

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационной безопасности

Кафедра инфокоммуникационных технологий

## **ВЕБ-ТЕХНОЛОГИИ В ИНФОКОММУНИКАЦИЯХ**

### **Язык разметки гипертекста HTML**



**Минск 2022**

## Содержание

Лабораторная работа №1 Язык разметки гипертекста HTML .....	3
Структура HTML-документа .....	3
Элемент <html> .....	4
Элемент <head> .....	4
Элемент <title> .....	5
Элемент <meta> .....	5
Элемент <style> .....	6
Элемент <link> .....	7
Элемент <script> .....	9
Элемент <body> .....	10
Элемент <!-- --> .....	11
HTML-теги .....	11
HTML-атрибуты .....	11
HTML-текст .....	12
HTML-элементы для заголовков .....	12
Элементы для форматирования текста .....	13
Элементы для ввода «компьютерного» текста .....	14
Элементы для оформления цитат и определений .....	14
Абзацы, средства переноса текста .....	15
HTML-ссылки .....	15
Структура ссылки .....	15
Абсолютный и относительный путь .....	16
Абсолютный путь .....	17
Относительный путь .....	17
Якоря .....	17
Как сделать изображение-ссылку .....	18
Как сделать ссылку на телефонный номер, скайп или адрес электронной почты .....	18
Атрибуты ссылок .....	18
HTML-изображения .....	20
Элемент <img> .....	20
Адрес изображения .....	21
Размеры изображения .....	21
Форматы графических файлов .....	22
Элемент <map> .....	22
Элемент <area> .....	23
Пример создания карты-изображения .....	24
HTML-таблицы .....	25
Как создать таблицу .....	25
Как создать строки (ряды) таблицы .....	26
Как сделать ячейку заголовка столбца таблицы .....	26
Как сделать ячейку тела таблицы .....	26
Как добавить подпись (заголовок) к таблице .....	26
Группирование строк и столбцов таблицы .....	27
Группировка разделов таблицы .....	27
Как объединить ячейки таблицы .....	28
Атрибуты элементов таблицы .....	28
HTML-списки .....	29
Маркированный список .....	29
Нумерованный список .....	29
Список описаний .....	30
Как создать вложенный список .....	31
Многоуровневый нумерованный список .....	31
Спецсимволы HTML .....	32
Задание к лабораторной работе №1 .....	33

## Лабораторная работа №1 Язык разметки гипертекста HTML

HTML-документ – это обычный текстовый документ, может быть создан как в обычном текстовом редакторе (Блокнот), так и в специализированном, с подсветкой кода (Sublime Text 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).

Начальный тег показывает, где начинается элемент, конечный – где заканчивается. Закрывающий тег образуется путем добавления слэша / перед именем тега: `<имя тега>...</имя тега>`. 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 «матрешки»).

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.

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.

