

## Лабораторная работа №2.

### «Блочная верстка с использованием CSS»

**Цель работы:** Изучить возможность верстки веб-страницы с помощью тегов *HTML5* и *CSS*.

Блочная верстка – это верстка страницы с использованием слоев или блоков, в основе используются теги *div*.

Основной принцип такой верстки в том, что вся страница разделяется на блоки, они могут накладываться друг на друга, им можно задавать отступы с краев, в данные блоки помещаются остальные элементы страницы. Ко всем блокам применяются стили *CSS*, которые показывают браузерам размер, расположение, отступы снаружи и внутри блока и много другое.

Стили мы будем прописывать с использованием тега `<style>` в части `<head>`.

#### Ход работы

Для начала нам необходимо понять какая структура страницы нам нужна, то есть нарисовать ее структуру.

Мы будем использовать простую структуру, которая встречается на большинстве сайтов. Она будет состоять из: шапки, горизонтального меню, левого сайдбара, основной части и подвала.

Представим необходимую структуру графически:



После того как мы нарисовали структуру в графическом варианте мы можем определиться с тем какие теги мы будем использовать для создания структуры страницы с использованием *HTML5*.

Создадим структуру страницы:

```
<html>
<head>
<title>Лабораторная работа №2 </title>
</head>

<body>
<header>
Шапка сайта
</header>
<nav>
Меню навигации горизонтальное
</nav>
<aside>
Левый сайдбар
</aside>
<div>
Основная часть
</div>
<div></div>
<footer>
Подвал
</footer>
</body>
</html>
```

Если мы посмотрим как выглядит наш код в браузере то увидим следующее

Шапка сайта  
Меню навигации горизонтальное  
Левый сайдбар  
Основная часть  
Подвал

На нужный результат совсем не похоже, кроме надписей. Для того чтобы наша страница приобретала нужный вид добавляем стили CSS в часть `<head>`.

```
<style>
body {
background: #f5f5f5;
color: #000000;
font-family: Arial, Times New Roman;
font-size: 16px;
}
header {
background: yellow;
height: 100px;
width: 100%;
}
nav {
background: #330044;
color: white;
width: 100%;
height: 50px;
}
aside {
background: #1baf5d;
float: left;
width: 10%;
```

```
height: 400px;
}
#content {
background: gray;
float: left;
width: 90%;
}
#clear {
clear: both;
}
}
footer {
background: #ff0404;
color: white;
height: 80px;
width: 100%;
}
</style>
```

Проверим результат в браузере



После применения стилей видно что наш сайт уже стал приобретать вид того что нам нужно. Теперь разберем код стилей.

Итак, начнем по порядку с тега `<header>` нам необходимо задать ему ширину в 100% от окна, это делается с помощью свойства `width: 100%`, и задаем высоту в 100 пикселей с помощью свойства `height: 100px`. И соответственно используя свойство `background: yellow`. Шапка готова. Значения ширины и высоты могут принимать значения в пикселях и в процентах. Что использовать зависит от конкретных ситуаций. Переходим дальше. Следующим элементом у нас является меню навигации. При стилизации блока меню используются те же самые свойства что и при стилизации шапки. Плюс мы добавили свойство `color: white`, оно позволяет указать цвет текста. В качестве значений для свойств задающих цвета могут выступать названия цветов (например, red, Green, black), так и шестнадцатичные значения цветов (например, #ff00ff, #000000 и т.д.).

Следующим блоком у нас является сайдбар созданный с помощью тега `<aside>`, к нему применяем свойства ширины, высоты, цвет заливки и добавляется свойство `float:left`, оно позволяет задать позиционирование блока по левому краю. При задании размеров блоков нужно правильно рассчитывать ширину блоков, которые находятся рядом. Вся ширина экрана это 100%, то есть для того чтобы нам разместить два блока рядом необходимо одному блоку задать ширину X второму задать ширину Y, но главное следует всегда просчитывать чтобы сумма значений X и Y всегда равнялась 100%. В противном случае может появиться пустое пространство, либо блоки не будут уместиться и будут не структурированы, то есть какой либо из блоков может переместиться ниже или выше другого, либо наложиться сверху.

Следующий блок это основная часть страницы, его мы создали с помощью тега `<div>` и присвоили ему идентификатор `#content`. данному идентификатору мы создали стили заливки фона (background), ширину (width), высоту (height), и обтекаемость (float).

Последним блоком является подвал, который мы создали с помощью тега `<footer>`. Стили для данного блока практически идентичны стилям которые мы использовали при стилизации шапки страницы.

Это небольшой пример того как можно сверстать веб-страницу с помощью блоков и стилей CSS. Большие и качественные проекты безусловно требуют гораздо более сложного кода и структуры тегов. Но всегда нужно с чего то начинать и данная работа поможет вам в этом.

### Практическое задание для выполнения.

Вам необходимо сверстать предложенный макет страницы используя HTML5 и CSS. Стили необходимо присоединить с помощью внешнего файла style.css. Его необходимо разместить в той же папке что и ваш HTML файл.

