

Министерство образования, науки и молодежной политики Нижегородской области  
Государственное бюджетное профессиональное образовательное учреждение  
«Арзамасский коммерческо-технический техникум»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ**  
**ОСНОВЫ САЙТОСТРОЕНИЯ**

Специальность СПО:  
**09.02.07 Информационные системы и программирование**

Арзамас  
2022 г.

**Рекомендованы к использованию**  
методическим объединением  
информационных дисциплин  
Протокол №\_\_\_\_  
от «\_\_»\_\_\_\_\_ 20\_\_ г

Председатель МО:  
\_\_\_\_\_ Н.И. Богомолова

Составлены в соответствии с требованиями  
к результатам освоения ППССЗ по  
специальности **09.02.07 Информационные  
системы и программирование**

Зам. директора по УиНМР  
\_\_\_\_\_ Н.В. Слюдова

*Н.Г. Саблукова*, к.п.н., преподаватель высшей квалификационной категории ГБПОУ  
«Арзамасский коммерческо-технический техникум»

Методические указания содержат задания к лабораторным работам, порядок их выполнения, рекомендации, перечень контрольных вопросов по каждой лабораторной работе, требования к знаниям и умениям. Приведен список основной литературы для подготовки к практическим работам.

Методические указания предназначены для обучающихся специальности **09.02.07 Информационные системы и программирование**.

## СОДЕРЖАНИЕ

Введение	4
Лабораторная работа № 1 «Использование основных тегов HTML»	5
Лабораторная работа № 2 «Работа со свойствами CSS»	11
Лабораторная работа № 3 «Работа с тегами формы в HTML и CSS»	18
Лабораторная работа № 4 «Создание и оформление таблиц в HTML и CSS»	22
Лабораторная работа № 5 «Верстка сайта на FlexBox»	23
Лабораторная работа № 6 «Верстка сайта на FlexBox»	31
Лабораторная работа № 7 «Создание выпадающего меню, использование свойств CSS3»	35
Лабораторная работа № 8 «Верстка адаптивного сайта»	39
Лабораторная работа № 9 «Верстка адаптивного сайта. Публикация сайта в сети Интернет»	45
Лабораторная работа № 10 «Зачетная работа»	49
Литература	51

## Введение

**Лабораторные работы** направлены на формирование у студентов практических умений верстки сайтов. Включенные в лабораторные работы задания стимулируют исследовательскую и творческую деятельность, развивают познавательные интересы, помогают научиться применять полученные знания на практике.

Содержанием лабораторных работ является выполнение заданий на применение языка HTML, CSS, блочной верстки при разработке сайтов.

Состав заданий для лабораторного занятия спланирован с расчетом, чтобы за отведенное время большинство обучающихся могли их выполнить качественно.

Выполнению лабораторных работ предшествует проверка знаний студентов – их теоретической готовности к выполнению задания.

Во время выполнения лабораторной работы используется индивидуальная и парная формы организации работы обучающихся. При индивидуальной форме организации занятий каждый обучающийся самостоятельно выполняет задание за отдельным персональным компьютером. При парной форме организации работ одна и та же работа выполняется микрогруппами по 2 человека за одним компьютером.

Каждая лабораторная работа оформляется и выполняется в электронном виде. В оформление работы входит запись номера лабораторной работы, темы, цели, ответов на контрольные вопросы.

Выполнение лабораторных работ по дисциплине ОУД. 18 Основы сайтостроения направлено на формирование общих компетенций:

ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе общечеловеческих ценностей, применять стандарты антикоррупционного поведения.

ОК 7. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.

ОК 9. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.

ОК 11. Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере.

## Лабораторная работа № 1 ИСПОЛЬЗОВАНИЕ ОСНОВНЫХ ТЕГОВ HTML

**ЦЕЛЬ РАБОТЫ:** научиться использовать основные теги языка HTML для разработки и оформления сайта.

Для выполнения работы необходимо *знать* базовые технологии (HTML и CSS), используемые при написании web-страниц; необходимо *уметь* использовать готовые прикладные компьютерные программы по профилю подготовки, разрабатывать web-страницы на языках HTML и CSS.

**ВРЕМЯ ВЫПОЛНЕНИЯ:** 90 минут.

**ОБОРУДОВАНИЕ:** ПК ИМВ, OS Windows, браузер (Google Chrome, Mozilla, Opera), Notepad++, Sublime Text

### КРАТКАЯ ТЕОРИЯ И МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

#### Основные понятия языка HTML

**HTML** – это язык разметки для гипертекста. Он состоит из тегов, которые могут быть одиночные и парные.

**Парные теги:** <название\_тега>... </название\_тега> -  
открывающийся тег      закрывающийся тег

**Одиночный тег:** <Название\_тега />

#### Структура HTML документа

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>Моя страница</title>  
    <meta charset="utf-8" >  
</head>  
<body>  
    ...  
</body>  
</html>
```

Сначала браузеру нужно указать версию языка HTML с помощью тега !DOCTYPE.

**<!DOCTYPE html>** - это строка для самой последней версии html, для ранних версий данный тег будет длиннее.

Раздел **<head>** - голова документа, в нем размещается служебная информация, например, объявляются meta-теги, которые могут иметь разное назначение (кодировка, описание страницы, ключевые слова), подключаются файлы каскадных листов стилей (css), подключаются файлы скриптов. Все что находится в данном разделе не видно пользователю, за исключением содержимого тега **<title>**, в котором указывается название веб-страницы.

Тег **<meta charset = "utf-8">** – определяет кодировку сайта.

В разделе **<body>** располагаются все элементы, который видит пользователь. Именно с этим разделом в основном и работает разработчик сайтов.

#### Основные теги HTML-документов

- 1) Для отображения **заголовков** служит парный тег **<h>** и указывается цифра от 1 до 6, которая соответствует уровню заголовка.

<h1>Заголовок</h1> - заголовок 1 уровня (самый большой заголовок)  
<h6>Заголовок</h6> - заголовок 6 уровня (самый маленький заголовок)

- 2) Для **абзацев** используется парный тег <p>, внутри него помещается тот текст, который нужно отобразить.  
<p>Здесь мы напишем первый абзац.</p>

3) **Теги выделения текста**

<b>Текст</b> - жирный текст;                      <i>Текст</i> - текст курсивом  
<u>Текст</u> - подчеркнутый текст;              <s>Текст</s> - зачеркнутый текст

- 4) **Спецсимволы** – это знаки, которых нет на клавиатуре.

&copy – копирайт    &larr и &rarr – стрелка влево и стрелка вправо  
&nbsp; – неразрывный пробел                              &lt; и &gt; – знаки < и >  
&laquo и &raquo – символы « и »  
*Например, &copy DM*

5) **Гиперссылки**

Относительные гиперссылки

<a href = "file.html">Текст, который видит пользователь</a>

Атрибутом для ссылки является адрес (href) той страницы, куда мы переходим.

Абсолютные ссылки – указывается полный путь до сайта.

<a href = "<http://yandex.ru>" target = "\_blank">Перейти на Яндекс</a>

Атрибут target = "\_blank" – позволяет открывать страницу в новой вкладке, если его не прописать, то страница будет отображаться в той же вкладке.

6) **Списки**

Маркированный список

```
<ul>
  <li>Первый элемент</li>
  <li>Второй элемент</li>
  <li>Третий элемент</li>
</ul>
```

Нумерованный список

```
<ol>
  <li>Первый элемент</li>
  <li>Второй элемент</li>
  <li>Третий элемент</li>
</ol>
```

- 7) **Изображения** добавляются с помощью тега <img>. У этого тега есть обязательный атрибут src, в значении указывается путь к изображению.



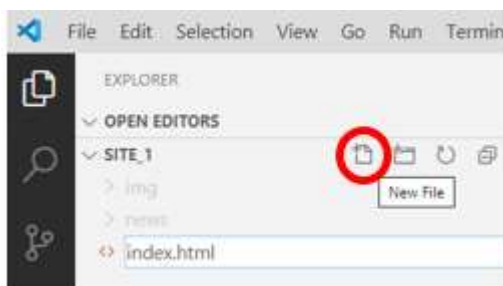
В примере выше у тега img прописаны следующие атрибуты:

src="img/my\_foto.jpg"                      (путь к файлу)  
alt="это моя фотография" (альтернативный текст, если картинка не загрузится)  
title = "это моя фотография" (всплывающая подсказка)  
width="300"                                      (ширина изображения)

## ПОРЯДОК ВЫПОЛНЕНИЯ И ФОРМА ОТЧЕТНОСТИ

### Задание 1. Создание главной страницы сайта.

1. Создайте папку с именем site\_1, в ней создайте 2 папки с именами img и news. Скопируйте из папки media картинки в папку img Вашего сайта.
2. Запустите программу VS Code и откройте папку site\_1 в этой программе. Чтобы открыть папку в программе можно просто перетащить папку в рабочую область программы или использовать команду File – Open Folder.
3. С помощью команды New file в корне папки site\_1 создайте новый файл index.html.



4. В файле index.html задайте структуру html-документа. Чтобы быстро задать структуру, наберите с клавиатуры восклицательный знак ! и нажмите кнопку **Enter**. Сработает плагин Emmet и появится необходимая структура.

```
<> index.html ×
<> index.html > html > head > meta
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  |   <title>Document</title>
7  </head>
8  <body>
9  |
10 </body>
11 </html>
```

5. Задайте название веб-страницы «Сайт группы \_\_\_» (в теге <title>).
6. В раздел <body> поместите поочередно теги, указанные в пунктах а-в. Должна получиться веб-страница, как на рисунке.
  - а. Тег заголовка первого уровня с названием главной страницы – темой сайта: **Сайт группы (номер группы)**.
  - б. Маркированный список, включающий следующие пункты: Главная, Галерея, Актив группы, Мероприятия, Регистрация.

## Сайт группы ИС

- Главная
- Галерея
- Актив группы
- Мероприятия
- Регистрация

Сохраните файл index.html и откройте его в браузере. *Сохранение файлов и проверку в браузере HTML-кода необходимо делать после каждого пункта заданий.*

7. Добавьте к каждому пункту списка ссылки на остальные страницы сайта: Главная (index.html), Галерея (gallery.html), Актив группы (asset.html), Мероприятия (news.html), Регистрация (check.html).

Пример оформления одного из пунктов списка с вложенной ссылкой:

```
<li><a href="index.html">Главная</a> </li>
```

Остальные пункты списка измените аналогично, используя соответствующие названия страниц. *В следующих пунктах лабораторной работы эти страницы будут созданы.*

## Сайт группы ИС

- [Главная](#)
- [Галерея](#)
- [Актив группы](#)
- [Мероприятия](#)
- [Регистрация](#)

8. Далее поместите тег добавления изображения – общей фотографии группы. Уменьшите размер изображения.
9. Ниже добавьте теги абзацев с описанием своей группы.
10. Далее добавьте несколько тегов <br> - тег перехода на новую строку.
11. Далее добавьте тег линии <hr> и поставьте тег копирайта с указанием информации об авторе сайта.



## Сайт группы ИС

- [Главная](#)
- [Галерея](#)
- [Актив группы](#)
- [Мероприятия](#)
- [Регистрация](#)



Мы студенты ГБПОУ "Арзамасский коммерческо-технический техникум".

Поступили в \_\_\_\_ году на специальность 09.02.07 Информационные системы и программирование.

Наш классный руководитель: \_\_\_\_\_

©ИС

Главная страница сайта

### Задание 2. Создание страницы «Галерея»

1. В корне сайта site\_1 создайте новую страницу с именем gallery.html.
2. Задайте структуру html-документа.
3. Добавьте теги, чтобы получилась веб-страница, как на рисунке. Можно скопировать из файла index.html необходимый код.

## Галерея фотографий

- [Главная](#)
- [Галерея](#)
- [Актив группы](#)
- [Мероприятия](#)
- [Регистрация](#)

4. Далее на страницу Галерея добавьте уменьшенные копии сохраненных картинок, при наведении на картинку мышью должно появляться название изображения.

## Галерея

- [Главная](#)
- [Галерея](#)
- [Актив группы](#)
- [Мероприятия](#)
- [Регистрация](#)



©ИС

Страница «Галерея»

5. **\*\*Добавьте теги ссылок для того, чтобы можно было посмотреть картинку в полном размере в отдельном окне.**

### **\*Задание 3. Создание страницы «Актив группы»**

На странице «Актив группы» после меню перечислите студентов, входящих в актив Вашей группы. Используйте теги нумерованного и маркированного списков, а также теги для выделения текста полужирным и курсивом (см. рисунок).

## Актив группы

- [Главная](#)
  - [Галерея](#)
  - [Актив группы](#)
  - [Мероприятия](#)
  - [Регистрация](#)
1. **Староста группы**
    - *Фамилия, имя*
  2. **Заместитель старосты**
    - *Фамилия, имя*
  3. **Учебный сектор**
    - *Фамилия, имя*
    - *Фамилия, имя*
  4. **Физорг**
    - *Фамилия, имя*

©ИС

Страница «Актив группы»

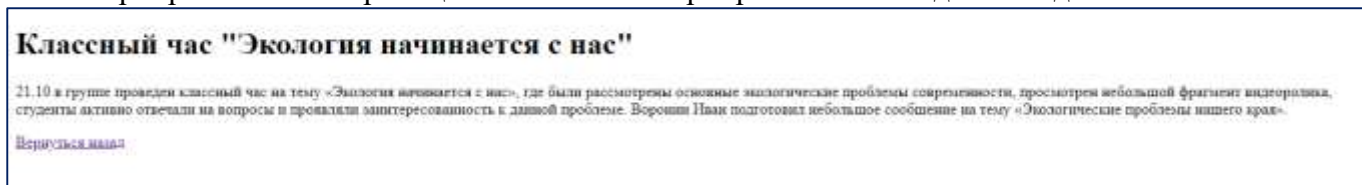
### **\*\*Задание 4. Создание страницы «Мероприятия»**

1. На странице «Мероприятия» после меню поместите теги для отображения названий мероприятий. Название мероприятия должно быть в виде ссылки. (см. рисунок).



Страница «Мероприятия»

2. При щелчке на названии мероприятия должна открыться новая веб-страница с описанием этого мероприятия. Все страницы с описанием мероприятия необходимо создавать в папке news.



Страница с мероприятием

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Опишите структуру HTML документа.
2. Какие основные теги использовались при создании сайта, для чего служит каждый тег?

### Лабораторная работа № 2 РАБОТА СО СВОЙСТВАМИ CSS

**ЦЕЛЬ РАБОТЫ:** научиться использовать каскадные таблицы стилей для оформления сайта.

Для выполнения работы необходимо *знать* базовые технологии (HTML и CSS), используемые при написании web-страниц; необходимо *уметь* использовать готовые прикладные компьютерные программы по профилю подготовки, разрабатывать web-страницы на языках HTML и CSS.

**ВРЕМЯ ВЫПОЛНЕНИЯ:** 90 минут.

**ОБОРУДОВАНИЕ:** ПК IBM, OS Windows, браузер (Google Chrome, Mozilla, Opera), Notepad++, Sublime Text

### КРАТКАЯ ТЕОРИЯ И МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

**CSS (Cascading Style Sheets)** – это каскадные таблицы стилей, которые применяются для описания внешнего вида веб-документа, написанного при помощи языка разметки HTML.

Стили удобно хранить во внешнем файле, например, style.css и подключать их к html-странице следующим тегом в разделе head:

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

### Синтаксис CSS:

```
Селектор {  
    Свойство 1: значение 1;  
    Свойство 2: значение 2;  
}
```

Селекторы могут быть следующих типов:

1. **Селектор тега** – связывает стиль с конкретным тегом на странице.

*Пример 1. Оформление всех тегов абзаца <p>*

```
p {  
    font-size: 18px; /*размер шрифта у абзаца*/  
    background-color: #eee; /*заливка абзаца*/  
}
```

2. **Селектор id** – используется для создания стиля для уникальной метки (идентификатора).

*Пример 2. Оформление только первого абзаца <p>*

Файл index.html

```
<p id = "first">Первый пункт </p>  
<p> Второй пункт </p>  
<p >Третий пункт </p>
```

Файл style.css

```
#first {  
    Color: red; /*цвет шрифта первого абзаца*/  
    Font-weight: bold; /*толщина шрифта первого абзаца*/  
}
```

3. **Селектор class** – используется для выделения подобных элементов.

*Пример 3. Выделение всех нечетных абзацев <p>*

Файл index.html

```
<p class = "odd">Первый пункт </p>  
<p> Второй пункт </p>  
<p class = "odd">Третий пункт </p>  
<p> Четвертый пункт </p>  
<p class = "odd" >Пятый пункт </p>
```

Файл style.css

```
.odd {  
    background-color: #eee; /*заливка нечетных абзацев*/  
}
```

### Вложенность стилей

При помощи CSS можно управлять различными вложенными конструкциями. Для

управления вложенностью в CSS существуют несколько специальных селекторов.

1) **Контекстные селекторы** – позволяют работать со вложенными тегами.

*Пример 4. Меню создано с помощью списка. Все теги li, которые лежат внутри списка теги, будут находиться в черной рамке.*

Файл index.html

```
<ul id = "menu">
  <li> <a href = "yandex.ru">Главная </a></li>
  <li><a href = "google.ru">Главная </a></li>
  <li><a href = "mail.ru">Главная </a></li>
</ul>
```

Файл style.css

```
#menu li {
  Border: 1px solid black;
}
```

Через пробел можно писать несколько тегов. Пробел будет означать, что стили применяются ко всем тегам, перечисленным через пробел.

*Пример 5. У ссылок меню будет убрано подчеркивание.*

```
#menu li a {
  Color: black;
  Text-decoration: none;
}
```

2) **Дочерние селекторы** – позволяют выбирать только те теги, которые являются прямыми потомками определенного элемента.

*Пример 6.*

Файл index.html

```
<p class="main">
  <a href=" yandex.ru ">Яндекс</a>
</p>
<p class="main">
  <em><a href="mail.ru">Почта</a></em>
</p>
```

Файл style.css

```
p. main > a {
  font-size: 20px;
  color: green;
  background: yellow;
}
```

В первом параграфе, ссылка является дочерним элементом для этого параграфа, т.к. вложена только в параграф. А во втором параграфе, ссылка для параграфа не будет являться дочерним элементом, т.к. она вложена в тег em, и соответственно будет являться дочерним элементом для тега em.

В разделе CSS кода после обращения к селектору параграфа p с классом main ставится знак больше, а затем тег a. В этом примере ТОЛЬКО к ссылке первого параграфа будет применен заданный стиль.

## Псевдоклассы и псевдоэлементы

Псевдоклассы и псевдоэлементы позволяют назначить CSS стили структурам, существование которых на веб-странице не обязательно.

Псевдокласс означает состояние элемента:

- hover – состояние, когда навели мышку.
- active – элемент активен, нажат элемент.
- visited – посещенная ссылка
- link – непосещенная ссылка
- focus – для разных элементов, например input

### Синтаксис

```
тег:псевдокласс {  
    свойство: значение;  
}
```

Псевдоэлементы:

- first-letter – первая буква
- first-child – первый пункт

## ПОРЯДОК ВЫПОЛНЕНИЯ И ФОРМА ОТЧЕТНОСТИ

### Задание 1. Подключение стилей css, оформление шапки сайта.

1. Откройте в VS Code папку site\_1, созданную в лабораторной работе №1, в ней создайте файл style.css. В этом файле будут храниться все стили Вашей работы.
2. Подключите этот файл ко всем страницам сайта (index.html, asset.html, news.html, gallery.html, check.html).

### Универсальный селектор

3. В файле style.css сначала добавьте стили, которые будут применяться ко всем селекторам вашего сайта: используйте при этом универсальный селектор \* , удалите внешние и внутренние отступы.

```
1 /*стили для всех селекторов*/  
2 *{  
3     margin: 0; /*удаляем внешние отступы*/  
4     padding: 0; /*удаляем внутренние отступы*/  
5 }
```

### Селекторы тегов. Стили для тела всех веб-страниц

4. Для селектора body установите неяркий цвет фона и параметры шрифта: цвет шрифта, размер 20px, гарнитуру (Arial, "Courier New", monospace).

```

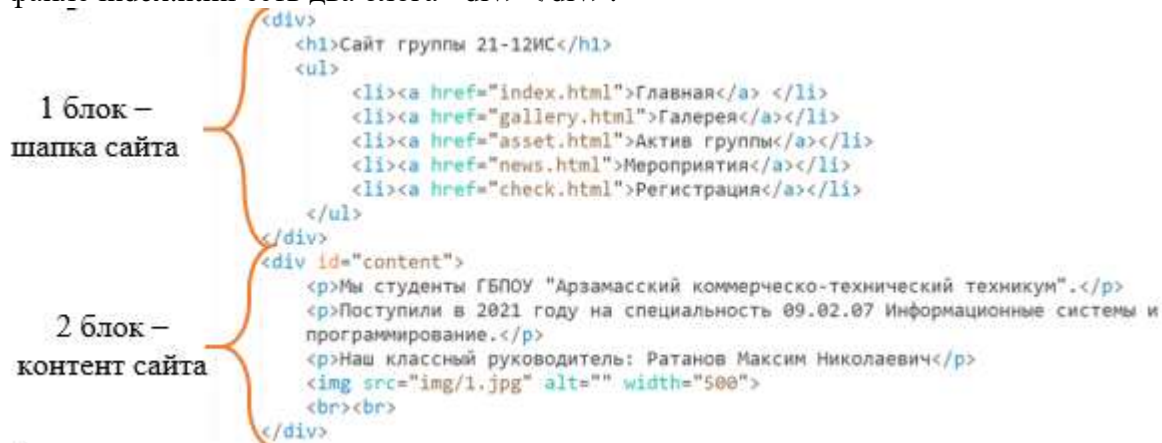
6  body{
7      background: #fae4d6;
8      color:#123;
9      font-size:20px;
10     font-family: Arial, "Courier New", monospace;
11 }

```

5. Установите для заголовка h1 свойства:
- цвет (отличный от цвета основного текста);
  - размер (32px);
  - выравнивание по центру;
  - все буквы должны быть заглавными;
  - верхний и нижний внутренние отступы по 20px.

### Селектор id.

В файле index.html есть два блока <div></div>.



Чтобы применить к каждому из блоков свои стили зададим для них id.

6. Для первого блока задайте id="header".

```

<div id="header">
  <h1>Сайт группы 21-12ИС</h1>
  <ul>
    <li><a href="index.html">Главная</a> </li>
    <li><a href="gallery.html">Галерея</a></li>
    <li><a href="asset.html">Актив группы</a></li>
    <li><a href="news.html">Мероприятия</a></li>
    <li><a href="check.html">Регистрация</a></li>
  </ul>
</div>

```

7. В файле style.css задайте для блока с id="header" следующие стили:

- Ширину 100%;
- Высоту 500px;
- Картинку для фоновой заливки;
- Свойство **background-size: cover**, позволяющее картинке растянуться во весь блок.



