

### Лекция 3. Создание меню средствами PHP, HTML, CSS

Без меню не обходится не один ресурс. Меню бывают простыми (одноуровневыми) и сложными (многоуровневыми). Кроме того, пункты меню могут быть как фиксированными (статическими), что опять же является простейшим вариантом, так формироваться исходя из имеющейся информации в БД (динамическими), что в определенных ситуациях является более предпочтительным вариантом, так без изменения кода позволит гибко настраивать сайт. Начнем рассматривать процесс создания меню с самого простого вариант.

В данной лекции рассмотрим несколько стратегий создания меню: от простейшего статического в виде гиперссылок и до динамического, которое автоматически будет формироваться на основе информации, хранящейся в базе данных.

#### 1. Статическое меню (одноуровневое)

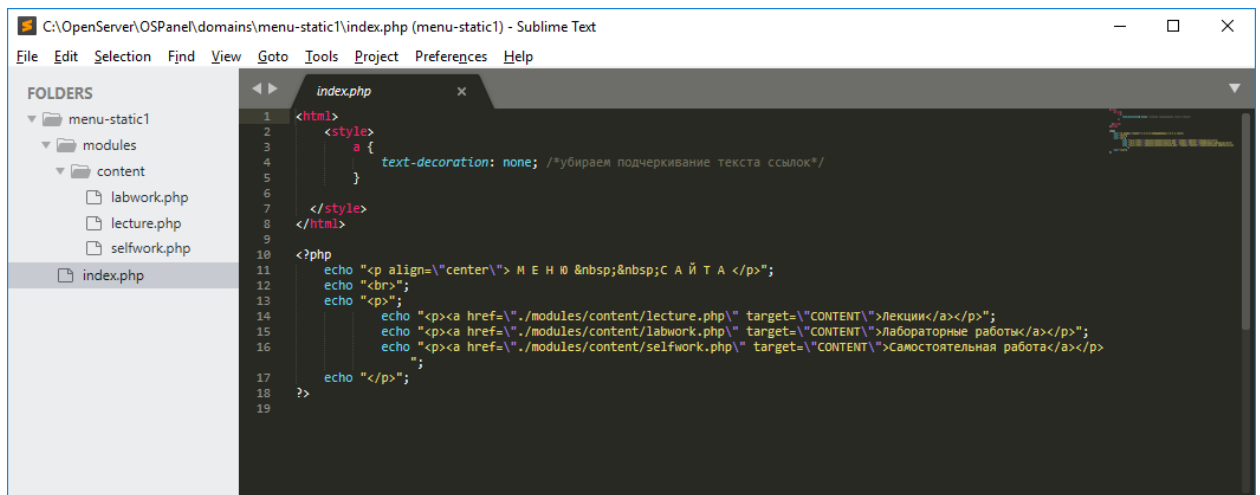
Для начала рассмотрим процесс создания простого одноуровневого меню с фиксированным количеством элементов. Надо сразу отметить, что подобное меню фактически реализуется без использования языка PHP – задействованы, как правило, HTML и CSS.

Фактически это наиболее простой вариант меню, которое удобно создавать в виде набора гиперссылок, отдельно расположенных или в виде списка. Чаще применяют последний вариант. Для придания внешнего вида целесообразно воспользоваться CSS и JS.

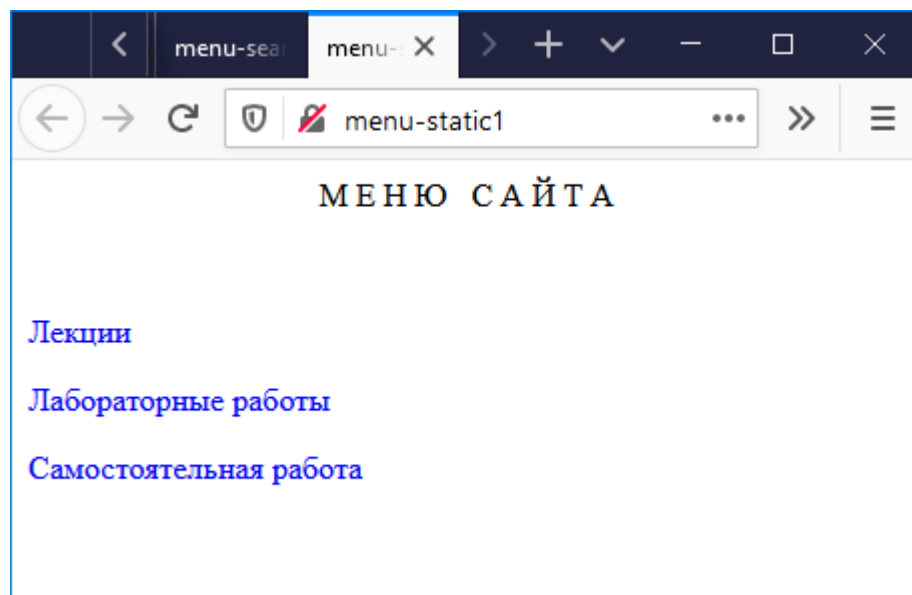
Пример организации меню с фиксированным числом элементов и с переходом на отдельный файл представлен на следующем листинге.

```
<html>
  <style>
    a {
      text-decoration: none; /*убираем подчеркивание текста
ссылок*/
    }
  </style>
</html>

<?php
  echo "<p align=\"center\"> М Е Н Ю &nbsp;&nbsp;&nbsp;С А Й Т А </p>";
  echo "<br>";
  echo "<p>";
      echo "<p><a href=\"./modules/content/lecture.php\"
target=\"CONTENT\">Лекции</a></p>";
      echo "<p><a href=\"./modules/content/labwork.php\"
target=\"CONTENT\">Лабораторные работы</a></p>";
      echo "<p><a href=\"./modules/content/selfwork.php\"
target=\"CONTENT\">Самостоятельная работа</a></p>";
  echo "</p>";
?>
```



```
1 <html>
2   <style>
3     a {
4       text-decoration: none; /*убираем подчеркивание текста ссылок*/
5     }
6   </style>
7 </html>
8
9
10 <?php
11   echo "<p align='center'> М Е Н Ю &nbsp;&nbsp;&nbsp;С А Й Т А </p>";
12   echo "<br>";
13   echo "<p>";
14     echo "<p><a href='./modules/content/lecture.php' target='CONTENT'>Лекции</a></p>";
15     echo "<p><a href='./modules/content/labwork.php' target='CONTENT'>Лабораторные работы</a></p>";
16     echo "<p><a href='./modules/content/selfwork.php' target='CONTENT'>Самостоятельная работа</a></p>";
17   echo "</p>";
18 ?>
```



Очевидно, что для корректного выполнения данного кода требуется наличие файлов, на которые ведут гиперссылки (в представленном коде это: lecture.php, labwork.php, selfwork.php, расположенных в modules/content/).

## 2. Статическое меню (одноуровневое, в виде списка)

Данный вариант достаточно часто применяется на практике при создании простых одноуровневых меню. Он в целом повторяет предыдущий вариант решения, при этом отличается только лишь вариант верстки.

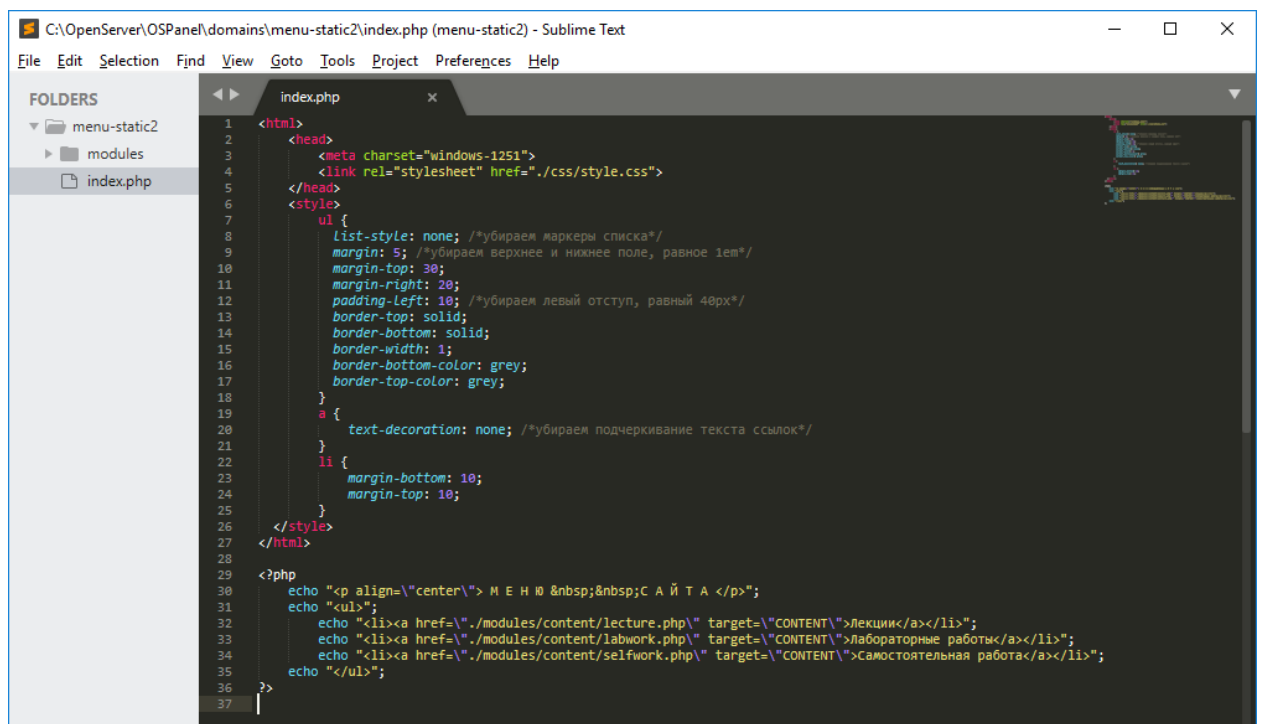
```
<html>
  <head>
    <meta charset="windows-1251">
    <link rel="stylesheet" href="./css/style.css">
  </head>
  <style>
    ul {
      list-style: none; /*убираем маркеры списка*/
      margin: 5; /*убираем верхнее и нижнее поле, равное 1em*/
      margin-top: 30;
      margin-right: 20;
      padding-left: 10; /*убираем левый отступ, равный 40px*/
      border-top: solid;
      border-bottom: solid;
```

```

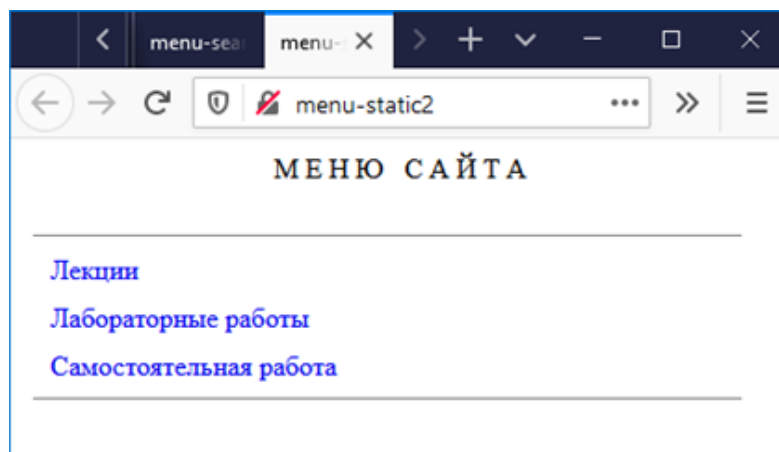
        border-width: 1;
        border-bottom-color: grey;
        border-top-color: grey;
    }
    a {
        text-decoration: none; /*убираем подчеркивание текста
ССЫЛОК*/
    }
    li {
        margin-bottom: 10;
        margin-top: 10;
    }
}
</style>
</html>

<?php
    echo "<p align=\"center\"> М Е Н Ю &nbsp;&nbsp;&nbsp;С А Й Т А </p>";
    echo "<ul>";
        echo "<li><a href=\"./modules/content/lecture.php\"
target=\"CONTENT\">Лекции</a></li>";
        echo "<li><a href=\"./modules/content/labwork.php\"
target=\"CONTENT\">Лабораторные работы</a></li>";
        echo "<li><a href=\"./modules/content/selfwork.php\"
target=\"CONTENT\">Самостоятельная работа</a></li>";
    echo "</ul>";
?>

