

Практическая работа №25

Установка и настройка локального сервера

1 Цель работы

- 1.1 Научиться устанавливать и применять локальный WAMP-сервер;
- 1.2 Изучить настройки php.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.24.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.2.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Скопировать папку с локальным сервером Denwer в папку C:\Temp\[название группы].

Открыть папку «denwer» и запустить на выполнение файл Run.exe, дождаться запуска всех служб локального сервера. Если выдает ошибку, замените букву виртуального диска в файле configuration на другую, например, R.

5.2 Открыть браузер, ввести в адресной строке:
`http://localhost/`

В случае корректной установки в браузере будет выведена страница с текстом «Ура, заработало!».

5.3 Для вывода подробной информации о настройках php в папке Denwer\home\localhost\www создать файл info.php со следующим содержимым:

```
<?php
// Показывать всю информацию, по умолчанию INFO_ALL
phpinfo();
// Показывать информацию только о загруженных модулях.
phpinfo(INFO_MODULES);
?>
```

5.4 В браузере открыть страницу `http://localhost/info.php` и изучить полученный результат.

5.5 Перейти в папку Denwer\usr\local\php5 и изучить файл конфигурации php.ini.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Для чего применяется локальный WAMP-сервер?
- 8.2 Из каких компонентов состоит WAMP-сервер?
- 8.3 Какая функция применяется для просмотра настроек php?
- 8.4 Для чего применяется Apache-сервер?
- 8.5 Каковы особенности разработки с использованием WAMP-сервера?
- 8.6 Какие WAMP-серверы существуют?

Практическая работа №26

Встраивание PHP-сценариев в веб-страницы

1 Цель работы

- 1.1 Научиться разрабатывать php-сценарии;
- 1.2 Научиться изолировать php-код от html-кода;
- 1.3 Закрепить навык тестирования и отладки приложений.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.25.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.3.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Создание php-сценария

Создать php-страницу со следующим содержимым:

```
<html>
<head>
  <title>Тест PHP</title>
</head>
<body>
  <?php
    echo '<p>Привет, мир!</p>';
  ?>
</body>
</html>
```

Протестировать результат в браузере. Открыть код страницы и убедиться, что php-сценарий транслировался в html-код.

- 5.2 Изолирование php-сценария

Модифицировать созданную php-страницу, заменив php-вставку на следующий код:

```
<?php if ($expression == true): ?>
```

Это будет отображено, если выражение истинно.

```
<?php else: ?>
```

В ином случае будет отображено это.

```
<?php endif; ?>
```

Перед запуском в браузере задать переменной `$expression` произвольное значение. Протестировать результат в браузере с различными значениями переменной `$expression`.

5.3 Применение условия в php-сценарии

Заменить в php-сценарии изолирование с использованием условий на один тэг `<?php ... ?>` с командой `echo` и операторами `if-else`. Протестировать результат в браузере.

5.4 Объявление переменных

Добавить в сценарий две переменных `$a` и `$b`, присвоить им числовые значения и вывести результат сложения этих переменных на экран, используя `echo`.

5.5 Добавление комментариев

Добавить в сценарий комментарий к каждой строке php-кода.

6 Порядок выполнения работы

6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Как встроить php-сценарий в html-код?

8.2 Какой способ является более эффективен при выводе больших блоков текста: изолирование с использованием условий или отправка текста с помощью функций `echo` или `print`?

8.3 Какая общая форма изолирования php-сценариев с помощью условий?

8.4 Какие пары тэгов могут применяться для обозначения PHP-кода?

8.5 Что такое «интерпретатор»?

8.6 Как добавить комментарии в php-сценарий?

Практическая работа №27

Включение внешних модулей в PHP-сценарии

1 Цель работы

- 1.1 Научиться разрабатывать php-сценарии;
- 1.2 Научиться создавать и встраивать в php-сценарий внешние php-модули;
- 1.3 Закрепить навык тестирования и отладки приложений.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.25.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – п.5.2.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Создание и подключение внешнего php-сценария

5.1.1 Создать файл vars.php со следующим сценарием:

```
<?php  
$color = 'зеленое';  
$fruit = 'яблоко';  
?>
```

5.1.2 Подключение php-сценария

Создать файл test.php со следующим сценарием:

```
<?php  
echo "До подключения: $color $fruit<br>";  
include 'vars.php';  
echo "После подключения: $color $fruit<br>";  
?>
```

5.1.3 Открыть файл test.php и объяснить полученный результат.

5.2 Для проверки того, что внешний модуль удалось подключить, добавить в test.php следующий код:

```
if ((include 'vars.php') == TRUE) {  
    echo 'Модуль успешно подключен';  
    ...  
}
```

```
}
```

5.3 Самостоятельно изучить в отдельных файлах подключение внешних модулей при помощи команд `include_once`, `require`, `require_once`. Объяснить отличия в выводе результата и подключении, если подключаемый файл будет отсутствовать.

5.4 Реализовать подключение внешнего модуля внутри функции:

```
<?php
```

```
function foo()
```

```
{
```

```
    global $color;
```

```
    include 'vars.php';
```

```
    echo "Тест: $color $fruit";
```

```
}
```

```
/* vars.php в той же области видимости, что и foo(), *
```

```
* поэтому $fruit НЕ доступен снаружи этой области      *
```

```
* $color доступен, поскольку переменная глобальная */
```

```
foo();           // A green apple
```

```
echo "Тест: $color $fruit"; // A green
```

```
?>
```

5.5 Создать следующий набор php-сценариев и объяснить полученную при выводе разницу:

Файл `return.php`

```
<?php
```

```
$var = 'PHP';
```

```
return $var;
```

```
?>
```

Файл `noreturn.php`

```
<?php
```

```
$var = 'PHP';
```

```
?>
```

Файл `testreturns.php`

```
<?php
```

```
$foo = include 'return.php';
```

```
echo $foo; // выведет 'PHP'
```

```
$bar = include 'noreturn.php';
```

```
echo $bar; // выведет 1
```

```
?>
```

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 В чем преимущество разработки и применения модулей?
- 8.2 В чем отличие между командами `include` и `require`?
- 8.3 В чем отличие между командами `include` и `include_once`?