

Лабораторная работа № 1

Тема работы: «Создание и форматирование веб-страницы. Внедрение ссылок и графических объектов в веб-документ»

1. Цель работы

Формирование умений создания простейших веб-страниц, использования различных тегов для форматирования текста в веб-документе, создания ссылок, в том числе с графическими элементами привязки, внедрение графических объектов в веб-документ.

2. Задание

Реализовать внедрение на HTML-страницу текста, ссылок и изображений.

3. Оснащение работы

ПК, редактор исходного кода, текстовый редактор, браузер.

4. Основные теоретические сведения

Каркас HTML-документа имеет следующую структуру:

```
<!DOCTYPE HTML>
<html>
<head>
СОДЕРЖАНИЕ ЗАГОЛОВКА
</head>
<body>
СОДЕРЖАНИЕ ТЕЛА ДОКУМЕНТА
</body>
</html>
```

Заголовок включает в себя несколько специализированных тегов, основными из которых являются `<title> ... </title>` и `<meta> ... </meta>`.

Тег `<title>` содержит заголовок документа, который будет выводиться в строке заголовка окна браузера.

Метатеги `<meta>` содержат специальную информацию, такую как тип кодировки, а также используются для хранения информации, предназначенной для браузеров и поисковых систем. Например, механизмы поисковых систем обращаются к метатегам для получения описания сайта, ключевых слов и других данных.

```
<meta name="description" content="Описание содержимого страницы">
<meta name="keywords" content="Ключевые слова через запятую">
<meta name="robots" content="index, follow"> // Индексация и переход
// по ссылкам разрешены
```

`<meta http-equiv="refresh" content="30">` // Страница будет перезагружена через 30 секунд

Атрибуты тега `<meta>` представлены в таблице 1.1.

Таблица 1.1 – Атрибуты тега `<meta>`

Атрибут	Описание, принимаемое значение
Charset	Указывает кодировку символов для текущего HTML-документа: <code><meta charset="UTF-8"></code>
Content	Содержит произвольный текст, который определяет значение, ассоциируемое с атрибутом <code>http-equiv</code> или <code>name</code> , в зависимости от их значения.
http-equiv	<p>Контролирует действия браузера на данной веб-странице (эквивалент HTTP заголовков). При отображении страницы браузер будет следовать инструкциям, заданным в атрибуте:</p> <p>default-style указывает предпочтительный стиль для использования на странице;</p> <p>content должен содержать идентификатор элемента <code><link></code>, который ссылается на таблицу стилей CSS, или идентификатор элемента <code><style></code>, содержащего таблицу стилей;</p> <p>refresh указывает время в секундах до перезагрузки страницы или время до перенаправления на другую страницу, если в атрибуте content после указания времени идет строка <code>"url=адрес_страницы"</code>.</p> <p>Автоматическая перезагрузка страницы через заданный промежуток времени, в данном примере, через 30 секунд: <code><meta http-equiv="refresh" content="30"></code></p> <p>Если необходимо сразу перебросить посетителя на другую страницу, то можно указать URL-адрес в параметре <code>url</code>: <code><meta http-equiv="refresh" content="0; url=http://mail.ru/"></code></p>
Name	<p>Ассоциируется со значением, содержащемся в атрибуте content. Не должен использоваться в случае, если для элемента уже заданы атрибуты <code>http-equiv</code>, <code>charset</code> или <code>itemprop</code>.</p> <p>application-name указывает название веб-приложения, используемого на странице;</p> <p>author указывает имя автора документа в свободном формате;</p> <p>description определяет краткое описание к содержимому страницы: <code><meta name="description" content="Описание содержимого страницы"></code></p> <p>generator указывает один из пакетов программного обеспечения, используемого для создания документа, например: <code><meta name="generator" content="WordPress 4.0"></code></p> <p>keywords содержит список ключевых слов, разделенных запятыми,</p>

	<p>соответствующих содержимому страницы, например:</p> <pre><meta name="keywords" content="Ключевые слова через запятую"></pre> <p>Также атрибут name может принимать следующие значения из расширенной спецификации, такие как creator, googlebot, publisher, robots, slurp, viewport, хотя ни одно из них еще не было официально принято.</p>
--	--

Элемент `<body>` – тело документа – определяет видимую часть документа и предназначен для хранения содержания веб-страницы (контента), отображаемого в окне браузера. Информацию, которую следует выводить в документе, следует располагать именно внутри контейнера `<body>`. К такой информации относится текст, изображения, теги, скрипты JavaScript и т.д.

