

Лабораторная работа №5. "Введение в PHP"

Contents

Упражнение 1: Установка набора дистрибутивов Denwer	1
Упражнение 2: Запуск Denwer	2
Упражнение 3. Создание первого сайта.....	3
Упражнение 4: Простейшие программы на PHP (на 60%).....	10
Упражнение 5: Создание базы данных MySQL (на 80%)	11
Упражнение 6. Простой вывод данных. Страницы blog.php и comments.php (на 100%).....	17
Упражнение 7: Обмен данными между серверными страницами (на дополнительные 2 балла).....	18

Упражнение 1: Установка набора дистрибутивов Denwer

В данном упражнении будет продемонстрирована установка набора дистрибутивов, в состав которого входят дистрибутивы Apache, PHP, MySQL, phpMyAdmin и других систем и приложений, необходимых для организации разработки серверных приложений с использованием средств языка программирования PHP.

1. Скачайте дистрибутив базового комплекта Denwer - можно взять с сайта - <http://www.denwer.ru/>.
2. Запустите инсталлятор. Вначале архив будет автоматически распакован во временную директорию, а затем автоматически запустится инсталлятор.
3. По умолчанию для установки комплекса используется директория C:\WebServers, нажмите Enter, чтобы согласиться с этим выбором. В указанном каталоге будут расположены абсолютно все компоненты системы, и вне его никакие файлы в дальнейшем не создаются (исключая ярлыки на Рабочем столе).
4. Далее вам предложат ввести имя виртуального диска, который будет связан с только что указанной директорией. Рекомендуется согласиться со значением по умолчанию (Z:). Важно, что диска с этим именем еще не должно содержаться в системе — чаще всего так и происходит с диском Z:.

5. После этого начнется копирование файлов дистрибутива. В ходе установки вам будет предложена установка ярлыков на Рабочий стол – это необходимо сделать для дальнейшего удобства работы.

6. В конце установки будет задан вопрос, как именно вы собираетесь запускать и останавливать комплекс (2 варианта):

- Автоматически создавать виртуальный диск при загрузке машины (а при остановке серверов этот диск не отключать);
- Создавать виртуальный диск только по явной команде старта комплекса (при щелчке по ярлыку запуска на Рабочем столе). И, соответственно, отключать диск от системы — при остановке серверов.

Рекомендуемый вариант – второй.

Упражнение 2: Запуск Denwer

Чтобы начать использовать Денвер, сделайте следующие действия:

1. Запустите Денвер, воспользовавшись ярлыком Start Denwer на Рабочем столе. Если вы не создавали ярлыки, то можно запустить Денвер по команде `C:\WebServers\denwer\Run.exe`.
2. Откройте браузер и перейдите по адресу <http://localhost>.
3. Вы должны увидеть главную страницу Денвера.
4. Если после запуска Денвера <http://localhost> не открывается, проверьте, не блокируется ли Денвер вашим антивирусом или фаерволом. Например, были замечены проблемы с NOD32 в Windows XP (в нем нужно добавить процесс `Z:\usr\local\apache\bin\httpd.exe` в список исключений, это можно сделать в окне IMON/Настройка/Разное/Исключение).
5. Изучите информацию страницы <http://localhost/> Обратите особое внимание на разделы посвященные тестированию и работе с имеющимися утилитами.

Упражнение 3. Создание первого сайта.

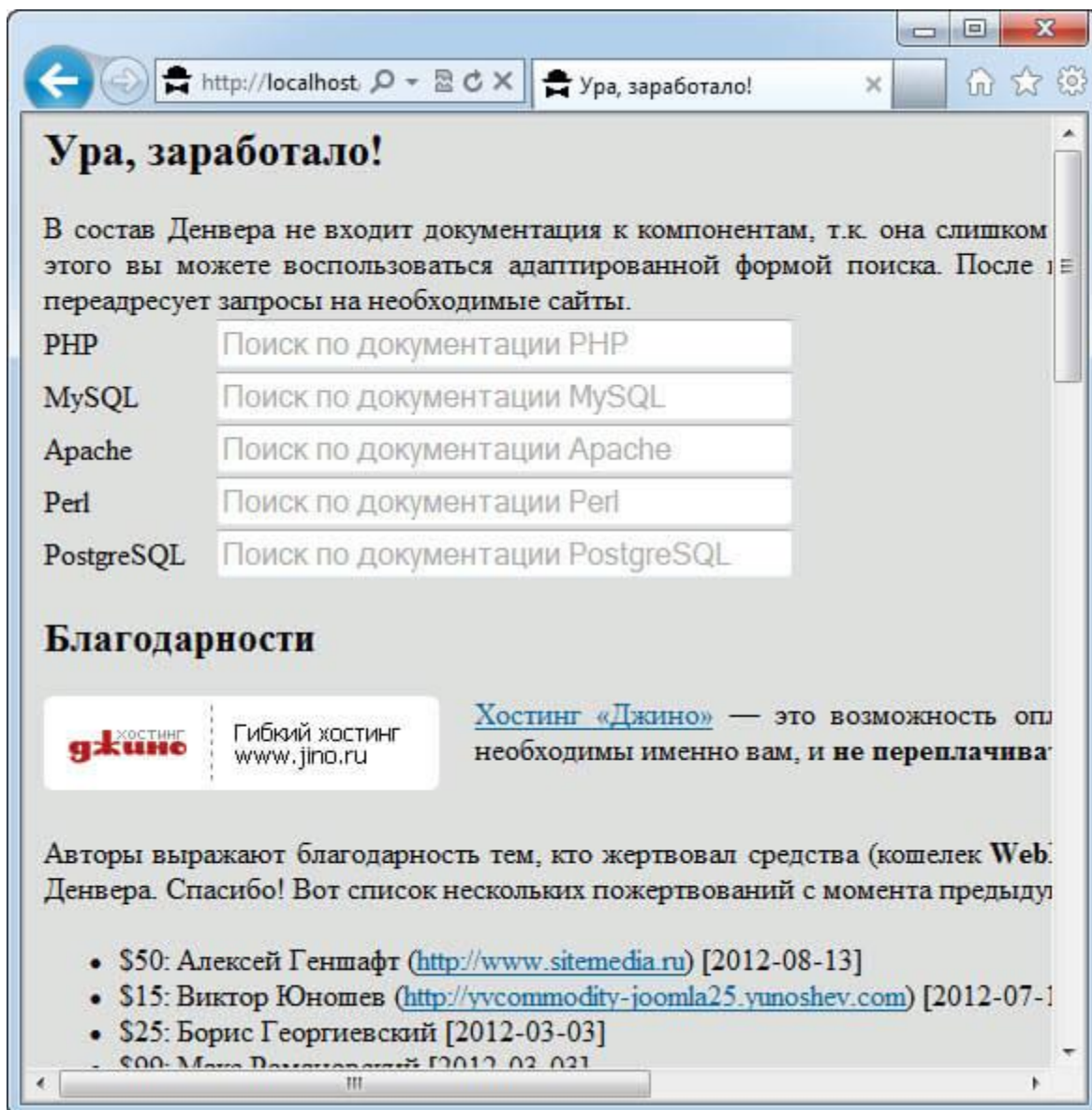
Для начала работы нам необходимо запустить Denwer. Делается это двойным кликом по значку «Start Denwer».



