

Метод POST

Создадим два файла: файл form.php будет содержать форму:

```
<?
$forma = '';
$forma .= "
<h1> Страница регистрации </h1>
<input type=hidden name=action value=post>
<form method=\"post\" action=\"page1_1.php\">
Имя:<br>
<input type=\"text\" name=\"name\"><br>
<input type=submit name=\"okbutton\" value='Enter'><br>
</form>";
echo $forma;
?>
```

файл page1_1.php – обработчик введенных в форме данных:

```
<?
$u_name = $_POST['name'];
echo "<b>Ваше имя:</b> your name " . ($u_name) . "<br>";
?>
```

Чтобы использовать метод POST атрибуту method тега form присваивается значение "post". При нажатии кнопки "Enter" браузер производит следующие действия: 1) Просматривает входящие в форму элементы и формирует из их атрибутов name и value данные формы. Допустим, введено имя Alex. В этом случае данные формы – name=Alex&okbutton=Enter. 2) Устанавливает соединение с сервером, отправляет на сервер запрос документа, указанного в атрибуте action (page1_1.php) тега <form>, используя метод отправки данных, указанный в атрибуте method (в данном случае – POST) передавая в запросе данные формы. 3) Сервер анализирует полученный запрос, формирует ответ, отправляет его браузеру и закрывает соединение. 4) Браузер отображает полученный от сервера документ.

PHP автоматически заполняет массив \$_POST:

```
$_POST['name'] = 'Alex'
$_POST['okbutton'] = 'Enter'
```

Именно к элементу \$_POST['name'] идет обращение в файле page1_1.php

Результат:

Ваше имя: Alex

Метод GET

Изменим значение атрибута method тега form на "get" (хотя метод GET используется по умолчанию). PHP с GET-формой поступает аналогично – заполняет массив \$_GET.

В этот раз браузер отправляет GET-запрос, который равносильно вводу в адресной строке адреса:

```
http://адрес-сайта/page1_1.php?name=Alex&okbutton=Enter
```

Результат аналогичен.

magic_quotes_gpc и stripslashes

magic_quotes_gpc boolean («магические кавычки») – это функция установки magic_quotes для GPC(Get / Post / Cookie операций). Когда magic_quotes включены (on), все ' (одиночные кавычки), " (двойные), \ (слэши) и пустоты экранируются бэкслэшами автоматически. get_magic_quotes_gpc() получает текущую активную установку конфигурации «магических кавычек» gpc.

stripslashes – удаляет экранирование символов.

Рассмотрим пример применения этих функций.

Оставим файл form.php (см. выше методы GET и POST) неизменным. Файл page1_1.php придется изменить следующим образом:

```
<?
if(get_magic_quotes_gpc()){
$u_name = stripslashes($_POST['name']);
echo "<b>get_magic_quotes_gpc() подключен</b><br><b>Если использовать stripslashes():</b> your name " . stripslashes($u_name) . "
}
//
$u_name = $_POST['name'];
echo "<b>Если не использовать stripslashes():</b> your name " . ($u_name) . "<br>";
?>
```

Если ввести в форму имя D'Artanian, то в итоге получим такие результаты, в комментариях не нуждающиеся:

```
get_magic_quotes_gpc() подключен
Если использовать stripslashes(): your name D'Artanian
Если не использовать stripslashes(): your name D\Artanian
```

Массив \$_REQUEST

Массив `$_REQUEST` объединяет в себе три массива (`$_GET`, `$_POST` и `$_COOKIE`). Поэтому для создания программ в не зависимости от метода передачи данных, полезно использовать именно его: к примеру, `$u_name = $_POST['name']` мож заменить на `$u_name = $_REQUEST['name']`.

Бывают случаи, когда требуется разделить методы приема, но тогда Вы сами решите для себя какой массив использовать: `$_GET[...]` или `$_POST[...]`.