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

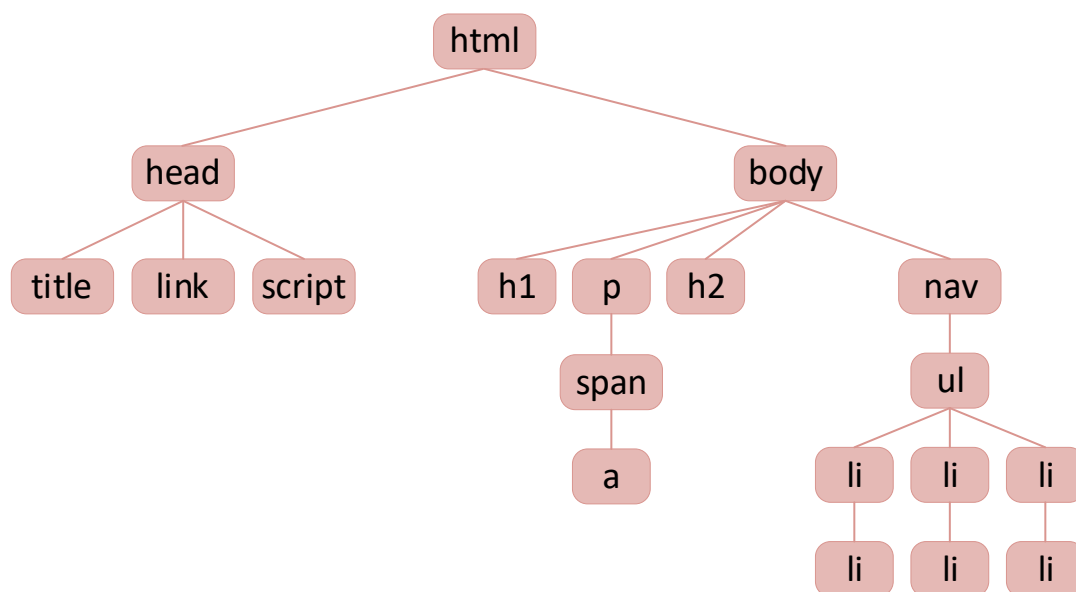
### Структура HTML-документа

Язык HTML следует правилам, которые содержатся в файле объявления типа документа (Document Type Definition, или DTD). DTD представляет собой XML-документ, определяющий, какие теги, 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>
```

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



Отношения между множественными вложенными элементами подразделяются на родительские, дочерние и сестринские.

**Предок** – элемент, который включает в себе другие элементы. На рисунке 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 `<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 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 `<head>` и `<body>` – элементы одного уровня, так же, как и элементы `<h1>`, `<h2>` и `<p>` являются между собой сестринскими.

### 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 доступны [глобальные атрибуты](#).

### 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](#).

## 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](#).

## 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">
```

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">
```

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

1. Индексация и переход по ссылкам разрешены. 2. Индексация разрешена, переход по ссылкам запрещен. 3. Индексация и переход по ссылкам запрещены. 4. Для автоматической перезагрузки страницы через заданный промежуток времени нужно воспользоваться значением refresh. Страница будет перезагружена через 30 секунд. 5. Чтобы перебросить посетителя на другую страницу, нужно указать URL-адрес в параметре url.

```
1 <meta name="robots" content="index, follow">
2 <meta name="robots" content="index, nofollow">
3 <meta name="robots" content="noindex, nofollow">
4 <meta http-equiv="refresh" content="30">
5 <meta http-equiv="refresh" content="0; url=http://bsuir.by/">
```

Для элемента <meta> доступны атрибуты, описанные в таблице ниже, а также [глобальные атрибуты](#).

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

	<p>таблицу стилей CSS, или идентификатор элемента <code>&lt;style&gt;</code>, содержащего таблицу стилей.</p> <p><code>refresh</code> указывает время в секундах до перезагрузки страницы или время до перенаправления на другую страницу, если в атрибуте <code>content</code> 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>&lt;meta http-equiv="refresh" content="30"&gt;</code>.</p> <p>Если необходимо сразу перебросить посетителя на другую страницу, то можно указать URL-адрес в параметре <code>url</code>:  <code>&lt;meta http-equiv="refresh" content="0; url=http://mail.ru/"&gt;</code>.</p>
name	<p>Ассоциируется со значением, содержащемся в атрибуте <code>content</code>. Не должен использоваться в случае, если для элемента уже заданы атрибуты <code>http-equiv</code>, <code>charset</code> или <code>itemprop</code>.</p> <p><code>application-name</code> указывает название веб-приложения, используемого на странице.</p> <p><code>author</code> указывает имя автора документа в свободном формате.</p> <p><code>description</code> определяет краткое описание к содержимому страницы, например:  <code>&lt;meta name="description" content="Описание содержимого страницы"&gt;</code>.</p> <p><code>generator</code> указывает один из пакетов программного обеспечения, используемого для создания документа, например:  <code>&lt;meta name="generator" content="WordPress 4.0"&gt;</code>.</p> <p><code>keywords</code> содержит список ключевых слов, разделенных запятыми, соответствующих содержимому страницы, например:  <code>&lt;meta name="keywords" content="Ключевые слова через запятую"&gt;</code>.</p>

### 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](#).

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 <code>media query</code> .
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 <code>type="text/css"</code> .

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>
```

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

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

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

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

### 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>
```

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">
```

Для элемента доступны атрибуты, приведенные в таблице ниже, а также [глобальные атрибуты](#).