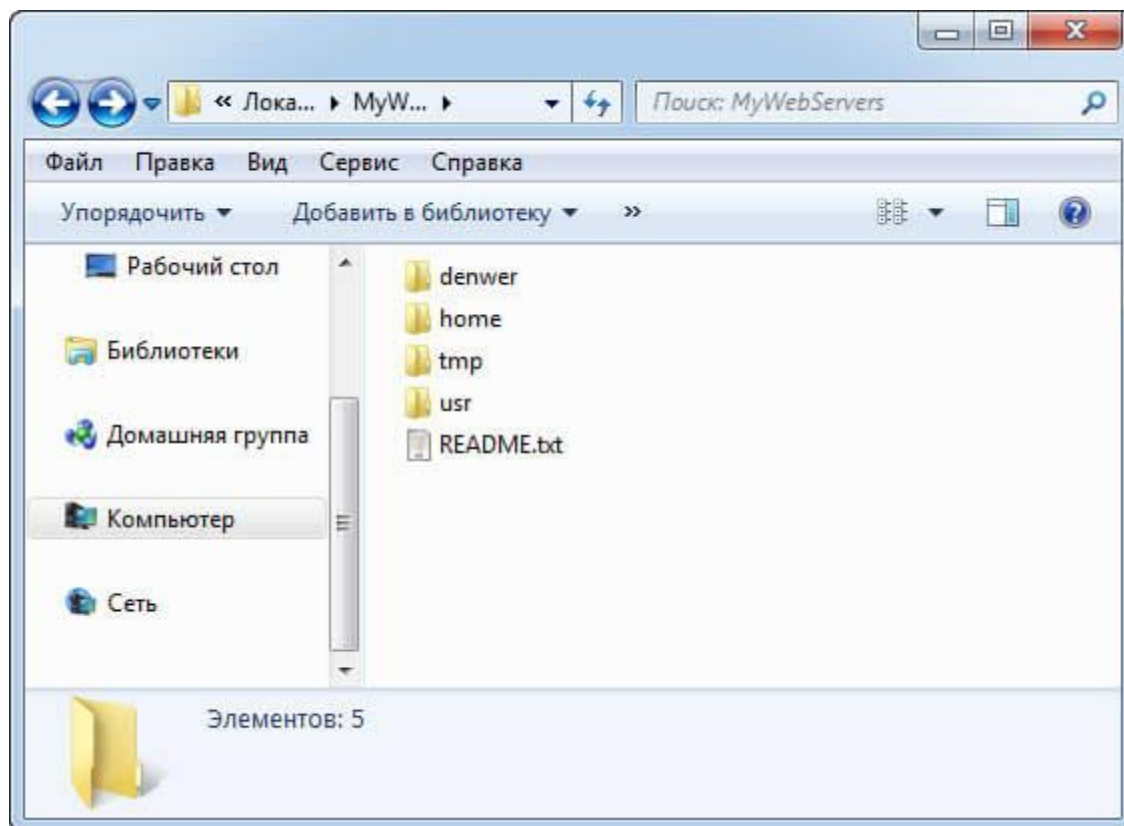
После запуска в правом нижнем углу появятся два значка:



