Лабораторная работа №3.

Использование 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 в отдельном документе.

Попробуем сверстать структуру, встречающуюся на очень многих сайтах.

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.

<!DOCTYPE html>

<html>

<head>

<title>Лабораторнаяработа №2 </title>

<link rel="stylesheet" href="style.css">

<link rel="stylesheet" href="C:\Users\Анна\Desktop\ПРОГРАММА ПГУ\Лекции\Лекция 4\zero.css">

</head>

<body>

<header> Шапка сайта </header>

<nav> навигация </nav>

<aside> Левый сайдбар </aside>

<div id="content"> Основная часть </div>

<div></div>

<footer> Подвал </footer>

</body>

</html>

На нужный результат совсем не похоже, кроме надписей. Для того чтобы наша страница приобретала нужный вид добавляем стили CSS, предварительно создав для этого документ style.css

body {

background: #f5f5f5;

color: #000000;

font-family: Arial, Times New Roman;

font-size: 16px;

}

header {

background: #D9D9D9;

height: 80px;

width: 100%;

padding-top: 30px;

text-align: center;

}

nav {

background: #BEBBBB;

width: 100%;

height: 40px;

text-align: center;

padding-top: 8px;

}

aside

{

background: #E9E7E7;

display: inline-block;

float:left;

width: 10%;

height: 400px;

padding: 15px 0 0 15px;

}

#content {

padding: 15px 0 0 15px;

background: #F2F2F2;

display: inline-block;

float: left;

width: 90%;

height: 400px;

}

footer {

display: inline-block;

background: #D9D9D9;

height: 80px;

width: 100%;

text-align: center;

padding-top: 30px;

}

Подключите этот документ к html документу в теге <head>:

**<link** rel="stylesheet" href="style.css"**>**

Теперь разберёмся, что же произошло:

Начнём 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: #f5f5f5, задаём цвет фона. 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.

Следующим блоком у нас является сайдбар созданный с помощью тега <aside>, к нему применяем свойства ширины, высоты, цвет заливки и добавляется свойство display: inline-block, оно позволяет задать ему свойство, чтобы блок отображался в одну линию, занимая только 10% экрана.

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 Х 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 умещаться и будут не структурированы, то есть какой либо из блоков может переместиться ниже или выше другого, либо наложиться сверху.

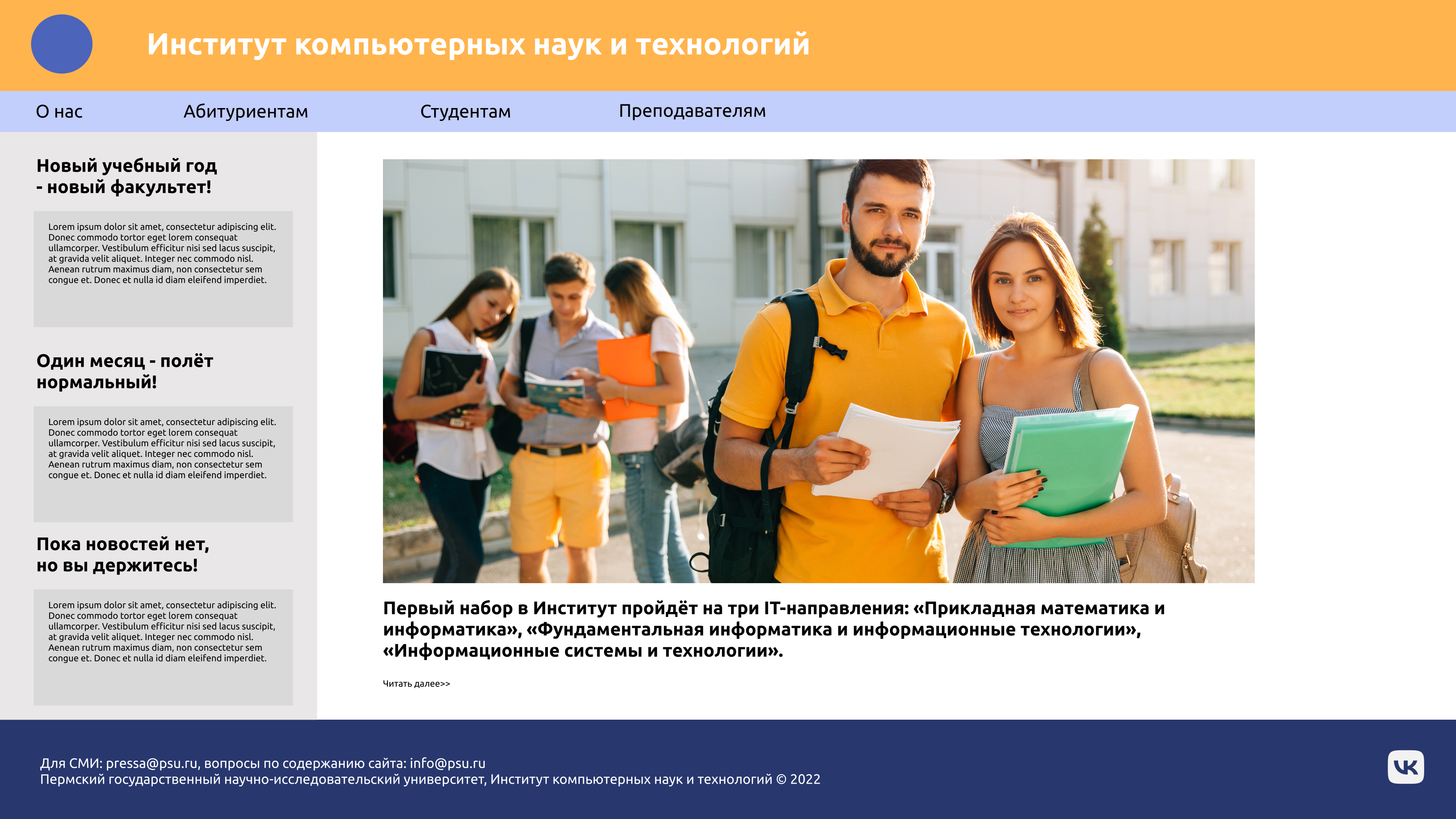
Следующий блок - это основная часть страницы, его мы создали с помощью тега <div> и присвоили ему идентификатор #content . данному идентификатору мы создали стили заливки фона (background), ширину (width), высоту (height), и display: inline-block;.

