

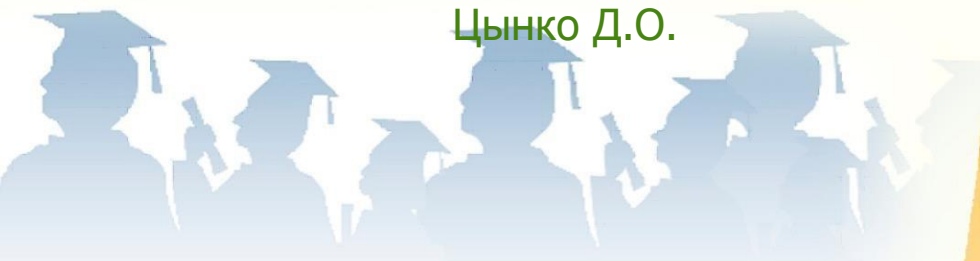


ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
УПРАВЛЕНИЕ ЦИФРОВЫХ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

Кафедра «Информационные технологии»

**Методические указания к
лабораторным работам
по дисциплине
«Технология обработки
информации»**

Авторы
Борисова Е.В.;
Зубарева Е.Г.;
Цынко Д.О.



Ростов-на-Дону, 2023

Аннотация

Данный ресурс предназначен для студентов всех форм обучения направления 09.03.03 «Прикладная информатика» и 09.03.02 «Информационные системы и технологии»; соответствует требованиям, предъявляемым Федеральным государственным образовательным стандартом высшего образования.

Составители

доцент к.т.н. кафедры
«Информационные технологии» Борисова Е.В.

старший преподаватель кафедры
«Информационные технологии» Зубарева Е.Г.

программист кафедры
«Информационные технологии» Цынка Д.О.

Оглавление

ВВЕДЕНИЕ	4
Лабораторная работа № 1	5
Лабораторная работа № 2	6
Лабораторная работа № 3	7
Лабораторная работа № 4	9
Лабораторная работа № 5	11
Лабораторная работа № 6	15
Лабораторная работа № 7	20
Лабораторная работа № 8	22
Индивидуальное задание	28
Список литературы	29

ВВЕДЕНИЕ

Основные задачи данного ресурса:

- сформировать устойчивый интерес к изучаемой дисциплине, способствовать развитию научного мировоззрения и творческого потенциала, позволяющего будущему специалисту эффективно осуществлять доступ к требуемым информационным ресурсам;
- сформировать навыки создания, обработки и хранения информации для решения прикладных задач в различных областях профессиональной деятельности;
- создать необходимые условия самообучения и самостоятельной работы, направленные на мобилизацию индивидуальных возможностей и ресурсов, обучаемых в рамках профессиональной деятельности.

Лабораторная работа № 1

Создание структуры проекта. Выбор текстового редактора. Базовая разметка страницы

Создание структуры проекта

Создание любого проекта начинается с создания базовой структуры папок. Эта структура является однотипной:

В удобном для вас месте создайте папку с вашим будущим проектом. В корне этой папки создайте папку “static” (здесь будут храниться статические файлы: изображения, стили) и документ “index.html” (это базовый файл с которым нам доведётся работать). В папке “static” создайте папку “css” и “images”. Последним штрихом добавляем файл “style.css” в папку “css”.

Примечание: Кроме того, более динамические сайты используют для интерактивных элементы которые создаются с помощью JavaScript. И файлы со скриптами хранятся еще в папке “js”, которая находится в папке “static”.

Выбор текстового редактора

HTML - страницу можно сверстать используя Блокнот или Notepad++, но на данный момент существуют редакторы кода, имеющие огромный функционал в виде подсветки синтаксиса, открытие целых директорий, множества файлов, разбиение на несколько окон и т.д. Одни из самых популярных это SublimeText, Brackets, Atom. Весь код, представленный в дальнейшем в описании лабораторных работ будет оформлен в SublimeText3.

Базовая разметка страницы

Откройте файл “index.html” и перепишите вручную текст с картинки. Всё, кроме того что заключено в теги вида <!-- какой-то комментарий -->(текст в данных тегах не будет отображаться в веб браузере). И посмотрите, что в браузере получилось.



The image shows a file explorer on the left and a code editor on the right. The file explorer shows a folder named 'site' containing subfolders 'static', 'css', and 'images', and a file 'index.html'. The 'static' folder contains 'css' and 'images' subfolders, and the 'css' folder contains 'style.css'. The code editor shows the content of 'index.html' with the following code:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <!--
5       Здесь происходит всё, что связано с настройкой,
6       подгрузка стилей, скриптов, настройка кодировки и т.д.
7     -->
8
9     <!-- задаем заголовок который будет отображаться на вкладке браузера -->
10    <title>Hello world</title>
11  </head>
12
13  <body>
14    <!-- в этом блоке описывается вся структура макета -->
15
16    <!--
17      h1, h2..h6 - это теги которые задают заголовки
18      1 - это самый крупный, а 6 самый мелкий.
19    -->
20    <h1>Здравствуй мир!</h1>
21  </body>
22 </html>
    
```

Рисунок 1 – Пример кода

В результате вы написали базовое приложение типа “Hello world”. Самым главным, что стоит запомнить из этого примера, что теги - это элементы языка вида <имя тега> и они обязательно должны закрываться: </имя тега>. В дальнейшем содержимое тегов может форматироваться. Указанная структура документа является базовой и обязательной.

Лабораторная работа № 2

Форматирование. Работа с текстом.

Дальнейшую работу продолжим в документе, созданном в лабораторной работе №1.

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Hello world</title>
5      <!--Для того, чтобы браузеры понимали кириллические символы введите!-->
6      <meta charset="utf-8">
7    </head>
8    <!--
9      .....      Примечание
10     теги можно писать как полностью большими так и полностью маленькими буквами,
11     разницы никакой не будет. Тоесть <H1> тоже самое, что и <h1>
12     -->
13   <body>
14     <!--div - это обычный контейнер который группирует элементы в единое целое -->
15     <!-- к div-ам мы еще еще более подробнее в дальнейшем -->
16     <div align="center">
17       <h1>Здравствуй мир!</h1>
18
19       <!-- -->
20       <br>
21
22       <h2>Сегодня мы вспомним химию</h2>
23       <!--разбивает текст на параграфы -->
24       <p>
25         <b>Сахар</b> – <u>бытовое название сахарозы</u> (C<sub>12</sub>H<sub>22</sub>O<sub>11</sub>).
26         <i>Тростниковый</i> и <i>свекловичный сахар</i> <sup>(сахарный песок, рафинад)</sup>
27
28         <!--
29           В - задает полужирный шрифт
30           U - задает нижнее подчеркивание
31           I - курсивный шрифт
32           SUB - подстрочный индекс
33           SUP - подстрочный индекс
34         -->
35       </p>
36
37       <p>
38         является важным пищевым продуктом. <font color="red" size="4">Обычный</font> сахар относится к
39         <strike>углеводам</strike>,
    
```

Рисунок 2 – Работа с различными тэгами

Клиентские интернет-технологии

```

40 <!--
41   strike - зачеркивание
42   font - Тег для задание свойств шрифта
43
44   На практике задание свойств текста через теги задается с помощью CSS,
45   но для базового ознакомления будет полезным ознакомиться с этим. В следующей
46   лабораторной работе мы сделаем те же самые вещи с помощью CSS
47
48   -->
49 <!-- BR - перевод строки -->
50 <!--
51   Вы уже обратили, что в одном из тегов была такая вещь, как align="center",
52   это мы назначали тегу напрямую стиль, который задает центральное позиционирование,
53   всему что находится в этом теге, кроме того, можно задать позиционирование: left(слева), right(
54   справа), top(сверху), bottom(снизу)
55
56   -->
57 <!--
58   Font- это тег, позволяющий форматировать нам текст, на этом примере мы задали цвет через
59   свойство "color" и размер шрифта "size", кроме того цвет можно задавать не только
60   зарезервированным словом, а его кодом, к примеру #000000 задаст черный цвет, подобрать код
61   цвета через палитру вы можете на сайте "getcolor.ru"
62
63   -->
64 </p>
65 <p>
66   которые считаются ценными питательными веществами, обеспечивающими организм необходимой энергией.
67 </p>
68 </div>
69 </body>
70 </html>
    
```

Рисунок 3 - Работа с различными тэгами

В комментариях вы можете увидеть описание каждого нового тега. Открывая страницу в браузере, вы получите такой результат:

Здравствуй мир!