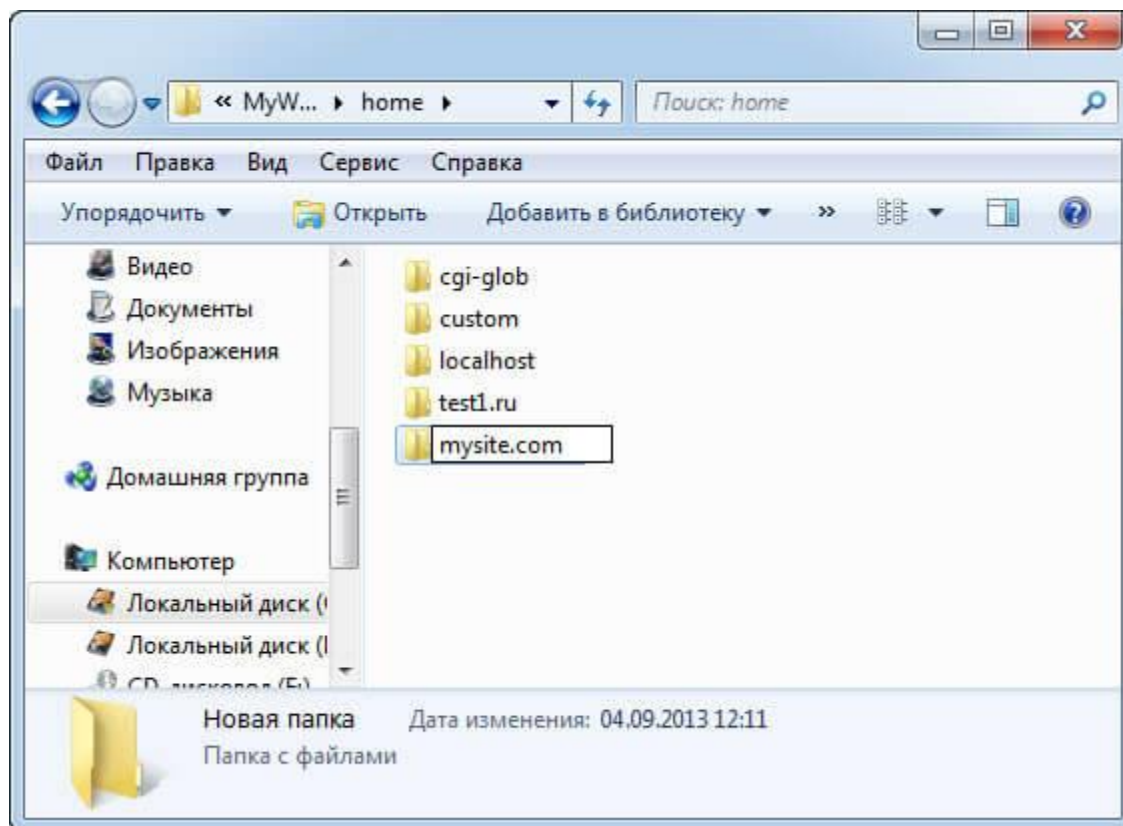
Значит Denwer запущен успешно. Чтобы проверить это в любом браузере в адресной строке наберите «<http://localhost>» и должна открыться следующая страница:



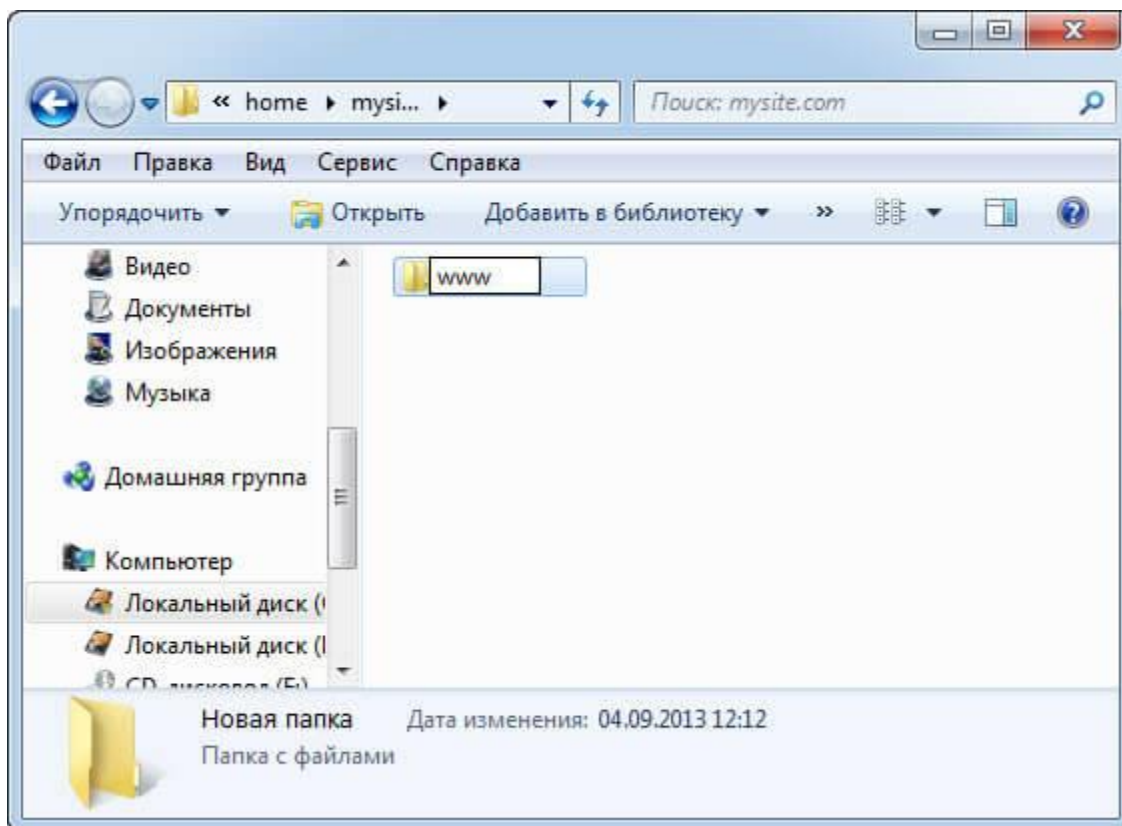
Далее переходим на диск C в папку куда мы в прошлый раз установили Denwer. Там вы увидите следующие директории:



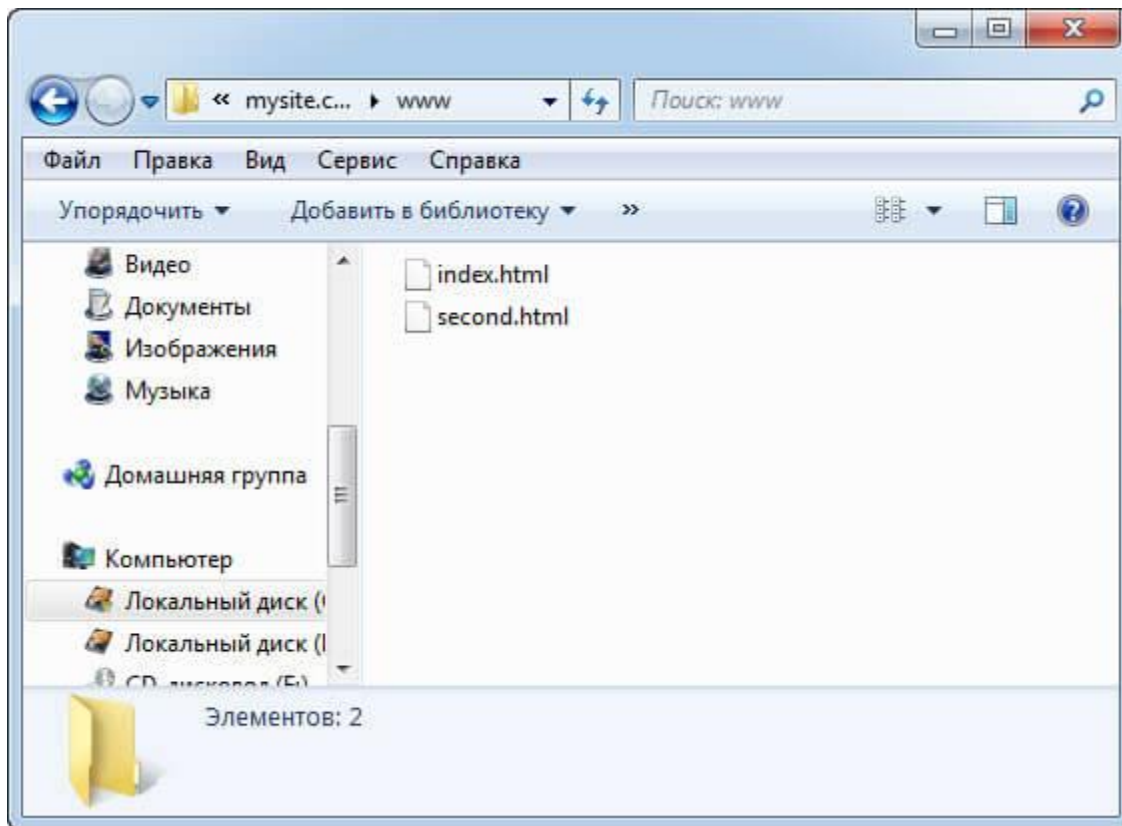
Далее переходим в папку «home» и там создаем папку с названием нашего сайта:



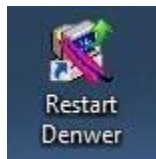
После того как папка создана. Заходим в нее и создаем папку «www»:



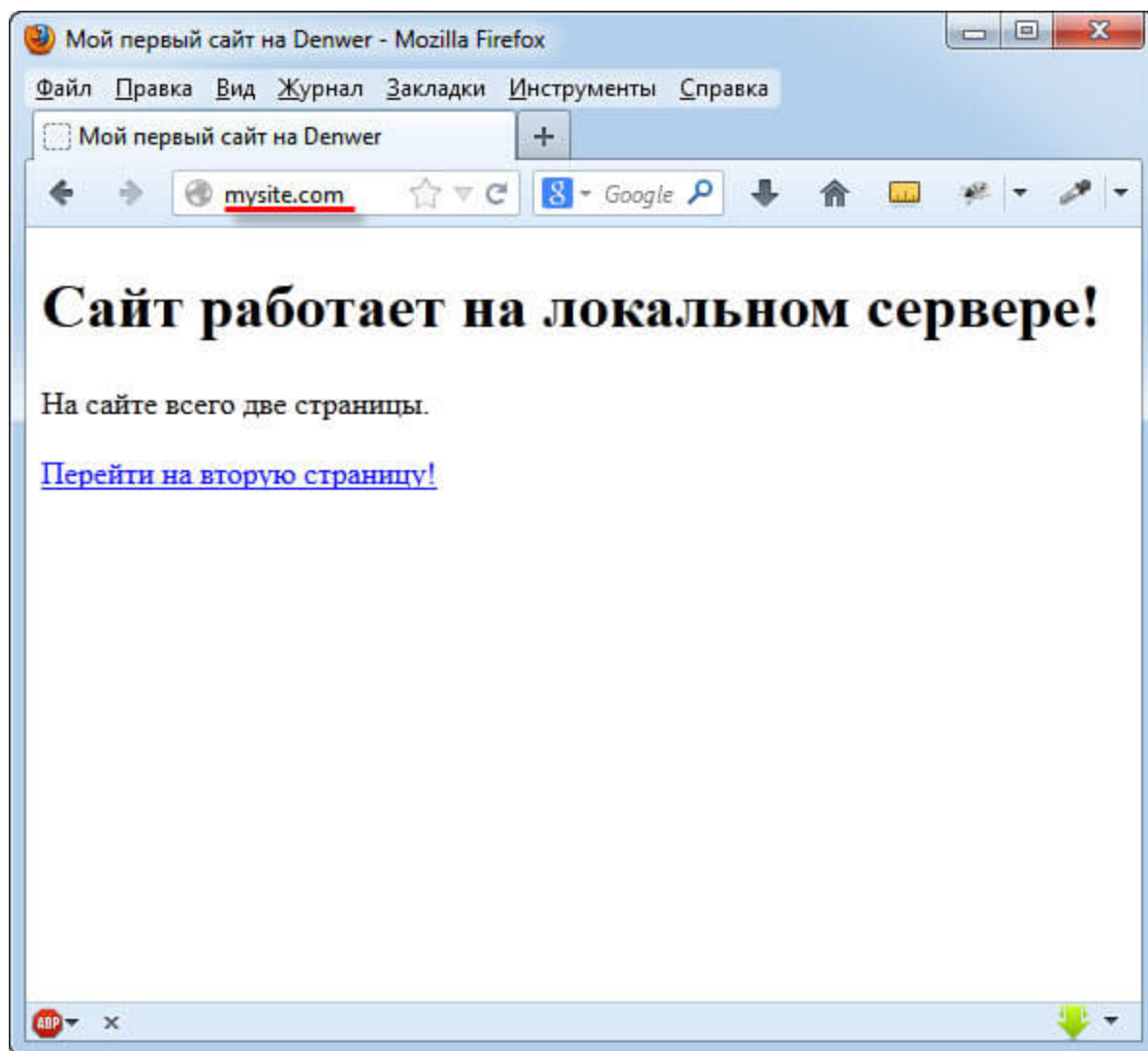
Сейчас мы создали все необходимые папки, для того чтобы запустить сайт с локального компьютера. В данную папку «www» можно заливать все необходимые файлы. Скачайте и скопируйте в данную папку следующие два файла(index.html и second.html) — <https://disk.yandex.ru/d/Aouad2yebfmwww>



