used for sending passwords or other sensitive information!

## When to use POST?

Information sent from a form with the POST method is **invisible to others** (all names/values are embedded within the body of the HTTP request) and has **no limits** on the amount of information to send.

Moreover POST supports advanced functionality such as support for multi-part binary input while uploading files to server.

However, because the variables are not displayed in the URL, it is not possible to bookmark the page.





PHP

## PHP Form Validation Example

### Form Validation

\* required field

Name:  \*

E-mail:  \*

Website:

Comment:

Gender: ☐ Female ☐ Male ☐ Other \*

**Your Input:**





# PHP

## Field

## Validation Rules

Name	Required. + Must only contain letters and whitespace
------	--

E-mail	Required. + Must contain a valid email address (with @ and .)
--------	---

Website	Optional. If present, it must contain a valid URL
---------	---

Comment	Optional. Multi-line input field (textarea)
---------	---

Gender	Required. Must select one
--------	---------------------------



The name, email, and website fields are text input elements, and the comment field is a textarea.

The HTML code looks like this:

```
Name: <input type="text" name="name">  
E-mail: <input type="text" name="email">  
Website: <input type="text" name="website">  
Comment: <textarea name="comment" rows="5" cols="40"></textarea>
```

## Radio Buttons

The gender fields are radio buttons and the HTML code looks like this:

```
Gender:  
<input type="radio" name="gender" value="female">Female  
<input type="radio" name="gender" value="male">Male  
<input type="radio" name="gender" value="other">Other
```



# The Form Element

The HTML code of the form looks like this:

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

When the form is submitted, the form data is sent with method="post".

**What is the `$_SERVER["PHP_SELF"]` variable?**

The `$_SERVER["PHP_SELF"]` is a super global variable that returns the filename of the currently executing script.





So, the `$_SERVER["PHP_SELF"]` sends the submitted form data to the page itself, instead of jumping to a different page. This way, the user will get error messages on the same page as the form.

## What is the `htmlspecialchars()` function?

The `htmlspecialchars()` function converts special characters into HTML entities. This means that it will replace HTML characters like `<` and `>` with `&lt;` and `&gt;`. This prevents attackers from exploiting the code by injecting HTML or Javascript code (Cross-site Scripting attacks) in forms.

## PHP Forms

PHP Form Handling

PHP Form Validation

PHP Form Required

PHP Form URL/E-mail

PHP Form Complete

UserName :  Please fill ur Name

Email :  Please fill Ur Email

Password :

Please fill Ur Pass Gender :

☐ Female ☐ male Please fill Ur gender :

Years : 1999 ▼

Ur Name is

Ur email is

Ur pass is

Ur gender is

Ur years is 1999

[https://www.youtube.com/watch?v=WJaU1ImHjCQ&list=PLrCBCI\\_Do84jHBv\\_mmGfjOwbAn7xgjKEA&index=6](https://www.youtube.com/watch?v=WJaU1ImHjCQ&list=PLrCBCI_Do84jHBv_mmGfjOwbAn7xgjKEA&index=6)



## PHP MySQL Database

With PHP, you can connect to and manipulate databases.

MySQL is the most popular database system used with PHP.

The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows.

Databases are useful for storing information categorically. A company may have a database with the following tables:

- Employees
- Products
- Customers
- Orders





# PhP

## PHP + MySQL Database System

### Database Queries

A query is a question or a request.

We can query a database for specific information and have a recordset returned.

Look at the following query (using standard SQL):

```
SELECT LastName FROM Employees
```