Сегодня мы вспомним химию

Сахар — бытовое название сахарозы ($C_{12}H_{22}O_{11}$). *Тростниковый и свекловичный сахар* (сахарный песок, рафинад) является важным пищевым продуктом. **Обычный сахар** относится к углеводам, которые считаются ценными питательными веществами, обеспечивающими организм необходимой энергией.

Рисунок 4 – Результат работы кода

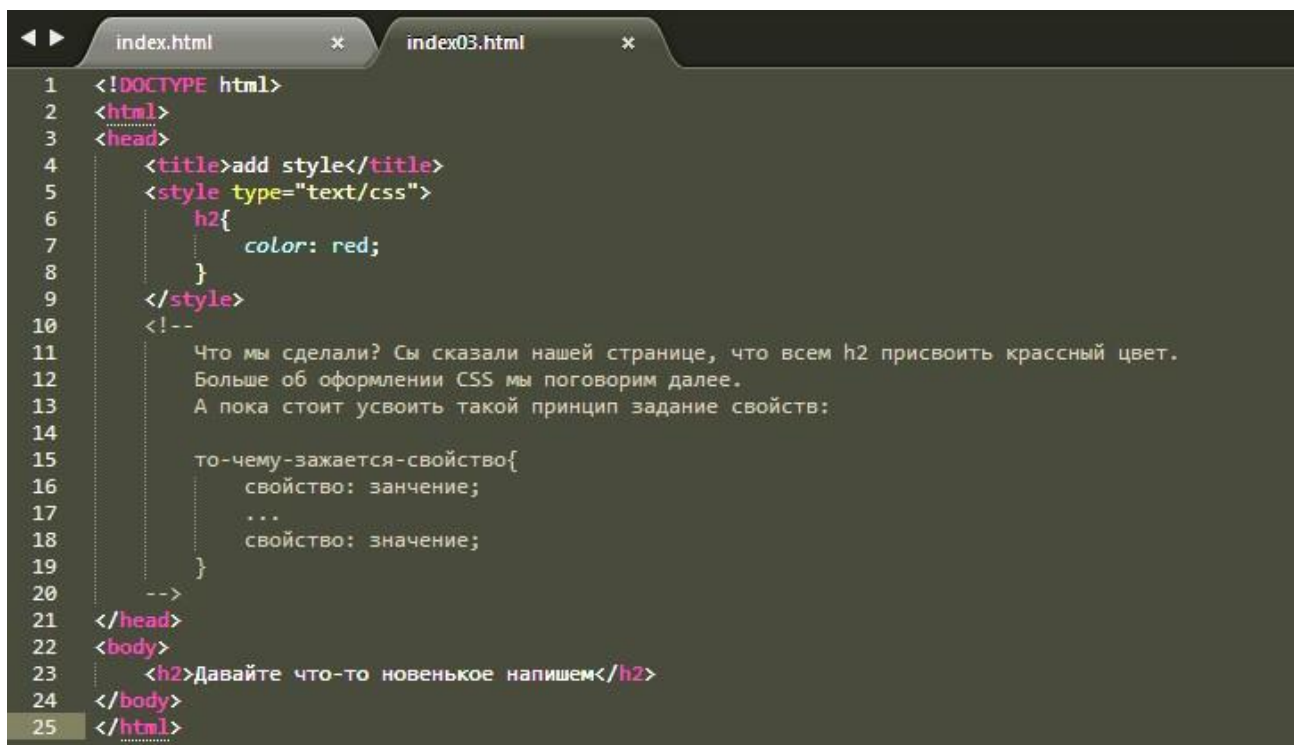
Лабораторная работа № 3

Объявление стилей. Базовые свойства стилей стилей.

Рассмотрим форматирование текста в HTML-документе с помощью таблицы каскадных стилей (далее CSS - Cascading Sheet Styles).

Существуют различные способы объявления стилей: в самих тегах, в теге <style></style>, и в отдельном файле. Мы же рассмотрим последние два метода.

Для начала создайте новый документ «index03.html» и задайте ему базовую структуру, как в лабораторной работе № 1 и новое содержимое блока <style> как в примере:

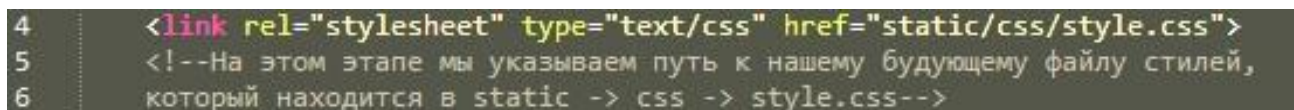


```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>add style</title>
5      <style type="text/css">
6          h2{
7              color: red;
8          }
9      </style>
10     <!--
11         Что мы сделали? Мы сказали нашей странице, что всем h2 присвоить красный цвет.
12         Больше об оформлении CSS мы поговорим далее.
13         А пока стоит усвоить такой принцип задание свойств:
14
15         то-чему-задается-свойство{
16             свойство: значение;
17             ...
18             свойство: значение;
19         }
20     -->
21 </head>
22 <body>
23     <h2>Давайте что-то новенькое напишем</h2>
24 </body>
25 </html>
    
```

Рисунок 5 - Объявление стилей

Чтобы ещё меньше нагружать наш документ стилями, мы подключим внешние стили. Для этого вместо блока `<style>` вставьте эту строчку:

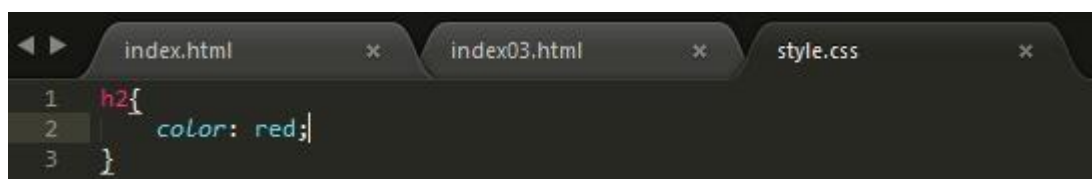


```

4  <link rel="stylesheet" type="text/css" href="static/css/style.css">
5  <!--На этом этапе мы указываем путь к нашему будущему файлу стилей,
6  который находится в static -> css -> style.css-->
    
```

Рисунок 6 – Подключение внешнего стиля

Затем открываем файл “style.css” из “static/css” и добавляем следующий код:



```

1  h2{
2      color: red;
3  }
    
```

Рисунок 7 - Изменение цвета заголовка

Обновив страницу, вы обнаружите, что заголовок так и останется красным. Так и должно быть, так как вы логику описания стилей перенесли в отдельный файл.

