МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

ГЛАВНОЕ УПРАВЛЕНИЕ ПО ОБРАЗОВАНИЮ

МОГИЛЕВСКОГО ОБЛАСТНОГО ИСПОЛНИТЕЛЬНОГО КОМИТЕТА

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«МОГИЛЕВСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

|  |  |
| --- | --- |
| Специальность | 2– 40 01 01 |
| Учебная группа | ПО-455 |
|  |  |

Учебная дисциплина Программные средства

создания интернет-

приложений

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**СОЗДАНИЕ И ФОРМАТИРОВАНИЕ ТЕКСТА НА HTML-СТРАНИЦЕ. СЕМАНТИЧЕСКАЯ РАЗМЕТКА**

Выполнил Савич А.О.

Проверил Денисовец Д. А.

2022

**1 Цель работы**

1.1 Научиться организовывать системы ссылок сайта, создавать списки и таблицы на HTML-странице, внедрять аудио- и видеоконтент в HTML-документ

**2 Индивидуальное задание**

**Вариант 8**

1 Преобразуйте домашнюю страницу, разработанную в лабораторной работе №1 в простой домашний сайт, связав отдельные части с помощью гиперссылок.

Обязательные страницы:

об авторе;

мой колледж (о МГПК);

моя группа;

моя будущая профессия;

мои увлечения или хобби;

любая другая информация;

фоновая картинка и свой цвет фона, на тот случай, если картинки не отображаются;

отступы от краев станицы не менее 9 пикселей;

возможность прокрутки страницы с фиксацией фонового рисунка.

<!DOCTYPE html>

<html>

<head>

<title>Заголовок</title>

<meta name="hs" content="bla"></meta>

<style type="text/css">

h1{

font-size:30px;

color:purple;

}

body{

background-image: url(bg.jpg); /\* Путь к фоновому изображению \*/

background-color: skyblue; /\* Цвет фона синее небо\*/

font-size: 10px;

font-family: Arial;

font-weight: bold;

color:green

}

}

</style>

</head>

<body>

<P align="center" >

<h1>Главная</h1>

<a href="lab5.2.html">"О авторе"</a>

<a href="lab5.3.html">"МГПК"</a>

<a href="lab5.6.html">группа</a>

<a href="lab5.4.html">проф</a>

<a href="lab5.5.html">хобии</a>

<a href=

"mailto:example@foo.com?subject=СОЗДАНИЕ И ФОРМАТИРОВАНИЕ ТЕКСТА НА HTML-СТРАНИЦЕ. СЕМАНТИЧЕСКАЯ РАЗМЕТКА">Написать текст письма</a>

</P>

<a href=

"mailto:example@foo.com?subject=СОЗДАНИЕ И ФОРМАТИРОВАНИЕ ТЕКСТА НА HTML-СТРАНИЦЕ. СЕМАНТИЧЕСКАЯ РАЗМЕТКА">Загрузить архив</a>

</P>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>Заголовок</title>

<meta name="hs" content="bla"></meta>

<style type="text/css">

h1{

font-size:30px;

color:purple;

}

body{

background-image: url(bg.jpg); /\* Путь к фоновому изображению \*/

background-color: skyblue; /\* Цвет фона синее небо\*/

font-size: 6px;

font-family: Arial;

font-weight: bold;

color:green

}

}

</style>

</head>

<body>

<P align="center" >

<h1>О авторе</h1>

<h1 align="center" >савич андрей олегович

по -455 </h1>

<p align="center">короткая информация</p>

<p align="center" > базовый уровень знакомства

</p> <P align="center">в далеком будущем

</P>

</P>

<p><a href="lab5.1.html">НАЗАД </a></p>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>Заголовок</title>

<meta name="hs" content="bla"></meta>

<style type="text/css">

h1{

font-size:30px;

color:purple;

}

body{

background-image: url(bg.jpg); /\* Путь к фоновому изображению \*/

background-color: skyblue; /\* Цвет фона синее небо\*/

font-size: 6px;

font-family: Arial;

font-weight: bold;

color:green

}

}

</style>

</head>

<body>

<P align="center" >

<h1>О КОЛЛЕДЖЕ</h1>

<p>А собственно не лучше ли дать сайту колледжа все это описать- <a href="http://politex.mogilev.by">сайт колледжа</a></p>

<p><a href="lab5.1.html">НАЗАД </a></p>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>Заголовок</title>

<meta name="hs" content="bla"></meta>

<style type="text/css">

h1{

font-size:30px;

color:purple;

}

body{

background-image: url(bg.jpg); /\* Путь к фоновому изображению \*/

background-color: skyblue; /\* Цвет фона синее небо\*/

font-size: 6px;

font-family: Arial;

font-weight: bold;

color:green

}

}

</style>

</head>

<body>

<P align="center" >

<h1>проф</h1>

<p>Будушая профессия - техник программист</a></p>

<p><a href="lab5.1.html">НАЗАД </a></p>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>Заголовок</title>

<meta name="hs" content="bla"></meta>

<style type="text/css">

h1{

font-size:30px;

color:purple;

}

body{

background-image: url(bg.jpg); /\* Путь к фоновому изображению \*/

background-color: skyblue; /\* Цвет фона синее небо\*/

font-size: 6px;

font-family: Arial;

font-weight: bold;

color:green

}

}

</style>

</head>

<body>

<P align="center" >

<h1>хобби</h1>

<p>какие к черту хобби кроме программирования и компов у меня могут быть</p>

<p><a href="lab5.1.html">НАЗАД </a></p>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>Заголовок</title>

<meta name="hs" content="bla"></meta>

<style type="text/css">

h1{

font-size:30px;

color:purple;

}

body{

background-image: url(bg.jpg); /\* Путь к фоновому изображению \*/

background-color: skyblue; /\* Цвет фона синее небо\*/

font-size: 6px;

font-family: Arial;

font-weight: bold;

color:green

}

}

</style>

</head>

<body>

<P align="center" >

<h1>проф</h1>

<p>группа обычная как и все другие</a></p>

<p><a href="lab5.1.html">НАЗАД </a></p>

</body>

</html>

2 Используйте следующие виды гиперссылок:

ссылки на другие HTML – страницы (с переходом на новую страницу и без перехода);

ссылки на конец и начало страницы, переходы по параграфам документа;

ссылку, позволяющую написать письмо, причем в качестве темы – тема практической работы. При наведении на ссылку появляется текст «Написать текст письма»;

ссылку, позволяющую загрузить архив практической работы, при наведении на ссылку появляется текст «Загрузить архив».

**Задание 2:**

**Вариант 8**

1 Используя текстовый редактор Sublime Text 3, создайте список следующего вида:

III Зима

● Декабрь

● Январь

● Февраль

IV Весна

iii Март

iv Апрель

v Май

<!DOCTYPE html>

<html>

<head>

<title>Заголовок</title>

</head>

<body>

<ol>

<li>зима</li>

<ul><li>Декабрь</li><li>Январь </li><li>Февраль</li></ul>

<li>Весна</li>

<ul><li>Март </li><li>Апрель</li><li>Май</li></ul>

</ol>>

</body>

</html>

2 Используя текстовый редактор Sublime Text 3, создайте таблицу, содержащую ячейки различных цветов.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  |  |  | | |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |
|  | |  |  |  |  | |
|  |  |  |  |  |  |  |
|  | | |  |  |  | | |

<!DOCTYPE html>

<html>

<head>

<title>Заголовок</title>

<script type="text/javascript">

function colorize() {

var

r = ('0'+(Math.random()\*255|0).toString(16)).slice(-2),

g = ('0'+(Math.random()\*255|0).toString(16)).slice(-2),

b = ('0'+(Math.random()\*255|0).toString(16)).slice(-2);

return '#' +r+g+b;

}

function change(){

var x = document.getElementById("1");

x.style.background = colorize();

x.style.color = colorize();

