

# **МДК 02.02. Web-программирование. Язык PHP**

Работа с формой

# Суперглобальные переменные PHP

Доступны в любом месте скрипта.

## **\$GLOBALS**

Содержит ссылку на каждую переменную, доступную в данный момент в глобальной области видимости скрипта. Ключами этого массива являются имена глобальных переменных.

## **\$\_SERVER**

Переменные, установленные web-сервером

# Суперглобальные переменные PHP

## **`$_GET`**

Ассоциативный массив данных. Содержит переменные, передаваемые скрипту с помощью метода GET.

## **`$_POST`**

Ассоциативный массив данных. Содержит переменные, передаваемые скрипту с помощью метода POST.

## Суперглобальные переменные PHP

### `$_COOKIE`

Переменные, передаваемые скрипту через HTTP cookies.

### `$_FILES`

Переменные, передаваемые скрипту через HTTP post-загрузку файлов.

# Суперглобальные переменные PHP

## `$_REQUEST`

Переменные, передаваемые скрипту через механизмы ввода GET, POST и COOKIE

## `$_SESSION`

Переменные, зарегистрированные на данный момент в сессии скрипта

## Способы передачи параметров сценарию

Наиболее распространенными методами передачи данных между браузером и сценарием являются GET и POST.

**Метод GET.** Данные методом GET передаются путем их добавления к URL-адресу вызываемого сценария, предназначенного для обработки полученной информации.

## Пример

```
<?php  
echo 'Привет '.$_GET["name"].!';  
?>
```

## Использование метода GET

### Файл forma\_get.php

```
<html>
<head><title>Форма для передачи данных методом
GET</title></head>
<body>
<form action="get_method.php" method="get">
Введите имя <input type="text" name="num" value=""><br>
<input type="submit" name="bsubmit" value="Отправить">
</form>
</body>
</html>
```



## Файл `get_method.php`

```
<html>
  <head><title>Проверка метода GET</title></head>
  <body>
    <?php
      if(isset($_GET['num'])) {
        echo $_GET['num'];
      }
    ?>
  </body>
</html>
```

## Использование метода POST

Способ позволяет скрыто передавать переменные.

Файл `forma_post.php`

```
<html>
  <head><title>Форма для передачи данных методом POST </title>
</head>
  <body>
    <form action="forma_post.php" method="post">
      Введите имя <input type="text" name="num" value="">
      <br>
      <input type="submit" name="bsubmit" value="Отправить">
    </form>
  </body>
</html>
```

## Файл post\_method.php

```
<html>  
  <head><title>Проверка метода POST </title></head>  
  <body>  
    <?php  
    if(isset($_POST['num'])) {  
      echo $_POST['num'];  
    }  
    ?>  
  </body>  
</html>
```

## Пример

```
<?php
echo <<< _FORM
<form action="post.php" method="post">
Имя: <input type="text" name="username"><br>
Email: <input type="text" name="email"><br>
<input type="submit" name="submit" value="Отправить!">
</form>
 _FORM;
if(isset($_POST['username'])){echo $_POST['username'];}
if(isset($_POST['email'])){echo $_POST['email'];}
?>
```

## Метод POST

### Пример форма\_post.php

```
<!DOCTYPE>  
<html>  
<body>  
<form action="formamethod.php" method="post">  
Имя: <input type="text" name="username"><br>  
Email: <input type="text" name="email"><br>  
<input type="submit" name="submit" value="Отправить!">  
</form>  
</body>  
</html>
```

## Метод POST

### Пример post\_method.php

```
<?php
if (isset($_POST['username']))
{
    echo $_POST['username'];
}
if (isset($_POST['email']))
{
    echo $_POST['email'];
}
?>
```

## Схема кодирования данных из формы

1. Для каждого элемента формы, имеющего имя, заданное параметром **name**, формируется пара **name = value**, где **value** - значение элемента, введенное пользователем или назначенное по умолчанию. При отсутствии значения, соответствующая пара имеет вид: **name =**.
2. Все пары объединяются в строку через разделитель **&**. Символы, не допустимые в составе URL (русские символы, пробелы, служебные символы) заменяются последовательностью, состоящей из символа **%** и их 16-го ASCII кода.
3. Закодированная информация передается серверу одним из методов (**GET** или **POST**).