Атрибут	Описание, принимаемое значение
<code>crossorigin</code>	<p>Указывает, должен ли использоваться CORS (технология браузеров, которая позволяет предоставить веб-странице доступ к ресурсам другого домена) при извлечении изображения с сайта.</p> <p><code>anonymous</code> – в кросс-доменный запрос браузер автоматически добавляет заголовок <code>Origin</code>, содержащий имя домена, с которого осуществлен запрос. Если сервер не ответит CORS-заголовком <code>Access-Control-Allow-Origin: *</code> (или имя домена вместо звездочки), то загрузка изображения будет заблокирована.</p> <p><code>use-credentials</code> – если сервер не предоставляет учетные данные с помощью <code>Access-Control-Allow-Credentials: true</code>, то загрузка изображения будет заблокирована.</p>



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	<p>Атрибут определяет отношения между текущим документом и документом, на который идет ссылка.</p> <p><b>alternate</b> – ссылка на тот же документ, но в другом формате (например, страницы для печати, перевод, зеркало, лента в формате RSS или Atom),</p> <pre>&lt;link rel="alternate" stylesheet="type="text/css" title="small_font" href="small-font.css"&gt;.</pre> <pre>&lt;link rel="alternate" type="application/rss+xml" title="my_RSS" href="index.xml"&gt;.</pre> <pre>&lt;link rel="alternate" type="application/atom+xml" title="News" href="/atom.xml?type=news"&gt;.</pre> <p><b>archives</b> – указывает на то, что документ по ссылке представляет исторический интерес. Ссылка может указывать на коллекцию записей, документов и других материалов.</p> <p><b>author</b> – ссылка на страницу об авторе документа или на страницу с контактными данными автора.</p> <p><b>bookmark</b> – ссылка на ближайшего предка статьи, являющегося связующим звеном, или на раздел статьи, наиболее тесно связанных с элементом, если нет предка.</p> <p><b>external</b> – используется для указания того, что страница, на которую ведет ссылка не являются частью данного сайта.</p> <p><b>first</b> – указывает ссылку, ведущую на первый документ из последовательности документов.</p> <p><b>help</b> – ссылка на документ со справкой.</p> <p><b>icon</b> – определяет путь к иконке, которая будет использована для текущего документа.</p> <p><b>last</b> – указывает ссылку, ведущую на последний документ в последовательности документов.</p> <p><b>license</b> – ссылка на сведения об авторских правах для документа.</p> <p><b>next</b> – указывает, что этот документ является частью серии, и что ссылка ведет на следующий документ в этой серии.</p> <pre>&lt;link rel="next" href="/next.html" type="text/html" title="Следующая страница"&gt;.</pre> <p><b>nofollow</b> – указывает на то, что ссылка не одобрена автором страницы или что ссылка носит коммерческий характер.</p> <p><b>noreferrer</b> – указывает на то, что заголовок запроса клиента, содержащий <b>url</b> источника запроса, не должен передаваться при переходе по ссылке.</p> <p><b>pingback</b> – указывает адрес пингбэк-сервера, что дает возможность для блога автоматически оповещать сайты, ссылающиеся на него.</p> <p><b>prefetch</b> – указывает, что следует заранее кэшировать файл, на который ведет ссылка.</p>



