Предмет: «Технология разработки программного обеспечения»

Тема: «Лабораторная работа №5. Разработка модулей программного обеспечения

Цель: применить на практике знания по разработке модулей программного обеспечения

Отведенное время: 8 часов (2-я пара).

Теоретические сведенья

До этого момента, рассматривая HTML и CSS, мы говорили только о статических страницах, то есть таких, которые, как бы пользователь с ними не взаимодействовал, всегда остаются неизменными а, чтобы содержимое или оформление их поменялось, хозяину страницы надо править код вручную.

**Динамические страницы и что для них нужно**

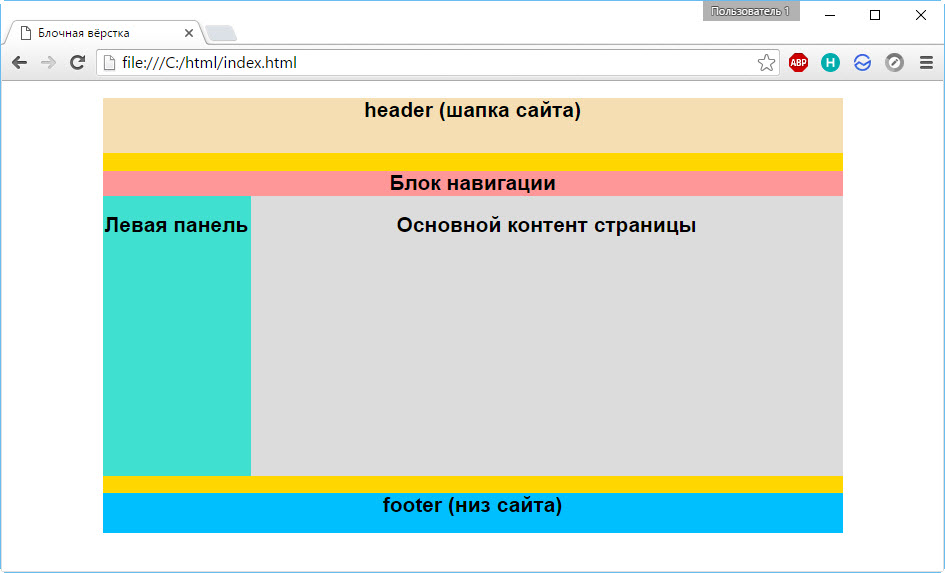
Помимо статических есть ещё динамические страницы. Таких сейчас в Интернете большинство. Информация в них подгружается из внешних источников, например базы данных или других файлов. Содержимое и форматирование таких страниц может меняться в зависимости от действий пользователя. Для правки динамических сайтов не обязательно вмешиваться в их код – достаточно изменить контент в специально для него предназначенном файле или базе данных, которая, к слову, представляет собой тоже файл, только определённым образом структурированный.

Для создания динамических сайтов только HTML и CSS недостаточно. Тут используются ещё и языки программирования, а также базы данных и языки запросов к ним. Чаще всего современные динамические сайты используют в своей работе HTML, CSS, PHP, JavaScript, SQL. Первые две аббревиатуры вам уже знакомы не понаслышке, SQL применяется для доступа к базам данных, JavaScript — клиентский язык, команды которого обрабатываются браузером и зачастую используются, чтобы показывать вам всякую красоту вроде шторок или плавно открывающихся фотографий, а вот PHP — это серверный язык программирования, который работает, в том числе, с содержимым сайта и делает его динамическим, с ним мы сегодня и соприкоснёмся.

**Пример использования команды include**

В предыдущей статье я рассказывал о блочной вёрстке сайта и приводил в пример простейшую страницу (документ index.html и привязанный к нему файл style.css).

Сейчас мы разделим документ index.html на несколько файлов, каждый из которых будет содержать свою часть страницы, что поможет ещё сильней разделить код, улучшить структуру шаблона и, по сути, сделать страницу динамической. Для этой цели мы будем использовать язык PHP, а точнее лишь одну его директиву — функцию include(), которая включает один файл в другой.



1. Смените разрешение созданного в статье о блочной вёрстке файла index с .html на .php, чтобы документ назывался index.php. Тип файла .PHP указывает серверу, что документ был написан или использует вставки на одноимённом языке программирования.

2. В папке со страницей создайте директорию blocks.

3. Всю вспомогательную информацию (верх, низ, навигацию и боковую панель сайта) вынесем в отдельные файлы, которые разместим в папке blocks.