```
#header{
    width: 100%;
    height: 500px;
    background: url('img/header.jpg');
    background-size: cover;
}
```



## Задание 2. Оформление команд меню. Использование контекстных селекторов и псевдоклассов

1. Задайте для меню (первого списка ul) произвольный класс, например, class="nav".
2. Для меню задайте ширину, высоту и выравнивание блока с меню по центру:

```
.nav{
    width: 80%;
    height: 40px;
    margin: 0 auto;
}
```

3. Преобразуем вертикальное меню в горизонтальное. Для этого для пунктов меню – тегов li применим свойство float: left – это свойство задает обтекание элементов слева.

Используем для задания контекстный селектор:

```
.nav li{
    float:left;
}
```

Т.е. стили будут применяться только к тем тегам li, которые находятся внутри блока .nav

4. Добавьте в предыдущий селектор стили, чтобы получилось оформление сайта, как на рисунке:



5. Измените стили у ссылок (используйте контекстные селекторы).





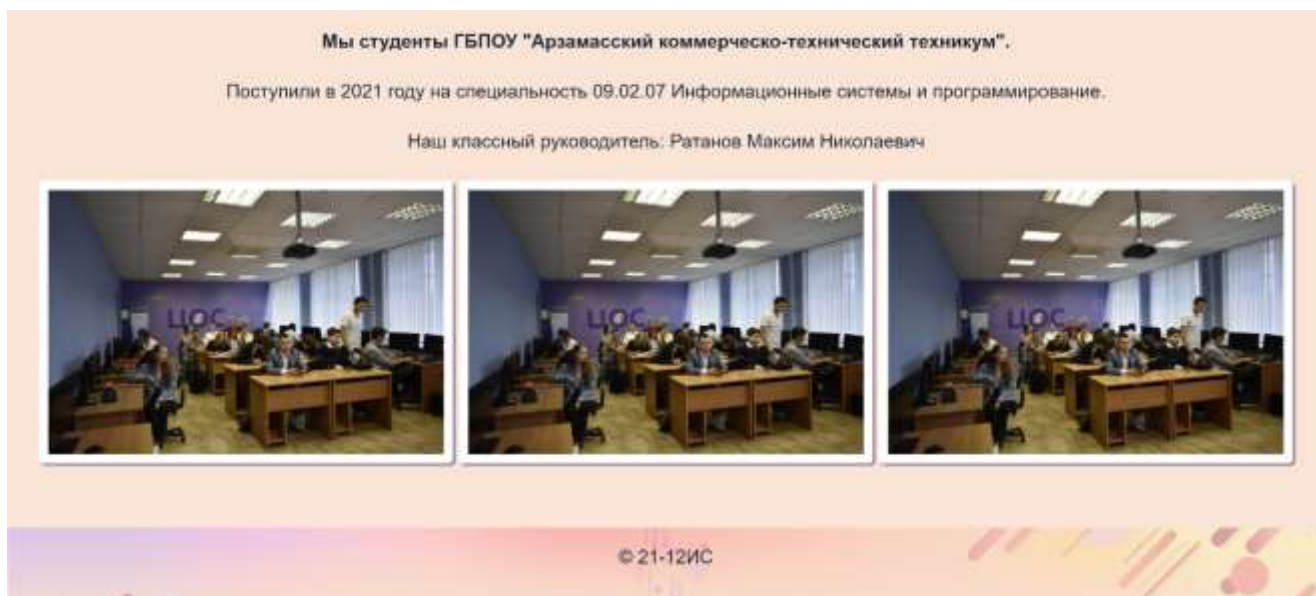
6. Задайте другой стиль для ссылок меню при наведении на них мыши (использован псевдокласс *hover* - состояние, когда навели мышку):

```
.nav li a: hover{  
    color: #fff;  
    background: #7a638d;  
    text-decoration: none;  
}
```

### Задание 3. Оформите контент и футер главной страницы.

1. Самостоятельно оформите контент и футер главной страницы сайта, чтобы получилось, как на рисунке.

2. При наведении мышкой на картинки, рамка должна исчезнуть, но сама картинка не должна при этом смещаться, т.е. должна оставаться неподвижной (для выполнения этого задания используйте цвет рамки прозрачный – *transparent*).



Веб-страница целиком:



### Лабораторная работа № 3 РАБОТА С ТЕГАМИ ФОРМЫ В HTML И CSS

**ЦЕЛЬ РАБОТЫ:** научиться добавлять основные элементы форм на web-страницу; использовать свойства CSS для оформления форм.

Для выполнения работы необходимо *знать* базовые технологии (HTML и CSS), используемые при написании web-страниц; необходимо *уметь* использовать готовые прикладные компьютерные программы по профилю подготовки, разрабатывать web-страницы на языках HTML и CSS.

**ВРЕМЯ ВЫПОЛНЕНИЯ:** 90 минут.

**ОБОРУДОВАНИЕ:** ПК ИМВ, OS Windows, браузер, Notepad++, Sublime Text

#### КРАТКАЯ ТЕОРИЯ И МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

**Формы** необходимы, чтобы получать от пользователя какую-либо информацию.

**Задание формы:**

```
<form method = "post" enctype = "multipart/form-data">  
<fieldset>
```

```
<legend>Регистрация нового пользователя</legend>
  <!-- Здесь задаются элементы формы --!>
</fieldset>
</form>
```

Значение атрибута `method="post"` означает, что запрос на сервер будет отправлен методом `post`. Атрибут `enctype="multipart/form-data"` необходим для того, чтобы загружать файлы.

Для того, чтобы отделить форму или ее часть, можно использовать тег `<fieldset>`, а внутри тэга `<legend>` поместить название данной формы или ее части. Тогда браузер выделит эту часть рамкой, в которой будет находиться название.

### Элементы формы

1) `<input type = "text" name = "second_name" size = "50">` - обычное однострочное **поле ввода** (можно использовать для ввода фамилии, имени, отчества и т.д.).

Атрибут `name = "second_name"` задает имя поля

Атрибут `size = "50"` задает размер поля

Обычно при помощи дополнительного тега `<label>` устанавливается связь между определенным элементом формы и той надписи, которая соответствует данному элементу.

*Например, код для поля ввода фамилии может выглядеть следующим образом:*

```
<label>
```

    Фамилия

```
    <input type = "text" name = "second_name" size = "50">
```

```
</label>
```

или

```
<label for="s_name"> Фамилия</label>
```

```
<input type = "text" name = "second_name" id = "s_name" size = "50">
```

Чтобы отобразить текст в текстовом поле, на кнопке или связанный текст используется атрибут **Value**.

2) `<input type = "password" name = "pass">` - однострочное поле для ввода **пароля**.

3) `<input type="radio" name = "gender">` - выбор единственного варианта в виде **переключателя**.

Причём у переключателей из одной группы должно быть *одинаковое имя* и разные значения, которые задаются с помощью `value`. Если задать атрибут `checked="checked"`, то есть возможность отметить любой пункт по умолчанию.

*Например,*

```
<input type="radio" name="proba" value="yes" checked>Да
```

```
<input type="radio" name="proba" value="no"> Нет
```

4) `<input type = "checkbox" name = "confirm">` - выбор нескольких вариантов в виде **флажков**.

Чекбокс не подразумевает выбор одного элемента из нескольких. Поэтому если в одной форме есть несколько чекбоксов, то имена у них должны быть *разными*.

5) Элемент `<select>` представляет собой **выпадающий список**, элементы которого указываются в тэге `<option>`.

*Например, код для выбора года рождения:*

```
<select name = "year">
  <option value="1990"> 1990 </option >
  <option value="1991"> 1991 </option >
  ...
</select>
```

- 6) Тег `<textarea>` - **многострочное поле ввода** для большого количества информации. С помощью атрибута `cols` можно задать ширину этого поля, а с помощью атрибута `rows` указать число строк данного поля.

*Например,*

```
<textarea name = "opinion" cols="40" rows="4">
</textarea>
```

- 7) `<input type="file" name = "ava">` - добавление файла .
- 8) `<input type = "submit" value= "Сохранить">` - кнопка для отправки введенных данных на сервер.
- 9) `<input type = "reset" value = "Не нажимай">` - кнопка для очистки формы
- 10) `<input type = "button" value = "Жми">` - просто кнопка

## ПОРЯДОК ВЫПОЛНЕНИЯ И ФОРМА ОТЧЕТНОСТИ

### Задание 1. Создание страницы «Регистрация» (рис. 1).

1. В папке с сайтом, созданным в лабораторных работах 1,2 создайте страницу Регистрации – файл `check.html`. Задайте структуру веб-документа.
2. В разделе `body` добавьте теги, которые задают форму (рис. 1).

```
<body>
  <h1>Регистрация</h1>
  <form action="">

  </form>
</body>
```

Рис. 1

3. Внутри тега `form` поместите теги, которые добавляют на страницу текстовое поле ввода для фамилии:

```
<form action="">
  <label for="surname">Введите фамилию</label>
  <input type="text" id="surname" name="surname" size="30">
</form>
```

Рис. 2

4. Ниже аналогично добавьте поля ввода для следующих элементов:
  - Имя

- Отчество
- Пароль – он должен вводиться звездочками или кружочками
- Адрес электронной почты
- Адрес сайта (в поле ввода уже должен быть указан протокол http://)

Добавьте в поля подсказки.

Поля Фамилия, имя, пароль, адрес электронной почты сделайте обязательными для заполнения.

Разместите элементы так, как показано на рисунке 3.

Рис. 3

5. Ниже добавьте теги Переключатели для выбора пола.
6. Ниже добавьте тег Выпадающий список для выбора года рождения (выпадающего списка задайте 10 пунктов).
7. Ниже добавьте теги Флажки для выбора нескольких вариантов Ваших интересов.
8. Ниже поставьте тег для добавления фотографии на сайт.
9. Ниже добавьте тег для задания многострочного поля ввода «О себе».
10. Ниже добавьте кнопки «Зарегистрироваться» и «Очистить форму».
11. Добавьте группировку для всех элементов и заголовков формы: Регистрация нового пользователя

Рис. 4

**Задание 2. Самостоятельно оформите форму стилями CSS. Включите страницу с регистрацией в созданный ранее сайт.**

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для чего предназначена форма на сайте?
1. Какие значения может принимать атрибут type у тега input?

### Лабораторная работа № 4

## СОЗДАНИЕ И ОФОРМЛЕНИЕ ТАБЛИЦ В HTML И CSS

**ЦЕЛЬ РАБОТЫ:** научиться добавлять таблицы на web-страницу; использовать свойства CSS для оформления таблиц.

Для выполнения работы необходимо *знать* базовые технологии (HTML и CSS), используемые при написании web-страниц; необходимо *уметь* использовать готовые прикладные компьютерные программы по профилю подготовки, разрабатывать web-страницы на языках HTML и CSS.

**ВРЕМЯ ВЫПОЛНЕНИЯ:** 90 минут.

**ОБОРУДОВАНИЕ:** ПК ИМВ, OS Windows, браузер, Notepad++, Sublime Text

### КРАТКАЯ ТЕОРИЯ И МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

**Таблица** в html состоит из некоторого количества столбцов и строк, которые формируют ячейки в таблице.

Таблица в html задается с помощью тега:

```
<table>
  <tr>
    <td>столбец </td>
    <td>столбец </td>
    <td>столбец </td>
  </tr>
</table>
```

<table> - это контейнер таблицы

<tr> - это строка

<td> - это столбец (ячейка)

<th> - то же самое, что и тег <td>, только в этой ячейке текст будет полужирным, и выровненный посередине ячейки.

Для объединения ячеек используются следующие атрибуты:

rowspan - объединение по вертикали

colspan – объединение по горизонтали

Например, для объединения двух ячеек в одной строке используется следующий html-код: <td rowspan="2">Объединен по вертикали</td>

### Стилевое оформление таблиц

#### Граница у таблицы и ячеек