	<p><code>prev</code> – указывает, что этот документ является частью серии, и что ссылка ведет на предыдущий документ в этой серии.</p> <p><code>&lt;link rel="prev" href="/next.html" type="text/html" title="Предыдущая страница"&gt;</code>.</p> <p><code>search</code> – указывает, что ссылаемый документ содержит интерфейс поиска и связанных с ним ресурсов.</p> <p><code>sidebar</code> – 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.</p> <p><code>stylesheet</code> – ссылка на внешний файл, который будет использоваться в качестве таблицы стилей для данного документа.</p> <p><code>tag</code> – указывает на то, что метка, на которую ведет гиперссылка, относится к данному документу.</p> <p><code>up</code> – указывает, что страница является частью иерархической структуры, и что гиперссылка ведет на более высокий уровень ресурса в структуре.</p>
sizes	<p>Указывает размер иконок для визуального отображения. Атрибут <code>sizes</code> используется только совместно с <code>rel="icon"</code>, и может принимать следующие значения:</p> <p>ширина×высота – определяет список размеров, разделенных пробелами, каждый размер должен быть в формате – ширина×высота (размеры иконки задаются в пикселях), например:</p> <p><code>&lt;link rel="icon" href="favicon.png" sizes="16x16 32x32" type="image/png"&gt;</code>.</p> <p><code>any</code> – иконка может масштабироваться до любого размера.</p>
title	<p>Определяет заголовок ссылки или название набора альтернативных таблиц стилей. Значение атрибута – текст.</p>
type	<p>Определяет MIME-тип документа, на который идет ссылка. В данном случае он принимает значение <code>"text/css"</code>.</p>

### 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](#).

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	<p>Определяет, будет ли использоваться CORS при загрузке внешних скриптов (с использованием атрибута <code>src</code>).</p> <p><code>anonymous</code> – перед загрузкой скрипта в кросс-доменный запрос браузер автоматически добавляет заголовок <code>Origin</code>, при этом не передаются параметры доступа (<code>cookie</code>, сертификат <code>X.509</code>, логин/пароль для базовой аутентификации по <code>HTTP</code>).</p>

	Если в ответе сервера отсутствует заголовок <code>Access-Control-Allow-Origin: имя домена</code> , скрипт не будет загружен. <code>use-credentials</code> – перед загрузкой скрипта в кросс-доменный запрос браузер автоматически добавляет заголовок <code>Origin</code> с указанием параметров доступа ( <code>cookie</code> , <code>SSL-сертификат</code> или пары <code>логин/пароль</code> ). Если в ответе сервера отсутствует заголовок <code>Access-Control-Allow-Credentials: true</code> , скрипт не будет загружен.
<code>defer</code>	Интерпретация сценариев откладывается до окончания отображения документа на устройстве пользователя.
<code>nonce</code>	Обеспечивает безопасность, защищая от атак с внедрением межсайтового скриптинга ( <code>XSS</code> , <code>cross site scripting</code> ). Устанавливает правила использования встроенных скриптов с помощью <code>nonce</code> -значений и хэшей. Во время рендеринга страницы браузер для каждого инлайн-скрипта вычисляет хэши и сравнивает с перечисленными в <code>CSP</code> . Загрузка с ресурсов, не входящих в «белый список», блокируется.
<code>src</code>	Указывает на месторасположение файла со сценарием, значение атрибута – это <code>url</code> файла, содержащего <code>JavaScript</code> -программу.
<code>type</code>	Используются для объявления языка сценария, использованного при составлении содержимого элемента.

### Элемент `<body>`

В разделе `<body>` располагается все содержимое документа.

Для элемента доступны атрибуты, приведенные в таблице ниже, а также [глобальные атрибуты](#).

Атрибут	Описание, принимаемое значение
<code>onafterprint</code>	Событие, срабатывающее после отправки страницы на печать или после закрытия окна печати.
<code>onbeforeprint</code>	Событие, срабатывающее перед отправкой страницы на печать.
<code>onbeforeunload</code>	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.
<code>onhashchange</code>	Событие срабатывает, когда меняется <code>hash</code> -часть <code>URL</code> , например, когда пользователь перейдет с адреса <code>example.domain/test.aspx#page1</code> на <code>example.domain/test.aspx#page2</code> .
<code>onmessage</code>	Событие происходит, когда сообщение получено через источник события.
<code>onoffline</code>	Событие вызывается браузером в том случае, когда браузер определит, что соединение с интернет пропало.
<code>ononline</code>	Событие вызывается браузером в том случае, когда соединение с интернет возобновилось.
<code>onpagehide</code>	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 <code>onload</code> .
onunload	Событие срабатывает если страница не загрузилась по каким-либо причинам, либо при закрытии окна браузера.

