

ЛАБОРАТОРНАЯ РАБОТА №11

Студент: Кинстлер Александра

Тема: Работа с URL-адресами. Передача данных через GET-запросы

1. Цель работы

Изучить механизм передачи данных между страницами через параметры URL, научиться использовать массивы `$_GET` и `$_SERVER`, формировать динамические ссылки и обрабатывать параметры безопасно.

2. Задачи

1. Изучить структуру URL и query-string.
2. Научиться передавать параметры через метод GET.
3. Научиться обрабатывать данные из `$_GET` с фильтрацией.
4. Освоить получение информации о запросе через `$_SERVER`.
5. Реализовать динамические ссылки и простую навигацию.
6. Реализовать мини-скрипты на PHP.

3. Теоретическая часть

3.1. URL и строка запроса

URL состоит из:

`protocol://host/path?param=value¶m2=value2`

Строка после ? называется query string.

3.2. Массив `$_GET`

Позволяет получить все значения из строки запроса:

`$_GET['key']`

3.3. Массив \$_SERVER

Позволяет узнать данные о запросе:

- `$_SERVER['REQUEST_URI']` – полный URL после домена
- `$_SERVER['HTTP_HOST']` – хост
- `$_SERVER['QUERY_STRING']` – строка запроса

3.4. Защита данных

Текст фильтруется через:

`htmlspecialchars()`

Числа приводятся к типу:

`(int)`

4. Ход выполнения работы

4.1. Страница index.php (формирование ссылки)

```
$ nano index.php
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>Передача данных через URL</title>
</head>
<body>
    <h2>Передача данных через ссылку</h2>
    <a href="page.php?name=Damir&age=25">
        Показать информацию о пользователе
    </a>
</body>
```

```
</html>
```

4.2. Обработка параметров на странице page.php

```
$ nano page.php
```

```
<?php
```

```
if (isset($_GET['name']) && isset($_GET['age'])) {  
    $name = htmlspecialchars($_GET['name']);  
    $age = (int) $_GET['age'];  
  
    echo "<h3>Информация о пользователе</h3>";  
    echo "Имя: $name <br>";  
    echo "Возраст: $age лет <br>";  
}  
else {  
    echo "Параметры не переданы!";  
}  
  
echo "<hr>";  
echo "Текущий URL: " . $_SERVER['REQUEST_URI'] . "<br>";  
echo "Хост: " . $_SERVER['HTTP_HOST'] . "<br>";  
echo "Строка запроса: " . $_SERVER['QUERY_STRING'] . "<br>";  
  
?>
```

4.3. Динамическая навигация (пагинация)

```
$ nano pagination.php
```

```
<?php
```

```
$page = isset($_GET['page']) ? (int) $_GET['page'] : 1;
```

```
echo "<h3>Вы на странице № $page</h3>";
```

```
for ($i = 1; $i <= 5; $i++) {  
    echo "<a href='?page=$i'>Страница $i</a> ";  
}  
?>
```

4.4. Страница search.php (форма поиска)

```
$ nano search.php
```

```
<form method="get" action="">  
    <label>Введите поисковый запрос:</label>  
    <input type="text" name="q" placeholder="Например: PHP">  
    <input type="submit" value="Поиск">  
</form>
```

```
<?php  
if (!empty($_GET['q'])) {  
    $query = htmlspecialchars($_GET['q']);  
    echo "<p>Результаты поиска по запросу: <b>$query</b></p>";  
}  
?>
```

5. Самостоятельные задания

5.1. calc.php — работа с двумя числами

```
$ nano calc.php
```

```
<?php

if (isset($_GET['a']) && isset($_GET['b'])) {

    $a = (int) $_GET['a'];
    $b = (int) $_GET['b'];

    echo "Сумма: " . ($a + $b) . "<br>";
    echo "Разность: " . ($a - $b) . "<br>";
    echo "Произведение: " . ($a * $b) . "<br>";

    if ($b != 0) {
        echo "Частное: " . ($a / $b);
    } else {
        echo "Частное: деление на ноль невозможно";
    }
}

echo "Передайте параметры а и b!";
}

?>
```

Пример вызова:

calc.php?a=8&b=4

5.2. profile.php — приветствие и возврат назад

```
$ nano profile.php
```

```
<?php
```

```
if (!empty($_GET['name']) && !empty($_GET['city'])) {
```

```
    $name = htmlspecialchars($_GET['name']);
```

```
    $city = htmlspecialchars($_GET['city']);
```

```
    echo "Привет, $name! Добро пожаловать из города $city. <br><br>";
```

```
    if (!empty($_SERVER['HTTP_REFERER'])) {
```

```
        echo "<a href=\" . $_SERVER['HTTP_REFERER'] . \">Вернуться назад</a>";
```

```
}
```

```
} else {
```

```
    echo "Недостаточно параметров!";
```

```
}
```

```
?>
```

6. Контрольные вопросы

1. Пользовательские функции создаются программистом; встроенные — уже есть в PHP.
2. Значение по умолчанию задаётся так:

```
function f($x = 10) {}
```

3. Передача по значению копирует данные, по ссылке — передается адрес (&).
4. Оператор return возвращает значение и завершает выполнение функции.
5. Рекурсия — вызов функции самой себя; применяется при обходе деревьев, вычислении факториала.
6. Примеры:
 - строковые: strlen(), strtolower(), explode()
 - массивы: array_push(), count()
7. Анонимные функции появились в PHP 5.3, стрелочные — в PHP 7.4.

7. Вывод по работе

В ходе лабораторной работы изучены основы передачи данных через URL-адреса, освоена работа с суперглобальными массивами \$_GET и \$_SERVER, реализованы динамические ссылки, пагинация и обработка параметров. Получены практические навыки безопасной фильтрации входящих данных и создания простых PHP-скриптов.

Внизу прикреплены скриншоты вывода.



Передача данных через ссылку

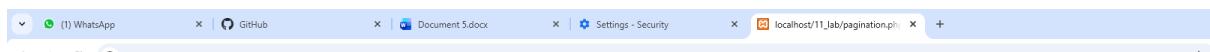
[Показать информацию о пользователе](#)



Информация о пользователе

Имя: Damir
Возраст: 25 лет

Текущий URL: /11_lab/page.php?name=Damir&age=25
Хост: localhost
Строка запроса: name=Damir&age=25



Вы на странице № 3

[Страница 1](#) [Страница 2](#) [Страница 3](#) [Страница 4](#) [Страница 5](#)