}

</script>

</head>

<body>

<table border="2" cellspacing="0" onmousedown="change()"id="1" >

<TR>

<TH width="30" height="30" colspan="3" class="col1"> </TH>

<TH width="30" height="30" class="col6"> </TH>

<TH width="30" height="30" class="col4"> </TH>

<TH width="30" height="30" colspan="3" class="col12"> </TH>

</TR>

<TR>

<TH width="30" height="30" class="col9"> </TH>

<TH width="30" height="30" class="col5"> </TH>

<TH width="30" height="30" rowspan="2" class="col10"> </TH>

<TH width="30" height="30" rowspan="2" class="col12"> </TH>

<TH width="30" height="30" class="col7"> </TH>

<TH width="30" height="30" rowspan="2" class="col8"> </TH>

<TH width="30" height="30" class="col5"> </TH>

<TH width="30" height="30" class="col3"> </TH>

</TR>

<TR>

<TH width="30" height="30" rowspan="2" class="col8"> </TH>

<TH width="30" height="30" class="col1"> </TH>

<TH width="30" height="30" class="col6"> </TH>

<TH width="30" height="30" class="col1"> </TH>

<TH width="30" height="30" class="col9"> </TH>

</TR>

<TR>

<TH width="30" height="30" colspan="2" class="col3"> </TH>

<TH width="30" height="30" class="col11"> </TH>

<TH width="30" height="30" rowspan="2" class="col3"> </TH>

<TH width="30" height="30" class="col9"> </TH>

<TH width="30" height="30" colspan="2" class="col12"> </TH>

</TR>

<TR>

<TH width="30" height="30" class="col1"> </TH>

<TH width="30" height="30" class="col3"> </TH>

<TH width="30" height="30" class="col5"> </TH>

<TH width="30" height="30" class="col12"> </TH>

<TH width="30" height="30" class="col9"> </TH>

<TH width="30" height="30" class="col2"> </TH>

<TH width="30" height="30" class="col11"> </TH>

</TR>

<TR>

<TH width="30" height="30" colspan="3" class="col7"> </TH>

<TH width="30" height="30" class="col2"> </TH>

<TH width="30" height="30" class="col4"> </TH>

<TH width="30" height="30" colspan="3" class="col8"> </TH>

</TR>

</table>

</body>

</html>

3 Контрольные вопросы

1 Объясните, как создать гиперссылку на другую страницу?

**HTML-ссылки** создаются с помощью элементов <a>, <area> и <link>. Ссылки представляют собой связь между двумя ресурсами, одним из которых является текущий документ.

Ссылки можно поделить на две категории:

**ссылки на внешние ресурсы** — создаются с помощью элемента <link> и используются для расширения возможностей текущего документа при обработке браузером;

**гиперссылки** — ссылки на другие ресурсы, которые пользователь может посетить или загрузить.

Как сделать гиперссылки на сайте

**СОДЕРЖАНИЕ:**

1. Структура ссылки

2. Абсолютный и относительный путь

3. Якорь

4. Как сделать изображение-ссылку

5. Как сделать ссылку на телефонный номер, скайп или адрес электронной почты

6. Атрибуты ссылок

1. Структура ссылки

Гиперссылки создаются с помощью элемента <a></a>. Внутрь помещается текст, который будет отображаться на веб-странице. Текст ссылки отображается в браузере с подчёркиванием, цвет шрифта — синий, при наведении на ссылку курсор мыши меняет вид.

[https://favicon.yandex.net/favicon/htmlacademy.ru?size=32&stub=2](https://yandex.ru/an/count/WWOejI_zO2i1vGy051ro_89ZGChxdWK0Am8nOkTNOm00000uqh_10Uo1pPlkyV_NeW600Vlmb0M80PFtzwkA0P01bAIfjzg0W802c06CfActMhW1mewcbIJ00GBO0QxddQG1u07KlOO7w07C0VW1vewinQmFy0AAWRhBa1k00uIaYlS4Y0FErQYb2zKk-0JNXqU81U2XF905ruT7e0NwrXMe1Rk84h05kuWIk0MxY1B01Q6PkmJ81O-SaG7G1VsS1w06kWAe1he2k0RwpWF91i_TkfV0CdaJgGSEqA0eltsFLxW7j0RW1uR2W8262E08rgBM1-W91u0A0GRe39i6u0s2W821W82029WEpxlhlOdOpBmJo130i9A20Q4HdvcPcPcPoOWI0P0I0QWJ_CBEY_Igqv0NW1I0e0BG5C2L-MtW507e58G2e1QmfQgQ1iaMy3-O5uxqmmAe5mcu5m705xNM0Q0PkWAm6RWP____0VWPnvMa784Q__y_pB15Pxom6lxBh9VvswN1dm6270qqI44wGarlOKz8I6Gtg1u1i1y1o1_NyuvIgI1vj22nvh_1Fv0Ypguea2BIhYYG8jIkA90YrQuea2Bno2YG8lB8A90Y-DOea2BvrYZL8l__V_yB045auE1G1bsAeODfBlhtZX7XpdazJiDi0DKfe5fYf9FT8asErJQ11m00~1?stat-id=3&test-tag=119297011669521&banner-sizes=eyI3MjA1NzYwNjE3NTUzMzMyMiI6IjU5M3gzMDAifQ%3D%3D&format-type=118&actual-format=10&pcodever=670937&banner-test-tags=eyI3MjA1NzYwNjE3NTUzMzMyMiI6IjcxMjc1MyJ9&order-banners-options=eyI3MjA1NzYwNjE3NTUzMzMyMiI6MjA0OH0&width=593&height=300)

htmlacademy.ru

РЕКЛАМА

**Вы станете JavaScript-разработчиком**

Выучите JavaScript и развивайтесь в любом направлении разработки. Оставьте заявку!

|Как проходит обучение на курсе

|Наставники

|Программа

|Стоимость

Узнать больше

Обязательным параметром элемента <a> является атрибут href, который задает URl-адрес веб-страницы.

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

HTML

Ссылка состоит из двух частей — **указателя** и **адресной части**. **Указатель ссылки** представляет собой фрагмент текста или изображение, видимые для пользователя. **Адресная часть** ссылки пользователю не видна, она представляет собой адрес ресурса, к которому необходимо перейти.

Адресная часть ссылки состоит из URl. **URl** (Uniform Resource Locator) — унифицированный адрес ресурса. При создании адресов для разделения слов между собой рекомендуется использовать дефис, а не символ подчёркивания. В общем виде URl имеющий следующий формат:

метод доступа://имя сервера:порт/путь

HTML

**Метод доступа**, или протокол, осуществляет обмен данными между рабочими станциями в разных сетях. Наиболее распространенные протоколы передачи данных:

file обеспечивает чтение файла с локального диска:

file:/gallery/pictures/summer.html

HTML

http предоставляет доступ к веб-странице по протоколу HTTP:

http://site.ru/

HTML

https — специальная реализация протокола HTTP, использующая шифрование (как правило, SSL или TLS):

https://site.ru/

HTML

ftp осуществляет запрос к FTP-серверу на получение файла:

ftp://pgu/directory/library

HTML

mailto запускает сеанс почтовой связи с указанным адресатом и хостом:

mailto: nika@gmail.com

HTML

**Имя сервера** описывает полное имя машины в сети, например, site.ru. Если имя сервера не указано, то ссылка считается локальной, т.е. она относится к той же машине, на которой находится HTML-документ, содержащий ссылку.

**Номер порта ТСР**, на котором функционирует веб-сервер. Представляет собой число, которое необходимо указывать, если метод требует номер порта (отдельные сервера могут иметь свой отличительный номер порта). Если порт не указан, по умолчанию используется порт 80. Стандартными портами являются:

21 — FTP

23 — Telnet

70 — Gopher

80 — HTTP

**Путь** содержит имя папки, в которой находится файл.

2. Абсолютный и относительный путь

Когда в ссылке указывается только имя файла, браузер предполагает, что файл находится в той же папке, что и документ, содержащий гиперссылку. На практике веб-сайты содержат сотни документов, которые размещают в отдельные папки, чтобы ими было легче управлять. Чтобы создать ссылку на файл, находящийся вне папки, содержащей текущий документ, необходимо указать расположение файла или путь. HTML поддерживает два вида пути: абсолютный и относительный.