После того как мы скопировали необходимо перезапустить Denwer для того чтобы он увидел новые файлы. Делается это с помощью значка на Рабочем столе «Restart Denwer»:



Сейчас наш сайт готов к запуску. Для этого заходим в любой браузер и вводим название нашего сайта. У меня название сайта «mysite.com»:



Упражнение 4: Простейшие программы на PHP (на 60%)

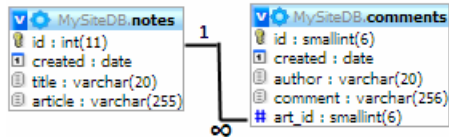
Создайте новый .php файл **examples.php** и выполните предложенные далее задания.

1. Переменной \$a необходимо присвоить значение 10, переменной \$b присвоить значение 20. Выведите значения переменных на экран.
2. Затем переменной \$c присвойте значение суммы этих переменных (переменной \$a и переменной \$b). Выведите значение переменной \$c на экран.
3. Далее увеличьте значение переменной \$c в три раза и выведите полученный результат на экран.
4. Разделите переменную \$c на разность переменных \$b и \$a, выведите результат на экран.
5. Введите новые переменные \$p и \$b. Присвойте переменной \$p значение «Программа», а переменной \$b значение «работает».
6. Затем сложите переменные, содержащие эти слова («Программа» и «работает»), при этом слова должны быть разделены пробелом (' '). Результат необходимо присвоить переменной \$result.
7. Далее с помощью оператора «.=» необходимо к строке «Программа работает» добавить слово «хорошо». Результат необходимо присвоить переменной \$result.
8. Есть две переменные: \$q = 5 и \$w = 7. Создайте скрипт, в результате выполнения которого эти две переменные «обмениваются» значениями – переменная \$q получает значение 7, переменная \$w получает значение 5, при этом не создавая новых переменных (вариант \$q = 7 и \$w = 5 не рассматривается). Выведите результат на экран.
9. Напишите **RНР цикл**, который выводит числа от 23 до 78.
10. Напишите **RНР цикл**, который выводит нумерованный список из 10 пунктов.
11. Создайте массив из 100 случайных чисел. Вывести массив, при помощи цикла while, а потом при помощи foreach.
- 12*. Напишите **скрипт**, который будет, в зависимости от дня недели, выводить надпись. Например: сегодня среда. Примечание: используйте оператор switch
13. Создать **функцию в RНР** — getPlus10(), которая будет принимать число и распечатывать сумму этого числа и 10. Выведите результат расчетов на экран.

Вставьте ссылку на этот файл из файла index.html. Продемонстрируйте результат.

Упражнение 5: Создание базы данных MySQL (на 80%)

В ходе выполнения данной лабораторной работы необходимо создать в MySQL новую базу данных с названием «**MySiteDB**» и добавить в нее две таблицы: **notes** и **comments**. **Notes** содержит заметки блога; **comments** – комментарии к этим заметкам. Схема данных :



Создание БД «MySiteDB»

В этом упражнении реализуется запрос на создание новой базы данных.

1. Создайте новый php документ, который будет называться **create_db.php**.
2. Создайте соединение с сервером localhost. Имя сервера **localhost**, пользователь **root**, пароля нет.
3. Создайте базу данных:
 - 3.1. Сформируйте запрос на создание базы **MySiteDB** с использованием SQL;
 - 3.2. Реализуйте запрос на создание БД с помощью функции `mysqli_query()`.
4. Сохранить документ, выполнить запрос.
5. С помощью утилиты **PhpMyAdmin** убедитесь, что создана новая база данных. Для этого запустите утилиту: <http://localhost/Tools/phpMyAdmin> (или <http://localhost> и выберите PhpMyAdmin из списка утилит).
6. Вторично выполните запрос, чтобы убедиться, что соединение есть, а база не создается (т.к. она была уже создана ранее, в ходе предыдущего выполнения скрипта).

