

Практическая работа №2

Тема: «Клиентские и серверные технологии. Denwer»

Содержание работы:

- [Задание 1](#)
- [Задание 2](#)

Задание 1. Конфигурирование веб-сервера

Скопируйте папку [WebServers](#) на свой USB-накопитель или компьютер. Папка содержит набор веб-разработчика Denwer, базовый пакет которого состоит из компонентов:

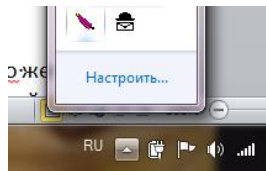
- Веб-сервер Apache с поддержкой SSI, SSL, mod_rewrite, mod_php.
- Интерпретатор PHP с поддержкой GD, MySQL, SQLite.
- СУБД MySQL с поддержкой транзакций (mysqld-max).
- Система управления виртуальными хостами, основанная на шаблонах.
- Система управления запуском и завершением.
- Панель phpMyAdmin для администрирования СУБД.
- Ядро интерпретатора Perl без стандартных библиотек (поставляются отдельно).
- Эмулятор sendmail и сервера SMTP с поддержкой работы совместно с PHP, Perl, Parser и др.
- Установщик.

Сразу после установки доступен полностью работающий веб-сервер Apache, работающий на локальном компьютере, на котором может работать неограниченное количество сайтов, что очень эффективно для разработки и отладки сценариев PHP без загрузки его файлов на удаленный сервер. Для запуска практически всех утилит «Денвера» используется приложение Run в подкаталоге /denwer корневого каталога установки «Денвера». При запуске создается виртуальный диск (по умолчанию Z:), где хранятся все файлы проектов.

Особенностью Denwer является автоматическая правка системного файла hosts, являющегося локальным аналогом DNS-сервера, что позволяет обращаться к локальным сайтам, работающим под управлением Денвера, по именам, совпадающим с именем папки, расположенной в каталоге home.

Чтобы начать использовать Денвер, выполните следующие действия:

1. Запустите Денвер, запустив файл Run.exe в папке WebServers\denwer.
2. После этого у Вас должны появиться такие знаки в области уведомлений:



3. Откройте браузер и перейдите по адресу <http://localhost> либо же просто localhost (современные браузеры поддерживают и такой формат). Вы должны увидеть главную страницу Денвера.
4. Чтобы увидеть как Ваш сайт будет выглядеть в браузере, файлы следует переместить в каталог `WebServers\home\localhost\www`.

В целом, локальный веб-сервер не требует каких-то особых настроек. Однако иногда возникает необходимость в этом. Например, если Вы при создании сайта будете использовать кодировку windows-1251, проще говоря, кириллицу, то никаких проблем не будет. А если для вашей работы Вам понадобится кодировка utf-8 или ещё какая-то другая, то возникнут проблемы.

- Чтобы этого не случилось, создайте в корне вашего сайта (в папке `www`) файл `.htaccess` (он создаётся в обычном блокноте) и пропишите следующее:

```
.htaccess - Блокнот
Файл  Правка  Формат  Вид  Справка
AddDefaultCharset utf-8
AddCharset utf-8 *
<IfModule mod_charset.c>
CharsetSourceEnc utf-8
CharsetDefault utf-8
</IfModule>
```

Настройка web-сервера Apache сводится к редактированию конфигурационного файла `httpd.conf`, который находится по адресу `\usr\local\apache\conf\httpd.conf`. Откройте файл, ознакомьтесь с его структурой, основными параметрами.

Внимание! Перед изменением файла `httpd.conf` рекомендуется создать его копию `http_original.conf`

Также в этом файле тоже можно указать **кодировку**. Для этого найдите `DefaultCharset` и измените его значение.

Скопируйте файл [example.php](http://localhost/example.php) на локальный сервер и откройте его в браузере <http://localhost/example.php>. Если все работает – Вы все выполнили правильно!

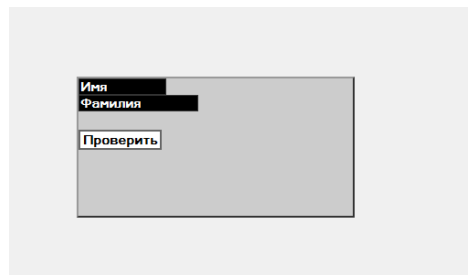
Задание 2. Клиентские и серверные технологии

Веб-разработчик работал над проектом, в котором использовались такие веб-технологии, как CSS, JavaScript, PHP. Однако из-за технических неполадок

компьютер разработчика сломался, часть информации была утеряна. Удалось восстановить лишь фрагменты файлов.