```



Отметим, что при помощи стилей изменялись параметры стандартных элементов *ul*, *li*, *a*. Выглядит меню в таком случае следующим образом.



Отметим, что программная реализация многоуровневого статического меню фактически выполняется точно также, т.е. не приводит ни к каким изменениям в логике разработки и выполнения скрипта. Визуально вложенность реализуется через настройки параметров верстки.

Также при реализации меню популярностью пользуется следующая стратегия: Элементы меню записываются с какую-либо структуру данных, например, массив. В таком случае можно написать универсальный обработчик, который будет из массива извлекать и выстраивать (отображать) сами элементы меню. Данный обработчик удобно реализовывать в виде функции, на вход которой опраляется непосредственно сам массив. Если необходимо реализовывать одноуровневое меню, достаточно будет одномерного массива строк (строка в таком случае является отдельным элементом меню. Например:

Лекции	Лабораторные работы	Самостоятельная работа
--------	---------------------	------------------------

Если же необходимо выстраивать двухуровневое меню, то уже необходимо будет использовать двумерный массив, при чем некоторое удобство добавит возможность использования ассоциативных двумерных массивов, например, следующей структуры (ключевые поля выделены красным цветом):

	Лекции	Лабораторные работы	Самостоятельная работа
1	Лекция 1	Лабораторная работа 1	
2	Лекция 2	Лабораторная работа 2	
3	Лекция 3		

Необходимо отметить, что элементами меню первого уровня в данном случае являются «ключи» столбцов.

Из вышеприведенного массива элементов меню на практике «должно получиться» меню следующего вида.

Лекции  
 Лекция 1  
 Лекция 2  
 Лекция 3  
 Лабораторные работы  
 Лабораторная работа 1  
 Лабораторная работа 2  
 Самостоятельная работа

### 3. Создание динамического меню (с использованием БД)

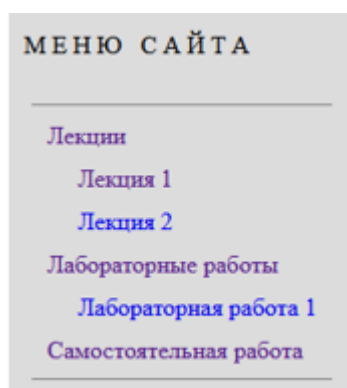
Динамическое меню в отличие от статического, зависит от того, какая информация, касающаяся формирования меню, имеется в таблицах в БД. Так, например, если реализуется условный проект, содержащий учебные материалы в виде лекций и лабораторных работ (данные материалы будут храниться в таблицах lectures и labworks), то формироваться по лекциям и лабораторным работам должно исходя из количества и, например, названий лекций и лабораторных работ. Для этого в БД должны быть созданы таблицы lectures и labworks, в которых, например, отдельное поле будет определять названия пунктов меню. Пусть для обеих таблиц данное поле будет называться name\_menu.

Итак, пусть таблицы lectures и labworks будут иметь следующую структуру.

Сервер: 127.0.0.1:3306 » База данных: my_site » Таблица: lectures										
Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры										
Структура таблицы Связи										
	#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	lecture_id	int(10)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2	menu_name	varchar(64)	cp1251_general_ci		Нет	Нет			
<input type="checkbox"/>	3	topic	varchar(256)	cp1251_general_ci		Нет	Нет			
<input type="checkbox"/>	4	name	varchar(256)	cp1251_general_ci		Нет	Нет			
<input type="checkbox"/>	5	content	text	cp1251_general_ci		Нет	Нет			
<input type="checkbox"/> Отметить все С отмеченными:										

Сервер: 127.0.0.1:3306 » База данных: my_site » Таблица: labworks										
Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры										
Структура таблицы Связи										
	#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	labwork_id	int(10)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2	menu_name	varchar(64)	cp1251_general_ci		Нет	Нет			
<input type="checkbox"/>	3	lecture_id	int(11)			Нет	Нет			
<input type="checkbox"/>	4	name	varchar(256)	cp1251_general_ci		Нет	Нет			
<input type="checkbox"/>	5	content	text	cp1251_general_ci		Нет	Нет			
<input type="checkbox"/> Отметить все С отмеченными:										

Далее реализуем двухуровневое меню следующего вида.



В данном двухуровневом меню родительские категории (Лекции, Лабораторные работы, Самостоятельная работа) будут реализованы в статичном виде, а вот вложенные (дочерние) категории будем реализовывать динамически, т.е. в зависимости от информации в соответствующих таблицах БД.

При разработке будем придерживаться следующего подхода: родительские и дочерние пункты меню формируются через отдельные php-файлы menu.php (для родительского меню) и menu2.php (для дочерних пунктов меню). Для реализации вложенности, очевидно, необходимо будет вызывать файл menu2.php из файла menu.php. При этом, так как в явной форме передавать параметры при вызове файла нельзя, то их будем сохранять в сессии, а уже вызываемый файл menu2.php в зависимости от значения параметров в сессии, будет формировать соответствующие элементы дочернего меню.

Родительское меню будем формировать в виде списков гиперссылок. Однако перед строками, реализующими подключение (вызов) файла menu2.php, в сессию записывается параметр \$\_SESSION['menu'].

```
<ul>
    <li><a href="./modules/content/lecture.php"
target="CONTENT">Лекции</a></li>
    <div>
        <?php
            $_SESSION['menu']='lecture';
            include './modules/menu/menu2.php';
        ?>
    </div>
    <li><a href="./modules/content/labwork.php"
target="CONTENT">Лабораторные работы</a></li>
    <div>
        <?php
            $_SESSION['menu']='labwork';
            include './modules/menu/menu2.php';
        ?>
    </div>
    <li><a href="./modules/content/selfwork.php"
target="CONTENT">Самостоятельная работа</a></li>
</ul>
```

Алгоритм формирования меню второго уровня (дочернего) можно описать следующей последовательностью действий.

Сначала выполняется подключение к БД (хранится в отдельном файле).

```
include './connect/bd.php';
```

Далее в переменную \$table записывается название таблицы, из которой будут извлекаться элементы дочернего меню. Переменной \$table присваивается значение lectures, если \$\_SESSION['menu']=='lecture'. Если же \$\_SESSION['menu']=='labwork', то в переменную \$table записывается labworks.

```
if ($_SESSION['menu']=='lecture') $table=lectures;
if ($_SESSION['menu']=='labwork') $table=labworks;
```

Ну и наконец выполняется запрос следующего вида SELECT menu\_name FROM \$table с последующим формированием элементов дочернего меню на основе полученных результатов запроса.

```
$result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if($result)
    {
        $rows = mysqli_num_rows($result); // количество полученных
строк
        for ($i = 0 ; $i < $rows ; ++$i)
```

```

    {
        $row = mysqli_fetch_row($result);
        if ($_SESSION['menu']=='lecture') echo "<li
class=\"menu2\"><a
href='./modules/content/lecture.php?id=$i'>$row[0]</a></li>";
        if ($_SESSION['menu']=='labwork') echo "<li
class=\"menu2\"><a
href='./modules/content/labwork.php?id=$i'>$row[0]</a></li>";
    }
}

```

Таким образом, файл `menu.php` и `menu2.php` могут быть следующими.