```
table {
  border: 2px solid #000;
  border-collapse: collapse;
}
td {
  border: 1px solid #00F;
}
```

## ПОРЯДОК ВЫПОЛНЕНИЯ И ФОРМА ОТЧЕТНОСТИ

Задание 1. Создайте веб-страницу с таблицей по образцу:

№ п/п	Наименование товара	Количество, кг	Цена за 1 кг, руб	Стоимость, руб
1.	Томаты свежие	10	69	690
2.	Огурцы свежие	5	48	240
<b>ИТОГО</b>				<b>930</b>

Задание 2. Создайте страницу электронного журнала группы по образцу:

### Электронный журнал по информатике

Фамилия, имя студента	Март							Месяц
	2	9	12	16	19	24	30	
Иванов Иван	5		4	3				
Петров Петр		3		3	5		4	

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как задается таблица в HTML?
2. Какие атрибуты используются для объединения ячеек в таблице?

### Лабораторная работа № 5 ВЕРСТКА САЙТА НА FLEXBOX

**ЦЕЛЬ РАБОТЫ:** научиться верстать макет сайта с помощью блоков и технологии FlexBox, использовать CSS для оформления основных компонентов сайта.

Для выполнения работы необходимо *знать* базовые технологии (HTML и CSS), используемые при написании web-страниц; необходимо *уметь* использовать готовые прикладные компьютерные программы по профилю подготовки, разрабатывать web-страницы на языках HTML и CSS.

**ВРЕМЯ ВЫПОЛНЕНИЯ:** 90 минут.

**ОБОРУДОВАНИЕ:** ПК ИМВ, OS Windows, браузер (Google Chrome, Mozilla, Opera), Notepad++, Sublime Text

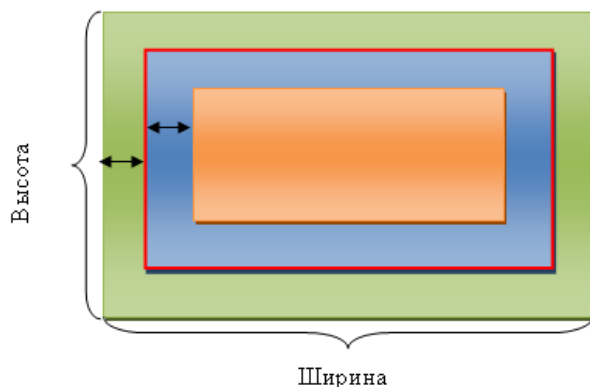
### КРАТКАЯ ТЕОРИЯ И МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ



Блочная верстка сайтов в настоящее время является самой популярной.

Основным тегом для структурирования страницы является тег `<div>`, с помощью которого можно делить web-страницу на блоки. Внутри данного тега можно разместить другие HTML теги, которые требуются для создания содержимого сайта (ссылки, текст, изображения и тд.). В отличие от строгих привязок стандартных HTML-тегов к своему содержимому («p» — к абзацам, «a» — к ссылкам, «img» — к изображениям), тэг `div` является нейтральным.

При блочной верстке необходимо обязательно учитывать внешние и внутренние отступы:



Внутренние отступы – отступы от края элемента до рамки (padding) .

Внешние отступы – отступы до другого объекта (margin).

Чтобы заставить браузер выводить элементы веб-страницы в определенной последовательности и четко заданных местах используются такие свойства CSS, как **float** и **clear**.

Свойство **float** определяет, по какой стороне будет выравниваться элемент, при этом остальные элементы будут обтекать его с других сторон. Выравнивание элемента происходит по краям родительского элемента или по краям окна браузера.

Значения свойства **float**:

- **left** - Выравнивает элемент по левому краю, остальные элементы обтекают его справа.
- **right** - Выравнивает элемент по правому краю, остальные элементы обтекают его слева.
- **none** - Выравнивание элемента не задается. (по умолчанию)

Свойство **clear** запрещает обтекание элемента с левой и/или правой стороны. Если для элемента установлено обтекание с помощью свойства **float**, то **clear** отменяет обтекание данного элемента с указанных сторон.

Значения:

- **both** - Запрещает обтекание элемента с левой и правой стороны.
- **left** - Запрещает обтекание элемента с левой стороны.
- **right** - Запрещает обтекание элемента с правой стороны.
- **none** - Запрет на обтекание элемента не задается (по умолчанию)

Позиционированием называют определение конкретного месторасположения на странице того или иного элемента. Чтобы позиционировать какой-либо элемент к нему применяют свойство **position** и одно из его возможных значений:

- **absolute** - Абсолютное позиционирование элемента.
- **relative** - Относительное позиционирование элемента.
- **fixed** - Фиксированное позиционирование элемента.
- **static** - Статическое позиционирование элемента. (Элемент отображаются как обычно.)

Абсолютно позиционированный элемент (**position: absolute**) выводится из общего потока и, несмотря на другие элементы и их взаимное расположение, занимает указанное место на странице



от края/краёв окна браузера. При таком способе позиционирования один элемент может накладываться поверх другого.

Для того чтобы позиционировать элемент от края/краёв окна браузера, используются следующие свойства CSS:

- **bottom** - Расстояние от нижнего края окна браузера.
- **left** - Расстояние от левого края окна браузера.
- **right** - Расстояние от правого края окна браузера.
- **top** - Расстояние от верхнего края окна браузера.

Относительное позиционирование (**position: relative**) определяет место элемента относительно краёв элемента родителя и не выводится из общего потока. Так же, как и в случае с абсолютным позиционированием расстояния от края/краёв родительского элемента задаётся с помощью свойств: **bottom, left, right, top**.

Фиксированное позиционирование (**position: fixed**) похоже на абсолютное, элемент выводится из общего потока, его координаты рассчитываются от краёв окна браузера, но при прокрутке страницы элемент не меняет своего положения.

### Flexbox

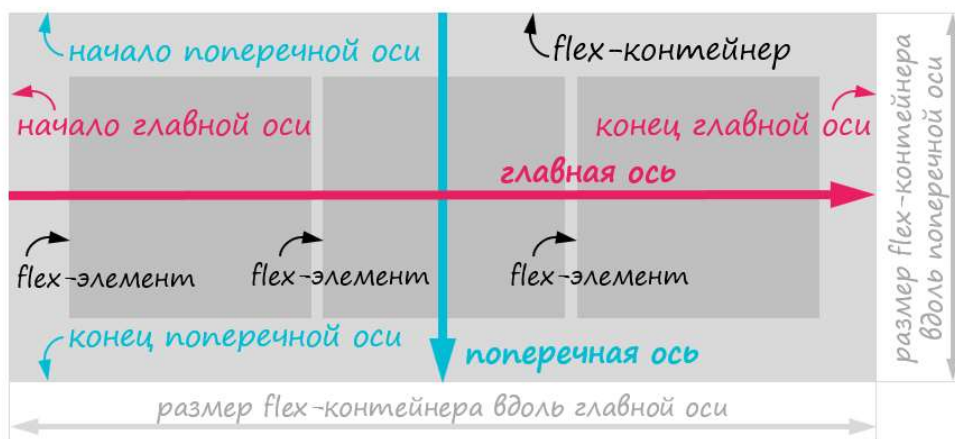
**CSS Flexbox предназначен для создания гибких макетов.** С помощью этой технологии можно очень просто и гибко расставить элементы в контейнере, распределить доступное пространство между ними, и выровнять их тем или иным способом даже если они не имеют конкретных размеров. CSS Flexbox позволяет создать адаптивный дизайн намного проще, чем с использованием Float и позиционирования.

Создание CSS разметки с помощью Flexbox начинается с установки необходимому HTML элементу CSS-свойства **display** со значением flex или **flex-inline**.

После этого данный элемент становится **flex-контейнером**, а все его **непосредственные** дочерние элементы – **flex-элементами**.



На рисунке представлена схема устройства flex-контейнера:



## Свойства родительских элементов (flex-container)

1) **display** - создаёт flex контейнер, инлайновый или блочный, в зависимости от заданного значения.

```
.container {  
  display: flex; /* или inline-flex */  
}
```

2) **flex-direction** - устанавливает направление главной оси и определяет направление flex элементов размещенных в flex контейнере



Значения:

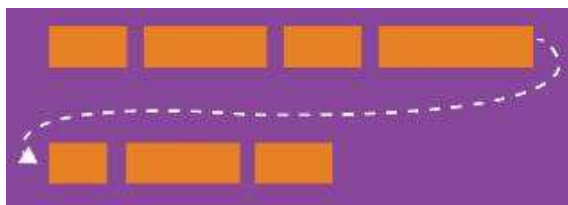
row (стандартное положение) — слева направо.

row-reverse — элементы располагаются справа налево.

column — тоже самое, что и row, только сверху вниз.

column-reverse — тоже самое, что и row-reverse, но снизу вверх.

3) **flex-wrap** - дает возможность дочерним элементам при необходимости переходить на другую строку



Значения:

nowrap — это значение по-дефолту, при котором все flex элементы будут выстраиваться в одну линию.

wrap — flex элементы будут переноситься на несколько строк, от верха к низу.

wrap-reverse — flex элементы будут переноситься на несколько строк снизу вверх.

4) **justify-content** – выравнивание дочерних элементов вдоль главной оси

Значения:

flex-start — дефолтное состояние, при котором элементы расставляются от начала строки.

flex-end — состояние, в котором элементы размещены с конца строки.

center — элементы центрированы вдоль строки.

space-between — элементы равномерно распределены по строке, первый элемент находится в начале строки, последний в конце.

space-around — элементы равномерно распределены по строке с равным местом вокруг них.

space-evenly — элементы распределены таким образом, что свободное пространство между любыми двумя элементами равномерно, как и место до границы края контейнера.

5) **align-items** – выравнивание дочерних элементов вдоль поперечной оси

Значения:

flex-start — всё размещается с начала поперечной оси

flex-end — все элементы размещаются с конца поперечной оси

center — элементы центрируются по поперечной оси

baseline — элементы выравниваются по базовой линии

stretch — это дефолтное состояние, при котором элементы заполняют контейнер, с учетом min-width и max-width.

б) **align-content** - выравнивает и распределяет строки контейнера, когда есть свободное пространство в поперечной оси. Это свойство не приносит эффекта, когда есть только одна строка flex элементов