Теги форматирования могут быть вложенными друг в друга. При этом нужно внимательно следить, чтобы один контейнер находился целиком в другом контейнере.

HTML теги (имя тега, описание, значение свойства display) представлены в таблице 1.2.

Таблица 1.2 – HTML теги

Текст		
<code><h1></h1></code> - <code><h6></h6></code>	заголовки шести уровней	block
<code><p></p></code>	параграфы в тексте	block
<code>
</code>	перенос текста на новую строку	none
<code><hr></code>	горизонтальная линия	block
<code><wbr></code>	возможное место разрыва длинной строки	none
<code><blockquote></blockquote></code>	большая цитата	block
<code><cite></cite></code>	источник цитирования	inline
<code><q></q></code>	краткая цитата	inline
<code><code></code></code>	фрагмент программного кода	inline
<code><kbd></kbd></code>	текст, вводимый пользователем с клавиатуры	inline
<code><pre></pre></code>	выводит текста с пробелами и переносами	block
<code><samp></samp></code>	результат выполнения сценария	inline
<code><var></var></code>	выделяет переменные из программ	inline
<code></code>	перечёркивает текст, помечая как удаленный	inline
<code><s></s></code>	перечёркивает неактуальный текст	inline
<code><dfn></dfn></code>	выделяет термин курсивом	inline
<code></code>	выделяет важные фрагменты текста курсивом	inline
<code><i></i></code>	выделяет текст курсивом без акцента	inline
<code></code>	выделяет полужирным важный текст	inline
<code></code>	задает полужирное начертание отрывка текста,	inline

	без дополнительного акцента	
<ins></ins>	подчёркивает изменения в тексте	inline
<u></u>	выделяет отрывок текста подчёркиванием, без дополнительного акцента	inline
<mark></mark>	выделяет фрагменты текста желтым фоном	inline
<small></small>	отображает текст шрифтом меньшего размера	inline
	подстрочное написание символов	inline
	надстрочное написание символов	inline
<time></time>	дата / время документа или статьи	inline
<abbr></abbr>	аббревиатура или акроним	none
<address></address>	контактные данные автора документа или статьи	block
<bdi></bdi>	изолирует текст, читаемый справа налево	inline
<bdo></bdo>	задаёт направление написания текста	inline
<ruby></ruby>	контейнер для Восточно-Азиатских символов и их расшифровки	inline
<rp></rp>	тег для скобок вокруг символов	none
<rt></rt>	расшифровка символов	block

Все атрибуты (vlink, text, link, bgcolor, background, alink), ранее используемые с элементом <body>, считаются устаревшими и запрещены к использованию в HTML5, поэтому для определения цвета заднего фона, цвета текста или ссылок, а так же для добавления фонового изображения и т.д. рекомендуется использовать CSS.

Обязательных параметров у тега <body> нет.

Для элемента <body> доступны глобальные атрибуты.

Глобальные атрибуты, приведенные в таблице 1.3., могут быть использованы для любого HTML-элемента, хотя некоторые из них могут не оказывать на элементы никакого влияния.

Таблица 1.3 – Глобальные атрибуты

Атрибут	Описание, принимаемое значение
Accesskey	Генерирует сочетания клавиш для доступа к текущему элементу. Состоит из разделенного пробелами списка символов. Браузер в первую очередь выбирает те клавиши, которые существуют на раскладке клавиатуры. Применяется к следующим элементам: <a>, <area>, <button>, <input>, <label>, <legend>, <textarea>. Принимаемые значения: перечень названий клавиш.
Class	Определяет имя класса для элемента (используется для определения класса в таблице стилей). Принимаемые значения: имя класса.
Contenteditable	Определяет, может ли пользователь редактировать содержимое

