

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 №8: Реализация поиска по сайту

В ходе выполнения данной лабораторной работы будут изучены основные функции работы со строками, а также поиск информации по web-сайту (по одному и нескольким словам поискового запроса).

Упражнение 1: Реализация поиска по сайту

В ходе выполнения данного упражнения с использованием функций работы со строками необходимо реализовать возможность поиска по ключевому слову заметки на главной странице сайта (возможны два варианта реализации: поиск по одному слову и поиск по фразе).

1. Поиск по одному ключевому слову: Для реализации поиска по одному слову можно использовать оператор LIKE и заменители символов %.

Вариант реализации кода:

```
//Поиск по одному слову
$user_search = $_GET['usersearch'];
if (!empty($user_search))
{
    $query_usersearch = "SELECT * FROM notes
        WHERE title LIKE '%$user_search%'
        OR article LIKE '%$user_search%'";
    $result_usersearch = mysqli_query($link, $query_usersearch);
    while ($array_usersearch = mysqli_fetch_array($result_usersearch))
    {
        echo $array_usersearch['id'];
        echo $array_usersearch['title'];
        echo $array_usersearch['article'];
    }
}
```

2. Реализация поиска по фразе:

- 2.1. Фразу надо разбить на отдельные слова (подстроки) и пометить в массив подстрок с помощью функции **explode()**;

```
$search_query = "SELECT * FROM tableName"
$where_clause = ' '; //условие поиска
$user_search = $_GET['usersearch']; // получаем данные из поля
поиска
$search_words = explode(' ', $user_search);

foreach($search_words as $word)
{
```

```

        // Формируем условие поиска
        $where_clause .= " fieldName LIKE '%$word%' OR "
    }
    if (!empty($$where_clause))
    {
        $search_query .= " WHERE $$where_clause ";
    }

```

- 2.2. Для того, чтобы в конце строки запроса не была оператора OR, можно использовать функцию ***implode()***, создающую строку из массива подстрок, переданного ей в качестве аргумента. Представленный ранее код можно изменить следующим образом:

```

$search_query = "SELECT * FROM tableName"
$where_list = array();
$user_search = $_GET['usersearch'];
$search_words = explode(' ', $user_search);

foreach($search_words as $word)
{
    //В конец массива добавляется новый элемент
    $where_list[] = "article LIKE '%$word%'";
}
$where_clause = implode(' OR ', $where_list);

if (!empty($where_clause))
{
    $query_usersearch .= " WHERE $where_clause";
}

```

Вариант реализации кода:

```

//Поиск по фразе (по содержанию заметки)

$user_search = $_GET['usersearch'];
$where_list = array();
$query_usersearch = "SELECT * FROM notes";
$search_words = explode(' ', $user_search);

foreach($search_words as $word)
{
    $where_list[] = " article LIKE '%$word%'";
}

```

```

    }
    $where_clause = implode ( ' OR ', $where_list);
    if (!empty($where_clause))
    {
        $query_usersearch .= " WHERE $where_clause";
    }

    $res_query = mysqli_query($link, $query_usersearch);
    while ($res_array = mysqli_fetch_array($res_query))
    {
        echo $res_array['id'], "<br>";
        echo $res_array['article'], "<br>", "<hr>", "<br>";
    }

```

Упражнение 2: Обработка строки поиска

Строка поиска должна содержать несколько слов, разделенных одним пробелом. Но надо учитывать, что пользователь может вводить слова поискового запроса через запятую (например, «заметка, моя, новая»). Такую строку необходимо перед передачей в запрос в базе данных обработать и привести к необходимому виду. Минимальная обработка осуществляется в два этапа:

- Замена запятых на пробелы;
- Удаление лишних пробелов между словами строки.

1. Замена запятых на пробелы осуществляется с помощью функции ***str_replace()***. Эта функция заменяет строку поиска на строку замены. Обязательными являются три аргумента: что заменить, чем заменить, где заменить. Следовательно, в данном случае вызов функции будет выглядеть следующим образом:

```

str_replace ( ' , ' , ' ', $user_search);

```

2. Удаление лишних пробелов между словами строки поискового запроса: в том случае, когда запятые были заменены на пробелы, появились лишние пробелы (т.е. более одного) между словами строки запроса. Если их не удалить, то в запросе они будут рассматриваться как пустые элементы массива, на основании которого формируется запрос к базе данных. Следовательно, при таком запросе будут выдаваться все записи базы данных. Для удаления пустых элементов массива можно сделать следующее:

- Создать новый массив, в котором будут сохраняться только действительные (непустые) критерии поиска. На основании этого массива будет строиться запрос к базе данных.
- Для создания этого массива можно пройти в цикле *foreach* все элементы уже существующего созданного ранее массива, используя управляющую конструкцию *if* найти все непустые элементы и скопировать их в новый массив.

2.1. В новый массив *\$final_search_words* помещаются непустые элементы уже существующего массива *\$search_words*.

```
//Извлечение критериев поиска в массив
//Замена запятых на пробелы
$clean_search = str_replace(',', ' ', $user_search);
$search_words = explode(' ', $user_search);

//Создаем еще один массив с окончательными результатами
$final_search_words = array();

//Проходим в цикле по каждому элементу массива $search_words.
//Каждый непустой элемент добавляем в массив с названием
//$final_search_words

if (count($search_words) > 0)
{
    foreach($search_words as $word)
    {
        if (!empty($word))
        {
            $final_search_words[] = $word;
        }
    }
}
```

2.2. Далее реализация работы с массивом такая же, разница – в

используемом массиве (работа происходит с новым массивом ***\$final_search_words***).

```
foreach ($final_search_words as $word)
{
    $where_list[] = " article LIKE '%$word%'";
}
$where_clause = implode (' OR ', $where_list);
if (!empty($where_clause))
{
    $query_usersearch .= " WHERE $where_clause";
}
```

Вариант реализации кода:

```
//Поиск по фразе (по содержанию заметки)
$user_search = $_GET['usersearch'];
$where_list = array();
$query_usersearch = "SELECT * FROM notes";
$clean_search = str_replace(',', ' ', $user_search);
$search_words = explode(' ', $user_search);

//Создаем еще один массив с окончательными результатами
$final_search_words = array();

//Проходим в цикле по каждому элементу массива $search_words.
//Каждый непустой элемент добавляем в массив $final_search_words
if (count($search_words) > 0)
{
    foreach($search_words as $word)
    {
        if (!empty($word))
        {
            $final_search_words[] = $word;
        }
    }
}
```

```

    }
    //работа с использованием массива $final_search_words
    foreach($final_search_words as $word) список имеющихся на web-узле
    {изображений и ссылками на эти изображениями), см. рис. 9.1.
        $where_list[] = " article LIKE '%$word%'";
    }
    $where_clause = implode(" and ", $where_list);
    if (!empty($where_clause))
    {
        $sql = "select * from photo where $where_clause";
    }
    $res_query = mysql_query($sql);
    while ($res_row = mysql_fetch_row($res_query))
    {
        echo $res_row[0]."".\$res\_row\[2\]."  
";
    }
}
?>

```

Это страница для работы с изображениями