```
.container {  
  align-content: flex-start | flex-end | center | space-between | space-around | stretch;  
}
```

## ПОРЯДОК ВЫПОЛНЕНИЯ И ФОРМА ОТЧЕТНОСТИ

### Задание 1. Подготовка к работе и создание структуры верхней части сайта

В результате выполнения работы должен получиться промо-сайт мероприятия: «Посади свое дерево».



Перед тем как верстать сайт необходимо внимательно проанализировать его структуру: выделить основные блоки; определить, какими тегами отображается тот или иной элемент сайта.

Подготовьтесь к работе:

- создайте папку PromoSite;
  - в папке PromoSite создайте файлы index.html и style.css, а также папку img;
  - в папку img скопируйте графические изображения из материалов к работе.
2. Изучите структуру верхней части сайта

## Структура верхней части сайта

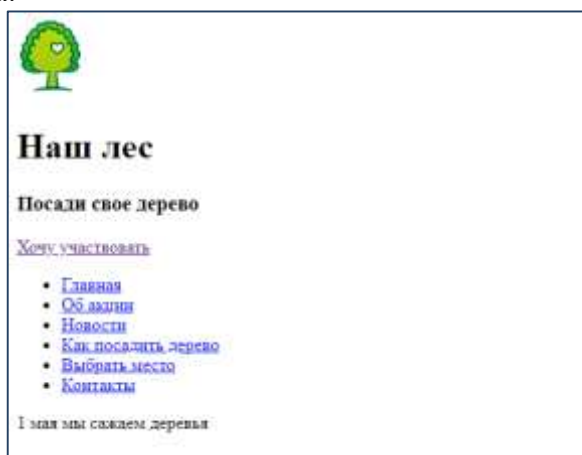


3. В файле index.html запишите теги, чтобы отобразились необходимые элементы на сайте.

### Пояснения:

- Главный блок `div class="main"` должен включать все остальные блоки сайта
- Шапка сайта `div class="header"` содержит два блока: `div class="logo"` и `div class="link"`
- В `div class="logo"` содержится картинка (логотип) и название сайта с девизом
- В `div class="link"` содержится ссылка `Хочу участвовать`
- Меню сайта `div class="nav"` содержит маркированный список, в каждом пункте которого имеется ссылка
- Баннер сайта `div class="banner"` содержит только текст «1 мая мы сажаем деревья»

Т.к. стили у сайта пока не заданы, то внешний вид вашего сайта после данных действий будет отличаться от образца.



4. Далее будем оформлять сайт, используя стили css

- сначала сбросьте для всех элементов внутренние и внешние отступы, задайте свойство `box-sizing: border-box;`



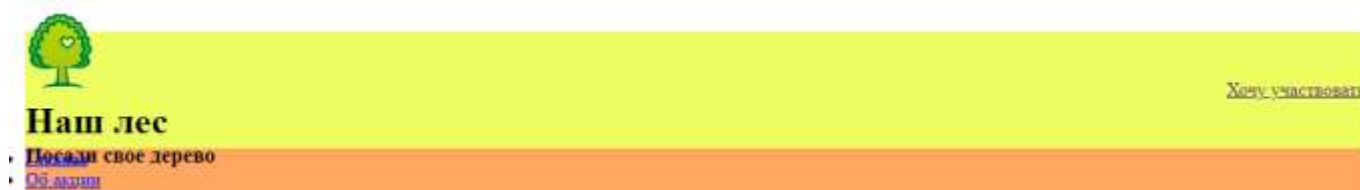
- Далее задайте стили для основных блоков: .main, .header, .nav и .banner согласно следующей таблице:

Блок	Стили	Комментарии
.main	<ul style="list-style-type: none"> <li>• Ширина 80%</li> <li>• Минимальная высота 100%</li> <li>• Заливка #FFFE0</li> <li>• Блок должен располагаться по горизонтали по центру с отступом от верхней границы 30px</li> </ul>	<p>Главный блок будет занимать 80% от тела сайта.</p> <p>Высота блока определяется его содержимым, при добавлении элементов родительский блок растягивается в высоту.</p> <p>Для расположения по центру используйте свойство <b>margin: 30px auto;</b></p>
.header	Ширина 100%, высота 100px	Чтобы данные блоки были видны, задайте им произвольную заливку. В конце работы заливку нужно будет убрать.
.nav	Ширина 100%, высота 40px	
.banner	Ширина 100%, высота 500px	

## Задание 2. Оформление шапки сайта



1. Для оформления шапки будем использовать flex-верстку.
2. Задайте для родительского блока header свойство display: flex
3. Задайте для родительского блока header свойства для выравнивания элементов по основной и поперечной осям:  
justify-content: space-between;  
align-items: center;



4. Задайте для дочерних блоков свойство flex-grow, так чтобы область для логотипа стала в 2 раза больше области для ссылки.

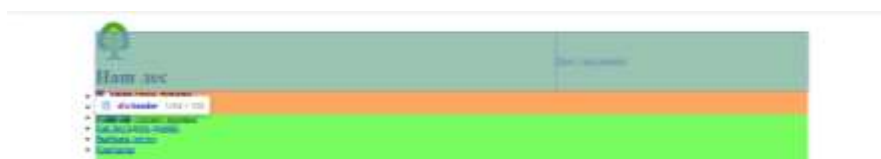
```

.header
  width: 100%;
  height: 100px;
  background: #ffefd6;
  display: flex;
  justify-content: space-between;
  align-items: center;

.logo
  flex-grow: 2;

.link
  flex-grow: 1;

```



### Оформление тегов, входящих в блок logo

- Добавьте для картинки-логотипа обтекание текста слева
- Добавьте для заголовков h1 и h3 цвет, шрифт, размер
- Добавьте необходимые отступы
- *Параметры текста и отступов можете использовать свои*



### Оформление тегов, входящих в блок link

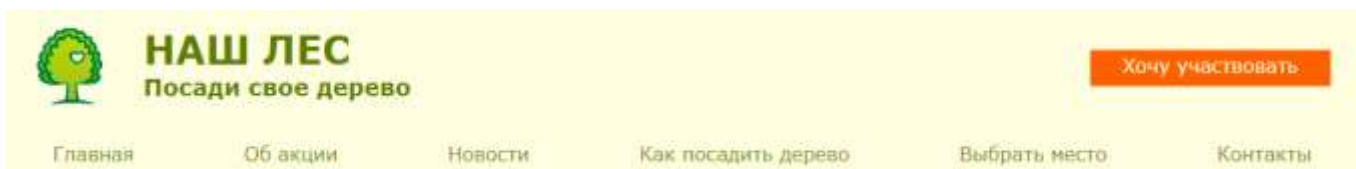
- Добавьте для ссылки свойство display: block
- Задайте для ссылки заливку, рамку, ширину, высоту, параметры шрифта
- Используя технологию flex переместите получившуюся кнопку вправо.
- Добавьте стили при наведении мышкой на ссылку.



Стили при наведении:



### Задание 3. Оформление меню сайта



1. Создайте горизонтальное меню с помощью flex-верстки, пункты меню равномерно распределены по всей строке
2. Добавьте оформление ссылок
3. Добавьте стили для ссылок при наведении

### Задание 4. Оформление баннера



1. Для баннера установите на задний фон картинку
2. Текст выровняйте по горизонтали по центру
3. Задайте параметры шрифта, отступы (в образце использован шрифт Comic Sans MS).

### Общие свойства для сайта

- Для тела сайта добавьте ширину и высоту в 100%, фоновую заливку изображением.
- Стили для тела сайта разместите в начале файла под стилями сброса.

```
# style.css > ...
1  *{
2  |   margin: 0;
3  |   padding: 0;
4  |   box-sizing: border-box;
5  | }
6  body{
7  |   width:100%;
8  |   height:100%;
9  |   background-image: url('img/1.jpg');
10 | }
```

- Удалите заливку для блоков header, nav и banner.
- Проверьте, как верхняя часть сайта адаптируется под различную ширину экрана
- Замените все свойства height на min-height и снова проверьте адаптивность
- Проверьте правильность (валидность) html-страницы по сайте: <https://validator.w3.org/>
- Проверьте правильность css-кода на сайте: <https://jigsaw.w3.org/css-validator/>

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем суть блочной верстки сайта?
2. Опишите свойства float, clear, position и их использование в блочной верстки сайта.

### Лабораторная работа № 6 ВЕРСТКА САЙТА НА FLEXBOX

**ЦЕЛЬ РАБОТЫ:** научиться верстать макет сайта с помощью блоков и технологии FlexBox, использовать CSS для оформления основных компонентов сайта.

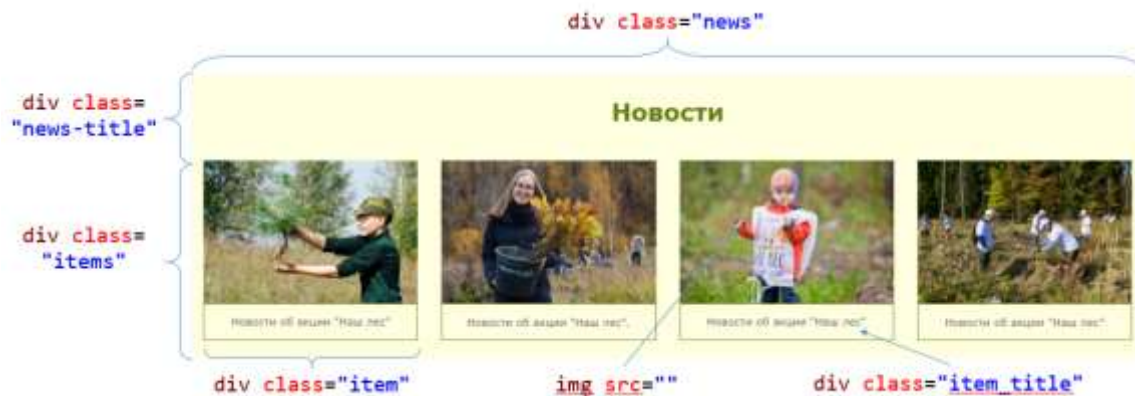
Для выполнения работы необходимо *знать* базовые технологии (HTML и CSS), используемые при написании web-страниц; необходимо *уметь* использовать готовые прикладные компьютерные программы по профилю подготовки, разрабатывать web-страницы на языках HTML и CSS.

**ВРЕМЯ ВЫПОЛНЕНИЯ:** 90 минут.

**ОБОРУДОВАНИЕ:** ПК ИМВ, OS Windows, браузер (Google Chrome, Mozilla, Opera), Notepad++, Sublime Text

### ПОРЯДОК ВЫПОЛНЕНИЯ И ФОРМА ОТЧЕТНОСТИ

#### Задание 1. Оформление блока Новости



1. В файле index.html запишите теги, чтобы отобразились необходимые элементы блока Новости.

**Пояснения:**

- a. Продолжайте писать код внутри главного блока `div class="main"`
- b. Блок новостей `div class="news"` должен включать 2 блока: `div class="news-title"` с текстом Новости и `div class="items"` – блок контейнер для новостей
- c. В блок контейнер `div class="items"` следует добавить 4 блока отдельных новостей `div class="item"`
- d. Внутри каждого блока `div class="item"` содержится картинка и блок `div class="item_title"` с ссылкой на новость

2. Задайте стили для блока Новости согласно следующим подсказкам:

Блок	Стили
.news	<ul style="list-style-type: none"> <li>• Ширина 100%</li> <li>• Минимальная высота 300px</li> <li>• Выравнивание текста по центру</li> </ul>
.news-title	Стили для заголовка Новости задайте самостоятельно

Добавить сверху и снизу одинаковые отступы для блока Новости



Блок	Стили
. items	<ul style="list-style-type: none"> <li>• Ширина 100%</li> <li>• Сделать блок flex-контейнером</li> <li>• Самостоятельно определить и задать тип выравнивания по основной и поперечной осям</li> <li>• Задать перенос блоков</li> </ul>



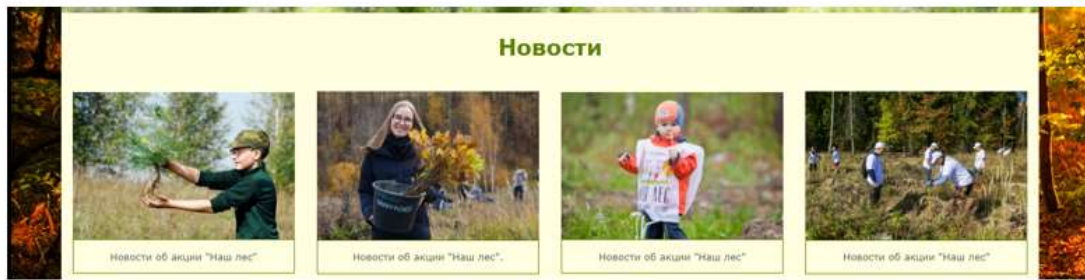


Блок	Стили	Комментарии
.item	display: flex; flex-direction: column;	Каждый блок с новостью также является flex-контейнером, в котором элементы выровнены по вертикальной оси
	margin: 1rem;	Внешние отступы между блоками (заданы в относительных единицах, можно задать в px)
	flex: 1 1 calc(25% - 2rem);	<ul style="list-style-type: none"> <li>• Все блоки с новостями занимают одинаковую ширину (flex-grow = 1, flex-shrink = 1)</li> <li>• Ширина каждого блока рассчитывается по формуле <math>\text{calc}(25\% - 2\text{rem})</math>, где 25% ширина блока от ширины родителя (<math>100\% / \text{на } 4 \text{ блока} = 25\%</math>) – 2rem – вычитаем отступ слева и справа от блоков</li> </ul>
	min-width: 170px;	Устанавливаем минимальную ширину для отдельного блока. Если не задать минимальную ширину, то будет некорректное отображение блоков при уменьшении размеров экрана (проверьте это!)



Блок .item\_title и ссылку внутри него оформите самостоятельно

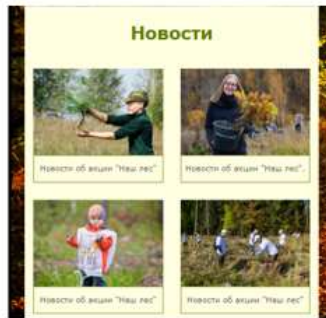
3. Проверьте корректность отображения блока Новости при различной ширине экрана



Широкоформатная версия сайта



Планшет



Мобильная версия

## Задание 2. Оформление блока с контентом.



### 1. Наберите html-код блока по образцу:

```

<div class="content">
  <div class="section">
    <p class="act">Акция "Наш лес. Посади свое дерево"</p>
    <p>Целью акции является восстановление леса. </p>
    <p>Посади свое дерево - это всероссийский проект в рамках которого ежегодно весной и осенью жители России, Украины и Белоруссии высаживают деревья в городах, селах и поселениях, и тем самым, совершенствуют среду обитания!</p>
    <p>1 мая все желающие смогут посадить дерево в любом месте города и района. Участие в акции, саженцы и инвентарь бесплатные. За один день акции вместе мы высадим более 1 млн. деревьев. Подарим лесу жизнь!</p>
    <p>Зарегистрируйтесь на сайте и приезжайте с семьей и друзьями на удобную для вас площадку.</p>
    <a href="#">Принять участие</a>
    <a href="video.html">Смотреть видео</a>
  </div>
  <div class="aside">
    <p class="bold">Как посадить дерево</p>
    <p class="green">ШАГ 1</p>
    <p>Нажимаем на ссылку "Принять участие" или кнопку "Хочу участвовать"</p>
    <p class="green">ШАГ 2</p>
    <p>Выбираем любую площадку в лесу, парке или сквере</p>
    <p class="green">ШАГ 3</p>
    <p>Регистрируемся на участие через e-mail или социальную сеть</p>
  </div>
</div>

```

## 2. Оформление CSS выполните самостоятельно

### Задание 3. Оформление подвала сайта. Выполните самостоятельно



### Лабораторная работа № 7 СОЗДАНИЕ ВЫПАДАЮЩЕГО МЕНЮ. ИСПОЛЬЗОВАНИЕ СВОЙСТВ CSS3

**ЦЕЛЬ РАБОТЫ:** научиться верстать макет сайта с помощью блоков и технологии FlexBox, использовать CSS для оформления основных компонентов сайта.

Для выполнения работы необходимо *знать* базовые технологии (HTML и CSS), используемые при написании web-страниц; необходимо *уметь* использовать готовые прикладные компьютерные программы по профилю подготовки, разрабатывать web-страницы на языках HTML и CSS.

**ВРЕМЯ ВЫПОЛНЕНИЯ:** 90 минут.

**ОБОРУДОВАНИЕ:** ПК IBM, OS Windows, браузер (Google Chrome, Mozilla, Opera), Notepad++, Sublime Text

### КРАТКАЯ ТЕОРИЯ И МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

#### Новые теги HTML 5

##### **А) Семантические теги**

В HTML 5 используются специальные семантические теги, которые ведут себя как тег блока div.

**Семантические теги** – это теги, которые предназначены для того чтобы компьютерные программы (поисковые системы, сборщики информации, речевые браузеры и т.д.), понимали какой тип информации заложен в данных тегах.

*Примеры семантических тегов:*

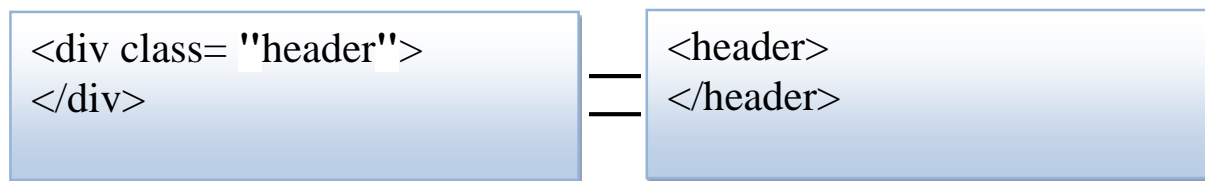
`<header> </header>` - шапка

`<nav> </nav>` - горизонтальное меню

`<aside> </aside>` - левый или правый блок

`<section> </section>` - контент

<footer> </footer> - подвал



Разницы между тегами **<div class="header">** и **<header>** при верстке сайта нет. Однако, например, для поисковых систем, разница будет большой. Поисковые машины или роботы не понимают **<div class="header">**, для них это типовой тег разметки – замени его на **<div class="abrakadabra">** и смысл не поменяется. Другое дело **<header>**, робот, обнаружив этот тег, воспринимает его именно как шапку сайта или раздела.

### Б) Новые атрибуты у тегов форм

<input type = "email"> - ввод электронной почты

<input type = "date"> - календарь

<input type = "source"> - поиск

<input type = "number"> - переключатель

<input type = "range"> - ползунок

### В) Теги для дополнительных технологий и API

<audio> </audio> – воспроизведение аудиофайлов без использования сторонних программ (плагинов, расширений).

<video> </video> – воспроизведение видеофайлов, без использования сторонних программ.

<source /> – указание пути к аудио/видео файлам, находится внутри тегов audio и video.

<canvas> </canvas> – создание специальной области на сайте, в которой можно создавать или добавлять изображения и, с помощью языка программирования JavaScript, манипулировать ими.

## 1. Новые свойства CSS3

### 1. Закругление углов у блока - свойство **border-radius**

border-radius: 2px; /\*закругление для всех углов 2px\*/

border-radius: 2px 5px 5px 2px; /\*разные закругления у углов\*/

border-radius: 5px/10px; /\*закругление со скосом\*/

2. **Градиентная заливка** – использование у свойства **background** или **background-image** значений: **linear-gradient()** или **radial-gradient()**

background: linear-gradient(to right, aqua, yellow); /\*линейный градиент вправо от голубого к желтому\*/

### 3. **Прозрачность элемента** - свойство **opacity**

opacity: 0; /\*полностью прозрачный элемент\*/

opacity: 1; /\*полностью непрозрачный элемент\*/

Удобный эффект, находит массу применений - полупрозрачный фон, изменение прозрачности при наведении и т.п.

### 4. **Тени**

Тени имеют следующие параметры: смещение по горизонтали, по вертикали, размытие и цвет



Свойство **box-shadow** – тени у блоков

Свойство **text-shadow** – тень у текста

```
box-shadow: 10px 5px 5px #000; /*тень смещена по оси X на 10px, по оси Y на 5px, радиус размытия 5px, черного цвета*/
```

## 5. Плавные переходы – свойство **transition**

Свойство **transition** включает 4 параметра:

- **transition-property** — указывает, для какого стиля будет действовать переход;
- **transition-duration** — определяет длительность анимации;
- **transition-timing-function** — скорость хода анимации;
- **transition-delay** — время ожидания перед началом перехода.

```
transition: all 1s ease 0s; /*переход для всех свойств длительностью 1s, начало анимации медленное, к середине скорость повышается и в конце вновь снижается, задержка анимации 0s*/
```

## 6. Трансформация объектов – свойство **transform**

**Перемещение объекта – transform: translate()**

```
transform: translate(10px, 5px); /*объект сдвинется на 10px вправо и на 5px вниз*/
```

Чтобы сдвинуть в противоположную сторону, надо добавить значениям знак "-" (например, -10px)

**Поворот объекта – transform: rotate()**

```
transform: rotate(90deg); /*поворот объекта по часовой стрелке на 90°*/
```

Если поставить со знаком "-", то поворот будет против часовой

**Изменение размера у объекта – transform: scale()**

```
transform: scale(1.5); /*увеличение объекта в полтора раза*/
```

Если поставить "-", то объект уменьшится на указанное значение

**Новые свойства нужно использовать с префиксами (приставками).**

Префиксы для браузеров:

- moz - мазилла
- webkit – вебкитовские браузеры (Chrome, Safari)
- ms – интернет эксплорер
- o – опера

```
-webkit-transform: rotate(90deg);  
-moz-transform: rotate(90deg);  
transform: rotate(90deg);
```

## ПОРЯДОК ВЫПОЛНЕНИЯ И ФОРМА ОТЧЕТНОСТИ

### Задание 1. Проверка лабораторной работы № 5,6

Проверьте HTML и CSS код по образцу (см. файлы HTML и CSS). В своих файлах замените теги

div с классом на соответствующие семантические теги (см. в образцах).

Например, `<div class = "main">` замените на тег `<main>`.

Исправьте свои существенные ошибки, которые привели к неправильному отображению веб-страницы в браузере. Если в Вашей работе все элементы сайта отображаются верно, а разница с образцами только в отдельных элементах оформления, то оставьте свои стили и их значения (например, шрифт, цвет, отступы и т.п.)

Добавьте в блок `<head>` метатеги, определяющие ключевые слова и описание, необходимое для поисковых систем:

```
<meta name="keywords" content="наш лес, дерево, акция, лес">  
<meta name="description" content="Наш лес" >  
<meta name="author" content="admin">
```

Проверьте свои исправленные файлы на валидность по следующим ссылкам:

index.html - [https://validator.w3.org/unicorn/?ucn\\_lang=ru#validate-by-uri+task\\_conformance](https://validator.w3.org/unicorn/?ucn_lang=ru#validate-by-uri+task_conformance)

style.css - [https://jigsaw.w3.org/css-validator/#validate\\_by\\_upload](https://jigsaw.w3.org/css-validator/#validate_by_upload)

## Задание 2. Создание выпадающего меню

1. В верхнее меню в пункт **Об акции** добавьте вложенное меню по образцу:

```
<ul>  
  <li><a href="#">Главная</a></li>  
  <li><a href="#">Об акции</a>  
    <ul class="submenu">  
      <li><a href="">История акции</a></li>  
      <li><a href="">Условия акции</a></li>  
      <li><a href="">Этапы акции</a></li>  
    </ul>  
  </li>  
  <li><a href="#">Как посадить дерево</a></li>  
  <li><a href="#">Выбрать место</a></li>  
  <li><a href="#">Запись</a></li>  
  <li><a href="#">Контакты</a></li>  
</ul>
```

2. В файл css добавьте стили для вложенного меню:

```

.submenu {
  position: absolute;
  display: none; /*вложенное меню не видно*/
  padding: 0;
  width: 200px;
  left: 110px;
  top: 10px;
  z-index: 10;
}
.submenu li a{
  display: block;
  line-height: 40px;
}
nav ul li:hover .submenu{
  display: block; /*при наведении вложенное меню появляется*/
  background: #ffffe0;
}

```

3. Добавьте для меню новые свойства CSS3, используйте при этом префиксы.

### Задание 3. Добавление видео на сайт

1. Создайте вторую страницу сайта (можно скопировать главную страницу и переименовать ее), удалите содержимое из блока <section> (также нужно удалить правый блок).
2. Вставьте в контент теги для добавления видео:

```

<video controls>
  <source src=" Как посадить своё дерево.mp4" type="video/mp4">
</video>

```

Здесь *controls* – появление управляющих кнопок под видео, *type* – определяет настройки для выбранного формата.

Формат *mp4* не понимает Opera, лучше использовать формат *ogv*

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для чего служат префиксы?
2. Как добавить видео на web-страницу?

## Лабораторная работа № 8 ВЕРСТКА АДАПТИВНОГО САЙТА

**ЦЕЛЬ РАБОТЫ:** научиться верстать адаптивный сайт

Для выполнения работы необходимо **знать** принципы и структуру устройства сети Интернет; необходимо **уметь** использовать готовые прикладные компьютерные программы по профилю подготовки.

**ВРЕМЯ ВЫПОЛНЕНИЯ:** 90 минут.

**ОБОРУДОВАНИЕ:** ПК IMB, OS Windows, браузер (Google Chrome, Mozilla, Opera), Notepad++, Sublime Text

### КРАТКАЯ ТЕОРИЯ И МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

**Адаптивная верстка** – html-верстка веб-страниц, при которой внешний вид, расположение и структура ее элементов зависят от типа и свойств устройства на котором отображается страница.

## Как сделать адаптивный дизайн сайта из фиксированного макета?

1. Необходимо добавить мета-тег **viewport**. Данный мета-тег прописывается в блоке <head> сайта.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, minimum-scale=1.0" />
```

2. Ширину основного контейнера необходимо задать с помощью свойства **max-width**.

