

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

С.В. Одиночкина

**Web-программирование: PHP
практикум**



Санкт-Петербург

УДК 004.655, 004.657, 004.62

С.В.Одиночкина

Web-программирование PHP - СПб: НИУ ИТМО, 2012. – 79 с.

В пособии излагаются методические рекомендации к выполнению лабораторных работ по дисциплине «Web-программирование PHP-технологии».

Предназначено для студентов, обучающихся по всем профилям подготовки бакалавров направления: 210700 Инфокоммуникационные технологии и системы связи.

Рекомендовано к печати Ученым советом факультета Инфокоммуникационных технологий, протокол №4 от 13 декабря 2011г.



В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

© Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 2012

© С.В.Одиночкина, 2012.

Лабораторная работа №4: Простой вывод данных. Страницы `blog.php` и `comments.php`.

Упражнение 1: Вывод данных из базы на страницу

В этом упражнении на главную страницу сайта необходимо вывести все заметки из таблицы БД `notes`.

1. Переименуйте `blog.html` в `blog.php` (старую `html`-страницу после переименования можно удалить).
2. Удалите записи на странице. Должны остаться только приветствие и навигационное меню.
3. 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/MySiteDB.php`»):

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

4. Далее необходимо вывести записи (строки) на страницу сайта из таблицы `notes`. Сначала надо реализовать запрос на выборку. Для этого:

4.1. выберите БД;

4.2. 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.

4.3. Реализуйте запрос на выборку.

5. 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.

6. 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>";
```

7. 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 2: Обмен данными между серверными страницами

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. 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»*.