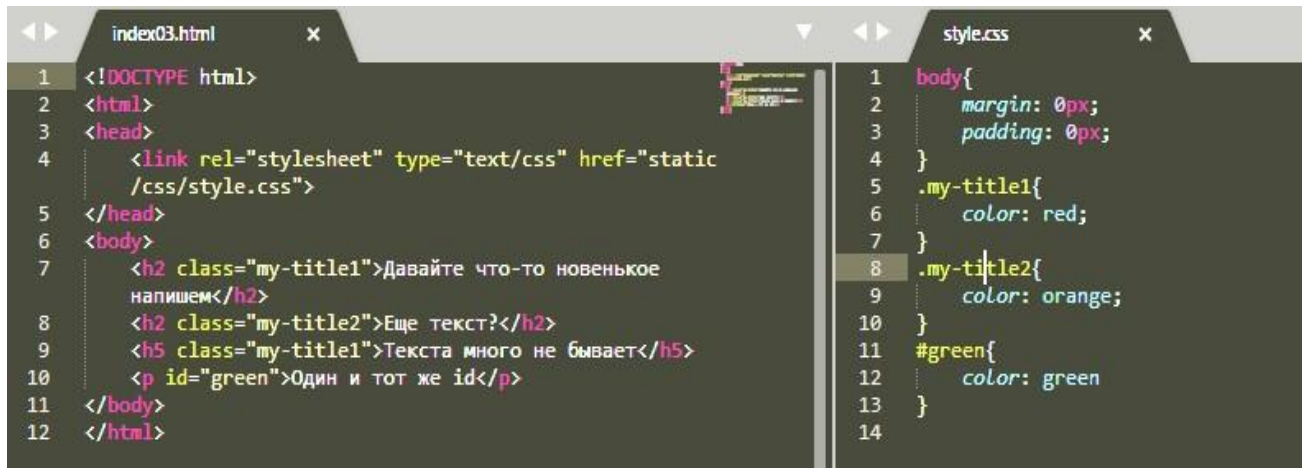
С объявлением мы разобрались, но что если мы не хотим делать так, чтобы все элементы h2 принимали красный цвет, или даже чтобы только один элемент имел собственный стиль? К нам на помощь приходят классы и идентификаторы.

Как работают классы? Для того чтобы тегу присвоить класс в самом теге вам надо написать `class="именна_классов_через_пробел"` (имена должны писаться латинскими символами и использовать из сторонних символов только дефис и нижнее подчеркивание).

В самом же файле стилей, стилю задаются свойства таким образом

.имя_класса{свойства как обычно}. С идентификатором(id) дела обстоят почти также. В теге объявляется через id="имя_id" а в стилях #имя_id{свойства}. А в чем же собственно разница? Разница в том, что id это уникальное имя которое следует объявлять только лишь раз один раз на странице, а класс сколько угодно раз.

Для большего понимания повторите следующий код и посмотрите результат:



```

index03.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <link rel="stylesheet" type="text/css" href="static
   /css/style.css">
5 </head>
6 <body>
7   <h2 class="my-title1">Давайте что-то новенькое
   напишем</h2>
8   <h2 class="my-title2">Еще текст?</h2>
9   <h5 class="my-title1">Текста много не бывает</h5>
10  <p id="green">Один и тот же id</p>
11 </body>
12 </html>

style.css
1 body{
2   margin: 0px;
3   padding: 0px;
4 }
5 .my-title1{
6   color: red;
7 }
8 .my-title2{
9   color: orange;
10 }
11 #green{
12   color: green
13 }
14
    
```

Рисунок 8 – Пример

Полученный результат для вас должен стать очевидным и понятным.

P.S. Если в дальнейшем css вызывает трудности, то можно воспользоваться сайтом <http://htmlbook.ru/css>. Здесь можно найти как описание тегов с подробным описанием поддержки тегов в различных браузерах, так и различные миниуроки.

Лабораторная работа № 4

Списки. Типы списков. Ссылки. Hover-эффекты.

Для отображения различных элементов в связанной структуре используют такой элемент, как списки. На следующем рисунке представлен код по созданию простой шапки сайта с использованием немаркированных списков.

Прежде чем повторять код, примите к сведению, что у элементов (элемент самого списка), есть html-атрибут «type». Этот атрибут может принимать различные значения, например «disc» (элементы списка будут иметь круглый черный маркер), «circle» (маркеры будут в виде кольца) и «square»(квадратные маркеры).

```

index04.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>lab-04</title>
5 <link rel="stylesheet" type="text/css" href="static/css/
  style.css">
6 </head>
7 <body>
8 <!--нав - наш тег для шапки сайта-->
9 <nav>
10 <ul class="list">
11 <!--ul - контейнер элементов списка -->
12 <!--li - элемент списка -->
13 <li>номер один</li>
14 <li>номер два</li>
15 <li>номер три</li>
16 </ul>
17 </nav>
18
19 <!--Блок с контентом-->
20 <content>
21 <!-- ol - контейнер нумерованого списка-->
22 <ol>
23 <li>Кот</li>
24 <li>Собака</li>
25 <li>Попугай</li>
26 </ol>
27 </content>
28 </body>
29 </html>

style.css
1 body{
2 margin: 0px;
3 padding: 0px;
4 }
5 nav{
6 height: 60px;
7 width: 100%;
8 background-color: #FEA610;/*оранжеавый фон*/
9 }
10 /*задаем стили контейнеру с классом list*/
11 ul.list{
12 position: absolute;
13 margin-left: calc((100% - 800px)/2);
14 width: 800px;
15 height: 30px;
16 list-style-type: none;/*убираем маркировку li-элементов*/
17 }
18 }
19
20 ul.list li{
21 background-color: grey;/*фон элементов серый*/
22 display: inline;/*li - элементы располагаются в ряд*/
23 }
24
25
26
27

```

Рисунок 9 – Списки

В результате вы будете иметь такой результат:

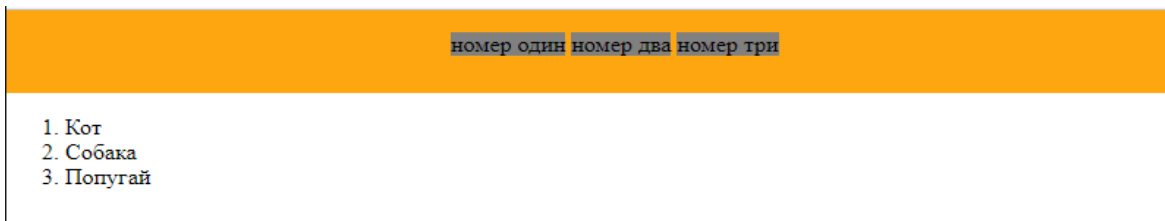


Рисунок 10 - Пример работы кода

Давайте теперь создадим на элементах списка ссылки и добавим к ней простую анимацию.

Ссылки описывает тег <a>. Основным атрибутом этого тега является «href», href – это ссылка на страницу. Ссылки бывают такого типа:

```

<a href="имя файла.html">Текст</a>
<a href="папки/где/лежит/страница.html">ссылки</a>
<a href="http://портал.com"></a>

```

Рисунок 11 – Ссылки

Кроме того, в ссылке можно оборачивать любой объект (тег), что-либо: текст, изображение, блок и т.д.