3. Необходимо перевести все статические единицы измерения в относительные единицы измерения: **px необходимо перевести в %, а шрифты задать в em**.

**Перевод ширины** контейнера или элемента из px в % осуществляется по формуле:

Размер контейнера (px) / размер основного контейнера (родителя) в (px) \* 100% = результат (%).

*Пример*

Размер основного контейнера 960px, в нем имеется контейнер 720px. Для него получим размер в % следующим образом:  $720/960*100=75\%$ .

**Перевод шрифта из px в em** осуществляется по формуле:

Размер шрифта (px) / 16px (стандартный размер) = размер шрифта (em)

*Пример*

Размер шрифта 32px, тогда  $32/16=2em$ .

4. Необходимо использовать **медиа-запросы**.

Медиа-запросы включают в себя медиа-тип (принтеры, смартфоны, экраны, телевизоры, проекторы и др.) и условия, которое может принимать в свою очередь истину или ложь (true, false). В зависимости от того верный ли медиа-тип и выполняется ли условие будут применяться различные стили css. Если условие верно, то будут применяться те стили, которые прописаны в этом медиа-запросе, если же будет ложным, то будут применяться обычные стили css.

Благодаря таким запросам и создаются различные отображения сайта для мобильных, планшетов и экранов мониторов. Медиа-запросы поддерживаются всеми современными браузерами.

**Медиа-запрос записывается следующим образом:**

```
@media screen and (max-width: 1000px) {  
  .class {  
    свойство: значение;  
  }  
}
```

Здесь:

**@media** – медиа-запрос;

**screen** – медиа-тип (также называют тип носителя);

**max-width: 1000px** – условие, которое должно выполняться (в данном примере стили будут применяться, если ширина окна меньше ширины 1000px);

## ПОРЯДОК ВЫПОЛНЕНИЯ И ФОРМА ОТЧЕТНОСТИ

В результате выполнения работы должна получиться часть адаптивного сайта, включающая шапку, блок с видео и галереей.

Текстовое и графическое содержимое сайта, а также цвета, параметры шрифтов, отступы CSS можете менять по своему усмотрению.





1. Подготовить необходимые файлы и папки для работы.
2. Подготовить начальный код в файлах index.html (структура страницы, подключение стилей) и style.css (сброс стилей).
3. Создать блок с шапкой сайта, используя соответствующий семантический тег. Залить блок градиентной заливкой.



4. Добавить в блок текстовый тег с названием сайта и кнопку.



### Шапка сайта

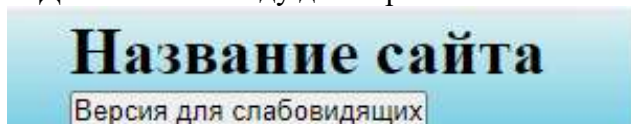
5. Добавить 3 необходимые команды в файл style.css, чтобы элементы расположились как на рисунке:



6. Добавить отступы слева и справа:



7. Добавить команду для переноса элементов в мобильной версии сайта:



6. Добавить оформление кнопки:



7. Добавить стили при наведении

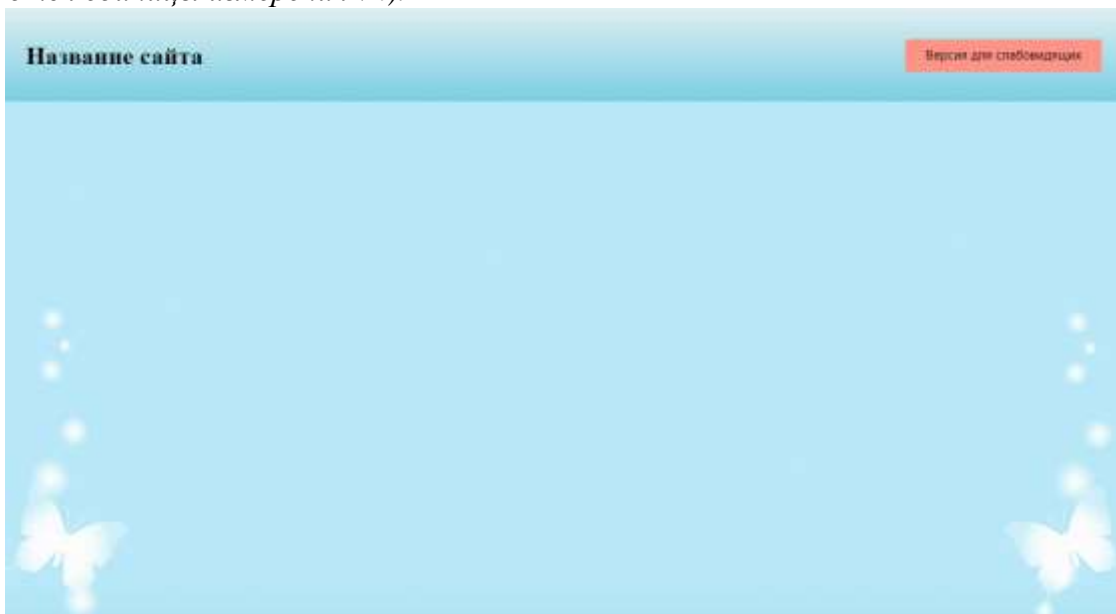


8. Добавьте плавный переход при наведении мыши: свойство transition (свойство необходимо добавить и в стили кнопки, и в стили при наведении)

## Блок с видео

1. Создать следующий блок с произвольным классом (вместо тега `div` используйте тег `section`). Залить блок фоновой картинкой.

Высота шапки сайта и блока с видео должно в сумме равняться высоте окна браузера (для этого используются единицы измерения `vh`).



2. Добавьте в созданный блок теги для вставки видео.



2. Разметите видео по центру блока.
3. Добавьте тень к видео

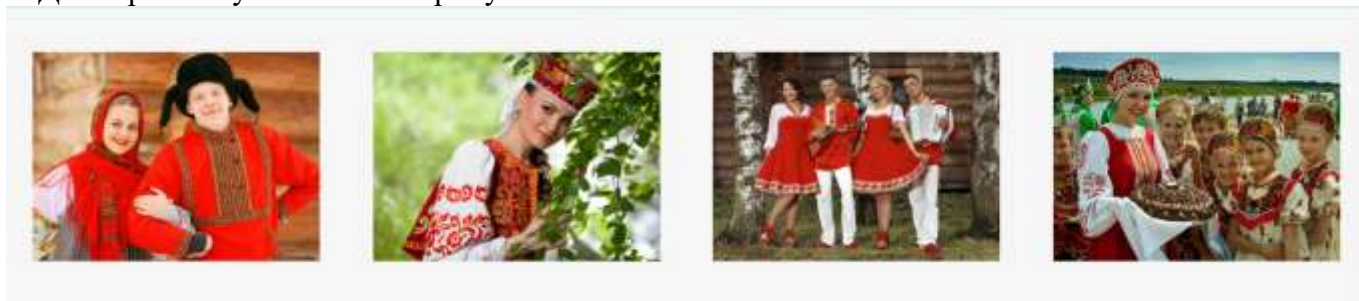


## Галерея сайта

1. Создать следующую секцию (блок) с произвольным классом. Добавить в нее внутренние блоки, в каждый из которых поместите тег добавления картинки.



2. Добавить в родительский блок необходимые свойства flex-верстки.
3. Для внутренних блоков установите максимальную ширину в 200-300px
4. Для картинок установить ширину в 100%





4. Добавьте команду, чтобы при уменьшении размеров экрана блоки переходили на новую строку.



5. Для внутренних блоков добавьте увеличение при наведении мыши. Сделайте увеличение плавным.



Лабораторная работа № 9  
**ВЕРСТКА АДАПТИВНОГО САЙТА. ПУБЛИКАЦИЯ САЙТА В СЕТИ ИНТЕРНЕТ**

**ЦЕЛЬ РАБОТЫ:** научиться переносить сайта на хостинг.

Для выполнения работы необходимо *знать* принципы и структуру устройства сети Интернет; необходимо *уметь* использовать готовые прикладные компьютерные программы по профилю подготовки.

**ВРЕМЯ ВЫПОЛНЕНИЯ:** 90 минут.

**ОБОРУДОВАНИЕ:** ПК ИМВ, OS Windows, браузер (Google Chrome, Mozilla, Opera), Notepad++, Sublime Text

### **КРАТКАЯ ТЕОРИЯ И МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ**

**Хостинг** — услуга по предоставлению места для физического размещения информации на сервере, постоянно подключенном к интернету.

Если необходимо разместить сайт – надо найти для нее место (как для книги на полке), в интернете такие места предоставляют специальные службы – **хостеры**. Они предоставят место для сайта на своем сервере – машине, на которой установлены специальные программы, и которая постоянно подключена к сети. Собственно, интернет и состоит из множества таких машин объединенных между собой.

**Существует два способа размещения сайта в интернете:** бесплатный и платный.

Бесплатный хостинг позволяет бесплатно выбрать доменное имя третьего уровня для своего сайта и выложить сайт в интернет. Бесплатный хостинг предполагает размещение на сайте рекламы со стороны хостинг-провайдера, которая «немного» искажает дизайн вашего сайта, и возможности у бесплатного варианта ограничены. Вы никогда не будете иметь большую посещаемость ресурса, расположенного на бесплатном хостинге, отсюда следует и невозможность заработать на таком сайте, если, конечно, вы ставите перед собой такую цель.

Доменное имя сайта — это имя, которое вы указываете в поисковой строке браузера, например, `www.mylsite.ru` или `mysite.ru`. Доменным именем первого уровня в данной записи является окончание `.ru`, второй уровень — это предыдущее название web-сайта: `mysite`. Доменное имя третьего уровня, которое предлагает бесплатный хостинг может выглядеть, к примеру, так: `photo.mylsite.ru` или `mypage.mylsite.ru`. Третьим уровнем доменного имени в данном случае является имя, которое предшествует `mysite.ru`

Платный хостинг дает возможность выбора доменного имени первого и второго уровня. Платный хостинг — это, когда вы платите деньги за место для своего web-сайта в сети интернет, но при этом являетесь полноправным хозяином своего ресурса, то есть никто не станет завешивать ваш сайт рекламой, пренебрегая дизайном.

### **ПОРЯДОК ВЫПОЛНЕНИЯ И ФОРМА ОТЧЕТНОСТИ**

#### **Задание 1.**

##### **Оформление контента сайта**

- Продолжайте работу с сайтом, созданным в лабораторной работе 8.
- Добавьте ниже еще одну секцию, задайте ей произвольный класс.
- Внутри данной секции поместите еще 2 блока.
- Первый внутренний блок должен содержать произвольный текст (заголовок и несколько абзацев).
- Второй внутренний блок должен содержать два поля ввода и кнопку Отправить.
- Добавьте стили, обязательные для блока.

