

## PHP and FORM

**\$\_GET**

HTML Form: use the attribute: method="get"

```
<form method="get" action="">
    <input type="text" name="keyword"/>
    <input type="submit" name="submit"
value="Search"/>
</form>
```

PHP Code:

```
if (isset($_GET['keyword'])) {
    $keyword = $_GET['keyword'];
    $sql = 'SELECT *
FROM article
WHERE title
        LIKE "%' . $keyword . '%"';
}
```

In GET method, data sent through url, use GET for non sensitive data, such as searching feature and pagination.

**\$\_POST**

HTML Form: use the attribute method="post"

```
<form method="post" action="">
    <input type="password" name="pass"/>
    <input type="submit" name="submit"
value="Login"/>
</form>
```

PHP Code:

```
if (isset($_POST['submit'])) {
    $sql = 'SELECT user, password
FROM user
WHERE password = "'. $_POST['pass'] . '"';
    $result = mysqli_query($sql);
}
```

in POST method, form data sent via HTTP HEADER, so it is not visible to the user. Use POST for sensitive data, such as passwords, bank account, or task that change the database data, such as editing or deleting data.

**\$\_FILES**

HTML Form: use the attribute  
enctype="multipart/form-data"

```
<form method="post" action=""
enctype="multipart/form-data">
    <input type="file" name="user_file"/>
    <input type="submit" name="submit"
value="Upload"/>
</form>
```

PHP Code:

```
if (isset($_POST['submit'])) {
    echo'<pre>'; print_r($_FILES); echo'</pre>';
}
```

## Fill input value with submitted data

In the following example, if the form is submitted, it will contain the search string keyword(GET Method)

```
<form method="get" action="">
    <input type="text" name="keyword"
value="<?=@$_GET['keyword'] ?: ""?>"/>
    <input type="submit" name="submit"
value="Search"/>
</form>
```

Use @ to remove the warning message when the variable \$\_GET['keyword'] undefined (e.g. for the first time form opened). An example for POST method

```
<form method="post" action="">
    <input type="text" name="nama"
placeholder="Nama Anda"
value="<?=@$_POST['nama'] ?: ""?>"
/>
    <input type="submit" name="submit"
value="Simpan"/>
</form>
```

## SELECT ELEMENT

Try to use a loop to create select elements

```
<form method="post" action="">
    <?php
    $month = array('January',
                    'February',
                    'March'
                );
    echo '<select name="month">';
    foreach ($month as $key => $month_name) {
        $selected = @$_POST['month'] == $key
            ? 'SELECTED="SELECTED"' : '';
        echo '<option value="'. $key . '" .
            $selected . '>' .
            $month_name .
            '</option>';
    }
    echo '</select>';
    ?>
    <input type="submit" name="submit"
value="Save"/>
</form>
```

Using the loop will make life easier, especially when we need to add or remove options, also, it is easier to add the attribute selected="selected". This attribute useful to automatically select the option according to the user input

## Example: Displaying data from a database

```
1. <?php
2. // SUBMIT
3. $query_result = false;
4. $error = $warning = array();
5. if (isset($_POST['submit']))
6. {
7.     if (!trim($_POST['book_title'])) {
8.         $error = 'Book title is required';
9.     }
10.
11.     if (!$error) {
12.         $conn = @mysqli_connect('localhost', 'root', '',
13. 'db_book');
14.         if (!$conn) {
15.             $error[] = mysqli_connect_error($conn);
16.         } else {
17.             $where = trim($_POST['book_title'])
18.                 ? 'book_title LIKE "%'
19.                     . $_POST['book_title'] . '%"
20.                 : '';
21.
22.             $sql = 'SELECT * FROM book WHERE ' . $where;
23.             $result = mysqli_query ($conn, $sql);
24.
25.             // RESULT
26.             if (!$result) {
27.                 $error[] = mysqli_error($conn)
28.                     . '<br/><strong>SQL Query</strong>: ' . $sql;
29.             } else {
30.                 $num_rows = mysqli_num_rows($result);
31.                 if (!$num_rows) {
32.                     $warning[] = 'Data not found';
33.                 } else {
34.                     $query_result = true;
35.                 }
36.             }
37.         } //
38.     }
39. }?>
40. <html>
41. <head> <title>PHP dan FORM</title> </head>
42. <body>
43. <?php
```

```
44. if ($error) {
45.     echo '<div>Error: ' . join($error, ', ') . '</div>';
46. }
47.
48. if ($warning) {
49.     echo '<div>' . join($warning, ', ') . '</div>';
50. }
51. ?>
52. <form action="" method="post">
53.     <div>
54.         <label>Book Title</label>
55.         <input type="text" name="book_title"
56.             value="<?=@$_POST['book_title'] ?: ''?>" />
57.     </div>
58.     <div>
59.         <input type="submit" name="submit" value="Submit" />
60.     </div>
61. </form>
62. <?php
63. if (isset($_POST['submit'])) {
64.
65.     if ($query_result)
66.     {
67.         echo 'Found ' . $num_rows . ' records';
68.         $thead = ' <tr>
69.                 <th>No</th>
70.                 <th>Book Title</th>
71.                 </tr>';
72.
73.         echo ' <table> <thead>' . $thead . '</thead>
74.             <tbody>';
75.
76.         $no = 1;
77.         while($row = mysqli_fetch_array($result)) {
78.             echo '<tr> <td>' . $no . '</td>
79.                 <td>' . $row['book_title'] . '</td>
80.                 </tr> ';
81.             $no++;
82.         }
83.         echo '</tbody></table>';
84.     }
85. }?>
86. </div>
87. </body>
88. </html>
```