Стандартно HTML задает тексту в ссылке подчеркивание и голубой цвет шрифта и изменение цвета шрифта ссылки на фиолетовый, после перехода по этой ссылке. Но с помощью CSS или тегов html атрибутов можно изменить эти стандартные свойства.

Кроме того, в CSS есть такое понятие, как псевдокласс. Псевдокласс в CSS - это ключевое слово, добавленное к селектору, которое определяет его особое состояние.

Например, `:hover` применит стиль, когда пользователь наводит курсор на элемент, указанный селектором. Как раз этот псевдокласс и применим к ссылке и посмотрим, что получится.

```

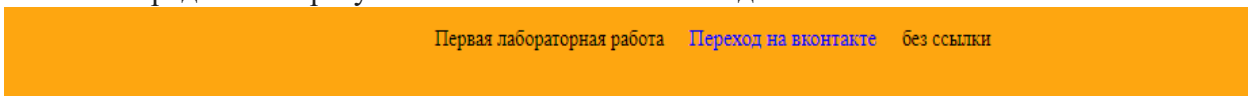
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>lab-04</title>
5    <link rel="stylesheet" type="text/css" href="static/css/
      style.css">
6  </head>
7  <body>
8
9    <nav>
10     <ul class="list">
11       <a href="index.html">
12         <li>Первая лабораторная работа</li>
13       </a>
14       <a href="https://vk.com">
15         <li>Переход на вконтакте</li>
16       </a>
17       <li>без ссылки</li>
18     </ul>
19   </nav>
20
21   <content>
22     <ol>
23       <li>Кот</li>
24       <li>Собака</li>
25       <li>Попугай</li>
26     </ol>
27   </content>
28
29 </body>
30 </html>
    
```

```

1  body{
2    margin: 0px;
3    padding: 0px;
4    font-size: 14px; /*задаем размер шрифта для всего документа*/
5  }
6  nav{
7    height: 60px;
8    width: 100%;
9    background-color: #FEA610;
10 }
11 ul.list{
12   position: absolute;
13   margin-left: calc((100% - 800px)/2);
14   width: 800px;
15   height: 30px;
16   list-style-type: none;
17 }
18 }
19 ul.list li{
20   margin-left: 15px;
21   display: inline;
22 }
23 a{
24   color: black;
25   text-decoration: none; /*отменяем нижнее подчеркивание*/
26 }
27 a:hover{
28   color: blue;
29   font-size: 15px
30 }
31
    
```

Рисунок 12 - Псевдокласс

Ниже представлен результат написанного вами кода:



1. Кот
2. Собака
3. Попугай

Рисунок 13 – Результат

Пример совсем не совершенен, но позволяет опробовать новые изученные свойства.

Лабораторная работа № 5

Блоки. Изображения. Позиционирование. Отступы.

Блок (атрибут `<div>`), это сущность которая напоминает контейнер (коробку), которая хранит в себе различные элементы с другими свойствами (размер, цвет и т.д.). Вложенные элементы в этот контейнер наследуют многие из свойств (такие, как шрифт, цвет, текстовое форматирование). В CSS есть свойства, которые применяются в основном к блоку и отвечают за способы расположения блоков.

Следующий пример покажет, как создать плитку цветных блоков, которой мы воспользуемся в дальнейшем.

```
index05.html x style.css x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>lab-5</title>
5   <link rel="stylesheet" type="text/css" href="static/css/style.css">
6 </head>
7 <body>
8   <div class="centr-cont">
9     <div class="block first"></div>
10    <div class="block second"></div>
11    <div class="block third"></div>
12    <div class="block fourth"></div>
13  </div>
14 </body>
15 </html>

1 body{
2   margin: 0px;
3   padding: 0px;
4   font-size: 14px; /*задаем размер шрифта для всего документа*/
5 }
6 .centr-cont{
7   height: 650px;
8   width: 750px;
9   background-color: #E7E4E4;
10  margin-left: calc((100% - 750px)/2);
11  /*операция калс выполняет мат операции,
12   очень важно между - и + с двух сторон ставить пробель*/
13 }
14 .block{
15  position: relative;
16  height: 300px;
17  width: 350px;
18  display: inline-block;
19  /*элементыэтого класса будут следовать в линии друг за другом,
20   пока им позволяет ширина окна браузера*/
21 }
22 .first{
23   background-color: #FAB933;
24 }
25 .second{
26   background-color: #FC625F;
27   margin-left: 46px;
28 }
29 .third{
30   background-color: #5FF6FC;
31   margin-top: 46px;
32 }
33 .fourth{
34   margin-left: 46px;
35   margin-top: 46px;
36   background-color: #69FC5F;
37 }
```

Рисунок 14 – Блоки

Результат:



Рисунок 15 - Результат

Теория

```
/*  
    Свойство padding позволяет задать величину поля  
    сразу для всех сторон элемента или определить ее только для указанных сторон.  
    Применяется для текста. И эти отступы применятся относительно того блока,  
    в котором он находится.  
  
    Пример:  
padding-left: 25px;  
padding: 10px 30px;  
padding: 20px;  
  
    Свойство margin Устанавливает величину отступа от каждого края элемента.  
    Отступом является пространство от границы текущего элемента до внутренней  
    границы его родительского элемента.  
  
    Пример:  
padding-left: 25px;  
padding: 10px 30px;  
padding: 20px;  
  
    Кроме того отступы можно задавать не только в пикселях(px),  
    но и процентах(%).  
*/
```

Позиционирование (css свойство- position)

Position - устанавливает способ позиционирования элемента относительно окна браузера или других объектов на веб-странице.

Способ объявления: position: absolute | fixed | relative | static | inherit

absolute

Указывает, что элемент абсолютно позиционирован, при этом другие элементы отображаются на веб-странице словно абсолютно позиционированного элемента и нет. Положение элемента задается свойствами left, top, right и bottom, также на положение влияет значение свойства position родительского элемента. Так, если у родителя значение position установлено как static или родителя нет, то отсчет координат ведется от края окна браузера.

Если у родителя значение position задано как fixed, relative или absolute, то отсчет координат ведется от края родительского элемента.

fixed

По своему действию это значение близко к absolute, но в отличие от него привязывается к указанной свойствами left, top, right и bottom точке на экране и не меняет своего положения при прокрутке веб-страницы. Браузер Firefox вообще не отображает полосы прокрутки, если положение элемента задано фиксированным, и оно не помещается целиком в окно браузера. В браузере Opera хотя и показываются полосы прокрутки, но они никак не влияют на позицию элемента.

relative

Положение элемента устанавливается относительно его исходного места. Добавление свойств `left`, `top`, `right` и `bottom` изменяет позицию элемента и сдвигает его в ту или иную сторону от первоначального расположения.

static

Элементы отображаются как обычно. Использование свойств `left`, `top`, `right` и `bottom` не приводит к каким-либо результатам.