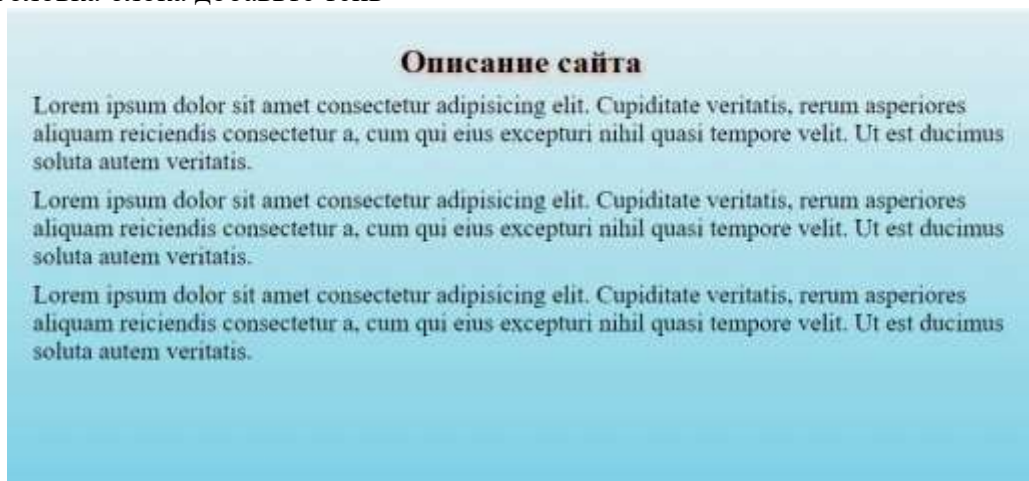




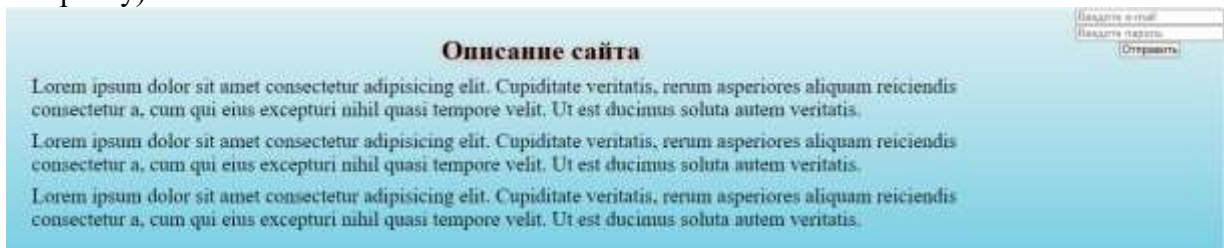
- В файле style.css добавьте одну команду, чтобы блоки разместились как на рисунке:



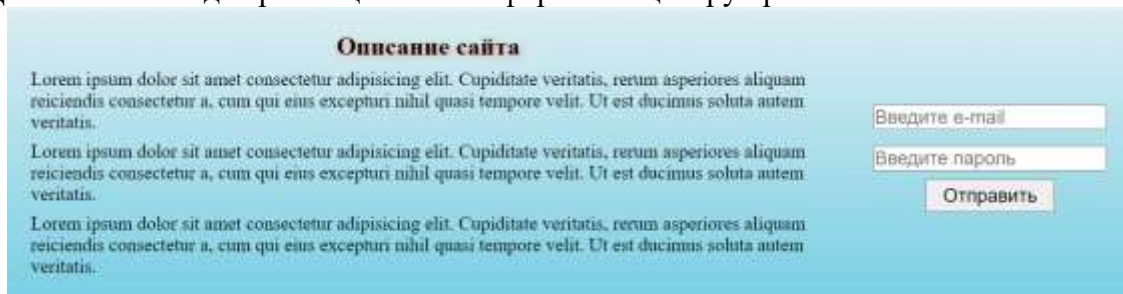
- Добавьте стили для оформления текста:
- У заголовка блока добавьте тень



- Добавьте стили для того, чтобы элементы формы расположились в столбик (используя flex-верстку)



- Добавьте стили для оформления элементов формы.
- Добавьте стили для размещения всей формы по центру правого блока



- Добавьте медиа-запросы для адаптивной верстки.

В примере `.content` – это класс всей секции, `.content_form` – для правого блока (вам нужно поправить на свои названия классов)



```
@media (max-width:900px){
  .content{
    flex-wrap: wrap;
  }
  .content_form{
    width: 100%;
  }
}
```



### Оформление меню сайта

- Добавьте в нужное место семантический блок для меню.
- Поместите в данный блок маркированный список со ссылками меню



- Залейте меню градиентной заливкой



Добавьте стили, чтобы ссылки меню расположились как на рисунке:



Добавьте оформление для ссылок и для ссылок при наведении.

### Ссылки-якоря

- Для каждого добавьте произвольное id
- У ссылок в значении адреса укажите соответствующие якоря (id)

Например, для блока Галерея задано id="gallery"

```
<section class="gallery" id="gallery">
```

В меню указывается соответствующий якорь:

```
<li><a href="#gallery">Галерея</a></li>
```

- Для создания статичного меню в стили меню добавьте следующие строки:

```
position: sticky;  
left: 0;  
top: 0;  
z-index: 1;
```

### Адаптивность

- Замените все теги height на min-height
- Добавьте медиа-запросы для уменьшения размера видео при ширине экрана менее 900px; при ширине экрана менее 720px

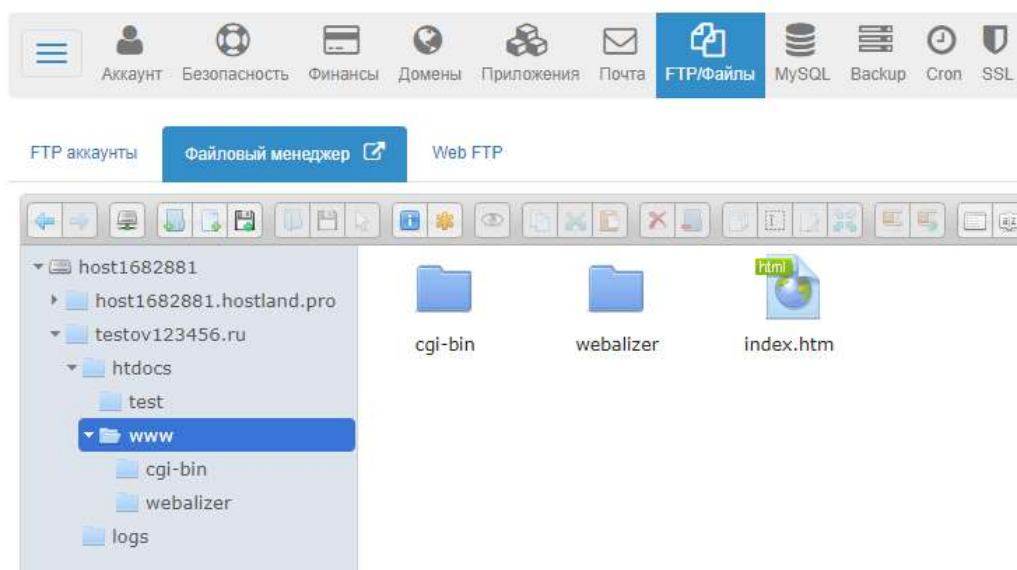
### Задание 2. Публикация сайта в сети Интернет

1. В браузере откройте сайт хостинг провайдера по адресу <https://www.hostland.ru/>
2. Нажмите на кнопку **Попробовать** для бесплатного хостинга в течении пробного периода (30 дней).
3. Выберите один из тарифов («Бриз», «Шторм», «Ураган»), заполните поля e-mail и код, поставьте галочку согласия с условиями договора и нажмите кнопку **Зарегистрировать**. На указанный адрес электронной почты придет письмо с данными для использования сервисов Hostland.ru
4. Нажмите кнопку **Вход** и введите в поля авторизации **логин и пароль Входа на панель управления**, полученные в письме.
5. Перейдите в меню **Домены**, нажмите на кнопку **Прикрепить домен**. В появившемся окне в полное Домен введите произвольное имя своего домена (например, testov123456.ru). Тип прикрепления оставьте **Как домен с собственной папкой**. Нажмите кнопку **Сохранить изменения**.
6. Если введенный вами домен не занят, то появится сообщение **Заявка на обработку доменного имени принята**. Дождитесь окончания данного действия (15-20 минут). Если домен занят, то измените имя домена.



Перейдите по ссылке Вашего домена и посмотрите, как выглядит тестовая страница.

7. Перейдите по ссылке с техническим именем, при этом должна открыться стартовая страница.
8. Перейдите в меню **FTP/файлы – Файловый менеджер** и далее откройте папку www в структуре сайта с вашим доменным именем.



В этой папке имеются:

- папка cgi-bin – служебная папка для хранения серверных скриптов
- папка webalizer – статистика сайта
- файл index.htm, содержащий тестовую страницу. Щелкните правой кнопкой по данному файлу и выберите команду **Редактировать – ACE Editor**. Просмотрите содержимое файла index.html и убедитесь, что он содержит информацию с тестовой страницы, которую Вы открывали в предыдущем пункте.

9. Удалите файл index.html и обновите содержимое тестовой страницы (должно появиться сообщение о запрете доступа к серверу).
10. Поместите на хостинг сайт, созданный в предыдущей лабораторной работе. Для этого просто перетащите содержимое своей папки в папку www на хостинге.
11. Обновите содержимое тестовой страницы. Если все выполнено правильно, то Ваш сайт должен отобразиться в Интернете.

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое хостинг?
2. Что такое доменное имя сайта?

## Лабораторная работа №10 Зачетная работа

**ЦЕЛЬ РАБОТЫ:** обобщить, систематизировать и проверить знания и умения по дисциплине.

Для выполнения работы необходимо *знать* базовые технологии (HTML и CSS), используемые при написании web-страниц; необходимо *уметь* использовать готовые прикладные компьютерные программы по профилю подготовки, разрабатывать web-страницы на языках HTML и CSS.

**ВРЕМЯ ВЫПОЛНЕНИЯ:** 90 минут.

**ОБОРУДОВАНИЕ:** ПК ИМВ, OS Windows, браузер (Google Chrome, Mozilla, Opera), Notepad++, Sublime Text

### **ПОРЯДОК ВЫПОЛНЕНИЯ И ФОРМА ОТЧЕТНОСТИ**

Каждому студенту на зачетном занятии выдается индивидуальный вариант лабораторной работы. Зачетная лабораторная работа содержит 3 задания:

- 1 задание – верстка структуры сайта или его отдельных блоков (шапка, меню) согласно схеме;
- 2 задание – заполнение отдельных блоков сайта базовыми элементами (текст, изображение, навигация, форма, логотип и т.д.) согласно схеме;
- 3 задание – добавление новых элементов на сайт согласно схеме, использование свойств CSS3.

Студент должен продемонстрировать знание основных тегов HTML и стилей CSS, понимание блочной верстки сайта с помощью технологии flex, умение разработки web-страницы на языках HTML и CSS.

## Литература

### Основная литература:

*Полуэктова Н. Р.* Разработка веб-приложений : учебное пособие для среднего профессионального образования / Н. Р. Полуэктова. — Москва : Издательство Юрайт, 2022. — 204 с. (электронно-библиотечная система <https://urait.ru/>)

### Интернет-ресурсы:

1. Учебники по HTML и CSS [Электронный ресурс]. Форма доступа: <http://htmlbook.name>.
2. <MyRusakov.ru/>. Уроки и статьи по созданию сайтов [Электронный ресурс]. Форма доступа: <http://MyRusakov.ru>.
3. Ruseller.com. Частная коллекция качественных материалов для тех, кто делает сайты [Электронный ресурс]. Форма доступа: <http://ruseller.com/>