Файл menu.php

```
<html>  
    <head>  
        <meta charset="windows-1251">  
        <link rel="stylesheet" href="./css/style.css">  
    </head>  
    <style>  
        ul {  
            list-style: none; /*убираем маркеры списка*/  
            margin: 5;  
            margin-top: 30;  
            margin-right: 20;  
            padding-left: 10;  
            border-top: solid;  
            border-bottom: solid;  
            border-width: 1;  
            border-bottom-color: grey;  
            border-top-color: grey;  
        }  
        а {  
            text-decoration: none; /*убираем подчеркивание текста  
ссылок*/  
        }  
        li {  
            margin-bottom: 10;  
            margin-top: 10;  
        }  
        .menu2{  
            padding-left: 20;  
        }  
    </style>  
</html>  
  
<p align=\"center\"> М Е Н Ю     С А Й Т А </p>  
<ul>  
    <li><a href=\"./modules/content/lecture.php\"  
target= \"CONTENT\">Лекции</a></li>  
    <div>  
        <?php  
            $_SESSION['menu']='lecture';  
            include './modules/menu/menu2.php';  
        ?>  
    </div>
```

```

        <li><a href="./modules/content/labwork.php"
target="CONTENT">Лабораторные работы</a></li>
    </div>
    <?php
        $_SESSION['menu']='labwork';
        include './modules/menu/menu2.php';
    <?>
    </div>
    <li><a href="./modules/content/selfwork.php"
target="CONTENT">Самостоятельная работа</a></li>
    </div>
</div>
</ul>

```

### Файл menu2.php

```

<?php

    include './connect/bd.php'; //подключаемся к БД
    //Выборка данных из БД в соответствии с запросом и вывод их во
    фрейм
    if ($_SESSION['menu']=='lecture') $table=lectures;
    if ($_SESSION['menu']=='labwork') $table=labworks;

    {
        $query ="SELECT menu_name FROM $table";
        $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
        if($result)
        {
            $rows = mysqli_num_rows($result); // количество полученных
            строк

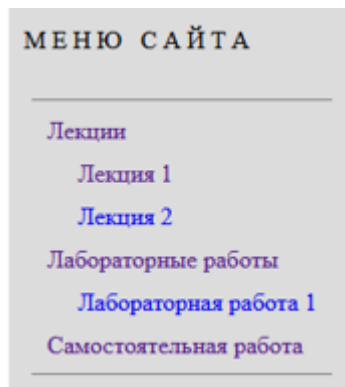
            for ($i = 0 ; $i < $rows ; ++$i)
            {

                $row = mysqli_fetch_row($result);
                if ($_SESSION['menu']=='lecture') echo "<li
class=\"menu2\"><a
href='./modules/content/lecture.php?id=$i'>$row[0]</a></li>";
                if ($_SESSION['menu']=='labwork') echo "<li
class=\"menu2\"><a
href='./modules/content/labwork.php?id=$i'>$row[0]</a></li>";
            }
        }
        mysqli_close($link);
    }
    ?>

```

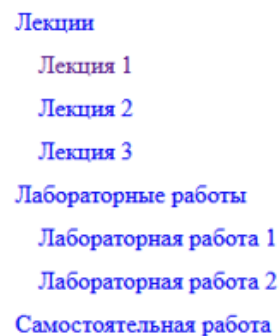
В итоге получим статически сформированное меню первого уровня и динамически сформированное меню второго уровня следующего вида (зависит от записей с соответствующих таблиц БД).





#### 4. Создание динамического многоуровневого меню (с использованием БД)

Концепция динамического многоуровневого меню состоит в том, что, как и в предыдущем случае, все пункты меню (их названия) хранятся в БД, однако должны учитываться уровни вложения. Так, например, представленное на следующем рисунке многоуровневое меню характеризуется двумя уровнями вложения.



На первом уровне (родительском) располагаются пункты меню *Лекции*, *Лабораторные работы*, *Самостоятельная работа*. А в них будут располагаться пункты меню второго уровня. Так, например, в родительском меню *Лекции* есть три пункта дочернего меню, или как говорится меню второго уровня: *Лекция 1*, *Лекция 2* и *Лекция 3*.

При этом необходимо отметить, что потенциально количество уровней вложения, обрабатываемое кодом, не должно быть ограничено.

Для хранения всех элементов многоуровневого меню понадобится отдельная таблица, например `menu` со следующей структурой.

Сервер: 127.0.0.1:3306 » База данных: my\_site » Таблица: menu

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(10)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	
2	title	varchar(32)	utf8_general_ci		Нет	Нет			
3	parent_id	int(10)		UNSIGNED	Нет	Нет			

↑ ☐ Отметить все С отмеченными:

Описание полей таблицы:

- `id` – идентификатор таблицы (как обычно идентификаторы `AUTO_INCREMENT` и `PRIMARY KEY`)
- `title` – заголовок категорий меню

– `parent_id` – идентификатор родительской категории. По умолчанию и если категория родительская, значит значение данного поля 0, для определенной записи. Если же категория дочерняя, то в данном поле указываем идентификатор категории родителя (то есть поля `id`).

Далее заполним базу данных контентом, вставим в таблицу несколько родительских, категорий и несколько уровней дочерних.

Сервер: 127.0.0.1:3306 » База данных: my\_site » Таблица: menu

Обзор Структура SQL Поиск Вставить Экспорт Импорт

Отображение строк 0 - 7 (8 всего, Запрос занял 0,0000 сек.)

```
SELECT * FROM `menu`
```

☐ Показать все | Количество строк: 50 | Фильтровать строки: Поиск в таблице

+ Параметры

				id	title	parent_id
<input type="checkbox"/>				1	Лекции	0
<input type="checkbox"/>				2	Лабораторные работы	0
<input type="checkbox"/>				3	Самостоятельная работа	0
<input type="checkbox"/>				7	Лекция 1	1
<input type="checkbox"/>				8	Лекция 2	1
<input type="checkbox"/>				9	Лекция 3	1
<input type="checkbox"/>				10	Лабораторная работа 1	2
<input type="checkbox"/>				11	Лабораторная работа 2	2

☐ Отметить все | С отмеченными:

Далее перейдем к созданию скрипта, задачей которого будет вывод блока многоуровневого меню.

Для того чтобы вывести многоуровневое меню, без ограничения по уровню вложенности необходимо использовать рекурсию. Рекурсия — это обращение функции к самой себе, другими словами, вызов функции внутри кода функции самой себя.

Данная задача (как и любая в программировании) имеет несколько решений. К примеру, одно из таких решений состоит в том, чтобы создать функцию (принимающая параметр - идентификатор родительской категории), которая в своем коде сформировала SQL запрос по выборке данных меню, по полю `parent_id`. То есть, при первом вызове функции, выбираются все записи, у которых поле `parent_id` = 0, затем `parent_id` = 1, далее `parent_id` = 2 (если таковой имеется) и т.д.

Далее в цикле получаем данные из результата отработки SQL запроса и в этом же цикле вызываем данную функцию (рекурсивно саму на себя) и передаем ей идентификатор родительской категории, полученный у записи на данной итерации цикла. То есть на первой итерации получаем в переменную ассоциативный массив, в этом массиве есть ячейка с ключом `parent_id`, вот ее значение и передаем при вызове функции.

Данный метод, несмотря на простоту и краткость реализации, характеризуется существенным недостатком — происходит многократное обращение к базе данных и выполнение запросов по выборке данных.

Поэтому решим поставленную задачу иначе. Первым делом необходимо обратиться к базе данных и извлечь все категории, которые в ней содержатся. Далее из этих категорий сформировать массив определенного вида. И на последнем этапе необходимо выполнить обработку данного массива и вывод категорий (в виде меню) на экран. Отметим, что как обработку массива, так и вывод меню на экран целесообразно выполнять в виде функций (в таком случае интеграция реализации многоуровневого динамического меню в проект будет проходить легче).

Рассмотрим реализацию указанных функций. Первая из них – `get_category()`, которая будет решать вопросы извлечения и сортировки всех категорий (уровней) меню. Рассмотрим представленный код функции.

```
function get_category()
{
    include './connect/bd.php';
    //запрос к базе данных
    $query = "SELECT * FROM menu";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if($result)
    {
        $arr_cat = array();
        if(mysqli_num_rows($result) != 0)
        {
            //В цикле формируем массив
            for($i = 0; $i < mysqli_num_rows($result); $i++)
            {
                $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
                //Формируем массив, где ключами являются id на
родительские категории
                if(empty($arr_cat[$row['parent_id']])) {
                    $arr_cat[$row['parent_id']] = array();
                }
                $arr_cat[$row['parent_id']][] = $row;
            }
        }
        //возвращаем массив
        return $arr_cat;
    }
}
```

***`mysqli_fetch_array()`** является расширенной версией функции **`mysqli_fetch_row()`**. Вдобавок к сохранению данных в обычном массиве, **`mysqli_fetch_array()`** может сохранять строки в ассоциативном массиве, где имена полей результирующей таблицы будут ключами элементов.*

И так, вначале подключается файл, содержащий подключение к БД. Далее формируется и выполняется SQL запрос по выборке всех данных из таблицы *menu*, далее проверяется, успешно ли выполнен запрос. Далее можно было бы просто при помощи цикла пройти по результату, полученному из базы данных и создать ассоциативный массив данных по категориям. Но такой массив очень неудобно обрабатывать, поэтому его необходимо отсортировать в удобном порядке. Нужно сделать так, чтобы на выходе данной функции, получился массив, **ключами которого были бы идентификаторы родительских категорий**. И в каждой ячейке данного массива, содержался бы еще массив – всех категорий (дочерних), у которых идентификатор родительской категории равен ключу соответствующего массива. Другими словами, отсортируем массив по дочерним категориям. Для этого используется цикл **for**, в котором будут поочередно проходить по каждой строке результирующей таблицы, полученной из базы данных. Затем в переменную `$row`, сохранятся массив данных – одной строки результирующей таблицы. А дальше формироваться массив `$arr_cat`, у которого ключи соответствуют идентификаторам родительских категорий. И наконец, в завершении возвращается массив `$arr_cat`, как результат работы функции в целом.

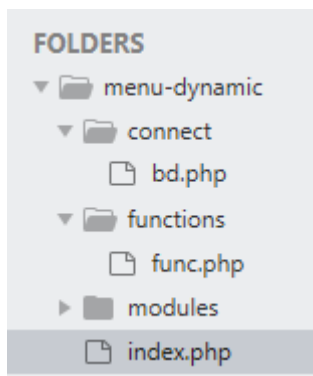
Далее необходимо создать функцию, которая будет выводить данные категории в виде многоуровневого меню. На данном этапе удобно воспользоваться рекурсией.

```
//вывод каталога с помощью рекурсии
function view_category($arr,$parent_id = 0)
{
    //Условия выхода из рекурсии
    if(empty($arr[$parent_id])) {
        return;
    }
    echo '<ul>';
    //перебираем в цикле массив и выводим на экран
    for($i = 0; $i < count($arr[$parent_id]);$i++) {
        echo '<li><a href="?category_id='.$arr[$parent_id][$i]['id'].
        '&parent_id='.$parent_id.'">'
        . $arr[$parent_id][$i]['title'].'</a>';
        //рекурсия - проверяем нет ли дочерних категорий
        view_category($arr,$arr[$parent_id][$i]['id']);
        echo '</li>';
    }
    echo '</ul>';
}
```

Далее остается только из необходимого места в коде осуществить вызов соответствующей функции и отправить результаты на экран, например, в соответствующий блок *div*.

```
<?php
    //Выводим каталог на экран с помощью рекурсивной функции
    echo '<div style="padding:10px; border-bottom:1px solid
#074776">';
    echo 'МЕНЮ САЙТА';
    echo '</div>';
    $result=get_category();
    echo "<div>";
    view_category($result);
    echo "</div>";
?>
```

Реализуем многоуровневое динамическое меню в виде проекта со следующей структурой.