### Элемент `<!-- -->`

Элемент `<!-- -->` добавляет комментарий в код документа. Текст комментария не отображается на странице. Разрешается внутрь комментария добавлять другие элементы. Вложенные комментарии (когда один комментарий расположен внутри другого) недопустимы.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">

<!-- Заголовок документа -->
<title>...</title>
</head>

<!-- Основная часть документа -->
<body>
...
</body>
</html>
```

### HTML-теги

HTML-теги – основа языка HTML. Теги используются для разграничения начала и конца элементов в разметке.

Каждый HTML-документ состоит из дерева HTML-элементов и текста. Каждый HTML-элемент обозначается начальным (открывающим) и конечным (закрывающим) тегом. Открывающий и закрывающий теги содержат имя тега.

Все HTML-элементы делятся на пять типов:

- пустые элементы – `<area>`, `<base>`, `<br>`, `<col>`, `<embed>`, `<hr>`, `<img>`, `<input>`, `<link>`, `<menuitem>`, `<meta>`, `<param>`, `<source>`, `<track>`, `<wbr>`;
- элементы с неформатированным текстом – `<script>`, `<style>`;
- элементы, выводящие неформатированный текст – `<textarea>`, `<title>`;
- элементы из другого пространства имен – MathML и SVG;
- обычные элементы – все остальные элементы.

[Полный список HTML-элементов.](#)

### HTML-атрибуты

HTML-атрибуты сообщают браузеру, каким образом должен отображаться тот или иной элемент страницы. Атрибуты позволяют сделать более разнообразными внешний вид информации, добавляемой с помощью одинаковых тегов.

Значение атрибута заключается в кавычки `"`. Названия и значения атрибутов не чувствительны к регистру, но, тем не менее, рекомендуется набирать их в нижнем регистре.

[Глобальные атрибуты](#) могут быть использованы для любого HTML-элемента, хотя некоторые из них могут не оказывать на элементы никакого влияния.

Например, атрибут `title` описывает содержимое элемента в виде всплывающей подсказки, которая появляется при наведении курсора на элемент. Вид такой подсказки зависит от браузера, настроек операционной системы и не может быть изменен с помощью HTML-кода или стилей.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Атрибут title</title>
</head>
<body>
<p title="А вот и я!">Пример всплывающей подсказки</p>
</body>
</html>
```

## HTML-текст

HTML-текст представлен в спецификации элементами для форматирования и группировки текста. Данные элементы являются контейнерами для текста и не имеют визуального отображения.

Элементы для форматирования текста несут смысловую нагрузку и обычно задают для текста, заключенного внутрь, стилевое оформление, например, выделяют текст жирным начертанием или отображают его шрифтом другого семейства (свойство `font-family`).

Грамотно отформатированный текст дает понять поисковым системам, какие слова несут важную смысловую нагрузку, по каким из них предпочтительно ранжировать веб-страницу в поисковой выдаче. Вся текстовая информация, отображаемая на сайте, размещается внутри элемента `<body>`.

## HTML-элементы для заголовков

Заголовки являются важными элементами веб-страницы, они упорядочивают текст, формируя его визуальную структуру. Элементы `<h1>...<h6>` должны использоваться только для выделения заголовков нового раздела или подраздела.

При использовании заголовков необходимо учитывать их иерархию, т.е. за `<h1>` должен следовать `<h2>` и т.д. Также не рекомендуется вкладывать в заголовки другие элементы.

### Элемент `<h1>`

Заголовок самого верхнего уровня, на странице рекомендуется использовать только один раз, по возможности частично дублируя заглавие страницы. Элемент `<h1>` должен быть уникальным для каждой страницы сайта.

Рекомендуется прописывать в начале статьи, используя ключевое слово в тексте заголовка. Размер шрифта в браузере равен `2em`, верхний и нижний отступ по умолчанию `0.67em`.