Вариант реализации создания БД MySiteDB

```
<?php
```

```
//Создать соединение с сервером
$link = mysqli_connect ("localhost", "root", "");
if ($link) {
    echo "Соединение с сервером установлено", "<br>";
} else {
    echo "Нет соединения с сервером";
}
//Создать БД MySiteDB
//Сначала формирование запроса на создание

$db = "MySiteDB";
$query = "CREATE DATABASE $db";
//Затем реализация запроса на создание. Важна последовательность
аргументов функции: соединение с сервером, SQL-запрос.
```

```

$create_db = mysqli_query($link, $query);
if ($create_db) {
    echo "База данных $db успешно создана";
} else {
    echo "База не создана";
}
?>

```

Создание пользователя admin

Необходимо создать нового пользователя базы данных с именем **admin** и паролем **admin** с правами администратора.

Пользователей можно добавлять двумя способами:

- при помощи SQL-запроса GRANT
 - в таблице назначения привилегий MySQL (Privileges) с помощью утилиты PhpMyAdmin.
- Выберите один из двух приведенных далее способов.

Способ 1: создание нового пользователя с помощью SQL-запроса GRANT

1. Создайте новый php-документ, который будет называться create_user.php;
2. Создайте соединение с сервером;
3. Сформируйте SQL-запрос на создание нового пользователя базы данных:

```

$query = "GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost'
IDENTIFIED BY 'admin'
WITH GRANT OPTION";
//*. * - глобальный уровень привилегий, применяется ко всем базам на сервере.

```

4. Реализуйте запрос.

Проверка создания пользователя. С помощью утилиты PhpMyAdmin убедитесь, что создан новый пользователь. Для этого запустите утилиту PhpMyAdmin и перейдите на вкладку Privileges. Изучите список пользователей.

Способ 2: создание нового пользователя с помощью утилиты PhpMyAdmin

1. Запустите утилиту PhpMyAdmin и перейдите на вкладку Privileges. Нажмите кнопку «Add a new user».
2. Введите имя пользователя (**admin**), имя сервера (**localhost**), пароль с подтверждением (**admin**). Предоставьте новому пользователю все права (**global privileges – Check All**).
3. Убедитесь, что новый пользователь создан корректно.
4. Все дальнейшие действия с базой данных будут проводиться под пользователем **admin** с паролем **admin** и соответствующими правами, если иное не указано в задании.

Создание таблицы notes

В данном упражнении будет продемонстрирован один из способов создания таблиц в ранее созданной базе данных на примере создания таблицы **notes**. Таблица **notes** содержит заметки автора блога. Данная таблица будет создана средствами PHP.

1. Создайте новый php-документ, который будет называться **create_tbl.php**;
2. Создайте соединение с сервером уже под созданным ранее пользователем **admin** с паролем **admin**.
3. Подключитесь к базе данных MySiteDB.
4. Сформируйте запрос на создание таблицы **notes**

```
.  
//Формирование запроса  
$query = "CREATE TABLE notes  
(id INT NOT NULL AUTO_INCREMENT,  
PRIMARY KEY (id),  
created DATE,  
title VARCHAR (20),  
article VARCHAR (255))";
```

6. Реализуйте запрос на создание таблицы.

С помощью утилиты PhpMyAdmin убедитесь, что создана новая таблица. Для этого запустите утилиту, перейдите к базе данных MySiteDB и просмотрите ее структуру. В ней должна появиться соответствующая таблица.

Вариант реализации создания таблицы notes

```
<?php  
//Соединение с сервером  
$link = mysqli_connect ('localhost', 'admin', 'admin');  
//Выбор БД  
$db = "mySiteDB";  
$select = mysqli_select_db($link, $db);  
if ($select){  
    echo "База успешно выбрана", "<br>";  
} else {  
    echo "База не выбрана";  
}  
  
//Создание таблицы  
//Формирование запроса  
$query = "CREATE TABLE notes  
(id INT NOT NULL AUTO_INCREMENT,  
PRIMARY KEY (id),  
created DATE,  
title VARCHAR (20),  
article VARCHAR (255))";  
//Реализация запроса
```

```

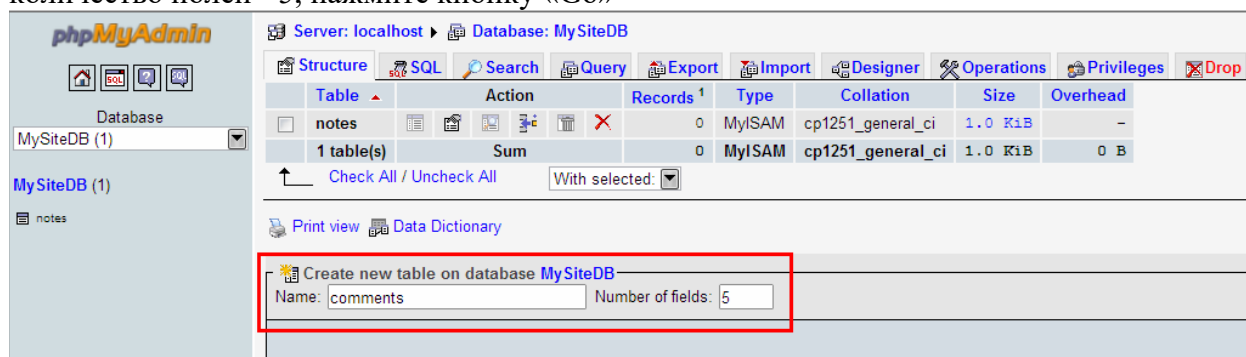
$create_tbl = mysqli_query ($link, $query);
if ($create_tbl){
    echo "Таблица успешно создана", "<br>";
} else {
    echo "Таблица не создана";
}
?>

```

Создание таблицы comments

В данном упражнении будет продемонстрирован другой способ создания таблиц в ранее созданной базе данных на примере создания таблицы **comments**. Таблица **comments** содержит комментарии пользователей к заметкам автора блога. Таблица будет создана с помощью утилиты PhpMyAdmin.