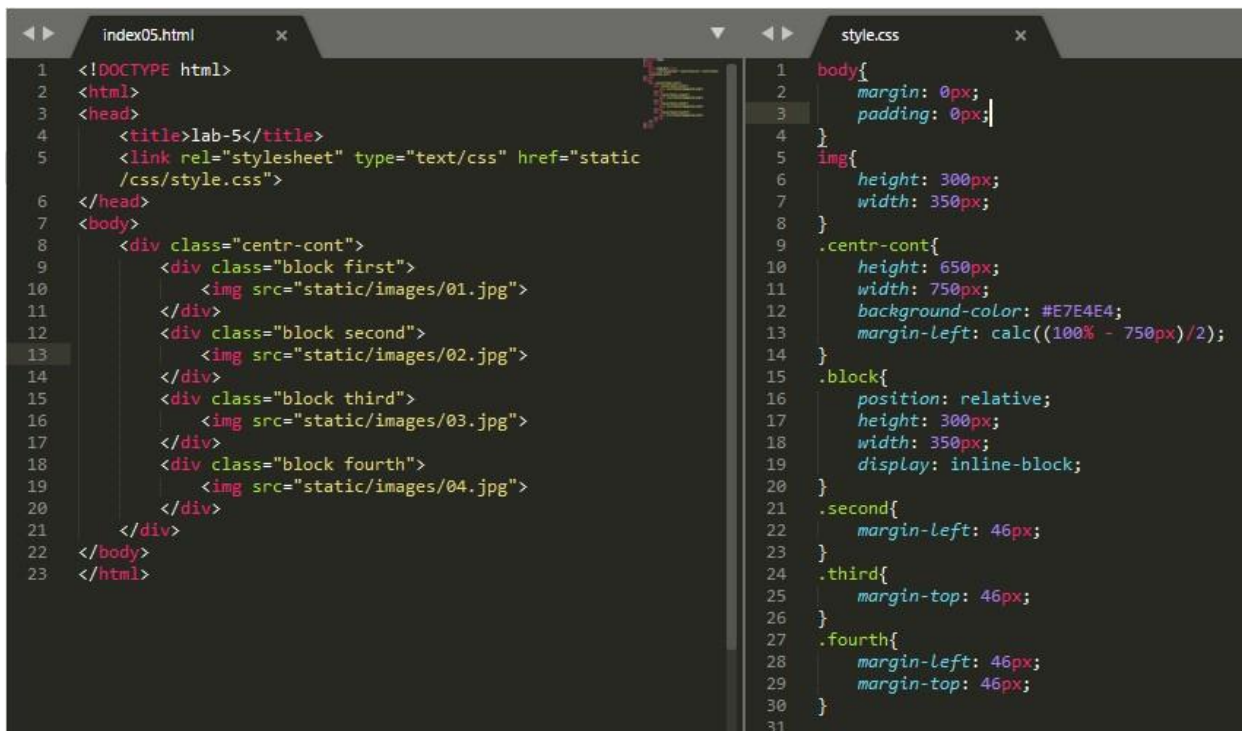
inherit

Наследует значение родителя.

Чтобы сделать еще красивее, добавим несколько изображений. Изображения очень похожи на ссылки, но имеют некоторые различия. Тег - `` (не закрывающийся), атрибут определяющий путь к файлу – `src="путь"`. Путь указывается также, как и у ссылки, только является важным разрешение файла (например `.jpg`)

Скачаем 4 изображения из интернета. Можно воспользоваться гугл, яндекс картинками. Или же сервисом со стоковыми изображениями, к примеру <https://unsplash.com>. Стоит иметь ввиду, что изображения с таких сервисов имеют огромный размер (до 15мб за изображение) и для быстрого отображения таких изображений на странице следует сжать изображение до некоторого минимума.

Скачанные изображения поместим в ранее созданную папку «`images`», а для большей простоты изображения переименуем в по образцу «`01.расширение_изображения`».



```

index05.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>lab-5</title>
5   <link rel="stylesheet" type="text/css" href="static
  /css/style.css">
6 </head>
7 <body>
8   <div class="centr-cont">
9     <div class="block first">
10      
11    </div>
12    <div class="block second">
13      
14    </div>
15    <div class="block third">
16      
17    </div>
18    <div class="block fourth">
19      
20    </div>
21  </div>
22 </body>
23 </html>

style.css
1 body{
2   margin: 0px;
3   padding: 0px;
4 }
5 img{
6   height: 300px;
7   width: 350px;
8 }
9 .centr-cont{
10  height: 650px;
11  width: 750px;
12  background-color: #E7E4E4;
13  margin-left: calc((100% - 750px)/2);
14 }
15 .block{
16  position: relative;
17  height: 300px;
18  width: 350px;
19  display: inline-block;
20 }
21 .second{
22  margin-left: 46px;
23 }
24 .third{
25  margin-top: 46px;
26 }
27 .fourth{
28  margin-left: 46px;
29  margin-top: 46px;
30 }
31
    
```

Рисунок 16 - Работа с изображениями

Результат:

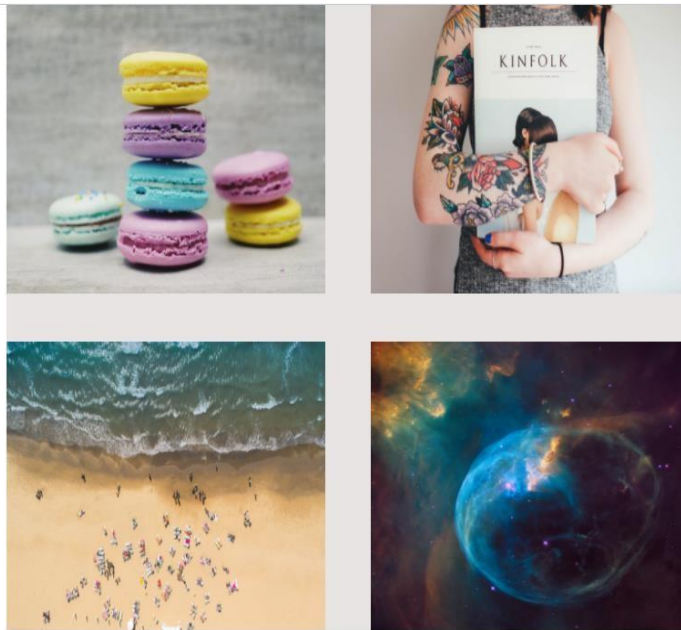


Рисунок 17 - Результат кода

П.С: Существует способ задания фона блоку через css-свойство «background-img», для ознакомления рекомендуем изучить его самостоятельно и опробовать на одном из блоков.

Лабораторная работа № 6

Таблицы.

Прежде чем приступить к выполнению лабораторной создайте файл «index0?.html» и файл стилей для него «style0?.css»

1. Создание простейшей таблицы

Создание таблицы начинается с тега `<table></table>`, далее, внутри него располагается все содержимое, а именно строки `<tr>` и уже внутри строк ячейки `<td>`. Чтобы понять, как все это работает обратите внимание на пример ниже:

Примечание: заголовочную ячейку можно создать тегом `<th>`, текст в такой ячейке располагается посередине и выделяется жирным шрифтом.

1. Для начала зададим базовые стили

```

1  /* Задаем рамку таблице */
2  table{
3      border:1px solid black;
4  }
5
6  /* Задаем рамку ячейке */
7  td{
8      border:1px solid black;
9  }
```

Рисунок 18 – Базовый стиль

не забудьте подключить файл стилей к вашему HTML файлу.