<code>&lt;h1&gt;Заголовок 1-го уровня&lt;/h1&gt;</code>	Заголовок 1-го уровня
<code>&lt;h2&gt;Заголовок 2-го уровня&lt;/h2&gt;</code>	Заголовок 2-го уровня
<code>&lt;h3&gt;Заголовок 3-го уровня&lt;/h3&gt;</code>	Заголовок 3-го уровня
<code>&lt;h4&gt;Заголовок 4-го уровня&lt;/h4&gt;</code>	Заголовок 4-го уровня
<code>&lt;h5&gt;Заголовок 5-го уровня&lt;/h5&gt;</code>	Заголовок 5-го уровня
<code>&lt;h6&gt;Заголовок 6-го уровня&lt;/h6&gt;</code>	Заголовок 6-го уровня

### Элемент `<h2>`

Представляет подзаголовки элемента `<h1>`. Размер шрифта в браузере равен `1.5em`, верхний и нижний отступ по умолчанию `0.83em`.

### Элемент `<h3>`

Показывает подзаголовки элемента `<h2>`. Размер шрифта в браузере равен `1.17em`, верхний и нижний отступ по умолчанию `1em`.

### Элементы <h4>, <h5>, <h6>

Обозначают подзаголовки четвертого, пятого и шестого уровня. Размер шрифта в браузере равен 1em / 0.83em / 0.67em, верхний и нижний отступ по умолчанию 1.33em / 1.67em / 2.33em соответственно.

Для всех элементов доступны [глобальные атрибуты](#).

### Элементы для форматирования текста

#### Элемент <b>

Задаёт полужирное начертания шрифта. Выделяет текст без акцента на его важность.

Для элемента доступны [глобальные атрибуты](#).

#### Элемент <em>

Отображает шрифт курсивом, придавая тексту значимость.

Для элемента доступны [глобальные атрибуты](#).

#### Элемент <i>

Отображает шрифт курсивом.

Для элемента доступны [глобальные атрибуты](#).

#### Элемент <small>

Уменьшает размер шрифта на единицу по отношению к обычному тексту.

Для элемента доступны [глобальные атрибуты](#).

#### Элемент <strong>

Задаёт полужирное начертание шрифта, относится к тегам логической разметки, указывая браузеру на важность текста.

Для элемента доступны [глобальные атрибуты](#).

#### Элемент <sub>

Используется для создания нижних индексов. Сдвигает текст ниже уровня строки, уменьшая его размер.

Для элемента доступны [глобальные атрибуты](#).

#### Элемент <sup>

Используется для создания степеней. Сдвигает текст выше уровня строки, уменьшая его размер.

Для элемента доступны [глобальные атрибуты](#).

#### Элемент <ins>

Выделяет текст в новой версии документа, подчеркивая его.

Для элемента доступны атрибуты `cite` и `datetime`.

#### Элемент <del>

Перечеркивает текст. Используется для выделения текста, удаленного из документа.

Для элемента доступны атрибуты `cite` и `datetime`.

#### Элемент <mark>

Применяется для выделения фрагментов текста в справочных целях, окрашивая блок символов желтым цветом.

Для элемента доступны [глобальные атрибуты](#).

### Элементы для ввода «компьютерного» текста

#### Элемент `<code>`

Служит для выделения фрагментов программного кода. Отображает текст моноширинным шрифтом.

Для элемента доступны [глобальные атрибуты](#).

#### Элемент `<kbd>`

Отмечает фрагмент как вводимый пользователем с клавиатуры. Отображает текст моноширинным шрифтом.

Для элемента доступны [глобальные атрибуты](#).

#### Элемент `<samp>`

Применяется для выделения результата, полученного в ходе выполнения программы. Отображает текст моноширинным шрифтом.

Для элемента доступны [глобальные атрибуты](#).

#### Элемент `<var>`

Выделяет имена переменных, отображая текст курсивом.

Для элемента доступны [глобальные атрибуты](#).

#### Элемент `<pre>`

Позволяет вывести текст на экран, сохранив изначальное форматирование. Пробелы и переносы строк при этом не удаляются.

Для элемента доступны [глобальные атрибуты](#).