1. Запустите браузер.
2. Запустите утилиту *phpMyAdmin*. В главном окне *PHPMyAdmin* выберите БД MySiteDB.
3. В поле “Create new table”, присвойте имя таблице – comments; количество полей - 5, нажмите кнопку «Go»






































4. Создание полей таблицы comments:

- 4.1. В открывшемся окне заполните необходимые поля таблицы и нажмите кнопку «Save».
- 4.2. Для поля id добавьте следующие атрибуты: обозначьте автоинкремент A_I и первичный ключ PRIMARY в поле со списком INDEX.

Server: localhost ▶ Database: MySiteDB ▶ Table: comments

Field	Type ?	Length/Values ¹	Default ²
id	SMALLINT		None
created	DATE		None
author	VARCHAR	20	None
comment	VARCHAR	256	None
art_id	SMALLINT		None

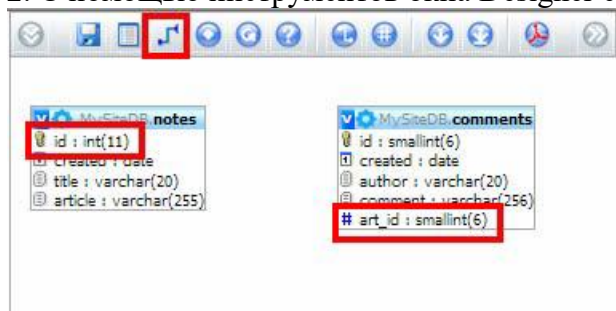
5. Полученный результат должен выглядеть следующим образом.

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	<u>id</u>	smallint(6)			No	None	auto_increment	      
<input type="checkbox"/>	created	date			No	None		      
<input type="checkbox"/>	author	varchar(20)	cp1251_general_ci		No	None		      
<input type="checkbox"/>	comment	varchar(256)	cp1251_general_ci		No	None		      
<input type="checkbox"/>	art_id	smallint(6)			No	None		      

Создание межтабличных связей

В данном упражнении необходимо создать связи между таблицами для поддержания целостности данных web-приложения.

- Для организации межтабличных связей выберите БД MySiteDB, вкладку Designer. Откроется окно схемы данных.
- С помощью инструментов окна Designer создайте связь «один ко многим».



Файл подключения базы данных

В ходе выполнения данного упражнения необходимо внести изменения в код автоматически созданного файла подключения базы данных для настройки корректной работы подключенной базы данных с кириллицей.

- Откройте файл MySiteDB.php и внесите в него следующие изменения:
 - Измените наименования переменных для удобства дальнейшей работы;
 - Измените функцию `mysql_pconnect()` на `mysqli_connect()`;

1.3. Внесите дополнения для корректной кодировки символов в базе данных:

```
<?php
    # FileName="Connection_php_mysql.htm"
    # Type="MYSQL"
    # HTTP="true"

    $localhost = "localhost";
    $db = "MySiteDB";
    $user = "admin";
    $password = "admin";
    $link = mysqli_connect($localhost, $user, $password) or
trigger_error(mysql_error(),E_USER_ERROR);

    //trigger_error выводит на страницу сообщение об ошибке. Первый параметр
    - сообщение об ошибке
    //в строковом виде, в данном случае возвращается функция mysql_error(),
    второй - числовой код //ошибки(почти всегда используется значение
    константы E_USER_ERROR, равное 256)
    //Следующие строки необходимы для того, чтобы MySQL воспринимал
    кириллицу.
    //Параметры функции mysqli_query(): идентификатор соединения с сервером
    и запрос SQL
    mysqli_query($link, "SET NAMES cp1251;") or die(mysql_error());
    mysqli_query($link, "SET CHARACTER SET cp1251;") or die(mysql_error());
?>
```


Упражнение 6. Простой вывод данных. Страницы blog.php и comments.php (на 100%)

Вывод данных из базы на страницу

На главную страницу сайта необходимо вывести все заметки из таблицы БД **notes**.

1. Создайте файл **blog.php**.

2. Создайте соединение с сервером. Оно у нас уже реализовано в файле **MySiteDB.php** – файл надо просто включить с помощью функции **require_once()**, в качестве параметра передав ей путь к файлу («Connections/MySiteBD.php»):

```
<?php require_once ("connections/MySiteDB.php"); ?>
```

3. Далее необходимо вывести записи (строки) на страницу сайта из таблицы notes.

Сначала надо реализовать запрос на выборку. Для этого:

3.1. выберите БД;

3.2. создайте SQL-запрос на выборку данных из таблицы (SELECT *fields* FROM *tableName*). Здесь SELECT – оператор выбора полей, FROM – оператор выбора таблицы-источника полей. ⇒ Если вам необходимо выбрать все поля таблицы (как в данном случае), то запрос можно построить так: *SELECT * FROM tablename*, где символ «*» обозначает все поля таблицы.

3.3. Реализуйте запрос на выборку.

4. Далее необходимо вывести запись на страницу сайта. Для этого используется функция **mysqli_fetch_array()**. Параметром функции является переменная, содержащая результат выполнения запроса к БД (в данном случае – реализации запроса на выборку); собственно функция получает по одной записи из таблицы за один раз. Каждая запись возвращается в виде массива.