2. Создадим простейшую таблицу и посмотрим на результат в браузере

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <link rel="stylesheet" href="style07.css">
6  </head>
7  <body>
8      <table>
9          <tr>
10             <td>Университет</td>
11             <td>Кол-во студентов</td>
12          </tr>
13          <tr>
14             <td>ДГТУ</td>
15             <td>46 000</td>
16          </tr>
17          <tr>
18             <td>ЮФУ</td>
19             <td>33125</td>
20          </tr>
21          <tr>
22             <td>РИНХ</td>
23             <td>21 636</td>
24          </tr>
25      </table>
26 </body>
27 </html>
    
```

Рисунок 19 - Создание таблицы

Университет	Кол-во студентов
ДГТУ	46 000
ЮФУ	33125
РИНХ	21 636

Рисунок 20 - Результат

Задание 1

Дополнить таблицу как показано ниже:

Университет	Кол-во студентов	Год основания	Адрес
ДГТУ	46 000	1930	пл.Гагарина,1
ЮФУ	33125	2006	ул.Б.Садовая,105
РИНХ	21 636	1931	ул.Б.Садовая,69

Самостоятельно освоите свойство **border-collapse**, в выполнении задания оно обязательно понадобится.

2. Работа с рамками, отступами внутри и между ячейками, ячейки-заголовки

Иногда бывают ситуации, когда нужно чтобы отображалась только нижняя рамка ячейки и т.д, такие эффекты достигаются за счет свойств, представленных ниже:

border-right,
border-left,
border-top,
border-bottom

Примечание: для того чтобы убрать рамку в значении свойства напишите **none**

Задание 2

В созданной таблице из Задания 1 оставьте рамки только справа и слева как на примере ниже:

Университет	Кол-во студентов	Год основания	Адрес
ДГТУ	46 000	1930	пл.Гагарина,1
ЮФУ	33125	2006	ул.Б.Садовая,105
РИНХ	21 636	1931	ул.Б.Садовая,69

Отступы внутри ячеек задаются с помощью свойства **padding**:

padding: 5px - одно значение задает отступ всем сторонам,
padding: 5px 10px - первое значение задает отступ сверху-снизу, второе справа-слева.

padding: 5px 10px 5px - первое значение задает отступ сверху, второе одновременно справа-слева, третье снизу.

padding: 5px 10px 5px 10px - первое сверху, второе справа, третье снизу, четвертое слева.

Задание 3

Задайте ячейкам отступы сверху-снизу (5px), справа-слева (10px) как на примере ниже:

Университет	Кол-во студентов	Год основания	Адрес
ДГТУ	46 000	1930	пл.Гагарина,1
ЮФУ	33125	2006	ул.Б.Садовая,105
РИНХ	21 636	1931	ул.Б.Садовая,69

Задание 4

1. Замените тег **<td>** в категориях на **<th>**.
2. В стилях задайте тегу **<th>** рамку и внутренний отступ (10px)

Университет	Кол-во студентов	Год основания	Адрес
ДГТУ	46 000	1930	пл.Гагарина,1
ЮФУ	33125	2006	ул.Б.Садовая,105
РИНХ	21 636	1931	ул.Б.Садовая,69

3. Объединение ячеек в строках-столбцах

Для объединения двух и более ячеек используют атрибуты colspan и rowspan для тега <td>

colspan — устанавливает число объединяемых ячеек по горизонтали

rowspan — аналогично, только по вертикали

Следуя инструкциям, создайте таблицу:

Браузер	Посещения	
	Количество	В процентах
Mozilla Firefox	163	59%
Google Chrome	78	28%
Safari	35	13%

1. Задайте базовую структуру таблицы и стили

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <link rel="stylesheet" href="style07.css">
6  </head>
7  <body>
8      <table>
9          <tr>
10             <th>Браузер</th>
11             <th>Посещения</th>
12          </tr>
13          <tr>
14             <th>Количество</th>
15             <th>В процентах</th>
16          </tr>
17          <tr>
18             <td>Mozilla Firefox</td>
19             <td>163</td>
20             <td>59%</td>
21          </tr>
22          <tr>
23             <td>Google Chrome</td>
24             <td>78</td>
25             <td>28%</td>
26          </tr>
27          <tr>
28             <td>Safari</td>
29             <td>35</td>
30             <td>13%</td>
31          </tr>
32      </table>
33 </body>
34 </html>
    
```

```

1  table{
2      border:1px solid black;
3      border-collapse: collapse;
4  }
5
6  td{
7      border:1px solid black;
8      padding: 10px;
9  }
10
11 th{
12     border:1px solid black;
13     padding:10px;
14 }
15
16
    
```

2. Начинаем работать с объединением ячеек

Первый тег `<th>` занимает две ячейки по горизонтали и две ячейки по вертикали, поэтому задаем:

```
<th colspan = "2" rowspan = "2"></th>
```

Второй занимает две ячейки по горизонтали:

```
<th colspan = "2"></th>
```

Тег `<th>` с названиями браузеров точно так же как и первый занимает две ячейки по горизонтали:

```
<th colspan = "2"></th>
```

Проделав все действия выше, вы должны получить такой код и результат изображенный выше:

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <link rel="stylesheet" href="style07.css">
6  </head>
7  <body>
8      <table>
9          <tr>
10             <th colspan="2" rowspan="2">Браузер</th>
11             <th colspan="2">Посещения</th>
12         </tr>
13         <tr>
14             <th>Количество</th>
15             <th>В процентах</th>
16         </tr>
17         <tr>
18             <td colspan="2">Mozilla Firefox</td>
19             <td>163</td>
20             <td>59%</td>
21         </tr>
22         <tr>
23             <td colspan="2">Google Chrome</td>
24             <td>78</td>
25             <td>28%</td>
26         </tr>
27         <tr>
28             <td colspan="2">Safari</td>
29             <td>35</td>
30             <td>13%</td>
31         </tr>
32     </table>
33 </body>
34 </html>
    
```

Задание 5

Создайте таблицу, изображенную ниже (размеры отступов кратны 5):

Город	Посещения	Страниц	Время
СПб	199	18,02	00:13:45
Москва	69	нет данных	00:00:44
Киев	5		00:18:07
Всего посещений			273

Лабораторная работа № 7

Формы

Тег `<form>` устанавливает форму на веб-странице. Форма предназначена для обмена данными между пользователем и сервером. Область применения форм не ограничена отправкой данных на сервер, с помощью клиентских скриптов можно получить доступ к любому элементу формы, изменять его и применять по своему усмотрению.

Для отправки формы на сервер используется кнопка Submit, того же можно добиться, если нажать клавишу `Enter` в пределах формы. Если кнопка Submit отсутствует в форме, клавиша `Enter` имитирует ее использование.

Атрибут `action` в форме принимает ссылку(URL) который ведет либо на локальный скрипт, либо на сервер которые примут на себя обработку данных формы.

Поля в форме именуются тегом `<input>` имеют один важный атрибут – `type`, который определяет тип поля, который будет использован. `Type` может принимать значения:

Клиентские интернет-технологии

Тип	Описание	Вид
button	Кнопка.	<input type="button" value="Кнопка"/>
checkbox	Флажки. Позволяют выбрать более одного варианта из предложенных.	<input type="checkbox"/> Пиво <input type="checkbox"/> Чай <input type="checkbox"/> Кофе
file	Поле для ввода имени файла, который пересылается на сервер.	<input type="button" value="Выберите файл"/> Файл не выбран
hidden	Скрытое поле. Оно никак не отображается на веб-странице.	
image	Поле с изображением. При нажатии на рисунок данные формы отправляются на сервер.	<input alt="Отправить" type="image"/>
password	Обычное текстовое поле, но отличается от него тем, что все символы показываются звездочками. Предназначено для того, чтобы никто не подглядел вводимый пароль.	<input type="password"/>
radio	Переключатели. Используются, когда следует выбрать один вариант из нескольких предложенных.	<input type="radio"/> Пиво <input type="radio"/> Чай <input type="radio"/> Кофе
reset	Кнопка для возвращения данных формы в первоначальное значение.	<input type="button" value="Сбросить"/>
submit	Кнопка для отправки данных формы на сервер.	<input type="button" value="Отправить"/>
text	Текстовое поле. Предназначено для ввода символов с помощью клавиатуры.	<input type="text"/>

Рисунок 21 - Работа с формами

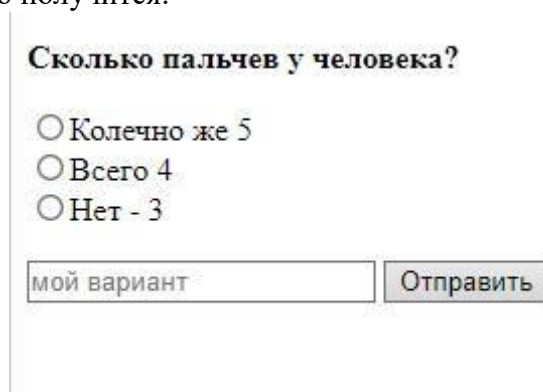
Чтобы понять данный материал выполните следующий код (так как у нас нет сервера, то никаких действий не выполнится после подтверждения формы):

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>form</title>
5  </head>
6  <body>
7    <form action="something_url">
8      <p>
9        <b>Сколько пальцев у человека?</b>
10     </p>
11
12     <p>
13       <input type="radio" name="answer" value="v1">Колечко же 5<br>
14       <input type="radio" name="answer" value="v2">Всего 4<br>
15       <input type="radio" name="answer" value="v3">Нет - 3</p>
16     <p>
17       <input type="text" name="answer" placeholder="мой вариант">
18       <input type="submit"></p>
19     </form>
20
21 </body>
22 </html>
    
```