### Элементы для оформления цитат и определений

#### Элемент `<abbr>`

Применяется для форматирования аббревиатур. Браузером обычно подчеркивается пунктирной линией. Расшифровка сокращения осуществляется с помощью атрибута `title`, она появляется при наведении курсора мыши на текст.

Для элемента доступны [глобальные атрибуты](#).

#### Элемент `<bdo>`

Используется для изменения текущего направления текста.

Для элемента доступен атрибут `dir`.

#### Элемент `<blockquote>`

Выделяет цитаты внутри документа, выделяя его отступами и переносами строк.

Для элемента доступен атрибут `cite`.

#### Элемент `<q>`

Используется для выделения коротких цитат. Браузерами заключается в кавычки.

Для элемента доступен атрибут `cite`.

#### Элемент `<cite>`

Применяется для выделения цитат, названий произведений, сносок на другие документы.

Для элемента доступны [глобальные атрибуты](#).

### Элемент <dfn>

Позволяет выделить текст как определение. Несмотря на наличие данного элемента, рекомендуется выделять текст силами CSS.

Для элемента доступен атрибут `title`.

## Абзацы, средства переноса текста

### Элемент <p>

Разбивает текст на отдельные абзацы, отделяя друг от друга пустой строкой. Браузер автоматически добавляет верхнее и нижнее внешнее поле `margin`, равное `1em`, при этом поля соседних абзацев «схлопываются».

Для элемента доступны [глобальные атрибуты](#).

### Элемент <br>

Переносит текст на следующую строку, создавая разрыв строки.

Для элемента доступны [глобальные атрибуты](#).

### Элемент <hr>

Используется для разделения контента на веб-странице. Отображается в виде горизонтальной линии.

Для элемента доступны [глобальные атрибуты](#).

## HTML-ссылки

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.

## Структура ссылки

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

Параметром элемента `<a>` является атрибут `href`, который задает URI-адрес веб-страницы.

```
<a href="http://bsuir.by">указатель ссылки</a>
```

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 подчеркивания. В общем виде URL имеющий следующий формат:

```
метод доступа://имя сервера:порт/путь
```



**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
```

**http** предоставляет доступ к веб-странице по протоколу HTTP:

```
http://bsuir.by/
```

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

```
https://bsuir.by/
```

**ftp** осуществляет запрос к FTP-серверу на получение файла:

```
ftp://pgu/directory/library
```

**mailto** запускает сеанс почтовой связи с указанным адресатом и хостом:

```
mailto: makeichik@bsuir.by
```

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

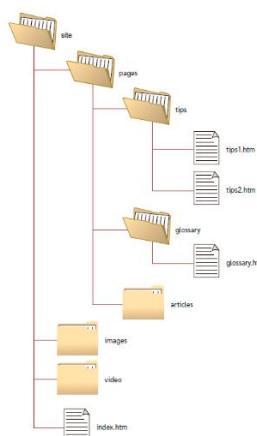
Номер порта TCP, на котором функционирует веб-сервер. Представляет собой число, которое необходимо указывать, если метод требует номер порта (отдельные сервера могут иметь свой отличительный номер порта). Если порт не указан, по умолчанию используется порт 80. Стандартными портами являются:

- 21 – FTP
- 23 – Telnet
- 70 – Gopher
- 80 – HTTP

Путь содержит имя папки, в которой находится файл.

### 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.



## Абсолютный путь

Абсолютный путь указывает точное местоположение файла в пределах всей структуры папок на компьютере (сервере). Абсолютный путь к файлу дает 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://bsuir.by/pages/tips/tips1.html  
//bsuir.by/pages/tips/tips1.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://bsuir.by/index.html
```

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

```
https://bsuir.by/
```

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.

## Относительный путь

Относительный путь описывает путь к указанному документу относительно текущего. Путь определяется с учетом 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.

## 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>
```

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://bsuir.by/css-shrifty/#about-color" class="site" target="_blank">color</a>
```

### 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://keksby.ru" target="_blank"></a>
```

### 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:+375291234567">+375 (29) 123-45-67</a>

ссылка на адрес электронной почты
<a href="mailto:makeichik@bsuir.by">makeichik@bsuir.by</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>
```

### Атрибуты ссылок

Элемент `<a>` поддерживает глобальные атрибуты и собственные.