	(контент). Позволяет преобразовать любое поле HTML в редактируемый элемент. Принимаемые значения: true/false.
Contextmenu	Добавляет к элементу контекстное меню, заданное тегом <menu>. Принимаемые значения: значение атрибута id элемента <menu>.
Dir	Определяет направление текста контента в элементах <bdo> и <bdi>. Принимаемые значения: ltr/rtl/auto.
Draggable	Определяет, может ли пользователь перетащить элемент. Принимаемые значения: true/false/auto.
Dropzone	Определяет область для приема перемещаемых элементов, сообщая браузеру пользователя, какие действия совершить при перемещении. Принимаемые значения: copy — содержимое перемещаемого элемента будет скопировано в область. move — содержимое перемещаемого элемента будет перемещено в новую область. link — при перемещении будет создана ссылка на первоначальные данные элемента.
Hidden	Указывает на то, что элемент должен быть скрыт. Принимаемые значения: hidden.
Id	Определяет уникальный идентификатор элемента. Принимаемые значения: id — идентификатор элемента.
Lang	Определяет код языка содержимого (контента) в элементе. Принимаемые значения: код языка.
Spellcheck	Указывает, подлежит ли содержимое элемента проверке орфографии и грамматики. Принимаемые значения: true/false.
Style	Указывает на код CSS, применяемую для оформления элемента. Принимаемые значения: код CSS.
TabIndex	Определяет порядок перехода к элементу при помощи клавиши TAB. Принимаемые значения: порядковый номер.
Title	Определяет дополнительную информацию об элементе, задавая всплывающую подсказку для страницы. Принимаемые значения: текст.
Translate	Разрешает или запрещает перевод текста внутри элемента. Принимаемые значения: yes/no.

Семантический HTML5 определяет точные теги для пояснения четкой роли контента на странице. Эта дополнительная информация помогает роботам/индексаторам, таким как Google и Bing лучше понять какой контент важен, какой является второстепенным, какой используется для навигации и так далее.

Элемент **<header>** группирует вводные и навигационные элементы, не является обязательным. Может содержать заголовки, оборачивать содержание раздела страницы, форму поиска или логотип. В HTML-документе может содержаться одновременно несколько элементов **<header>** и они могут располагаться в любой части страницы. Элемент **<header>** нельзя помещать внутрь элементов **<footer>**, **<address>** или другого элемента **<header>**.

Элемент **<nav>** предназначен для создания блока навигации веб-страницы или всего веб-сайта, при этом не обязательно должен находиться внутри **<header>**. На странице может быть несколько элементов **<nav>**. Не заменяет теги **** или ****, он просто их обрамляет. Не все группы ссылок на странице должны быть обернуты **<nav>**, этот элемент предназначен в первую очередь для разделов, которые состоят из главных навигационных блоков.

```
<header>
  <h1>Site description</h1>
  <nav>
    <ul>
      <li><a href="">About</a>
      <li><a href="">Forum</a>
      <li><a href="">Download</a>
    </ul>
  </nav>
</header>
```

В качестве элементов панели навигации можно использовать не только элементы списков:

```
<nav>
  <p><a href="">...</a></p>
  <p><a href="">...</a></p>
</nav>
```

Также можно добавлять заголовки внутрь элемента:

```
<nav>
  <h2>...</h2>
  <ul>
    <li><a>...</a></li>
    <li><a>...</a></li>
    <li><a>...</a></li>
  </ul>
</nav>
```

Элемент **<article>** используется для группировки записей — публикаций, статей, записей блога, комментариев. Представляет собой независимый обособленный блок, предназначенный для многократного использования, как правило, начинается с заголовка. Может дублироваться на других страницах сайта и содержать внутри другие элементы **<article>**, которые по содержанию имеют близкое отношение к содержанию внешней статьи. Если на странице присутствует только одна статья с заголовком и текстовым содержанием, она не нуждается в обёртке элементом **<article>**. Элемент рекомендуется использовать только в том случае, если содержание элемента будет явно указано в схеме документа.

```
<article>
  <header>
    <h2>...</h2>
  </header>
```

```

<p>...</p>
<footer>
Опубликовано в категории<a href="">Музыка</a>.
  <a href="">0 комментариев</a>
</footer>
</article>

```

Элемент **<section>** представляет собой универсальный раздел документа. Группирует тематическое содержимое и обычно содержит заголовок. Не является блоком-оберткой, для этих целей уместнее использовать элемент **<div>**. В качестве содержимого может выступать оглавление, разделы научных публикаций, программа мероприятия. Домашняя страница сайта также может быть поделена на секции – блок вводной информации, новости и контакты. Элемент рекомендуется использовать только в том случае, если содержимое элемента будет явно указано в схеме документа. Например, можно сделать на одной странице секцию статей, секцию комментариев, секцию галереи и т.д.

```

<article>
  <h1>...</h1>
  <section>
    <h2>...</h2>
    <p>...</p>
  </section>
  <section>
    <h2>...</h2>
    <p>...</p>
  </section>
  <p>...</p>
</article>