Необходимо:

- 1) На основании известной информации восстановить целостность файлов, определить какой фрагмент какому файлу принадлежит, и связать их между собой.
- 2) В результате Вы должны получить 4 файла, после чего следует проверить их работоспособность, а именно:
 - запустите файл index.html. Вы должны увидеть следующую форму:

A screenshot of a web form. It contains two input fields, the first labeled 'Имя' (Name) and the second labeled 'Фамилия' (Surname). Below these fields is a button labeled 'Проверить' (Check). The form is set against a light gray background.

- заполните форму;
- отправьте данные формы (нажмите кнопку «Проверить»);

Если Вы все правильно сделали, Вы должны увидеть на экране фразу: «Приветствую Вас [Имя] [Фамилия]». Получилось иначе – см. задание 3.

Известно, что:

- 1) Это были файлы:
 - index.html
 - main.css
 - current_script.js
 - user_checking.php
- 2) Утерянная информация обозначена знаками ?????????????????
- 3) Применялись только связанные стили.
- 4) JavaScript располагался только во внешнем файле.
- 5) **Фрагмент 1:**

```
//функция проверки полей, принимает два аргумента name1, и name2
function check_fields(name1, name2) {
    //если значения name1 и name2 не пусты
    if (document.getElementsByName(name1).value != "") &&
        (document.getElementsByName(name2).value != "")
    {
        //то пишем, что "Поля не пусты"
        document.write("Поля не пусты");
    }
    else // иначе
```

```

{
    //пишем это
    document.write("Одно или оба поля пусты");
}
}

```

Фрагмент 2:

```

????????????????????????????????????????????????????????????
//объявляем POST переменные
$lastName = $_POST['last_name']; //фамилия
$firstName = $_POST['first_name']; //имя

//если поле "Фамилия" и "Имя" не пусто
if (!empty($lastName) && !empty($firstName))
{
    //пишем
    echo "Приветствую Вас $lastName $firstName. <br />";
}
else //иначе
{
    //пишем
    echo "Как Вы посмели нажать эту кнопку?";
    //и обратно возвращаемся на index.html через 15 миллисекунд
    echo "<meta http-equiv='refresh' content='15; url=index.html' />";
}
?>

```

Фрагмент 3:

```

<html>
<head>
<title>Заголовок страницы</title>
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
</head>
<body>
<div id='form A'>
<form action='?????????????????????????????????????' method='POST'>
??????
??????
??????
??????
</form>
</div>
</body>
</html>

```

Фрагмент 4:

```

* {margin: 0; padding: 0;}
/* для html, body устанавливаем ширину, высоту и цвет заднего фона */
html, body {width: 100%; height: 100%; background-color: #f0f0f0; }

#form_A {
    /* Абсолютное позиционирование - будем перемещать его сами */
    position: absolute;
    /* Перемещаем направо и вниз на половину экрана */
    left: 50%; top: 50%;
    /* Выставляем ширину и высоту блока */
    width: 300px; height: 150px;
    /* Перемещаем налево на половину ширину, и вверх - на половину высоту */
    margin-left: -150px; margin-top: -75px;
    /* Устанавливаем цвет заднего фона в серый */
    background-color: #ccc;
}

```

```

/* Устанавливаем бордюры */
border-left: 2px solid #999; border-top: 2px solid #999;
border-right: 2px solid #333; border-bottom: 2px solid #333;
/* Благодаря ним добиваемся 3D эффекта объемности */
}

/* кнопка с классом button в блоке id=form_A */
#form_A .button {
/* устанавливаем задний фон в белый */
background-color: #fff;
/* цвет текста - черный */
color: #000;
/* стиль шрифта - жирный */
font-weight: bold;
/* бордюр вокруг на 2 пиксела темного цвета */
border: 2px solid #666;
}

/* стиль для кнопки, над которым стоит мышь */
#form_A .button:hover {
/* цвет заднего фона меняем на черный */
background-color: #000;
/* цвет текста - на белый */
color: #fff;
/* бордюр светлый */
border: 2px solid #bbb;
}

#form_A .field {
/* цвет заднего фона - черный */
background-color: #000;
/* цвет текста - белый*/
color: #fff;
/* жирный */
font-weight: bold;
/* однопиксельный бордюр темного цвета*/
border: 1px solid #333;
}

#form_A .field:hover {
/* цвет заднего фона меняем на белый */
background-color: #fff;
/* цвет текста - на черный */
color: #000;
/* бордюр делаем по-светлее*/
border: 1px solid #999;
}

```

Фрагмент 5:

```



```