РИС. 1. ПРИМЕР СТРУКТУРЫ ПАПОК

2.1. Абсолютный путь

**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:

протокол, например, http (опционально);

домен (доменное имя или IP-адрес компьютера);

папка (имя папки, указывающей путь к файлу);

файл (имя файла).

Существует два вида записи абсолютного пути — с указанием протокола (полный) и без него:

http://site.ru/pages/tips/tips1.html

//site.ru/pages/tips/tips1.html

HTML

Когда вы ссылаетесь на страницу на другом сайте, вы можете использовать только полный абсолютный путь.

Если файл находится в корневой папке, то путь к файлу будет следующим:

http://site.ru/index.html

HTML

При отсутствии имени файла будет загружаться веб-страница, которая задана по умолчанию в настройках веб-сервера (так называемый индексный файл).

http://site.ru/

HTML

Обычно в качестве индексного файла выступает документ с именем index.html. Наличие завершающего слеша / означает, что обращение идет к папке, если его нет — напрямую к файлу.

2.2. Относительный путь

**Относительный путь** описывает путь к указанному документу относительно текущего. Путь определяется с учётом местоположения веб-страницы, на которой находится ссылка. Относительные ссылки используются при создании ссылок на другие документы на одном и том же сайте. Когда браузер не находит в ссылке протокол http://, он выполняет поиск указанного документа на том же сервере.

Относительный путь содержит следующие компоненты:

папка (имя папки, указывающей путь к файлу);

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

Главное отличие относительного пути от абсолютного в том, что относительный путь не содержит имени корневой папки и родительских папок, что делает адрес короче, и в случае переезда с одного домена на другой не нужно прописывать новый абсолютный адрес. Но если сторонний ресурс будет ссылаться например, на ваши изображения с относительными адресами, то они не будут отображаться на другом сайте.

3. Якоря

**Якоря**, или внутренние ссылки, создают переходы на различные разделы текущей веб-страницы, позволяя быстро перемещаться между разделами. Это оказывается очень удобным в случае, когда на странице слишком много текста. Внутренние ссылки также создаются при помощи элемента <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>

HTML

Если нужно сделать ссылку **с одной страницы** сайта на определенный раздел **другой страницы**, то необходимо задать id для этого раздела страницы, а затем добавить его к абсолютному адресу ссылки:

<th id="about-color">color</th>

<a href="https://html5book.ru/css-shrifty/#about-color" class="site" target="\_blank">color</a>

HTML

4. Как сделать изображение-ссылку

Чтобы сделать кликабельное изображение, необходимо поместить элемент <img> внутрь элемента <a>. Чтобы ссылка открывалась в другом окне, нужно добавить атрибут target="\_blank" для ссылки.

<a href="http://www.fast-torrent.ru/film/gran-za-granyu-tv.html" target="\_blank"><img src="//html5book.ru/wp-content/uploads/2014/07/Fringe.jpg" alt="Fringe"></a>

HTML

РИС. 2. ИЗОБРАЖЕНИЕ-ССЫЛКА

5. Как сделать ссылку на телефонный номер, скайп или адрес электронной почты

У ссылок появились новые возможности — по клику можно не только переходить на другие страницы и скачивать файлы, но и совершать звонки на телефоны, отправлять сообщения или звонить по скайпу.

ссылка на телефонный номер

<a href="tel:+74951234567">+7 (495) 123-45-67</a>

ссылка на адрес электронной почты

<a href="mailto:example@mail.ru">example@mail.ru</a>

ссылка на скайп (позвонить)

<a href="skype:имя-пользователя?call">Skype</a>

ссылка на скайп (открыть чат)

<a href="skype:имя-пользователя?chat">Skype</a>

ссылка на скайп (добавить в список контактов)

<a href="skype:имя-пользователя?add">Skype</a>

ссылка на скайп (отправить файл)

<a href="skype:имя-пользователя?sendfile">Skype</a>

HTML

6. Атрибуты ссылок

Элемент <a> поддерживает глобальные атрибуты и собственные.

|  |  |
| --- | --- |
| ТАБЛИЦА 1. АТРИБУТЫ ЭЛЕМЕНТА <A> | |
| **Атрибут** | **Описание, принимаемое значение** |
| download | Дополняет атрибут href и сообщает браузеру, что ресурс должен быть загружен в момент, когда пользователь щелкает по ссылке, вместо того, чтобы, например, предварительно открыть его (как PDF-файл). Задавая имя для атрибута, мы таким образом задаем имя загружаемому объекту. Разрешается использовать атрибут без указания его значения:  <a href="/images/logo.png" download><img src="/images/logo.png" alt="logo"></a>  <a href="/images/logo.png" download="logo"><img src="/images/logo.png" alt="logo"></a>  <a href="files/20022014.pdf" download="Отчет Февраль 2014.pdf">Загрузить отчет за Февраль 2014</a> |
| href | Значением атрибута является URL-адрес документа, на который указывается ссылка. |
| hreflang | Определяет язык связанного веб-документа. Используется только вместе с атрибутом href. Принимаемые значения — аббревиатура, состоящая из пары букв, обозначающих код языка, например:  <a href="http://www.anysite.ru" hreflang="en">Anysite</a> |
| media | Определяет, для каких типов устройств файл будет оптимизирован. Значением может быть любой медиа-запрос. |
| ping | Отслеживает URL-адреса ресурсов, по которым переходил пользователь. |
| rel | Дополняет атрибут href информацией об отношении между текущим и связанным документом. Принимаемые значения: alternate — ссылка на альтернативную версию документа (например, печатную форму страницы, перевод или зеркало). author — ссылка на автора документа. bookmark — постоянный URL-адрес, используемый для закладок. help — ссылка на справку. license — ссылка на информацию об авторских правах на данный веб-документ. next/prev — указывает связь между отдельными URL. Благодаря этой разметке Google может определить, что содержание данных страниц связано в логической последовательности. nofollow — запрещает поисковой системе переходить по ссылкам на данной странице или по конкретной ссылке. noreferrer — указывает, что переходе по ссылке браузер не должен посылать заголовок HTTP-запроса (Referrer), в который записывается информация о том, с какой страницы пришел посетитель сайта. prefetch — указывает, что целевой документ должен быть кэширован, т.е. браузер в фоновом режиме загружает содержимое страницы к себе в кэш. search — указывает, что целевой документ содержит инструмент для поиска. tag — указывает ключевое слово для текущего документа. |
| target | Указывает на то, в каком окне должен открываться документ, к которому ведет ссылка. Принимает следующие значения: \_self — страница загружается в текущее окно; \_blank — страница открывается в новом окне браузера; \_parent — страница загружается во фрейм-родитель; \_top — страница загружается в полное окно браузера. |
| type | Указывает MIME-тип файлов ссылки, т.е. расширение файла, носит больше справочную информацию. |

2 Объясните, как создать внутреннюю гиперссылку?

Главной отличительной особенностью HTML-документов является создание специальных активных ссылок на другие документы, которые именуются гиперссылками.

При наведении мыши на них курсор принимает форму руки, а при клике в окне браузера открывается другая веб-страница или определённый документ. Гиперссылки могут представлять собой как текст, так и изображение.

Как сделать гиперссылку в HTML? Гиперссылка задаётся при помощи парного тега <a> (от английского слова **Anchor** — якорь). Параметр **href** тега <a> содержит адрес документа.

В HTML теги гиперссылки представлены в виде <a> и </a>, а ссылкой считается вся информация, которую <a> и </a> содержат:

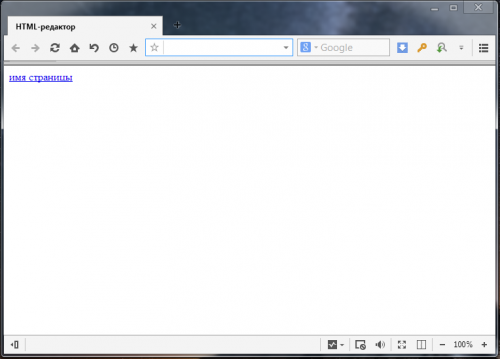
<a href=”адрес гиперссылки в html ”>текст ссылки</a>

Перед тем, как создать гиперссылку в HTML, следует знать, что её адрес может быть относительным и абсолютным. Относительный указывает расположение документа относительно текущей страницы.