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.

Вот так просто мы получили очень распространённую структура сайта.

## Творческая часть

Используя получившийся макет, попробуйте сверстать страницу следующего вида:



Макет доступен по ссылке:

<https://www.figma.com/file/b97WbB1SEg1KzoZxYROSRt/%D0%98%D0%9A%D0%9D?node-id=2%3A16>

Сохраните себе копию этого файла.

Чтобы скопировать стили, цвета, используйте возможности figma (cм. Практическую работу).

## План выполнения лабораторной работы:

### Структура файлов

Первым шагом создайте простую структуру файлов для наших файлов.

* Создаем папку с названием нашего проекта, например ИКН.
* В ней создаем пустой файл index.html.
* В папке проекта создаем папку css с пустым файлом styles.css.
* В папке проекта создаем пустую папку images.

### Предварительный осмотр

После создания структуры файлов открываем файл в figma. Важно внимательно осмотреть шаблон и оценить его. Нам нужно понять следующие вещи:

* Откуда мы возьмём изображения?
* Какими будут основные стили?
* Какой макет у нас получится?

Только после того, как вы мысленно себе ответите на эти вопросы, можно переходить к нарезке изображений и написанию кода.

### Изображения

Сохраним логотипы следующим образом:

/images/logo.png  
/images/footer-vk.png

Также можете использовать svg изображение иконки Вконтакте.

Общее правило при именовании изображений заключается в том, что мелкие и простые картинки, такие, как иконки, логотипы и т.д. сохраняются в формате png (если есть возможность – svg), а фотографии в формате jpg.

Сохраните фотографию для новости в формате jpg.

##### **Основные стили**

И только теперь можно начинать писать код. Начните это делать не с привычного HTML, а с переноса правил в CSS.

На данном этапе желательно перенести все визуальные стили из дизайна в CSS, которые будут применяться по умолчанию для каждого тега.

Пропишите основной цвет для фона (тег body), тип шрифта (Ubuntu, подключите его через Google fonts).

В дальнейшем вы будете дополнять файл по мере добавления новых элементов на страницу.

##### **Каркас HTML**

И вот, наконец, мы можем попрактиковаться в написании HTML кода. Используйте файл, получившийся в первом задании лабораторной, он уде размечает основное содержание страницы.

Вспомните о том, что HTML - это матрёшка. Значит, вам нужно добавить внутрь получившихся блоков элементы.

Например, добавьте внутрь хэдера два inline элемента: изображение логотипа (схематичное, в виде кружка, т.к. реального логотипа ещё не существует) и тег <h1>, в котором разместите заголовок сайта. Сохраните html документ и посмотрите, что у вас получится. Скорее всего, размеры шрифта и логотипа вас удивят.

Перейдите в документ css и, создав новые селекторы, опишите классы заголовка и логотипа, указав желаемые свойства (размер шрифта, цвет шрифта, для изображения – его высоту).

Аналогичным образом создайте меню внутри блока <nav>. Посмотрите, как было реализовано меню в предыдущей лабораторной работе (Городской портал). Скопируйте код отуда.

Внутри блока <aside> создайте несколько <div> одинаковой структуры. Каждый <div> содержит заголовок новости и текст, у текста есть фон.

Внутри блока #content разместите тег <img> с фотографией и два блока текста.

Внутри блока <footer> разместите текст и изображение иконки. Иконка должна быть выровнена по правому краю, это можно реализовать с помощью свойства position или float.

Разметив всю страницу, возвращайтесь к редактированию css. Постарайтесь сделать ваш макет максимально похожим на пример.