Функции, которые будут выполнять извлечение и необходимую сортировку, а также вывод меню на экран, будут храниться в виде отдельного файла `func.php` в каталоге `functions`.

Подключение к БД будет традиционно описано также в виде отдельного файла `bd.php`, который предполагается хранить в папке `connect`.

Стартовый файл `index.php` фактически будет задавать разметку, подключать файл с функциями и далее вызывать их в необходимой последовательности. Далее приведем листинг файла `index.php`.

Далее представим полные листинги всех трех файлов: `index.php`, `func.php` и `bd.php`.

#### Файл `index.php`

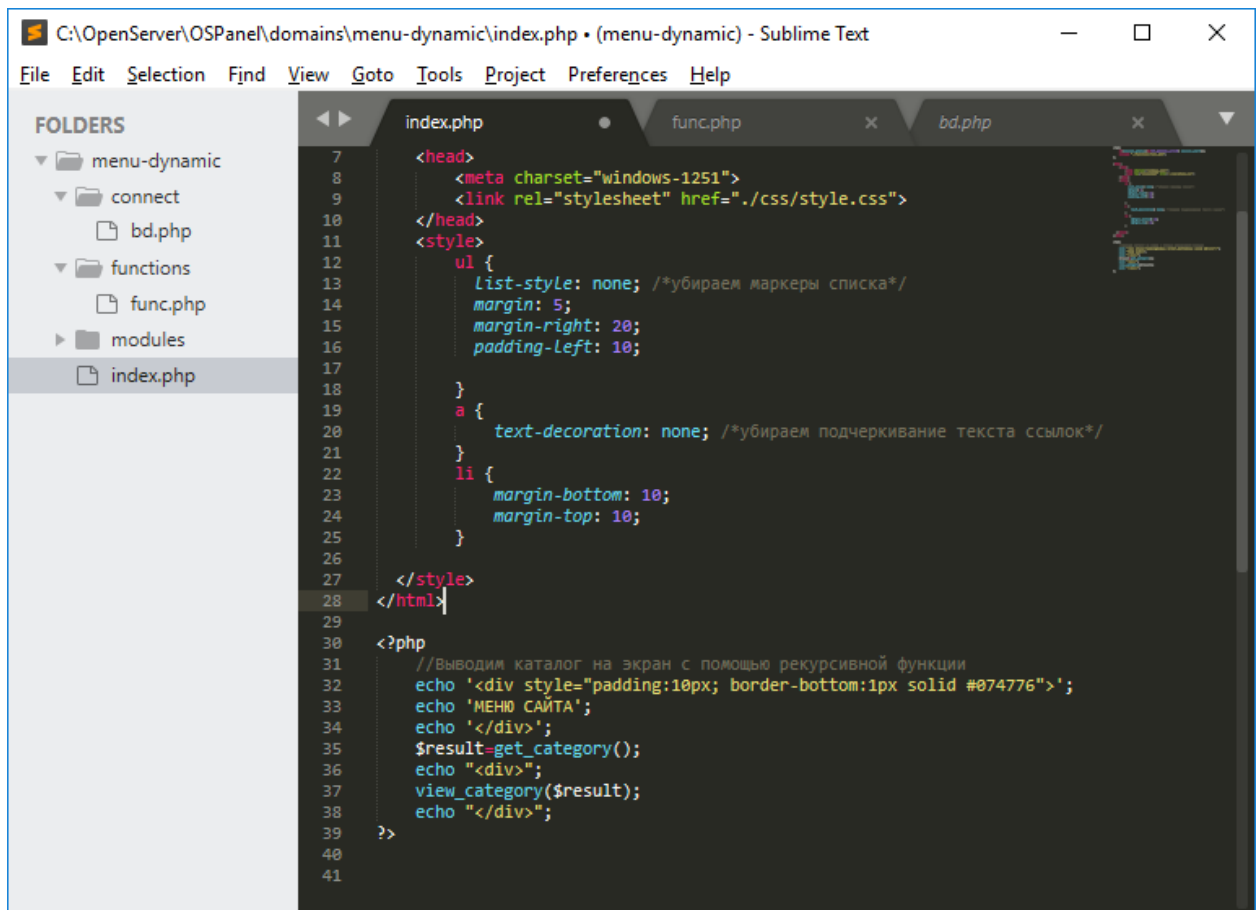
```
<?php
    if(session_status() != PHP_SESSION_ACTIVE) session_start();
    include '../functions/func.php';
?>

<html>
    <head>
        <meta charset="windows-1251">
        <link rel="stylesheet" href="../css/style.css">
    </head>
    <style>
        ul {
            list-style: none; /*убираем маркеры списка*/
            margin: 5;
            margin-right: 20;
            padding-left: 10;

        }
        a {
            text-decoration: none; /*убираем подчеркивание текста
ССЫЛОК*/
        }
        li {
            margin-bottom: 10;
            margin-top: 10;
        }

    </style>
</html>

<?php
    //Выводим каталог на экран с помощью рекурсивной функции
    echo '<div style="padding:10px; border-bottom:1px solid
#074776">';
    echo 'МЕНЮ САЙТА';
    echo '</div>';
    $result=get_category();
    echo "<div>";
    view_category($result);
    echo "</div>";
?>
```



Файл func.php

```

<?php
function get_category(){
    include './connect/bd.php';
    //запрос к базе данных
    $query ="SELECT * FROM menu";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if($result){
        $arr_cat = array();
        if(mysqli_num_rows($result) != 0){
            //В цикле формируем массив
            for($i = 0; $i < mysqli_num_rows($result); $i++){
                $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
                //Формируем массив, где ключами являются адишники на
родительские категории
                if(empty($arr_cat[$row['parent_id']])) {
                    $arr_cat[$row['parent_id']] = array();
                }
                $arr_cat[$row['parent_id']][] = $row;
            }
        }
        //возвращаем массив
        return $arr_cat;
    }
}
function view_category($arr,$parent_id = 0){
    //Условия выхода из рекурсии
    if(empty($arr[$parent_id])) {

```

```

        return;
    }
    echo '<ul>';
    //перебираем в цикле массив и выводим на экран
    for($i = 0; $i < count($sarr[$parent_id]);$i++) {
        echo '<li><a
href="./modules/content/view.php?category_id='.$sarr[$parent_id][$i]['id'].
    'parent_id='.$parent_id.'"target="CONTENT">'
    . $sarr[$parent_id][$i]['title'].'</a>';
        //рекурсия - проверяем нет ли дочерних категорий
        view_category($sarr,$sarr[$parent_id][$i]['id']);
        echo '</li>';
    }
    echo '</ul>';
}
?>

```

C:\OpenServer\OSPanel\domains\menu-dynamic\functions\func.php (menu-dynamic) - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- menu-dynamic
  - connect
    - bd.php
  - functions
    - func.php
  - modules
    - index.php

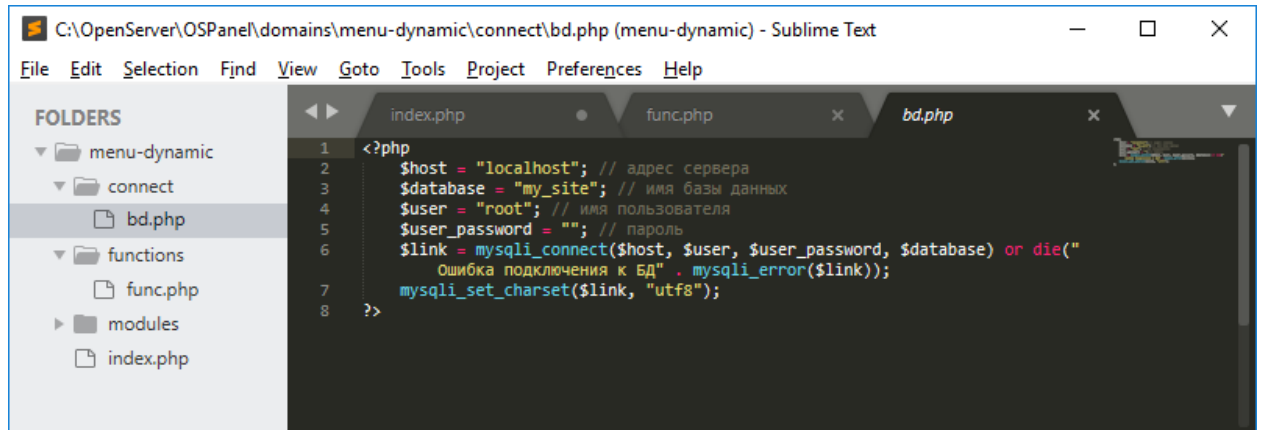
```

1  <?php
2
3  function get_category()
4  {
5      include './connect/bd.php';
6      //запрос к базе данных
7      $query = "SELECT * FROM menu";
8      $result = mysqli_query($link, $query) or die("Ошибка " . mysqli_error($link));
9
10     if($result)
11     {
12         $sarr_cat = array();
13         if(mysqli_num_rows($result) != 0)
14         {
15             //В цикле формируем массив
16             for($i = 0; $i < mysqli_num_rows($result);$i++)
17             {
18                 $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
19                 //формируем массив, где ключами являются адишники на
20                 //родительские категории
21                 if(empty($sarr_cat[$row['parent_id']])) {
22                     $sarr_cat[$row['parent_id']] = array();
23                 }
24                 $sarr_cat[$row['parent_id']][] = $row;
25             }
26             //возвращаем массив
27             return $sarr_cat;
28         }
29     }
30
31     function view_category($sarr,$parent_id = 0)
32     {
33         //Условия выхода из рекурсии
34         if(empty($sarr[$parent_id])) {
35             return;
36         }
37         echo '<ul>';
38         //перебираем в цикле массив и выводим на экран
39         for($i = 0; $i < count($sarr[$parent_id]);$i++) {
40             echo '<li><a href="./modules/content/view.php?category_id='.$sarr[$
41             parent_id][$i]['id'].
42             '&parent_id='.$parent_id.'"target="CONTENT">'
43             . $sarr[$parent_id][$i]['title'].'</a>';
44             //рекурсия - проверяем нет ли дочерних категорий
45             view_category($sarr,$sarr[$parent_id][$i]['id']);
46             echo '</li>';
47         }
48         echo '</ul>';
49     }
50 }
?>

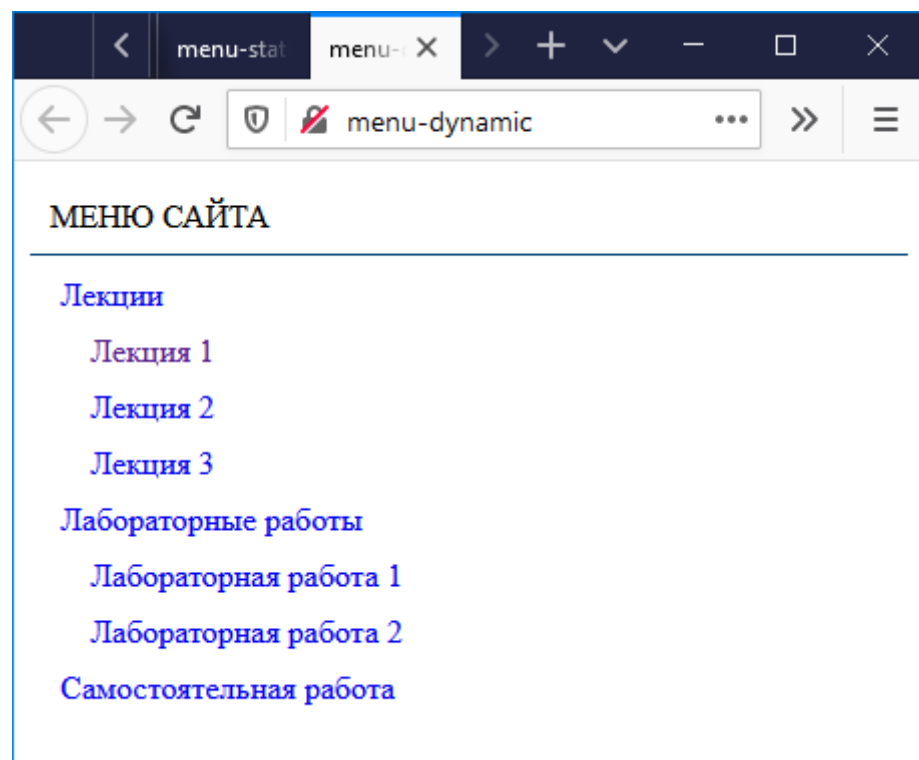
```