Сработает он в том случае, если конечный документ располагается в той же директории, что и текущий. Также, вы можете указать путь к файлу относительно корневой директории веб-сайта с помощью символа «**/**».

Например:

<a href=”/example.html”>имя страницы</a>



Абсолютный адрес представляет собой ссылку в полном виде на документ, расположенный в сети интернет.

Например:

<a href=”http://sitename.com/sitepage.html”>имя страницы</a>

Создание гиперссылок в HTML — процесс несложный. По умолчанию они подчёркиваются и выделяются синим цветом, а посещённые — фиолетовым.

Изменить цвет гиперссылки в HTML можно в параметрах тега <body>:

**link** – цвет ссылок веб-страницы;

**vlink** – цвет посещённых гиперссылок веб-страницы;

**alink** – цвет активных ссылок веб-страницы.

3 Объясните, как создать гиперссылку на электронный ящик?

Создание ссылки на адрес электронной почты делается почти также как и ссылка на веб-страницу. Только вместо URL указывается mailto:адрес электронной почты (пример 8.6).

**Пример 8.6. Ссылка на адрес электронной почты**

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

**<html>**

**<head>**

**<meta** http-equiv="content-type" content="text/html; charset=utf-8"**>**

**<title>**Адрес почты**</title>**

**</head>**

**<body>**

**<p><a** href="mailto:vlad@htmlbook.ru"**>**Задавайте вопросы по электронной почте**</a></p>**

**</body>**

**</html>**

В атрибуте href тега **<a>** вначале пишется ключевое слово mailto, затем через двоеточие желаемый почтовый адрес. Подобная ссылка ничем не отличается от ссылки на веб-страницу, но при нажатии на нее запускается почтовая программа, установленная по умолчанию. Поэтому в названии ссылки следует указывать, что она имеет отношение к электронной почте, чтобы читатели понимали, к чему приведет нажатие на нее.

Можно также автоматически добавить тему сообщения, присоединив к адресу электронной почты через символ вопроса (?) параметр subject=тема сообщения, как показано в примере 8.7.

**Пример 8.7. Задание темы сообщения**

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

**<html>**

**<head>**

**<meta** http-equiv="content-type" content="text/html; charset=utf-8"**>**

**<title>**Тема письма**</title>**

**</head>**

**<body>**

**<p><a** href="mailto:vlad@htmlbook.ru?subject=Вопрос по HTML"**>**Задавайте

вопросы по электронной почте**</a></p>**

**</body>**

**</html>**

При запуске почтовой программы поле Тема (Subject) будет заполнено автоматически.

4 Опишите общую структуру HTML документа.

**Основы HTML** содержат основные правила языка HTML, описание структуры HTML-страницы, отношения в структуре HTML-документа между HTML-элементами.

HTML-документ — это обычный текстовый документ, может быть создан как в обычном текстовом редакторе **(Блокнот)**, так и в специализированном, с подсветкой кода **(Notepad++, Visual Studio Code и т.п.)**. HTML-документ имеет расширение .html.

HTML-документ состоит из дерева HTML-элементов и текста. Каждый элемент обозначается в исходном документе начальным (открывающим) и конечным (закрывающим) тегом (за редким исключением).

РЕКЛАМА

**Начальный тег** показывает, где начинается элемент, конечный — где заканчивается. **Закрывающий тег** образуется путем добавления слэша / перед именем элемента: <имя элемента>…</имя элемента>. Между начальным и закрывающим тегами находится содержимое элемента — контент.

Элементы, представленные одиночными тегами, не могут хранить в себе содержимого напрямую, оно прописывается как значение атрибута, например, элемент <input type="button" value="Кнопка"> создаст кнопку с текстом **Кнопка** внутри.

Элементы могут вкладываться друг в друга, например, <p><i>Текст</i></p>. При вложении следует соблюдать порядок их закрытия **(принцип «матрёшки»)**, например, следующая запись будет неверной: <p><i>Текст</p></i>.

HTML-элементы могут иметь атрибуты (глобальные, применяемые для всех HTML-элементов, и собственные). Атрибуты прописываются в открывающем теге элемента и содержат имя и значение, указываемые в формате имя атрибута="значение". Атрибуты позволяют изменять свойства и поведение элемента, для которого они заданы.

Каждому элементу можно присвоить несколько значений class и только одно значение id. Множественные значения class записываются через пробел, <div class="nav top">. Значения class и id должны состоять только из букв, цифр, дефисов и нижних подчеркиваний и должны начинаться только с букв или цифр.

Браузер просматривает (интерпретирует) HTML-документ, выстраивая его структуру (DOM) и отображая ее в соответствии с инструкциями, включенными в этот файл (таблицы стилей, скрипты). Если разметка правильная, то в окне браузера будет отображена HTML-страница, содержащая HTML-элементы — заголовки, таблицы, изображения и т.д.

Процесс интерпретации **(парсинг)** начинается прежде, чем веб-страница полностью загружена в браузер. Браузеры обрабатывают HTML-документы последовательно, с самого начала, при этом обрабатывая CSS и соотнося таблицы стилей с элементами страницы.

HTML-документ состоит из двух разделов — заголовка — содержимое элемента <head> и содержательной части — содержимое <body>.

Структура веб-страницы

**СОДЕРЖАНИЕ:**

[1. Структура HTML-документа](https://html5book.ru/osnovy-html/?ysclid=l9o9rr5gvk24255502#part1)

[1.1. Элемент <html>](https://html5book.ru/osnovy-html/?ysclid=l9o9rr5gvk24255502#part2)

[1.2. Элемент <head>](https://html5book.ru/osnovy-html/?ysclid=l9o9rr5gvk24255502#part3)

[1.2.1. Элемент <title>](https://html5book.ru/osnovy-html/?ysclid=l9o9rr5gvk24255502#part4)

[1.2.2. Элемент <meta>](https://html5book.ru/osnovy-html/?ysclid=l9o9rr5gvk24255502#part5)

[1.2.3. Элемент <style>](https://html5book.ru/osnovy-html/?ysclid=l9o9rr5gvk24255502#part6)

[1.2.4. Элемент <link>](https://html5book.ru/osnovy-html/?ysclid=l9o9rr5gvk24255502#part7)

[1.2.5. Элемент <script>](https://html5book.ru/osnovy-html/?ysclid=l9o9rr5gvk24255502#part8)

[1.3. Элемент <body>](https://html5book.ru/osnovy-html/?ysclid=l9o9rr5gvk24255502#part9)

1. Структура HTML-документа

Язык HTML следует правилам, которые содержатся в файле объявления типа документа (Document Type Definition, или DTD). DTD представляет собой XML-документ, определяющий, какие элементы, атрибуты и их значения действительны для конкретного типа HTML. Для каждой версии HTML есть свой DTD.

**DOCTYPE** отвечает за корректное отображение веб-страницы браузером. DOCTYPE определяет не только версию HTML (например, html), но и соответствующий DTD-файл в Интернете.

<!DOCTYPE html> <!-- Объявление формата документа -->

<html>

<head> <!-- Техническая информация о документе -->

<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->

<title>...</title> <!-- Задаем заголовок документа -->

<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю таблицу стилей -->

<script src="script.js"></script> <!-- Подключаем сценарии -->

</head>

<body> <!-- Основная часть документа -->

</body>

</html>

HTML

Элементы, находящиеся внутри элемента <html>, образуют дерево документа, так называемую **объектную модель документа, DOM (document object model)**. При этом элемент <html> является корневым элементом.