```

Можно создавать родительские элементы **<section>** с вложенными элементами **<article>**, в которых есть один или несколько элементов **<article>**. Не все страницы должны быть устроены именно так, но это допустимый способ вложения элементов. Например, основная область контента страницы содержит два блока со статьями разной тематики. Можно сделать на этом акцент, поместив каждую статью одной тематики внутрь элемента **<section>**.

```

<section>
  <h1>Заметки о природе</h1>
  <article>
    <h2>...</h2>
    <p>...</p>
  </article>
  <article>
    <h2>...</h2>
    <p>...</p>
  </article>
</section>
<section>
  <h1>Исторические заметки</h1>
  <article>
    <h2>...</h2>
    <p>...</p>
  </article>
  <article>
    <h2>...</h2>
    <p>...</p>
  </article>
</section>

```

Элемент **<aside>** группирует содержимое, связанное с окружающим его контентом напрямую, но которое можно счесть отдельным (*т.е., удаление этого блока не повлияет на понимание основного содержимого*). Чаще всего элемент позиционируется как боковая колонка (как в книгах) и включает в себя группу элементов: **<nav>**, цифровые данные, цитаты, рекламные блоки, архивные записи. Не подходит для блоков, просто позиционированных в стороне.

```
<aside>
  <h2>...</h2>
  <p>...</p>
</aside>
<aside>
  <h2>...</h2>
  <p>...</p>
  <blockquote>
    <p>...<cite>...</cite>...</p>
    <p>...</p>
  </blockquote>
</aside>
```

Элемент **<footer>** представляет собой нижний колонтитул содержащей его секции или корневого элемента. Обычно содержит информацию об авторе статьи, данные о копирайте и т.д. Если используется как колонтитул всей страницы, содержимое дополняется сведениями об авторских правах, ссылками на условия использования, контактную информацию, ссылками на связанное содержимое и т.п.

В одном веб-документе может быть несколько элементов **<footer>**. Как каждая страница, так и каждая статья может иметь свой элемент **<footer>**, более того, **<footer>** можно поместить в элемент **<blockquote>**, чтобы указать источник цитирования.

```
<footer>
  <address>...</address>
  <small>©2014...</small>
</footer>
```

Элемент **<address>** используется для определения контактной информации автора/владельца документа или статьи. Для обозначения автора документа тег размещают внутри элемента **<body>**, для отображения автора статьи – внутри тега **<article>**. В браузере обычно отображается курсивом.

Элемент **<main>** представляет основное содержимое документа (содержимое элемента **<body>**). Контент, находящийся внутри элемента, должен быть уникальным и не повторяться во всех документах сайта, таких как навигационные ссылки, информация о копирайте, логотипы, формы поиска (в случае, если форма поиска является основной функцией документа).

Элемент **<main>** не может быть потомком таких элементов как **<article>**, **<aside>**, **<footer>**, **<header>** или **<nav>**.

```
<body>
<header>
  <h1>Пудель</h1>
  <nav>
    <ul>
      <li><a href="index.html">Главная</a></li>
      <li><a href="about.html">О породе</a></li>
```



```

        <li><a href="health.html">Здоровье</a></li>
    </ul>
</nav>
</header>
<main>
    <section>
        <header>
            <h2>О породе</h2>
            <nav>
                <ul>
                    <li><a href="#basic">Разновидности</a></li>
                    <li><a href="#app">Внешний вид</a></li>
                    <li><a href="#temp">Характер</a></li>
                </ul>
            </nav>
        </header>
        <section id="basic">
            <h3>Разновидности</h3>
            <p>...</p>
        </section>
        <section id="app">
            <h3>Внешний вид</h3>
            <p>...</p>
        </section>
        <section id="temp">
            <h3>Характер</h3>
            <p>...</p>
        </section>
        <footer>
            <a href="#basic">Разновидности</a>
            <a href="#app">Внешний вид</a>
            <a href="#temp">Характер</a>
        </footer>
    </section>
</main>
<footer>
    <small>Copyright      ©      <time      datetime="2016">2016</time>      Моя
собака.py</small>
</footer>
</body>

```

Элемент **<figure>** представляет автономный контент (необязательно с заголовком), являющийся самостоятельным элементом основного потока. Элемент может быть перемещен из основного содержимого документа в боковую колонку или приложение, не затрагивая поток документа. С помощью элемента **<figure>** можно добавлять краткие характеристики к иллюстрациям, фотографиям, диаграммам, фрагментам кода и т.д.

```

<figure>
    
    <figcaption>Осенний лес</figcaption>
</figure>

```

Элемент **<figcaption>** - потомок элемента **<figure>**, не принадлежит ни к одной категории контента. Элемент является блочным, по ширине равен ширине элемента **<figure>**, высота по умолчанию равна 18px.