Рисунок 22 - Работа с формами

В результате должно получиться:



Сколько пальцев у человека?

Колечно же 5
 Всего 4
 Нет - 3

Рисунок 23 - Результат

Задание:

- 1) Изучите тег, который также относится к форме “SELECT”
- 2) Напишите форму, которая спрашивает у пользователя запрашивает персональные данные (такие как Имя, Фамилия, телефон, возраст(с помощью <select>), адрес и секретный вопрос(например: «Любимы цвет»)). Форма должна содержаться в блоке шириной 400px автоматической высотой, цветом на усмотрение, с отступом сверху в 75px и располагаться по центру экрана.

Лабораторная работа № 8

Библиотека Bootstrap (Bootstrap framework)

Что такое Bootstrap?

- Bootstrap - это свободно распространяемая front-end библиотека, используемая для более быстрой и легкой разработки веб-страниц;
- Bootstrap включает основанные на HTML и CSS шаблоны для типографики, форм, кнопок, таблиц, навигации, модальных окон, слайдеров, и многие другие, такие как опциональные плагины JavaScript;
- Также Bootstrap дает возможность создавать адаптивный веб-дизайн.

Адаптивный веб-дизайн — дизайн веб-страниц, обеспечивающий правильное отображение сайта на различных устройствах, подключённых к интернету и динамически подстраивающийся под заданные размеры окна браузера.

Зачем использовать Bootstrap?

- Преимущества библиотеки Bootstrap: Простота использования: Любой, у кого есть базовые знания HTML и CSS может использовать Bootstrap;
- Адаптивные возможности: Адаптивный CSS библиотеки Bootstrap

подстраивается под телефоны, планшеты и большие мониторы;

- Используется метод Mobile-first: в Bootstrap 3.0, стили для мобильной разработки являются частью ядра библиотеки (т.е. они встроены в нее);
- Совместимость с различными браузерами: Bootstrap совместим со всеми современными браузерами (Chrome, Firefox, Internet Explorer, Safari, and Opera).

В лабораторных работах будет использоваться Bootstrap 3.3.7, есть новая версия Bootstrap 4, но ее использование требует понимания разметки с использованием Flexbox.

Где взять Bootstrap?

Есть два варианта, для того чтобы начать использовать Bootstrap на вашем собственном веб-сайте:

Вы можете:

- Скачать Bootstrap с сайта getbootstrap.com;
- Или добавить ссылку на Bootstrap из CDN (Content Delivery Network - Сеть доставки контента).

Скачать Bootstrap, если вы хотите скачать библиотеку Bootstrap и поместить ее в каталог вашего веб-сайта собственноручно, то Вам необходимо зайти на getbootstrap.com, и далее следовать предложенным там инструкциям.

Bootstrap CDN Если Вы не хотите скачивать Bootstrap сами, то Вы можете поместить ссылку на него из CDN (Content Delivery Network). (`<link rel = „stylesheet“ href = „ссылка CDN“>`).

Bootstrap 3.0 ориентирован на разработку для мобильных устройств

Bootstrap 3 создавался таким образом, чтобы была возможность создавать адаптивные страницы для мобильных устройств. Стили для разработки страниц для мобильных устройств являются частью ядра библиотеки.

Чтобы обеспечить правильное отображение и масштабирование на тач-скринах, добавляем следующий тег `<meta>` внутри элемента `<head>` :

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Значение атрибута `width=device-width` задает ширину страницы в соответствие с шириной экрана устройства (которая будет меняться в зависимости от устройства).

Значение атрибута `initial-scale=1` задает начальный масштаб страницы, когда страница загружается браузером впервые.

Контейнеры (Containers)

Bootstrap требует элемент `"container"` для всего содержимого сайта.

Существует два вида контейнеров, из которых моно выбирать:

- 1) Класс `.container` представляет собой адаптивный контейнер с

фиксированной шириной (справа и слева от контейнера есть некоторые отступы от границ окна).

2) Класс `.container-fluid` обеспечивает контейнер на всю ширину экрана (охватывает весь экран устройства без полей). Важно: Контейнеры не могут быть вложены в друг друга.

Важно: Контейнеры не могут быть вложены в друг друга.

Сетка Bootstrap

Сетка Bootstrap позволяет разместить контент в 12 колонок по ширине страницы. Также Вы можете объединять эти колонки, делая их более широкими. Пример сетки показан ниже:



Сетка Bootstrap является адаптивной, и колонки меняют свою ширину относительно размера экрана устройства: на большом экране содержимое страницы может выглядеть лучше при организации в три колонки, а на маленьком экране может быть лучше организовать блоки содержимого друг под другом.

Классы сетки Стек Bootstrap содержит 4 класса:

- `xs` (для телефонов)
- `sm` (для планшетов)
- `md` (для рабочих столов среднего размера)
- `lg` (для широких мониторов)

Классы, представленные выше, могут быть скомбинированы для создания более динамической и гибкой разметки.

Важно: Каждый класс имеет автомасштабирование, т.е. если Вы хотите установить одинаковую ширину блоков с `"xs"` и `"sm"` классами, Вам необходимо

установить ширину только для класса xs.

Правила использования сетки Bootstrap

- Строки должны быть помещены в класс `.container` (fixed-width) или `.containerfluid` (full-width) для надлежащего выравнивания и установки отступов;
- Используйте строки для создания горизонтальных групп элементов;
- Контент должен быть размещен в колонках, и только колонки могут быть непосредственными дочерними элементами строк;
- Предопределенные классы, такие как `.row` и `.col-sm-4` могут быть использованы для быстрой верстки страниц;
- Колонки имеют отступы относительно друг друга, которые создаются посредством свойства "padding". Эти отступы обнуляются в строках для первой и последней колонок посредством отрицательного значения `margin` для строк `.rows`;
- Колонки сетки создаются через определение количества колонок (из общего количества в 12 колонок), которые Вам необходимо объединить.