РИС. 1. ПРОСТЕЙШАЯ СТРУКТУРА ВЕБ-СТРАНИЦЫ

Чтобы разобраться во взаимодействии элементов веб-страницы, необходимо рассмотреть так называемые **«родственные отношения»** между элементами. Отношения между множественными вложенными элементами подразделяются на родительские, дочерние и сестринские.

**Предок** — элемент, который заключает в себе другие элементы. На рисунке 1 предком для всех элементов является <html>. В то же время элемент <body> является предком для всех содержащихся в нем элементов: <h1>, <p>, <span>, <nav> и т.д.

**Потомок** — элемент, расположенный внутри одного или более типов элементов. Например, <body> является потомком <html>, а элемент <p> является потомком одновременно для <body> и <html>.

**Родительский элемент** — элемент, связанный с другими элементами более низкого уровня, и находящийся на дереве выше их. На рисунке 1 <html> является родительским только для <head> и <body>. Элемент <p> является родительским только для <span>.

**Дочерний элемент** — элемент, непосредственно подчиненный другому элементу более высокого уровня. На рисунке 1 только элементы <h1>, <h2>, <p> и <nav> являются дочерними по отношению к <body>.

**Сестринский элемент** — элемент, имеющий общий родительский элемент с рассматриваемым, так называемые элементы одного уровня. На рисунке 1 <head> и <body> — элементы одного уровня, так же как и элементы <h1>, <h2> и <p> являются между собой сестринскими.

1.1. Элемент <html>

Является корневым элементом документа. Все остальные элементы содержатся внутри <html>...</html>. Все, что находится за пределами элемента, не воспринимается браузером как HTML-код и никак им не обрабатывается.

Для элемента доступен атрибут, описанный в таблице ниже, а также [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

|  |  |
| --- | --- |
| ТАБЛИЦА 1. АТРИБУТ ЭЛЕМЕНТА <HTML> | |
| **Атрибут** | **Описание, принимаемое значение** |
| manifest | С помощью значения атрибута указывается путь к документу кэша манифеста, например: <html manifest="about\_company.appcache"> |

1.2. Элемент <head>

Раздел <head>...</head> содержит техническую информацию о странице: заголовок, описание, ключевые слова для поисковых машин, кодировку и т.д. Введенная в нем информация не отображается в окне браузера, однако содержит данные, которые указывают браузеру, как следует обрабатывать страницу.

Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

1.2.1. Элемент <title>

Обязательным элементом раздела <head> является <title>. Текст, размещенный внутри элемента <title>, отображается в строке заголовка веб-браузера.

Длина заголовка должна быть не более 60 символов, чтобы полностью поместиться в заголовке. Текст заголовка должен содержать максимально полное описание содержимого веб-страницы.

Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

1.2.2. Элемент <meta>

Необязательным элементом раздела <head> является элемент <meta>. С его помощью можно задать описание содержимого страницы и ключевые слова для поисковых машин, автора HTML-документа и прочие свойства метаданных.

Элемент <head> может содержать несколько элементов <meta>, потому что в зависимости от используемых атрибутов они несут различную информацию.

<meta name="description" content="Описание содержимого страницы">

<meta name="keywords" content="Ключевые слова через запятую">

HTML

Описание содержимого страницы и ключевые слова одновременно можно указывать на нескольких языках, например, на русском и английском:

<meta name="description" lang="ru" content="Описание содержимого страницы">

<meta name="description" lang="en" content="Description">

<meta name="keywords" lang="ru" content="Ключевые слова через запятую">

<meta name="keywords" lang="en" content="Keywords">

HTML

С помощью элемента <meta> можно запретить или разрешить индексацию веб-страницы поисковыми машинами:

Индексация и переход по ссылкам разрешены:

<meta name="robots" content="index, follow">

HTML

Индексация разрешена, переход по ссылкам запрещен:

<meta name="robots" content="index, nofollow">

HTML

Индексация и переход по ссылкам запрещены:

<meta name="robots" content="noindex, nofollow">

HTML

Для автоматической перезагрузки страницы через заданный промежуток времени (например, через 30 секунд) нужно воспользоваться значением refresh:

<meta http-equiv="refresh" content="30">

HTML

Чтобы перебросить посетителя на другую страницу, нужно указать URL-адрес в параметре url:

<meta http-equiv="refresh" content="0; url=http://yandex.ru/">

HTML

Для элемента <meta> доступны атрибуты, описанные в таблице ниже, а также [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

|  |  |
| --- | --- |
| ТАБЛИЦА 2. АТРИБУТЫ ЭЛЕМЕНТА <META> | |
| **Атрибут** | **Описание, принимаемое значение** |
| charset | Указывает кодировку символов для текущего HTML-документа: <meta charset="UTF-8"> |
| content | Содержит произвольный текст, который определяет значение, ассоциируемое с атрибутом http-equiv или name, в зависимости от их значения. |
| http-equiv | Контролирует действия браузера на данной веб-странице (эквивалент HTTP заголовков). При отображении страницы браузер будет следовать инструкциям, заданным в атрибуте: default-style указывает предпочтительный стиль для использования на странице. Атрибут content должен содержать идентификатор элемента <link>, который ссылается на таблицу стилей CSS, или идентификатор элемента <style>, содержащего таблицу стилей. refresh указывает время в секундах до перезагрузки страницы или время до перенаправления на другую страницу, если в атрибуте content после указания времени идет строка "url=адрес\_страницы". Автоматическая перезагрузка страницы через заданный промежуток времени, в данном примере, через 30 секунд: <meta http-equiv="refresh" content="30"> Если необходимо сразу перебросить посетителя на другую страницу, то можно указать URL-адрес в параметре url: <meta http-equiv="refresh" content="0; url=http://mail.ru/"> |
| name | Ассоциируется со значением, содержащемся в атрибуте content. Не должен использоваться в случае, если для элемента уже заданы атрибуты http-equiv, charset или itemprop. application-name указывает название веб-приложения, используемого на странице. author указывает имя автора документа в свободном формате. description определяет краткое описание к содержимому страницы, например: <meta name="description" content="Описание содержимого страницы"> generator указывает один из пакетов программного обеспечения, используемого для создания документа, например: <meta name="generator" content="WordPress 4.0"> . keywords содержит список ключевых слов, разделенных запятыми, соответствующих содержимому страницы, например: <meta name="keywords" content="Ключевые слова через запятую">. 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. |

1.2.3. Элемент <style>

Внутри этого элемента задаются стили, которые используются на странице. Для задания стилей в HTML-документе используется язык CSS. Таких элементов на странице может быть несколько.

Для элемента доступны атрибуты, приведенные в таблице ниже, а также [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

|  |  |
| --- | --- |
| ТАБЛИЦА 3. АТРИБУТЫ ЭЛЕМЕНТА <STYLE> | |
| **Атрибут** | **Описание, принимаемое значение** |
| media | Определяет, для какого типа устройства предназначены те или иные каскадные таблицы стилей. Внутри таблицы стилей CSS специфичные правила для определенных устройств прописываются с помощью media query. |
| nonce | Генерируемая случайным образом на сервере строковая переменная, которая устанавливает правила использования встроенных стилей с целью защиты контента. Значение атрибута — строка текста. |
| type | Определяет MIME-тип подключаемого файла со стилями, в данном случае type="text/css". |

Элемент может содержать код форматирования как самих элементов веб-страницы, так и веб-страницы целиком.

<style>

.paper {

width: 200px;

height: 300px;

background-color: #ef4444;

color: #666666;

}

</style>

HTML

Чтобы подключить к элементу заданный стиль, необходимо через атрибут class (или id) присвоить элементу соответствующее название:

<div class="paper">

...

</div>

HTML

CSS-код можно встраивать непосредственно в элемент разметки в виде значение атрибута style, например:

<p style="color: #666666; background-color: #ef4444; padding: 20px;">

HTML

1.2.4. Элемент <link>

Задать стили для документа можно также при помощи другого способа — записать их в отдельный файл с расширением .css, например, style.css.