Файл, описывающий подключение к БД (bd.php), будет следующим.

```
<?php
    $host = "localhost"; // адрес сервера
    $database = "my_site"; // имя базы данных
    $user = "root"; // имя пользователя
    $user_password = ""; // пароль
    $link = mysqli_connect($host, $user, $user_password, $database) or
die("Ошибка подключения к БД" . mysqli_error($link));
    mysqli_set_charset($link, "utf8");
?>
```



В конечном варианте пользователь на своем экране будет видеть многоуровневое меню в следующем виде.



Далее, например, увеличим количество уровней вложения до трех, как показано в следующей таблице menu.



Сервер: 127.0.0.1:3306 » База данных: my\_site » Таблица:

Обзор Структура SQL Поиск Вставить

✔ Отображение строк 0 - 10 (11 всего, Запрос занял 0,0000 сек.)

`SELECT * FROM `menu``

☐ Профилирование [Построчное редактирование]

☐ Показать все | Количество строк: 50 | Фильтровать строки

+ Параметры

				id	title	parent_id
<input type="checkbox"/>				1	Лекции	0
<input type="checkbox"/>				2	Лабораторные работы	0
<input type="checkbox"/>				3	Самостоятельная работа	0
<input type="checkbox"/>				7	Лекция 1	1
<input type="checkbox"/>				8	Лекция 2	1
<input type="checkbox"/>				9	Лекция 3	1
<input type="checkbox"/>				10	Лабораторная работа 1	2
<input type="checkbox"/>				11	Лабораторная работа 2	2
<input type="checkbox"/>				13	Лекция 4	1
<input type="checkbox"/>				21	Дополнительные задания 1	10
<input type="checkbox"/>				22	Дополнительные задания 2	10

↑ ☐ Отметить все С отмеченными:

Выполнение проекта с таким содержимым таблицы меню пользователю отобразится следующее меню.

127.0.0.1 / 127 menu-dynamic × + - □ ×

← → ↻ menu-dynamic ☆ >> ≡

МЕНЮ САЙТА

Лекции
Лекция 1
Лекция 2
Лекция 3
Лекция 4
Лабораторные работы
Лабораторная работа 1
Дополнительные задания 1
Дополнительные задания 2
Лабораторная работа 2
Самостоятельная работа

Таким образом получили достаточно универсальную организацию меню, категории которого хранятся в отдельной таблице в БД. Количество уровней вложения зависит только от информации в соответствующей таблице БД. Изменяя ее, например, через панель администрирования, можно получать любой требуемый вариант меню.

## ПРАКТИКА ИНТЕГРАЦИИ РАЗРАБОТАННЫХ ВАРИАНТОВ РЕАЛИЗАЦИИ МЕНЮ В ПРОЕКТ «УМК»

Отметим, что реализовывать различные варианты меню будем в пределах структуры, проекта с регистрацией и авторизацией (рассмотрен в предыдущих лекциях). Отметим, что в основной части сайта будут использованы два блока `div`, один из которых (левый) предназначен для отображения непосредственно меню, а второй (правый) – для отображения контента после перехода пользователя по пунктам меню.



Таким образом разметка в наиболее простом виде может быть выполнена следующим образом (отметим, что стили для блоков `div` в листинге не представлены).

```
<?php
if (session_id()=='')
    session_start();

?>
<html>
  <head>
    <meta charset="windows-1251">
    <link rel="stylesheet" href="css/style.css">
```

```

</head>
<body>
    <div class="header">
        <?php
            include ("header.html");
        ?>
    </div>
    <div class="menu">
        <?php
            include ("modules/menu/menu.php");
        ?>
    </div>
    <div class="content">
        <iframe alloytransparency name="CONTENT"
align="left" width="100%" height="100%" style="border: 0;">Ваш браузер
не поддерживает iframe</iframe>
    </div>
    <div class="footer">
        <a class="footer_content">
            <?php
                include ("footer.html");
            ?>
        </a>
    </div>
</body>
</html>

```

### 1. Создание одноуровневого меню

Это наиболее простой вариант меню, которое удобно создавать в виде набора гиперссылок, отдельно расположенных или в виде списка. Чаще применяют последний вариант. Для придания внешнего вида целесообразно воспользоваться CSS и JS.

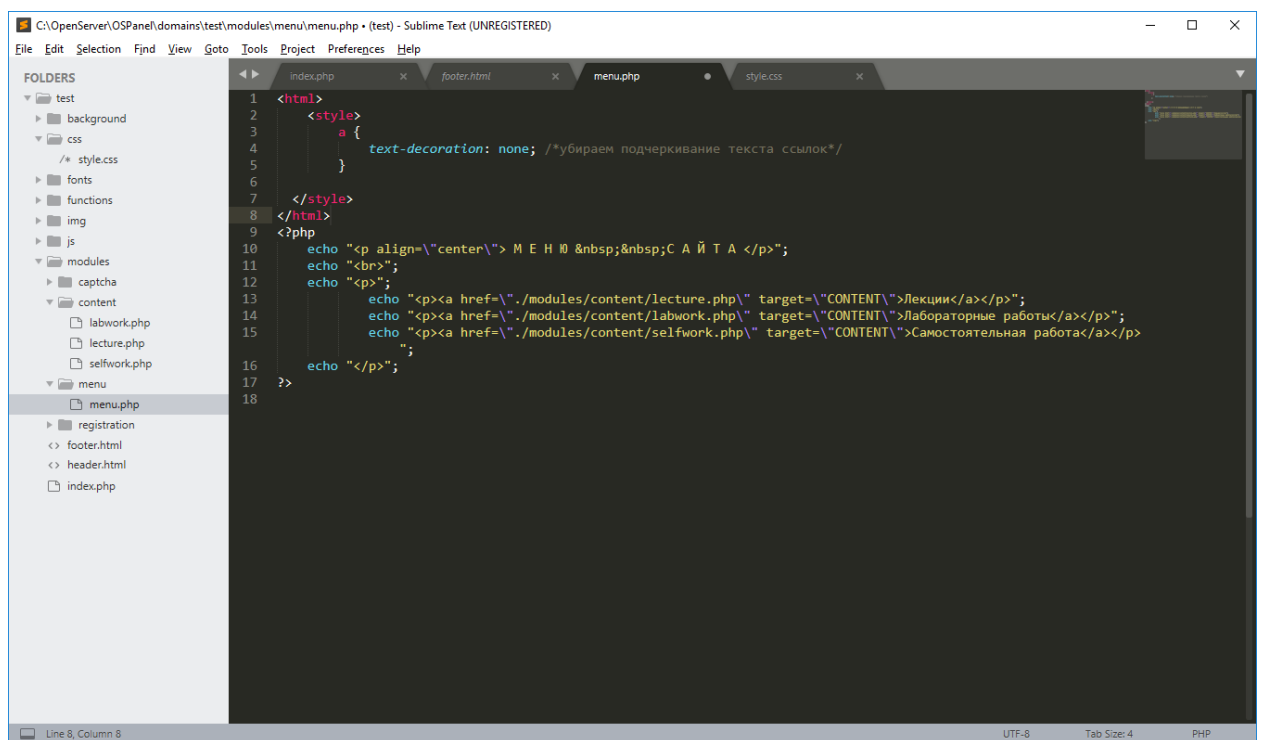
Пример организации меню с фиксированным числом элементов и с переходом на отдельный файл представлен на следующем листинге.

```

<html>
    <style>
        a {
            text-decoration: none; /*убираем подчеркивание текста
ссылок*/
        }
    </style>
</html>

<?php
    echo "<p align=\"center\"> М Е Н Ю &nbsp;&nbsp;&nbsp;С А Й Т А </p>";
    echo "<br>";
    echo "<p>";
        echo "<p><a href=\"./modules/content/lecture.php\"
target=\"CONTENT\">Лекции</a></p>";
        echo "<p><a href=\"./modules/content/labwork.php\"
target=\"CONTENT\">Лабораторные работы</a></p>";
        echo "<p><a href=\"./modules/content/selfwork.php\"
target=\"CONTENT\">Самостоятельная работа</a></p>";
    echo "</p>";
?>

```



```
1 <html>
2 <style>
3   a {
4     text-decoration: none; /*убираем подчеркивание текста ссылок*/
5   }
6 </style>
7 </html>
8 <?php
9   echo "<p align='center'> М Е Н Ю &nbsp;&nbsp;&nbsp;С А Й Т А </p>";
10  echo "<br>";
11  echo "<p>";
12    echo "<a href='./modules/content/lecture.php' target='CONTENT'>Лекции</a></p>";
13    echo "<a href='./modules/content/labwork.php' target='CONTENT'>Лабораторные работы</a></p>";
14    echo "<a href='./modules/content/selfwork.php' target='CONTENT'>Самостоятельная работа</a></p>";
15  echo "</p>";
16 >>
```

Очевидно, что для корректного выполнения данного кода требуется наличие файлов lecture.php, labwork.php, selfwork.php в modules/content/ .