Например, три одинаковые колонки на странице будут представлены тремя классами `.col-sm-4` (см. пример ниже).

```
<div class="row">
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-4">.col-sm-4</div>
</div>
```

Рисунок 24 - `.col-sm-4`

Две колонки разной ширины

Следующий пример показывает, как получить две колонки разной ширины для планшетов (которые останутся такими же при масштабировании до больших рабочих столов):

Базовая структура сетки Bootstrap

```
<div class="row">
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-8">.col-sm-8</div>
</div>
```

Рисунок 25 - Базовая структура сетки

Следующий код показывает базовую разметку сетки Bootstrap:

```

<div class="container">
  <div class="row">
    <div class="col-**-**">
    </div> |
  </div>

  <div class="row">
    <div class="col-**-**"></div>
    <div class="col-**-**"></div>
    <div class="col-**-**"></div>
  </div>

  <div class="row"> ... </div>
</div>
    
```

Рисунок 26 - Базовая разметка сетки Bootstrap

Таким образом, для создания необходимой разметки, создается контейнер (<div class="container">). Далее, создается строка (<div class="row">). Затем, добавляется желаемое количество колонок (теги с соответствующими классами .col-**-**). Важно, чтобы количество колонок в классах .col-**-** всегда было равно 12 для каждой строки

Свойства сетки

В скриншоте ниже приведены свойства сетки Bootstrap для различных размеров устройств:

	Экстра малые устройства, телефоны (<768px)	Малые устройства, Планшеты (>=768px)	Средние устройства, мониторы (>=992px)	Широкоформатные устройства, мониторы (>=1200px)
Поведение сетки	Все время горизонтальное	Свернуто на старте, горизонтальное при превышении указанного размера экрана	Свернуто на старте, горизонтальное при превышении указанного размера экрана	Свернуто на старте, горизонтальное при превышении указанного размера экрана
Ширина контейнера	Auto	750px	970px	1170px
Префикс класса	.col-xs-	.col-sm-	.col-md-	.col-lg-
Количество колонок	12	12	12	12
Ширина колонок	Auto	~62px	~81px	~97px
Размер отступов	30px (15px на каждую сторону колонки)	30px (15px на каждую сторону колонки)	30px (15px на каждую сторону колонки)	30px (15px на каждую сторону колонки)
Вложенность	Да	Да	Да	Да
Смещение	Да	Да	Да	Да
Порядок столбцов	Да	Да	Да	Да

Рисунок 27 - Свойства сетки Bootstrap

Клиентские интернет-технологии

Задание:

- На основе полученных выше теоретических знаний сверстать макет, представленный ниже в виде скриншота с использованием Bootstrap (используйте официальный сайт для получения информации об элементах, которые вы не знаете как сверстать)

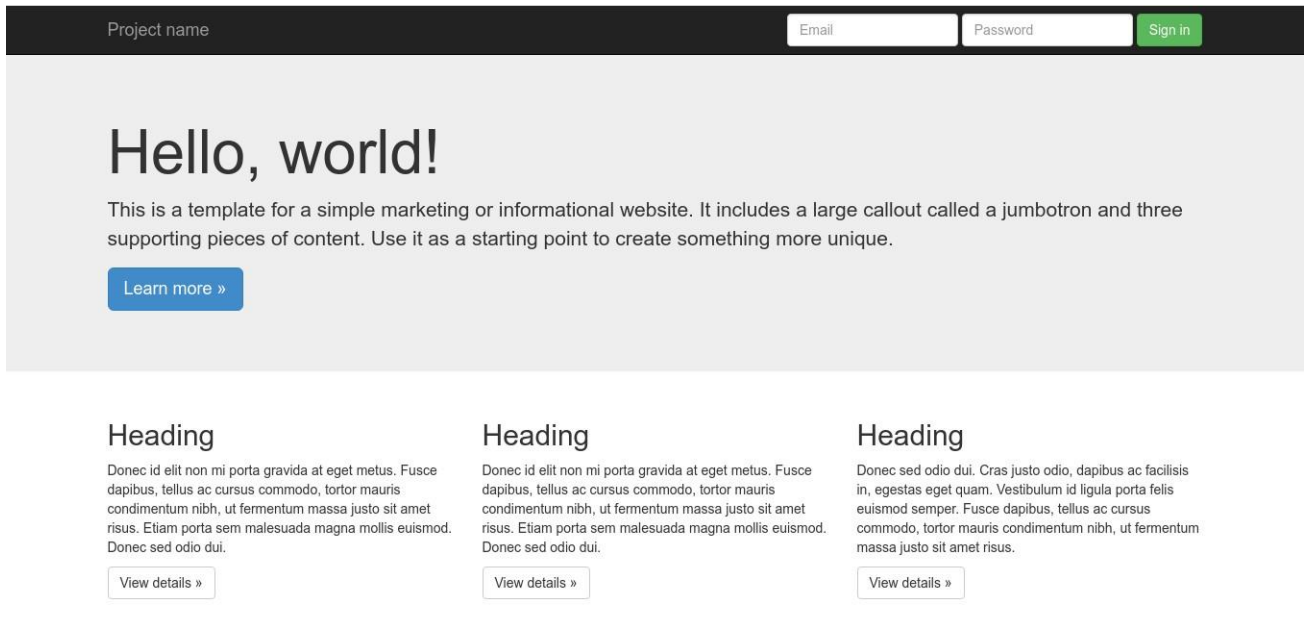


Рисунок 28 - Задание

Индивидуальное задание

Задание: Создать макет сайта на выбранную тематику.

https://github.com/ArioShaman/html_makets - данная ссылка ведет на репозиторий (хранилище) различных сверстанных макетов, многие из них используют гибкую адаптивную верстку, они созданы для примера верстки сложных элементов.

Для того, чтобы просмотреть макеты (код и результат), достаточно перейти по ссылке, скачать архив “Clone or download” -> “download ZIP” распаковать себе на компьютер и готово.

The screenshot shows the GitHub interface for the repository 'ArioShaman/html_makets'. At the top, there are navigation links for 'This repository', 'Search', 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below this, the repository name and statistics are displayed: 'Unwatch 1', 'Star 0', and 'Fork 0'. A navigation bar includes 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Insights', and 'Settings'. The repository description is 'Html makets for practices in front-end'. Below this, statistics show '4 commits', '1 branch', '0 releases', and '1 contributor'. A toolbar contains buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The file list includes 'blog', 'cooking', 'remgrad-company', 'shop', 'tovar-card', and 'README.md', each with a brief description and a 'just now' timestamp. A 'README.md' file is also visible at the bottom.

Рисунок 29 - GitHub

Список литературы

1. Кудряшев А.В. Введение в современные веб-технологии, Москва 2016
2. Штефен Вальтер, Создание приложений для Windows 8 с использованием HTML5 и JavaScript, Саратов: Профобразование, 2017
3. Никольский А.П. JavaScript на примерах, СПб.: Наука и Техника, 2017
4. Адамс Д.Р., Флloyd К.С., Основы работы с XHTML и CSS, М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016
5. Богданова С.В., Информационные технологии, Ставрополь, 2014, Учебное пособие
6. Соболев Б.В. и др., Информатика, Феникс, 2010
7. Кузнецова Л.В., Лекции по современным веб-технологиям, Москва, 2016
8. Кэгл К., XML, СПб.: Лори: Питер, 2006
9. Диков А.В., Веб-технологии HTML и CSS, Москва, 2012