Подключить файл со стилями к веб-странице можно двумя способами:

через директиву @import url

с использованием элемента <link>. Данный способ является предпочтительным.

<!DOCTYPE html>

<html>

<head>

<style>

@import url(style.css);

</style>

<meta>

<title> </title>

</head>

HTML

Элемент <link> определяет отношение между текущей страницей и другими документами. Таких элементов на странице может быть несколько.

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

HTML

Для элемента доступны атрибуты, приведенные в таблице ниже, а также [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

|  |  |
| --- | --- |
| ТАБЛИЦА 4. АТРИБУТЫ ЭЛЕМЕНТА <LINK> | |
| **Атрибут** | **Описание, принимаемое значение** |
| crossorigin | Указывает, должен ли использоваться CORS (технология браузеров, которая позволяет предоставить веб-странице доступ к ресурсам другого домена) при извлечении изображения с сайта. anonymous — в кросс-доменный запрос браузер автоматически добавляет заголовок Origin, содержащий имя домена, с которого осуществлён запрос. Если сервер не ответит CORS-заголовком Access-Control-Allow-Origin: \* (или имя домена вместо звёздочки), то загрузка изображения будет заблокирована. use-credentials — если сервер не предоставляет учётные данные с помощью Access-Control-Allow-Credentials: true, то загрузка изображения будет заблокирована. |
| href | Основной атрибут элемента, в качестве значения выступает путь к файлу со стилями. |
| hreflang | Определяет язык текста в документе, на который идет ссылка. |
| media | Определяет тип устройства, к которым должен быть применен ресурс ссылки. |
| nonce | Генерируемая случайным образом на сервере строковая переменная, которая устанавливает правила использования встроенных стилей с целью защиты контента. Значение атрибута — строка текста. |
| rel | Атрибут определяет отношения между текущим документом и документом, на который идет ссылка. alternate — ссылка на тот же документ, но в другом формате (например, страницы для печати, перевод, зеркало, лента в формате RSS или Atom), <link rel="alternate stylesheet" type="text/css" title="small\_font" href="small-font.css">. <link rel="alternate" type="application/rss+xml" title="my\_RSS" href="index.xml"> <link rel="alternate" type="application/atom+xml" title="News" href="/atom.xml?type=news"> archives — указывает на то, что документ по ссылке представляет исторический интерес. Ссылка может указывать на коллекцию записей, документов и других материалов. author ссылка на страницу об авторе документа или на страницу с контактными данными автора. bookmark ссылка на ближайшего предка статьи, являющегося связующим звеном, или на раздел статьи, наиболее тесно связанных с элементом, если нет предка. external используется для указания того, что страница на которую ведет ссылка не являются частью данного сайта. first указывает ссылку, ведущую на первый документ из последовательности документов. help ссылка на документ со справкой. icon определяет путь к иконке, которая будет использована для текущего документа. last указывает ссылку, ведущую на последний документ в последовательности документов. license ссылка на сведения об авторских правах для документа. next указывает, что этот документ является частью серии, и что ссылка ведет на следующий документ в этой серии. <link rel="next" href="/next.html" type="text/html" title="Следующая страница"> nofollow указывает на то, что ссылка не одобрена автором страницы или что ссылка носит коммерческий характер. noreferrer указывает на то, что заголовок запроса клиента, содержащий url источника запроса, не должен передаваться при переходе по ссылке. pingback указывает адрес пингбэк-сервера, что дает возможность для блога автоматически оповещать сайты, ссылающиеся на него. prefetch указывает, что следует заранее кэшировать файл, на который ведет ссылка. prev указывает, что этот документ является частью серии, и что ссылка ведет на предыдущий документ в этой серии. <link rel="prev" href="/next.html" type="text/html" title="Предыдущая страница"> search указывает, что ссылаемый документ содержит интерфейс поиска и связанных с ним ресурсов. sidebar указывает, что ссылаемый документ, если это возможно, будет показан в дополнительном контексте браузера, и некоторые браузеры при щелчке по гиперссылке открывают окно для добавления ссылки в панель закладок. stylesheet ссылка на внешний файл, который будет использоваться в качестве таблицы стилей для данного документа. tag указывает на то, что метка, на которую ведет гиперссылка, относится к данному документу. up указывает, что страница является частью иерархической структуры, и что гиперссылка ведет на более высокий уровень ресурса в структуре. |
| sizes | Указывает размер иконок для визуального отображения. Атрибут sizes используется только совместно с rel="icon", и может принимать следующий значения: ширинахвысота — определяет список размеров, разделенных пробелами, каждый размер должен быть в формате — ширинахвысота (размеры иконки задаются в пикселях), например: <link rel="icon" href="favicon.png" sizes="16x16 32х32" type="image/png">; any — иконка может масштабироваться до любого размера. |
| title | Определяет заголовок ссылки или название набора альтернативных таблиц стилей. Значение атрибута — текст. |
| type | Определяет MIME-тип документа, на который идет ссылка. В данном случае он принимает значение "text/css". |

1.2.5. Элемент <script>

Элемент <script> позволяет присоединять к документу различные сценарии. Текст сценария может располагаться либо внутри этого элемента, либо во внешнем файле. Если текст сценария расположен во внешнем файле, то он подключается с помощью атрибутов элемента.

Для элемента доступны атрибуты, приведенные в таблице ниже, а также [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

|  |  |
| --- | --- |
| ТАБЛИЦА 5. АТРИБУТЫ ЭЛЕМЕНТА <SCRIPT> | |
| **Атрибут** | **Описание, принимаемое значение** |
| async | Атрибут указывает на то, что сценарий будет выполняться асинхронно с остальной частью страницы (сценарий начнет выполняться одновременно с загрузкой страницы). |
| charset | Определяет кодировку символов |
| crossorigin | Определяет, будет ли использоваться CORS при загрузке внешних скриптов (с использованием атрибута src). anonymous — перед загрузкой скрипта в кросс-доменный запрос браузер автоматически добавляет заголовок Origin, при этом не передаются параметры доступа (cookie, сертификат X.509, логин/пароль для базовой аутентификации по HTTP). Если в ответе сервера отсутствует заголовок Access-Control-Allow-Origin: имя домена, скрипт не будет загружен. use-credentials — перед загрузкой скрипта в кросс-доменный запрос браузер автоматически добавляет заголовок Origin с указанием параметров доступа (cookie, SSL-сертификат или пары логин/пароль). Если в ответе сервера отсутствует заголовок Access-Control-Allow-Credentials: true, скрипт не будет загружен. |
| defer | Интерпретация сценариев откладывается до окончания отображения документа на устройстве пользователя. |
| nonce | Обеспечивает безопасность, защищая от атак с внедрением межсайтового скриптинга (XSS, cross site scripting). Устанавливает правила использования встроенных скриптов с помощью nonce-значений и хэшей. Во время рендеринга страницы браузер для каждого инлайн-скрипта вычисляет хэши и сравнивает с перечисленными в CSP. Загрузка с ресурсов, не входящих в «белый список», блокируется. |
| src | Указывает на месторасположение файла со сценарием, значение атрибута — это url файла, содержащего JavaScript-программу. |
| type | Используются для объявления языка сценария, использованного при составлении содержимого элемента. |

1.3. Элемент <body>

В разделе <body> располагается все содержимое документа.