В результате получим следующее.

## Создание МЕНЮ



## Создание МЕНЮ

МЕНЮ САЙТА

Это мои лабораторные работы

Лекции

Лабораторные работы

Самостоятельная работа

Романенко Дмитрий, 2020

## Создание МЕНЮ

МЕНЮ САЙТА

Это материалы для самостоятельной работы

Лекции

Лабораторные работы

Самостоятельная работа

Романенко Дмитрий, 2020

## 2. Создание одноуровневого меню на основе списков гиперссылок

Данный вариант достаточно часто применяется на практике при создании простых одноуровневых меню. Он в целом повторяет предыдущий вариант решения, при этом отличается только лишь содержимое файла `menu.php`.

```
<html>
  <head>
    <meta charset="windows-1251">
    <link rel="stylesheet" href="./css/style.css">
  </head>
  <style>
    ul {
      list-style: none; /*убираем маркеры списка*/
      margin: 5; /*убираем верхнее и нижнее поле, равное 1em*/
      margin-top: 30;
      margin-right: 20;
      padding-left: 10; /*убираем левый отступ, равный 40px*/
      border-top: solid;
      border-bottom: solid;
      border-width: 1;
      border-bottom-color: grey;
      border-top-color: grey;
    }
    a {
      text-decoration: none; /*убираем подчеркивание текста
ССЫЛОК*/
    }
    li {
      margin-bottom: 10;
      margin-top: 10;
    }
  </style>
</html>

<?php
  echo "<p align=\"center\"> М Е Н Ю &nbsp;&nbsp;&nbsp;С А Й Т А </p>";
  echo "<ul>";
    echo "<li><a href=\"./modules/content/lecture.php\"
target=\"CONTENT\">Лекции</a></li>";
    echo "<li><a href=\"./modules/content/labwork.php\"
target=\"CONTENT\">Лабораторные работы</a></li>";
    echo "<li><a href=\"./modules/content/selfwork.php\"
target=\"CONTENT\">Самостоятельная работа</a></li>";
    echo "</ul>";
?>
```

Отметим, что при помощи стилей изменялись параметры стандартных элементов *ul*, *li*, *a*. Для этого, чтобы не влиять на отображение данных элементов во всем проекте, параметры стилей прописывались именно в самом файле `menu.php`, а не в общем файле.

## Создание МЕНЮ



### 3. Создание динамического меню (на основе информации из БД)

Динамически, т.е. в зависимости от того, какая информация имеется в таблицах в БД (в данном случае в таблицах lectures и labworks) будут формироваться подменю по лекциям и лабораторным работам. Для этого в БД в таблицах lectures и labworks необходимо добавить поле, информация в котором и будет использоваться для формирования меню второго уровня. Пусть для обеих таблиц данное поле будет называться name\_menu.

Сервер: 127.0.0.1:3306 » База данных: my\_site » Таблица: lectures

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	lecture_id	int(10)	UNSIGNED	Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2	menu_name	varchar(64)	cp1251_general_ci	Нет	Нет			
<input type="checkbox"/>	3	topic	varchar(256)	cp1251_general_ci	Нет	Нет			
<input type="checkbox"/>	4	name	varchar(256)	cp1251_general_ci	Нет	Нет			
<input type="checkbox"/>	5	content	text	cp1251_general_ci	Нет	Нет			

☐ Отметить все С отмеченными:

Сервер: 127.0.0.1:3306 » База данных: my\_site » Таблица: labworks

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	labwork_id	int(10)	UNSIGNED	Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2	menu_name	varchar(64)	cp1251_general_ci	Нет	Нет			
<input type="checkbox"/>	3	lecture_id	int(11)		Нет	Нет			
<input type="checkbox"/>	4	name	varchar(256)	cp1251_general_ci	Нет	Нет			
<input type="checkbox"/>	5	content	text	cp1251_general_ci	Нет	Нет			

☐ Отметить все С отмеченными:

В таком случае скрипт menu2.php будет извлекать из соответствующей таблицы (в зависимости от того, по какому пункту меню первого уровня пользователь щелкнул)

содержимое и формировать из него, например, через список, пункты меню второго уровня. Пусть данный список будет отображать в линию непосредственно над блоком с выводимым контентом (материалы лекций или лабораторных работ).

#### Файл menu2.php

```
<html>
  <style>
    ul {
      list-style: none; /*убираем маркеры списка*/
      margin: 5; /*убираем верхнее и нижнее поле, равное 1em*/
      margin-top: 30;
      margin-right: 20;
      padding-left: 10; /*убираем левый отступ, равный 40px*/
      border-bottom: solid;
      border-width: 1;
      border-bottom-color: #66CCCC;
    }
    a {
      text-decoration: none; /*убираем подчеркивание текста
ССЫЛОК*/
    }
    li {
      display: inline; /* Отображать как строчный элемент */
      margin-right: 5px; /* Отступ слева */
      margin-bottom: 10;
      margin-top: 10;
      padding-bottom: 10px;
    }
  </style>
</html>

<?php
  include '../..../connect/bd.php'; //подключаемся к БД
  //Выборка данных из БД в соответствии с запросом и вывод их во
  фрейм
  if ($_SESSION['menu']=='lecture') $table=lectures;
  if ($_SESSION['menu']=='labwork') $table=labworks;
  {
    $query ="SELECT menu_name FROM $table";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if($result)
    {
      $rows = mysqli_num_rows($result); // количество полученных
      строк

      echo "<ul>";
      for ($i = 0 ; $i < $rows ; ++$i)
      {

        $row = mysqli_fetch_row($result);
        if ($_SESSION['menu']=='lecture') echo "<li><a
href='../modules/content/lecture.php?id=$i'>$row[0]</a></li>";
        if ($_SESSION['menu']=='labwork') echo "<li><a
href='../modules/content/labwork.php?id=$i'>$row[0]</a></li>";

      }
      echo "</ul>";
    }
  }
```



```

    }
}
// закрываем подключение
mysqli_close($link);
?>

```

Файл, создающий меню первого уровня в таком случае, не претерпит изменений.

Файл menu.php

```

<html>
  <head>
    <meta charset="windows-1251">
    <link rel="stylesheet" href="./css/style.css">
  </head>
  <style>
    ul {
      list-style: none; /*убираем маркеры списка*/
      margin: 5; /*убираем верхнее и нижнее поле, равное 1em*/
      margin-top: 30;
      margin-right: 20;
      padding-left: 10; /*убираем левый отступ, равный 40px*/
      border-top: solid;
      border-bottom: solid;
      border-width: 1;
      border-bottom-color: grey;
      border-top-color: grey;
    }
    a {
      text-decoration: none; /*убираем подчеркивание текста
ССЫЛОК*/
    }
    li {
      margin-bottom: 10;
      margin-top: 10;
    }
  </style>
</html>

<?php
  echo "<p align=\"center\"> М Е Н Ю &nbsp;&nbsp;&nbsp;С А Й Т А </p>";
  echo "<ul>";
    echo "<li><a href=\"./modules/content/lecture.php\"
target=\"CONTENT\">Лекции</a></li>";
    echo "<li><a href=\"./modules/content/labwork.php\"
target=\"CONTENT\">Лабораторные работы</a></li>";
    echo "<li><a href=\"./modules/content/selfwork.php\"
target=\"CONTENT\">Самостоятельная работа</a></li>";
  echo "</ul>";
?>

```

На практике после выбора пользователем пункта меню первого уровня (лекции, лабораторные работы) выполняется открытие соответствующего файла (lecture.php, labwork.php) в соответствующем блоке, а уже в самом файле сначала вызывается файл, формирующий меню второго уровня, а затем выполняются строки кода, выводящие контент из БД (в скриптах lecture.php и labwork.php вывод контента из БД не выполнялся).