Элемент **<time>** определяет время (24 часа) или дату по григорианскому календарю с возможным указанием времени и смещения часового пояса. Текст, заключенный в данный тег, не имеет стилового оформления браузером. Для тега до-

ступен атрибут **datetime**, в качестве содержимого которого указывается то, что будет видеть пользователь на экране своего компьютера:

```
<time datetime="2014-09-25"> 25 Сентября 2014</time>
```

С появлением данного тега, стало возможным задавать дату в стандартных временных форматах.

Чтобы дата могла считываться автоматически, она должна быть в формате YYYY-MM-DD. Время, которое также может указываться, задается в формате HH:MM с добавлением разделяющего префикса **T (time)**:

```
<time datetime="2014-09-25T12:00"> 25 Сентября 2014</time>
```

Элемент **<mark>**. Текст, помещенный внутрь тега **<mark>**, выделяется по умолчанию желтым цветом (цвет фона и цвет шрифта в выделенном блоке можно изменить, задав определенные css-стили) , как маркером на бумаге. С помощью данного тега можно отмечать важное содержимое, а также ключевые слова.

Элемент **<bdi>** отделяет фрагмент текста, который должен быть изолирован от остального текста для двунаправленного форматирования текста. Используется для текстов, написанных одновременно на языках, читающихся слева направо и справа налево.

Элемент **<wbr>** - одиночный тег, показывает браузеру место, где можно добавить разрыв длинной строки в случае необходимости.

Для создания ссылки необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку. Оба действия выполняются с помощью тега **<a>**, который имеет единственный обязательный атрибут **href**. В качестве значения используется адрес документа (URL). Общий синтаксис создания ссылок следующий:

```
<a href="http://site.ru">указатель ссылки</a>
```

Адрес ссылки может быть абсолютным и относительным.

Абсолютный путь указывает точное местоположение файла в пределах всей структуры папок на компьютере (сервере).

Абсолютный путь к файлу даёт доступ к файлу со сторонних ресурсов и содержит следующие компоненты:

- 1) протокол, например, http (опционально);
- 2) домен (доменное имя или IP-адрес компьютера);
- 3) папка (имя папки, указывающей путь к файлу);
- 4) файл (имя файла).

Относительные ссылки, как следует из их названия, построены относительно текущего документа или адреса и, соответственно, ведут отсчет от корня сайта или текущего документа.

Относительный путь содержит следующие компоненты:

- 1) папка (имя папки, указывающей путь к файлу);

2) файл (имя файла).

Путь для относительных ссылок имеет три специальных обозначения:

/ указывает на корневую директорию и говорит о том, что нужно начать путь от корневого каталога документов и идти вниз до следующей папки;

./ указывает на текущую папку;

../ подняться на одну папку (директорию) выше.

Якорем (внутренней ссылкой) называется закладка с уникальным именем на определенном месте веб-страницы, предназначенная для создания перехода к ней по ссылке. Якоря удобно применять в документах большого объема, чтобы можно было быстро переходить к нужному разделу.

Внутренние ссылки также создаются при помощи тега <a> с разницей в том, что атрибут href содержит имя указателя — так называемый якорь, а не URL-адрес. Перед именем указателя всегда ставится знак #.

Следующая разметка создаст оглавление с быстрыми переходами на соответствующие разделы:

```
<h1>Времена года</h1>
<h2>Оглавление</h2>
<a href="#p1">Лето</a> <!--создаём якорь, указав #id элемента-->
<a href="#p2">Осень</a>
<a href="#p3">Зима</a>
<a href="#p4">Весна</a>
<p id="p1">...</p> <!--добавляем соответствующий id элементу-->
<p id="p2">...</p>
<p id="p3">...</p>
<p id="p4">...</p>
```

Ссылку можно также сделать на закладку, находящуюся в другой веб-странице и даже другом сайте. Для этого в атрибуте href тега <a> надо указать адрес документа и в конце добавить символ решетки # и имя закладки.