5. Для вывода информации из массива по отдельным элементам необходимо придерживаться следующего синтаксиса:

```
//Вывод элементов массива
echo $note ['id'], "<br>";
echo $note ['created'], "<br>";
echo $note ['title'], "<br>";
echo $note ['article'], "<br>";
```

6. Сейчас из таблицы с помощью функции **mysqli_fetch_array()** выводится только одна запись. С помощью цикла необходимо сделать так, чтобы выводились все записи из таблицы. Для этого необходимо изменить часть кода следующим образом:

```
//Использование цикла while
while ($note = mysqli_fetch_array($select_note)){
    echo $note ['id'], "<br>";
    echo $note ['created'], "<br>";
```

```
echo $note ['title'], "<br>";
echo $note ['article'], "<br>";}
```

Здесь переменной с именем `$select_note` присваивается результат выполнения запроса к БД `mysqli_query()`.

Упражнение 7: Обмен данными между серверными страницами (на дополнительные 2 балла)

Каждая заметка на главной странице блога может быть прокомментирована. Для реализации этой функции необходимо сделать из заголовка каждой заметки гиперссылку, перейдя по которой посетитель попадет на страницу со списком комментариев к выбранной заметке. Кроме того, на этой же странице должна отображаться сама выбранная для комментирования заметка.

Следовательно, необходимо реализовать механизм обмена данными между страницами таким образом, чтобы при переходе по гиперссылке передавалась информация о том, какая именно заметка была выбрана. Для этого необходимо ввести некий идентификатор, значение которого будет совпадать с `id` комментируемой заметки, и который будет передаваться при переходе по гиперссылке.

1. Создание гиперссылки

1.1. Создайте новую страницу `comments.php`, которая будет содержать комментарии к выбранной заметке.

1.2. Реализуйте соединение с сервером.

1.3. Реализуйте подключение к БД.

1.4. Для передачи идентификатора заметки введем аргумент **note**. В качестве значения он будет получать значение поля **id** таблицы **notes**.

1.5. На странице **blog.php** найдите фрагмент кода, передающего заголовок заметки **title** (`echo $note ['title'];`). Его необходимо отредактировать таким образом, чтобы он стал гиперссылкой на страницу комментариев **comments.php**, а также передавал **id** выбранной заметки:

```
while ($note = mysqli_fetch_array($select_note)){
echo $note['id'], "<br>";
?>
<a href="comments.php?note=<?php echo $note['id']; ?>">
<?php echo $note ['title'], "<br>";?></a>
```

```
<?php
    echo $note ['created'], "<br>";
    echo $note ['content'], "<br>";
```

```
?> }
```

Здесь мы создаем гиперссылку на страницу **comments.php** и в этой гиперссылке передаем идентификатор **note**, значение которого равно значению элемента массива *\$note['id']*, т.е. значению *id* заметки.

2. Страница **comments.php**

2.1. Перейдите на страницу **comments.php**. На данной странице должны отображаться комментарии к выбранной записи, а также сама комментируемая запись (для удобства посетителя сайта).

2.2. Данную задачу можно выполнить по аналогии с выводом заметок на странице **blog.php**. Основное отличие заключается в том, что вначале необходимо со станицы **blog.php** получить переданный с помощью идентификатора **note** *id* заметки. Это делается с помощью метода `$_GET`:

```
//Переменной $note_id необходимо присвоить id заметки, переданной  
с помощью метода $_GET со страницы blog.php  
$note_id = $_GET['note'];
```

2.3. Далее необходимо вывести значения полей *created*, *title*, *content* из таблицы **notes** для заметки с полученным *id*. Для этого используется SQL запрос:
`SELECT... FROM... WHERE...`

В нем с помощью оператора **SELECT** выбираем необходимые поля таблицы; с помощью **FROM** определяем таблицу-источник выборки; **WHERE** задает условие отбора, по которому выбираем заметку с выбранным *id*:

```
//Формируем SQL-запрос на выборку с учетом переданного id заметки  
$query = "SELECT created, title, article FROM notes WHERE id = $note_id";
```

2.4. После формирования SQL-запроса его необходимо реализовать с помощью функции **mysqli_query()** и вывести данные на страницу с помощью функции **mysqli_fetch_array()**.

2.5. Затем аналогичным образом выведите комментарии к выбранной заметке. Обратите внимание, что SQL-запрос на выборку комментариев должен строиться следующим образом:

```
$query_comments = "SELECT * FROM comments WHERE art_id =  
$note_id";
```

В условии **WHERE** мы реализуем поддержку связи между таблицами, которые связаны по полям *id* (таблица **notes**) и *art_id* (таблица **comments**).

В переменной **\$note_id** содержится *id* выбранной заметки, следовательно, для выбора комментариев к этой заметке необходимо, чтобы значение поля *art_id* также было равно **\$note_id**.

3. Проверьте корректность данных между страницами **blog.php** и **comments.php**. При переходе по ссылке с **blog.php** на **comments.php** в адресной строке браузера должен отображаться *id* выбранной заметки, переданный с помощью идентификатора **note**.

4. Для того, чтобы выводились все комментарии, а не только первый – реализуйте цикл.

5* (необязательное задание). Если у заметки нет ни одного комментария – об этом надо сообщить.

5.1. Под областью комментариев добавьте надпись *«Эту запись еще никто не комментировал»*.

5.2. В коде программы создайте циклы с условием *if*: если хотя бы один комментарий существует – он должен быть выведен (т.е. элементы массива должны быть отображены); если количество комментариев равно нулю – должна выводиться надпись *«Эту запись еще никто не комментировал»*.