Для элемента доступны атрибуты, приведенные в таблице ниже, а также [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

|  |  |
| --- | --- |
| ТАБЛИЦА 5. АТРИБУТЫ ЭЛЕМЕНТА <BODY> | |
| **Атрибут** | **Описание, принимаемое значение** |
| onafterprint | Событие, срабатывающее после отправки страницы на печать или после закрытия окна печати. |
| onbeforeprint | Событие, срабатывающее перед отправкой страницы на печать. |
| onbeforeunload | Событие срабатывает, когда посетитель инициировал переход на другую страницу или нажал «закрыть окно». Позволяет отображать сообщение в диалоговом окне подтверждения, чтобы сообщить пользователю, хочет ли он остаться или покинуть текущую страницу. |
| onhashchange | Событие срабатывает, когда меняется hash-часть URL, например, когда пользователь перейдет с адреса example.domain/test.aspx#page1 на example.domain/test.aspx#page2. |
| onmessage | Событие происходит, когда сообщение получено через источник события. |
| onoffline | Событие вызывается браузером в том случае, когда браузер определит, что соединение с интернет пропало. |
| ononline | Событие вызывается браузером в том случае, когда соединение с интернет возобновилось. |
| onpagehide | Событие происходит, когда пользователь покидает страницу посредством навигации, например, нажав на ссылку, обновив страницу, заполнив форму и т.д. |
| onpageshow | Событие происходит, когда пользователь переходит на веб-страницу, после события onload. |
| onunload | Событие срабатывает если страница не загрузилась по каким-либо причинам, либо при закрытии окна браузера. |

5 Перечислите теги, применяемые для организации списков.

**HTML-списки** используются для группировки связанных между собой фрагментов информации. Существует три вида списков:

**маркированный список** — <ul> — каждый элемент списка <li> отмечается маркером,

**нумерованный список** — <ol> — каждый элемент списка <li> отмечается цифрой,

**список определений** — <dl> — состоит из пар термин <dt> — <dd> определение.

Каждый список представляет собой контейнер, внутри которого располагаются элементы списка или пары термин-определение.

Элементы списка ведут себя как блочные элементы, располагаясь друг под другом и занимая всю ширину блока-контейнера. Каждый элемент списка имеет дополнительный блок, расположенный сбоку, который не участвует в компоновке.

Создание HTML-списков

**СОДЕРЖАНИЕ:**

[1. Маркированный список <ul>](https://html5book.ru/html-lists/?ysclid=l9o9srdxip904639502#part1)

[2. Нумерованный список <ol>](https://html5book.ru/html-lists/?ysclid=l9o9srdxip904639502#part2)

[3. Список определений <dl>](https://html5book.ru/html-lists/?ysclid=l9o9srdxip904639502#part3)

[4. Вложенный список](https://html5book.ru/html-lists/?ysclid=l9o9srdxip904639502#part4)

[5. Многоуровневый нумерованный список](https://html5book.ru/html-lists/?ysclid=l9o9srdxip904639502#part5)

1. Маркированный список

**Маркированный список** представляет собой неупорядоченный список (от англ. Unordered List). Создаётся с помощью элемента <ul>. В качестве маркера элемента списка выступает метка, например, закрашенный кружок.

Браузеры по умолчанию добавляют следующее форматирование блоку списка:

ul {

padding-left: 40px;

margin-top: 1em;

margin-bottom: 1em;

}

CSS

Каждый элемент списка создаётся с помощью элемента <li> (от англ. List Item).

Для элемента <ul> доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

<ul>

<li>Microsoft</li>

<li>Google</li>

<li>Apple</li>

<li>IBM</li>

</ul>

HTML

Microsoft

Google

Apple

IBM

ФИГУРА. 1. МАРКИРОВАННЫЙ СПИСОК

2. Нумерованный список

**Нумерованный список** создаётся с помощью элемента <ol>. Каждый пункт списка также создаётся с помощью элемента <li>. Браузер нумерует элементы по порядку автоматически и если удалить один или несколько элементов такого списка, то остальные номера будут автоматически пересчитаны.

Блок списка также имеет стили браузера по умолчанию:

ol {

padding-left: 40px;

margin-top: 1em;

margin-bottom: 1em;

}

CSS

Для элемента <li> доступен атрибут value, который позволяет изменить номер по умолчанию для выбранного элемента списка. Например, если для первого пункта списка задать <li value="10">, то остальная нумерация будет пересчитана относительно нового значения.

Для элемента <ol> доступны следующие атрибуты:

|  |  |
| --- | --- |
| ТАБЛИЦА 1. АТРИБУТЫ ЭЛЕМЕНТА <OL> | |
| **Атрибут** | **Описание, принимаемое значение** |
| reversed | Атрибут reversed задает отображение списка в обратном порядке (например, 9, 8, 7…). |
| start | Атрибут start задает начальное значение, от которого пойдет отсчет нумерации, например, конструкция <ol start="10"> первому пункту присвоит порядковый номер «10». Также можно одновременно задавать тип нумерации, например, <ol type="I" start="10">. |
| type | Атрибут type задает вид маркера для использования в списке (в виде букв или цифр). Принимаемые значения: 1 — значение по умолчанию, десятичная нумерация. A — нумерация списка в алфавитном порядке, заглавные буквы (A, B, C, D). a — нумерация списка в алфавитном порядке, строчные буквы (a, b, c, d). I — нумерация римскими заглавными цифрами (I, II, III, IV). i — нумерация римскими строчными цифрами (i, ii, iii, iv). |

<ol>

<li>Microsoft</li>

<li>Google</li>

<li>Apple</li>

<li>IBM</li>

</ol>

HTML

Microsoft

Google

Apple

IBM

ФИГУРА. 2. НУМЕРОВАННЫЙ СПИСОК

3. Список определений

**Списки определений** создаются с помощью элемента <dl>. Для добавления термина применяется элемент <dt>, а для вставки определения — элемент <dd>.

Блок списка определений имеет следующие стили браузера по умолчанию:

dl {

margin-top: 1em;

margin-bottom: 1em;

}

CSS

Для элементов <dl>, <dt> и <dd> доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

<dl>

<dt>Режиссер:</dt>

<dd>Петр Точилин</dd>

<dt>В ролях:</dt>

<dd>Андрей Гайдулян</dd>

<dd>Алексей Гаврилов</dd>

<dd>Виталий Гогунский</dd>

<dd>Мария Кожевникова</dd>

</dl>

HTML

Режиссер:

Петр Точилин

В ролях:

Андрей Гайдулян

Алексей Гаврилов

Виталий Гогунский

Мария Кожевникова

ФИГУРА. 3. СПИСОК ОПРЕДЕЛЕНИЙ

4. Как создать вложенный список

Зачастую возможностей простых списков не хватает, например, при создании оглавления никак не обойтись без **вложенных пунктов**. Разметка для вложенного списка будет следующей:

<ul>

<li>Пункт 1.</li>

<li>Пункт 2.

<ul>

<li>Подпункт 2.1.</li>

<li>Подпункт 2.2.

<ul>

<li>Подпункт 2.2.1.</li>

<li>Подпункт 2.2.2.</li>

</ul>

</li>

<li>Подпункт 2.3.</li>

</ul>

</li>

<li>Пункт 3.</li>

</ul>

HTML

Пункт 1.

Пункт 2.

Подпункт 2.1.

Подпункт 2.2.

Подпункт 2.2.1.

Подпункт 2.2.2.

Подпункт 2.3.

Пункт 3.

ФИГУРА. 4. ВЛОЖЕННЫЙ СПИСОК

5. Многоуровневый нумерованный список

Многоуровневый список используется для отображения элементов списка на разных уровнях с различными отступами. Разметка для многоуровневого нумерованного списка будет следующей:

<ol>

<li>пункт</li>

<li>пункт

<ol>

<li>пункт</li>

<li>пункт</li>

<li>пункт

<ol>

<li>пункт</li>

<li>пункт</li>

<li>пункт</li>

</ol>

</li>

<li>пункт</li>

</ol>

</li>

<li>пункт</li>

<li>пункт</li>

</ol>

HTML

Такая разметка по умолчанию создаст для каждого вложенного списка новую нумерацию, начинающуюся с единицы. Чтобы сделать вложенную нумерацию, нужно использовать следующие свойства:

counter-reset сбрасывает один или несколько счётчиков, задавая значение для сброса;

counter-increment задаёт значение приращения счётчика, т.е. с каким шагом будет нумероваться каждый последующий пункт;

content — генерируемое содержимое, в данном случае отвечает за вывод номера перед каждым пунктом списка.

ol {

/\* убираем стандартную нумерацию \*/

list-style: none;

/\* Идентифицируем счетчик и даем ему имя li. Значение счетчика не указано - по умолчанию оно равно 0 \*/

counter-reset: li;

}

li:before {

/\* Определяем элемент, который будет нумероваться — li. Псевдоэлемент before указывает, что содержимое, вставляемое при помощи свойства content, будет располагаться перед пунктами списка. Здесь же устанавливается значение приращения счетчика (по умолчанию равно 1). \*/

counter-increment: li;

/\* С помощью свойства content выводится номер пункта списка. counters() означает, что генерируемый текст представляет собой значения всех счетчиков с таким именем. Точка в кавычках добавляет разделяющую точку между цифрами, а точка с пробелом добавляется перед содержимым каждого пункта списка \*/

content: counters(li, ".") ". ";

}

CSS

пункт

пункт

пункт

пункт

пункт

пункт

пункт

пункт

пункт

пункт

пункт

ФИГУРА. 5. МНОГОУРОВНЕВЫЙ НУМЕРОВАННЫЙ СПИСОК

6 Объясните, как создать нумерованный список?