Элемент <a> поддерживает глобальные атрибуты и собственные, таблица 1.4.

Таблица 1.4. – Атрибуты тега <a>

Атрибут	Значения	Описание
download	"пустой" назва- ние файла	Указывает браузеру, что материал по ссылке необходимо скачать. Если указано название файла, в диалоговом окне загрузки будет предложено сохранить файл именно под этим названием вместо названия файла с сайта.
Href	URI адрес	Определяет адрес документа, на который ведет ссылка.
Hreflang	language_cod	Указывает на язык документа по ссылке.

	<i>e</i>	
Media	<i>media_query</i>	Указывает на устройство, для которого оптимизирован документ по ссылке.
Rel	alternate author bookmark external help license next nofollow noreferrer noopener prev search tag	Определяет отношение между текущим документом и документом по ссылке. Указывает чем является документ по ссылке для текущего.
Target	_blank _parent _self _top <i>имя фрейма</i>	Определяет где будет открыт ссылаемый документ: _blank - в новой вкладке или окне; _parent - в родительском окне; _self - в текущем окне (используется по умолчанию); _top - на весь экран.
Type	<i>media_type</i>	Указывает на тип ссылаемого документа.

Для добавления к web–странице файла с изображением используется тег ``, с обязательным атрибутом `src`, который будет указывать на абсолютный или относительный путь к изображению.

`<p><center></center></p>`

Атрибуты тега ``, приведены в таблице 1.5.

Таблица 1.5. – Атрибуты тега ``

Атрибут	Описание, принимаемое значение
Alt	Атрибут <code>alt</code> добавляет альтернативный текст для изображения. Выводится на месте появления изображения до его загрузки или при отключенной графике, а также выводится всплывающей подсказкой при наведении курсора мыши на изображение. Синтаксис: <code>alt="описание изображения"</code> .
crossorigin	Атрибут <code>crossorigin</code> позволяет загружать изображения с ресурсов другого домена с помощью CORS-запросов. Изображения, загруженные в холст с помощью CORS-запросов, могут быть использованы повторно. Допускаемые значения: <code>Anonymous</code> — Cross-origin запрос выполняется с помощью HTTP-заголовка, при этом учетные данные не передаются. Если сервер не даёт

	<p>учетные данные серверу, с которого запрашивается контент, то изображение будет испорчено и его использование будет ограничено.</p> <p><code>use-credentials</code> — Cross-origin запрос выполняется с передачей учетных данных.</p> <p>Синтаксис: <code>crossorigin="anonymous"</code>.</p>
Height	<p>Атрибут <code>height</code> задает высоту изображения. Может указываться в <code>px</code> или <code>%</code>.</p> <p>Синтаксис: <code>height: 300px</code>.</p>
Ismap	<p>Атрибут <code>ismap</code> указывает на то, что картинка является частью изображения-карты, расположенного на сервере (изображение-карта — изображение с интерактивными областями). При нажатии на изображение-карту координаты передаются на сервер в виде строки запроса URL-адреса. Атрибут <code>ismap</code> допускается только в случае, если элемент <code></code> является потомком элемента <code><a></code> с действительным атрибутом <code>href</code>.</p> <p>Синтаксис: <code>ismap</code>.</p>
longdesc	<p>URL расширенного описания изображения, дополняющее атрибут <code>alt</code>.</p> <p>Синтаксис: <code>longdesc="http://www.example.com/description.txt"</code>.</p>
Src	<p>Атрибут <code>src</code> задает путь к изображению.</p> <p>Синтаксис: <code>src="flower.jpg"</code>.</p>
Sizes	<p>Задаёт размер изображения в зависимости от параметров отображения. Работает только при заданном атрибуте <code>srcset</code>. Значением атрибута является одна или несколько строк, указанных через запятую.</p>
Srcset	<p>Создаёт список источников для изображения, которые будут выбраны, исходя из разрешения экрана. Может использоваться вместе или вместо атрибута <code>src</code>. Значением атрибута является одна или несколько строк, разделённых запятой.</p> <pre></pre>
Usemap	<p>Атрибут <code>usemap</code> определяет изображение в качестве карты-изображения. Значение обязательно должно начинаться с символа <code>#</code>. Значение ассоциируется со значением атрибута <code>name</code> или <code>id</code> тега <code><map></code> и создает связь</p>