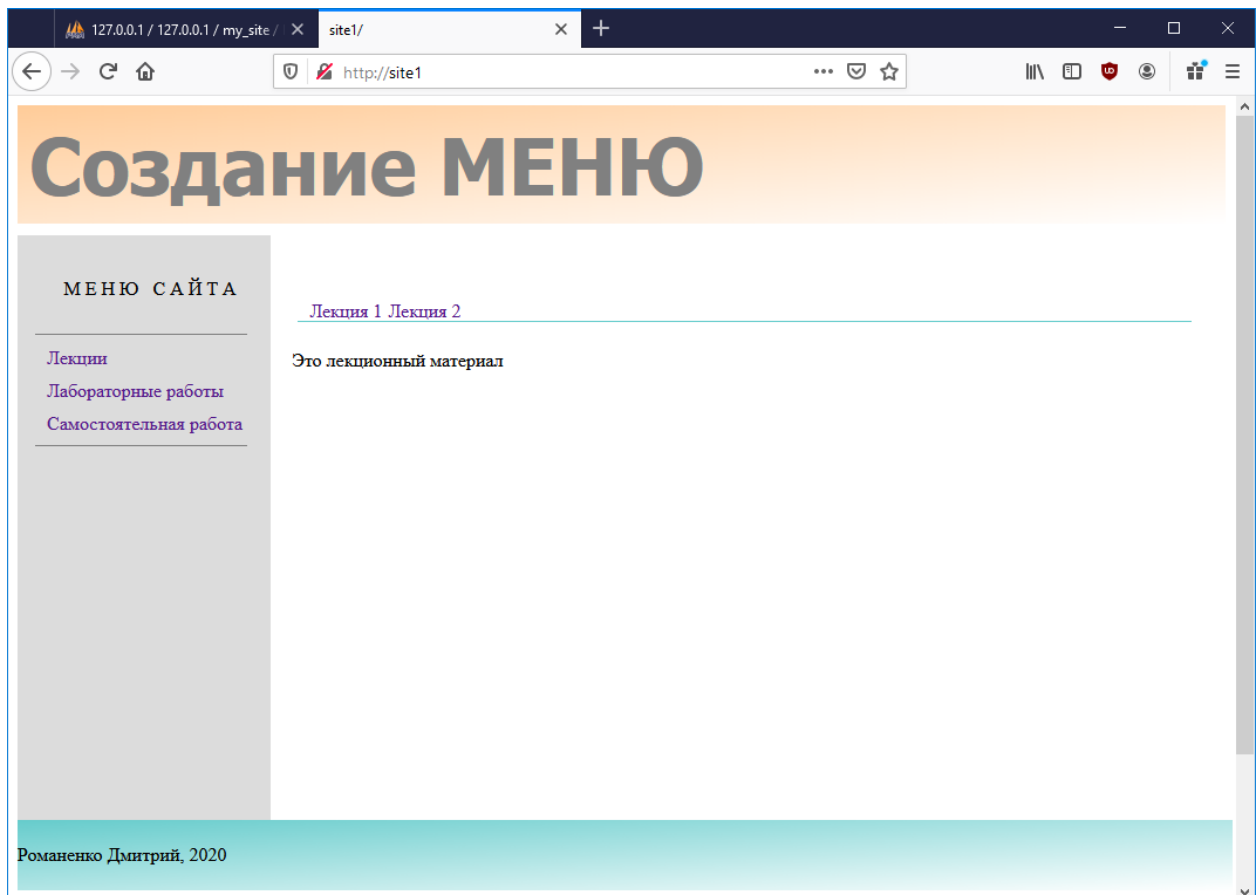
### Файл lecture.php

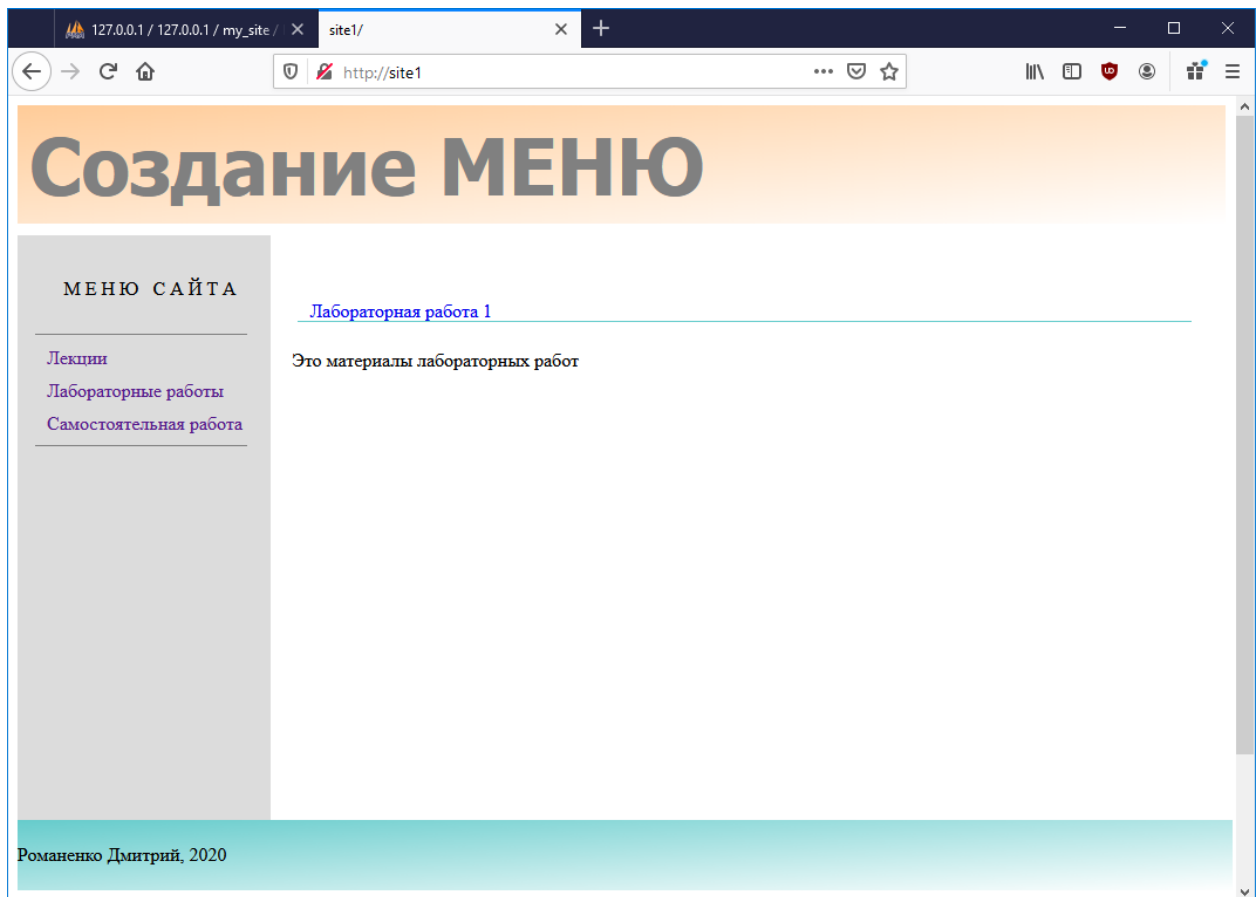
```
<?php
    $_SESSION['menu']='lecture';
    include '../modules/menu/menu2.php';
?>
<br>
<?php
    echo "Это лекционный материал";
?>
```

### Файл labwork.php

```
<?php
    $_SESSION['menu']='labwork';
    include '../modules/menu/menu2.php';
?>
<br>

<?php
    echo "Это материалы лабораторных работ";
?>
```





Рассмотрим другой вариант организации меню, использующего информацию из БД – меню второго уровня будет располагаться классически вслед за соответствующим меню первого уровня. Для этого необходимо для начала изменить верстку файла, отвечающего за формирование меню, т.е. `menu.php`, а именно для вложенных подменю добавим отдельные блоки `div`, которые вы сможете в дальнейшем при желании скрывать и раскрывать средствами JS.

#### Файл `menu.php`

```
<html>
  <head>
    <meta charset="windows-1251">
    <link rel="stylesheet" href="./css/style.css">
  </head>
  <style>
    ul {
      list-style: none; /*убираем маркеры списка*/
      margin: 5;
      margin-top: 30;
      margin-right: 20;
      padding-left: 10;
      border-top: solid;
      border-bottom: solid;
      border-width: 1;
      border-bottom-color: grey;
      border-top-color: grey;
    }
  </style>
</html>
```

[illegible]

Отметим, что в блоках `div` для каждого вложенного подменю пишется `php`-скрипт, который, определяет по какой таблице будем выбирать пункты подменю (`$_SESSION['menu']='lecture'` или `$_SESSION['menu']='labwork'`) и далее непосредственно подключаем файл `menu2.php`.

Файл menu2.php

```
<?php
    include './connect/bd.php'; //подключаемся к БД
    //Выборка данных из БД в соответствии с запросом и вывод их во
    фрейм
    if ($_SESSION['menu']=='lecture') $table=lectures;
    if ($_SESSION['menu']=='labwork') $table=labworks;

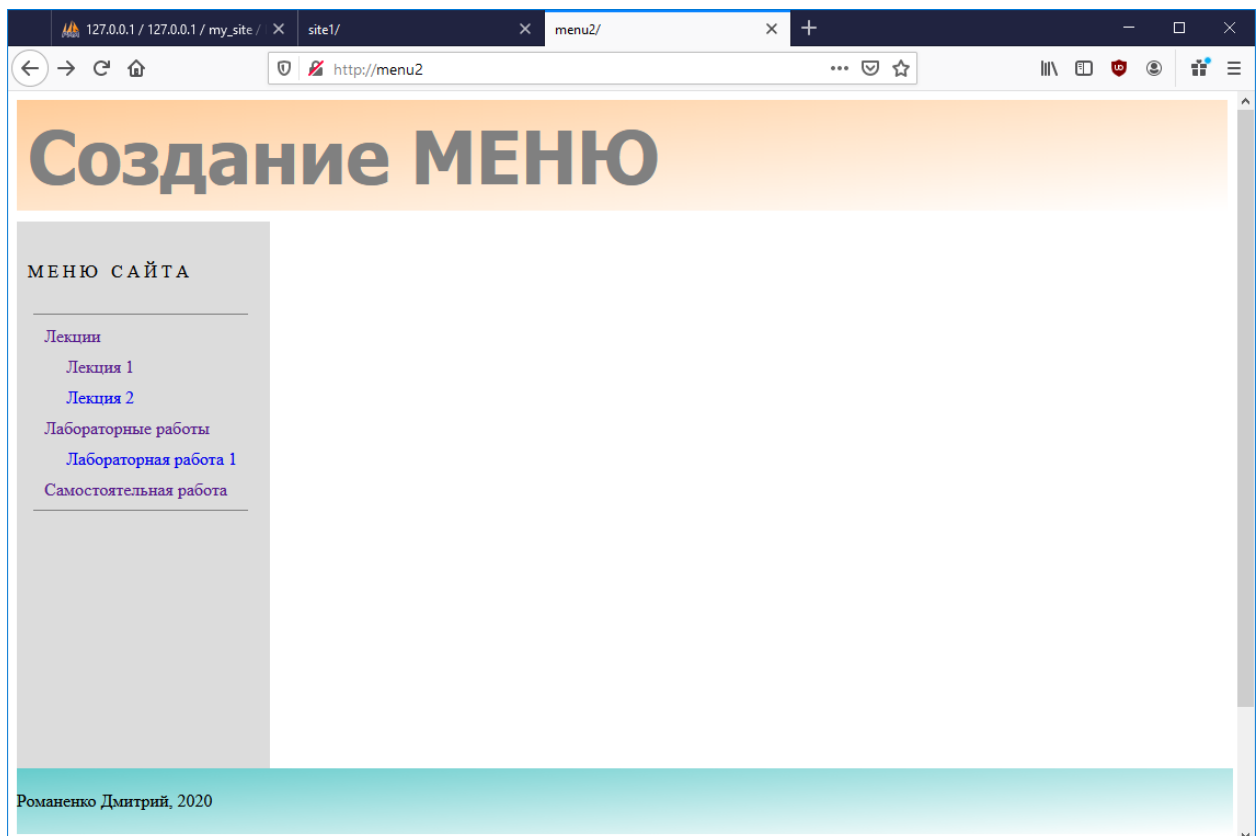
    {
        $query ="SELECT menu name FROM $table";
```

```

        $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
        if($result)
        {
            $rows = mysqli_num_rows($result); // количество полученных
строк
            for ($i = 0 ; $i < $rows ; ++$i)
            {

                $row = mysqli_fetch_row($result);
                if ($_SESSION['menu']=='lecture') echo "<li
class=\"menu2\"><a
href='./modules/content/lecture.php?id=$i'>$row[0]</a></li>";
                if ($_SESSION['menu']=='labwork') echo "<li
class=\"menu2\"><a
href='./modules/content/labwork.php?id=$i'>$row[0]</a></li>";
            }
        }
        // закрываем подключение
        mysqli_close($link);
    ?>