Нумерованные списки представляют собой набор элементов с их порядковыми номерами. Вид и тип нумерации зависит от атрибутов тега **<ol>**, который и применяется для создания списка. Каждый пункт нумерованного списка обозначается тегом **<li>**, как показано ниже.

**<ol>**

**<li>**Первый пункт**</li>**

**<li>**Второй пункт**</li>**

**<li>**Третий пункт**</li>**

**</ol>**

Если не указывать никаких дополнительных атрибутов и просто написать тег **<ol>**, то по умолчанию применяется список с арабскими числами (1, 2, 3,...), как показано в примере 11.3.

**Пример 11.3. Создание нумерованного списка**

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

**<html>**

**<head>**

**<meta** http-equiv="Content-Type" content="text/html; charset=utf-8"**>**

**<title>**Нумерованный список**</title>**

**</head>**

**<body>**

**<p><strong>**Работа со временем**</strong></p>**

**<ol>**

**<li>**создание пунктуальности (никогда не будете никуда опаздывать);**</li>**

**<li>**излечение от пунктуальности (никогда никуда не будете торопиться);**</li>**

**<li>**изменение восприятия времени и часов.**</li>**

**</ol>**

**</body>**

**</html>**

Результат данного примера показан на рис. 11.3.

*Рис. 11.3. Вид нумерованного списка*

Заметьте, что в нумерованном списке также добавляются автоматические отступы сверху, снизу и слева от текста.

В качестве нумерующих элементов могут выступать следующие значения:

арабские числа (1, 2, 3, ...);

прописные латинские буквы (A, B, C, ...);

строчные латинские буквы (a, b, c, ...);

прописные римские числа (I, II, III, ...);

строчные римские числа (i, ii, iii, ...).

Для указания типа нумерованного списка применяется атрибут type тега **<ol>**. Его возможные значения приведены в табл. 11.2.

|  |  |  |
| --- | --- | --- |
| *Табл. 11.2. Типы нумерованного списка* | | |
| **Тип списка** | **Код HTML** | **Пример** |
| Арабские числа | <ol type="1"> <li>...</li> </ol> | 1. Чебурашка 2. Крокодил Гена 3. Шапокляк |
| Прописные буквы латинского алфавита | <ol type="A"> <li>...</li> </ol> | A. Чебурашка B. Крокодил Гена C. Шапокляк |
| Строчные буквы латинского алфавита | <ol type="a"> <li>...</li> </ol> | a. Чебурашка b. Крокодил Гена c. Шапокляк |
| Римские числа в верхнем регистре | <ol type="I"> <li>...</li> </ol> | I. Чебурашка II. Крокодил Гена III. Шапокляк |
| Римские числа в нижнем регистре | <ol type="i"> <li>...</li> </ol> | i. Чебурашка ii. Крокодил Гена iii. Шапокляк |

Чтобы начать список с определенного значения, используется атрибут start тега **<ol>**. При этом не имеет значения, какой тип списка установлен с помощью type, атрибут start одинаково работает и с римскими и с арабскими числами. В примере 11.4 показано создание списка с использованием римских цифр в верхнем регистре, начинающихся с восьми.

**Пример 11.4. Нумерация списка**

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

**<html>**

**<head>**

**<meta** http-equiv="Content-Type" content="text/html; charset=utf-8"**>**

**<title>**Римские числа**</title>**

**</head>**

**<body>**

**<ol** type="I" start="8"**>**

**<li>**Король Магнум XLIV**</li>**

**<li>**Король Зигфрид XVI**</li>**

**<li>**Король Сигизмунд XXI**</li>**

**<li>**Король Хусбрандт I**</li>**

**</ol>**

**</body>**

**</html>**

Результат данного примера показан на рис. 11.4.

*Рис. 11.4. Нумерованный список с римскими числами*

7 Объясните, как создать маркированный список?

Маркированный список определяется тем, что перед каждым элементом списка добавляется небольшой маркер, обычно в виде закрашенного кружка. Сам список формируется с помощью контейнера **<ul>**, а каждый пункт списка начинается с тега **<li>**, как показано ниже.

**<ul>**

**<li>**Первый пункт**</li>**

**<li>**Второй пункт**</li>**

**<li>**Третий пункт**</li>**

**</ul>**

В списке непременно должен присутствовать закрывающий тег **</ul>**, иначе возникнет ошибка. Закрывающий тег **</li>** хотя и не обязателен, но советуем всегда его добавлять, чтобы четко разделять элементы списка.

В примере  11.1 приведен код HTML для добавления маркированного списка на веб-странице.

**Пример 11.1. Создание маркированного списка**

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

**<html>**

**<head>**

**<meta** http-equiv="Content-Type" content="text/html; charset=utf-8"**>**

**<title>**Маркированный список**</title>**

**</head>**

**<body>**

**<hr>**

**<ul>**

**<li>**Чебурашка**</li>**

**<li>**Крокодил Гена**</li>**

**<li>**Шапокляк**</li>**

**<li>**Крыса Лариса**</li>**

**</ul>**

**<hr>**

**</body>**

**</html>**

Результат данного примера показан на рис. 11.1.

*Рис. 11.1. Вид маркированного списка*

Обратите внимание на отступы сверху, снизу и слева от списка. Такие отступы добавляются автоматически.

Маркеры могут принимать один из трех видов: круг (по умолчанию), окружность и квадрат. Для выбора стиля маркера используется атрибут type тега **<ul>**. Допустимые значения приведены в табл. 11.1

|  |  |  |
| --- | --- | --- |
| *Табл. 11.1. Стили маркеров списка* | | |
| **Тип списка** | **Код HTML** | **Пример** |
| Список с маркерами в виде круга | <ul type="disc"> <li>...</li> </ul> | Первый  Второй  Третий |
| Список с маркерами в виде окружности | <ul type="circle"> <li>...</li> </ul> | Первый  Второй  Третий |
| Список с квадратными маркерами | <ul type="square"> <li>...</li> </ul> | Первый  Второй  Третий |

Вид маркеров может незначительно различаться в разных браузерах, а также при смене шрифта и размера текста.

Создание списка с квадратными маркерами показано в примере 11.2.

**Пример 11.2. Вид маркеров**

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

**<html>**

**<head>**

**<meta** http-equiv="Content-Type" content="text/html; charset=utf-8"**>**

**<title>**Маркированный список**</title>**

**</head>**

**<body>**

**<p><strong>**Изменение убеждений**</strong></p>**

**<ul** type="square"**>**

**<li>**изменение религиозной веры (на выбор: буддизм, конфуцианство, индуизм).

Специальное предложение - иудаизм и мусульманство вместе;**</li>**

**<li>**изменение веры в непогрешимость любимой партии;**</li>**

**<li>**убеждение в том, что инопланетяне существуют;**</li>**

**<li>**предпочтение политического строя, как самого лучшего в своем роде

(на выбор: феодализм, социализм, коммунизм, капитализм).**</li>**

**</ul>**

**</body>**

**</html>**

Результат данного примера показан на рис. 11.2.

*Рис. 11.2. Вид списка с квадратными маркерами*