	<p>между элементами <code></code> и <code><map></code>. Атрибут нельзя использовать, если элемент <code><map></code> является потомком элемента <code><a></code> или <code><button></code>.</p> <p>Синтаксис: <code>usemap="#мymap"</code>.</p>
Width	<p>Атрибут <code>width</code> задает ширину изображения. Может указываться в px или %.</p> <p>Синтаксис: <code>width: 100%</code>.</p>

```
<a href="/site/picture.jpg"></a>
```

Рисунки можно использовать в качестве ссылок.

```
<a href="/site/picture.jpg"></a>
```

Чтобы пометить или подписать фото на странице используется тег `<figure>`. Этот тег указывает на то, что внутри него помещается контент типа иллюстраций, фотографий, диаграмм и т.п. Тег `<figcaption>` (заголовок рисунка) позволяет добавить подпись к изображению.

```
<figure>
  
  <figcaption>Вид на море
</figcaption>
</figure>
```

Элемент `<picture>` представляет собой контейнер для хранения нескольких элементов `<source>`, которые поддерживают элемент ``. Это позволяет указывать разные изображения с учётом размера экрана, плотности пикселей, формата изображения и других параметров.

```
<picture>
  <source srcset="smaller_landscape.jpg" media="(max-width: 40em) and
(orientation: landscape)">
  <source srcset="smaller_portrait.jpg" media="(max-width: 40em) and
(orientation: portrait)">
  <source srcset="default_landscape.jpg" media="(min-width: 40em) and
(orientation: landscape)">
  <source srcset="default_portrait.jpg" media="(min-width: 40em) and
(orientation: portrait)">
  <img srcset="default_landscape.jpg" alt="My default image">
</picture>
```

Внутри <picture> содержится ноль или несколько элементов <source>, которые идут перед одним элементом .

5. Порядок выполнения работы

1. Создайте HTML-страницу, реализовав структуру с использованием семантических элементов HTML5, через 10 секунд после открытия которой произойдет перенаправление на сайт учебного заведения, в котором вы обучаетесь.

2. Создайте вторую HTML-страницу. Добавьте на страницу информацию о Ваших персональных данных, например: ФИО, учебное заведение, курс, группа, пол, возраст и т.д. Оформите введенную информацию с помощью тегов для форматирования текста.

3. Добавьте ссылку, которая будет открывать в новом окне сайт <https://html5book.ru/>.

4. Добавьте на страницу картинку, сделайте картинку ссылкой на первую страницу, которая будет открываться в новой вкладке. В случае, если картинка не загрузится, на ней должно быть написано: «Не удалось загрузить картинку». При наведении на картинку должна выводиться подсказка о том, что это ссылка на страницу с персональными данными.

5. Добавьте на страницу расписание движения городского транспорта по дням недели. Оформите расписание с помощью тегов форматирования текста. Вверху страницы разместите дни недели, при клике на которые должен осуществляться переход к выбранному дню недели с расписанием. Реализовать возможность пользователю заменить названия выходных дней.

6. Добавьте на HTML-страницу несколько изображений, которые будут иметь общую подпись.

7. Добавьте на HTML-страницу изображение, которое будет изменять размер и ориентацию, в зависимости от размера окна браузера.

6. Форма отчета о работе

Лабораторная работа № ____

Номер учебной группы _____

Фамилия, инициалы учащегося _____

Дата выполнения работы _____

Тема работы: _____

Цель работы: _____

Оснащение работы: _____

Результат выполнения работы: _____

7. Контрольные вопросы и задания

1. Опишите структуру HTML-документа.

2. Перечислите семантические элементы HTML5.
3. С помощью какого тега создается ссылка? Как создать ссылку на адрес электронной почты?
4. Дайте характеристику тегов, используемых для внедрения графического контента.

8. Рекомендуемая литература

1. **Гарднер, Л.** Разработка веб-сайтов для мобильных устройств / Л. Гарднер, Д. Григсби. – СПб.: Питер, 2013.
2. **Дронов, В.** HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / В. Дронов. – СПб.: БХВ-Петербург, 2014.
3. **Никсон, Р.** Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. 4-е изд. – СПб.: Питер, 2018.