Итак, создайте в каталоге blocks четыре файла: header.php, navigation.php, sidebar.php и footer.php. Заполните файлы кодом.

header.php:

<div id="header">

<h2>header (шапка сайта)</h2>

</div>

navigation.php

<div id="navigation">

<h2>Блок навигации</h2>

</div>

sidebar.php

<div id="sidebar">

<h2>Левая панель</h2>

</div>

footer.php

<div id="clear">

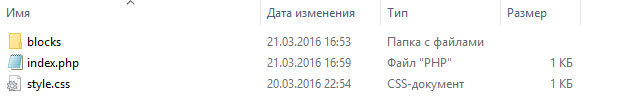
</div>

<div id="footer">

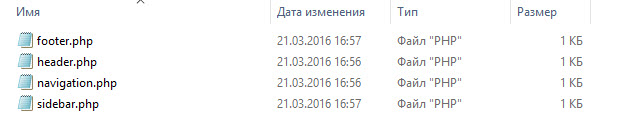
<h2>footer (низ сайта)</h2>

</div>

4. Проверьте структуру папки шаблона. В корне должны находиться файлы index.php, style.css и директория blocks.



Структура папки blocks должна быть такой.



В файле index.php удалите существующий код и напишите новый:

<!DOCTYPE html>

<html>

<head>

<title>Блочная вёрстка</title>

<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<div id="container">

<?php include ("blocks/header.php");?>

<?php include ("blocks/navigation.php");?>

<?php include ("blocks/sidebar.php");?>

<div id="content">

<h2>Основной контент страницы</h2>

</div>

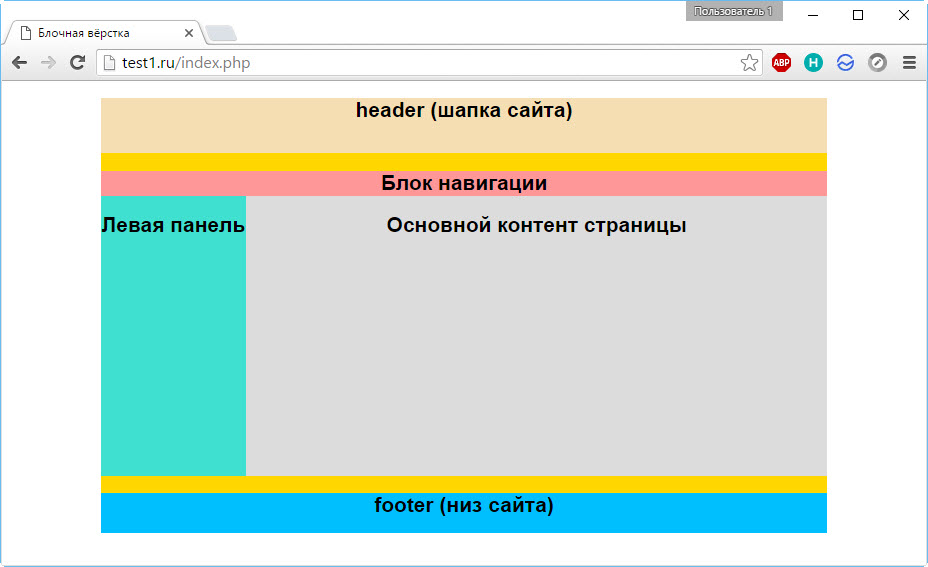
<?php include ("blocks/footer.php");?>

</div>

</body>

</html>

В браузере файл index.php выглядит точно так же, как и раньше, но структура шаблона при этом полностью изменилась. О том, что получилось, поговорим потом, а сейчас ответим на вопрос о загадочных командах вида <?php include ("имя\_файла");?>.



Как и HTML-код, код PHP тоже имеет своё обозначение начала и конца. Так вот начинать PHP-вставку нужно командой <?php, а заканчивать строкой ?>. Между этими командами пишется основной код. В нашем случае это всего лишь одна команда — include.

Функция include() вставляет в файл код из другого файла, давая возможность хранить разные части страницы в разных документах, тем самым надёжно отделяя их друг от друга.

В результате выполненных действий мы получили динамическую страницу index.php, части которой подгружаются из разных файлов. Благодаря этому можно создавать другие страницы, точно так же подгружая в них вспомогательные элементы из файлов папки blocks.

Такой подход хорош тем, что если вы захотите на сайте в 20-30 страниц изменить, скажем, название пункта меню, то в шаблоне с только что созданной структурой понадобится внести правки только в один файл — blocks/navigation.php, и меню изменится сразу на всех страницах, в которые он включен. Если же сайт был бы статическим, то для смены названия одного пункта меню вам пришлось бы вносить изменения в каждую из 20-30 страниц. Разница очевидна.

Задание

Необходимо сайт, который был выбран Вами в качестве разработке, создать используя такую структуру подключения. Сайт должен состоять минимум из 4-х страниц.

Домашнее задание

Установить OpenServer