```



При этом если добавить данные в БД, например, еще одну лекцию

Столбец	Тип	Функция	Null	Значение
lecture_id	int(10) unsigned	<input type="text"/>		<input type="text"/>
menu_name	varchar(64)	<input type="text"/>		Лекция 3
topic	varchar(256)	<input type="text"/>		Операции с переменными в PHP
name	varchar(256)	<input type="text"/>		Математические операции
content	text	<input type="text"/>		Математические операции

[Вперёд](#)







✓ Отображение строк 0 - 2 (3 всего, Запрос занял 0,0000 сек.)





`SELECT * FROM `lectures``

☐ Профилирование [\[Построчное редактирование\]](#) [\[Изменить\]](#) [\[Анализ SQL запроса\]](#) [\[Создать PHP-код\]](#) [\[Обновить\]](#)

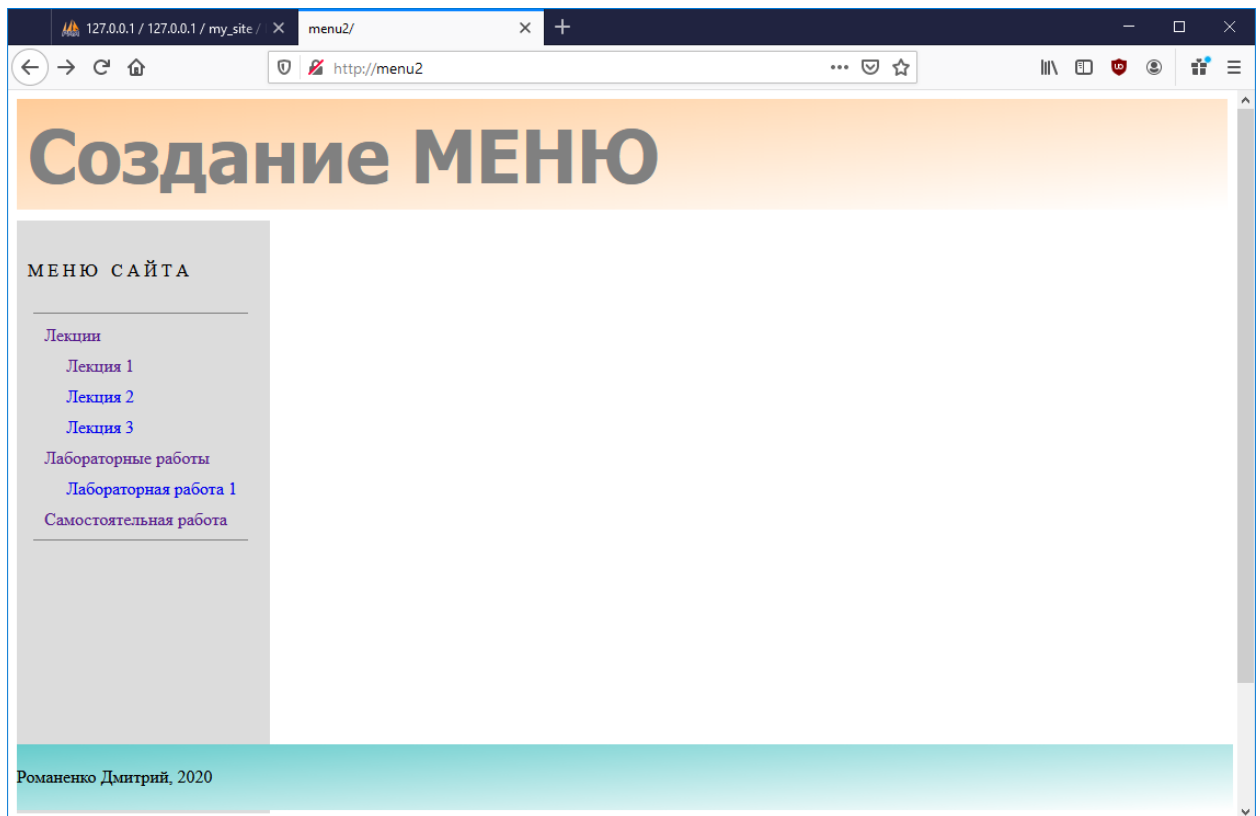
☐ Показать все | Количество строк:  | Фильтровать строки:  | Сортировать по индексу:

+ Параметры

	lecture_id	menu_name	topic	name	content
<input type="checkbox"/>  	1	Лекция 1	Основы PHP	Типы Данных	Четыре скалярных типа: • boolean (двоичные данные...
<input type="checkbox"/>  	2	Лекция 2	Основы PHP	Переменные в PHP	Как и в любом другом языке программирования, в PHP...
<input type="checkbox"/>  	3	Лекция 3	Операции с переменными в PHP	Математические операции	Математические операции

☐ Отметить все | С отмеченными:    

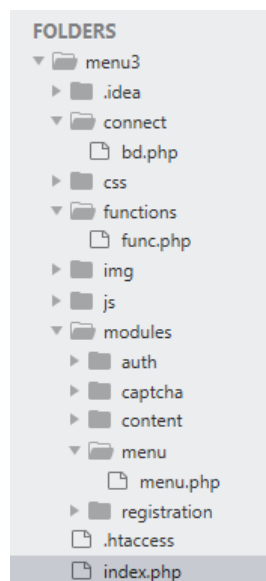
То при обновлении страницы получим следующий вариант меню.



#### 4. Создание многоуровневого меню

Для интеграции предложенного варианта реализации многоуровневого динамического меню фактически надо перенести функции, реализующие извлечение, сортировку и отображение в файл с функциями. Далее формирование меню (вызов соответствующих функций) целесообразно описать в виде отдельного файла, например, `menu.php`. С главной страницы `index.php` остается организовывать подключение всех необходимых модулей, в том числе и модуля (реализованного в виде одного файла) формирования меню. Такое искусственное разделение позволяет независимо работать над разработкой различных модулей.

При такой стратегии реализации структура проекта будет следующей (разработка других модулей, типа `content` и т.д. будут рассмотрены с следующих лекциях).



Далее представим листинги основных файлов проекта, реализующих формирование меню.

Файл *index.php*.

```
<?php
    if(session_status() != PHP_SESSION_ACTIVE) session_start();
?>
<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <link rel="stylesheet" href="css/style.css">
</head>
<body>

    <div class="header">
        Создание МЕНЮ
    </div>
<div>
    <div class="menu">
        <?php
            include ("modules/menu/menu.php");
        ?>
    </div>
    <div class="content">
        <iframe alloytransparency name="CONTENT" align="left"
width="100%" height="100%" style="border: 0;">Ваш браузер не
поддерживает iframe</iframe>
    </div>
</div>
<div class="footer">
    Романенко Дмитрий, 2020
</div>

</body>
</html>
```

Файл *menu.php*.

```
<html>
<head>
    <meta charset="windows-1251">
    <link rel="stylesheet" href="./css/style.css">
</head>
<style>
    ul {
        list-style: none; /*убираем маркеры списка*/
        margin: 5;
        margin-right: 20;
        padding-left: 10;

    }
    a {
        text-decoration: none; /*убираем подчеркивание текста
ссылок*/
    }
    li {
        margin-bottom: 10;
```



```

        margin-top: 10;
    }

</style>
</html>

<?php
    include './functions/func.php';
    //Выводим каталог на экран с помощью рекурсивной функции
    echo '<div style="padding:10px; border-bottom:1px solid
#074776">';
    echo 'МЕНЮ САЙТА';
    echo '</div>';

    $result=get_category();

    echo "<div>";
    view_category($result);
    echo "</div>";

?>

```

#### Файл *func.php*.

```

<?php

function get_category()
{
    include './connect/bd.php';
    //запрос к базе данных
    $query ="SELECT * FROM menu";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if($result)
    {
        $arr_cat = array();
        if(mysqli_num_rows($result) != 0)
        {
            //В цикле формируем массив
            for($i = 0; $i < mysqli_num_rows($result);$i++)
            {
                $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
                //Формируем массив, где ключами являются адишники на
родительские категории
                if(empty($arr_cat[$row['parent_id']])) {
                    $arr_cat[$row['parent_id']] = array();
                }
                $arr_cat[$row['parent_id']][] = $row;
            }
        }
        //возвращаем массив
        return $arr_cat;
    }
}

//вывод каталога с помощью рекурсии
function view_category($arr,$parent_id = 0)
{

```

```

//Условия выхода из рекурсии
if(empty($arr[$parent_id])) {
    return;
}
echo '<ul>';
//перебираем в цикле массив и выводим на экран
for($i = 0; $i < count($arr[$parent_id]);$i++) {
    echo '<li><a href="?category_id='.$arr[$parent_id][$i]['id'].'
    '&parent_id='.$parent_id.'">'
    .$arr[$parent_id][$i]['title'].'</a>';
    //рекурсия - проверяем нет ли дочерних категорий
    view_category($arr,$arr[$parent_id][$i]['id']);
    echo '</li>';
}
echo '</ul>';
}
?>

```

В итоге в браузере получим проект со следующим внешнем видом.