Атрибут	Описание, принимаемое значение
download	Дополняет атрибут <code>href</code> и сообщает браузеру, что ресурс должен быть загружен в момент, когда пользователь щелкает по ссылке вместо того, чтобы, например, предварительно

	<p>открыть его (как PDF-файл). Задавая имя для атрибута, мы таким образом задаем имя загружаемому объекту. Разрешается использовать атрибут без указания его значения:</p> <pre>&lt;a href="/images/logo.png" download&gt;&lt;img src="/images/logo.png" alt="logo"&gt;&lt;/a&gt;, &lt;a href="/images/logo.png" download="logo"&gt;&lt;img src="/images/logo.png" alt="logo"&gt;&lt;/a&gt;, &lt;a href="files/20022021.pdf" download="Отчет Февраль 2021.pdf"&gt;Загрузить отчет за Февраль 2021&lt;/a&gt;.</pre>
href	Значением атрибута является URL-адрес документа, на который указывается ссылка.
hreflang	<p>Определяет язык связанного веб-документа. Используется только вместе с атрибутом href. Принимаемые значения – аббревиатура, состоящая из пары букв, обозначающих код языка, например:</p> <pre>&lt;a href="http://www.anysite.ru" hreflang="en"&gt;Anysite&lt;/a&gt;.</pre>
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	<p>Дополняет атрибут href информацией об отношении между текущим и связанным документом. Принимаемые значения:</p> <ul style="list-style-type: none"> <li>alternate – ссылка на альтернативную версию документа (например, печатную форму страницы, перевод или зеркало).</li> <li>author – ссылка на автора документа.</li> <li>bookmark – постоянный URL-адрес, используемый для закладок.</li> <li>help – ссылка на справку.</li> <li>license – ссылка на информацию об авторских правах на данный веб-документ.</li> <li>next/prev – указывает связь между отдельными URL. Благодаря этой разметке Google может определить, что содержание данных страниц связано в логической последовательности.</li> <li>nofollow – запрещает поисковой системе переходить по ссылкам на данной странице или по конкретной ссылке.</li> <li>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.</li> <li>prefetch – указывает, что целевой документ должен быть кэширован, т.е. браузер в фоновом режиме загружает содержимое страницы к себе в кэш.</li> <li>search – указывает, что целевой документ содержит инструмент для поиска.</li> <li>tag – указывает ключевое слово для текущего документа.</li> </ul>
target	<p>Указывает на то, в каком окне должен открываться документ, к которому ведет ссылка. Принимает следующие значения:</p> <ul style="list-style-type: none"> <li>_self – страница загружается в текущее окно;</li> <li>_blank – страница открывается в новом окне браузера;</li> <li>_parent – страница загружается во фрейм-родитель;</li> <li>_top – страница загружается в полное окно браузера.</li> </ul>

type	Указывает MIME-тип файлов ссылки, т.е. расширение файла, носит больше справочную информацию.
------	--

## HTML-изображения

HTML-изображения добавляются на веб-страницы с помощью элемента `<img>`. Использование графики делает веб-страницы визуально привлекательнее. Изображения помогают лучше передать суть и содержание веб-документа.

Элементы `<map>` и `<area>` позволяют создавать карты-изображения с активными областями.

### Элемент `<img>`

Элемент `<img>` представляет изображение и его резервный контент, который добавляется с помощью атрибута `alt`. Так как элемент `<img>` является строчным, то рекомендуется располагать его внутри блочного элемента, например, `<p>` или `<div>`.

Элемент `<img>` имеет обязательный атрибут `src`, значением которого является абсолютный или относительный путь к изображению:

```

```

Для элемента `<img>` доступны следующие атрибуты:

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 дает учетные данные серверу, с которого запрашивается контент, то изображение будет испорчено и его использование будет ограничено. <code>use-credentials</code> – Cross-origin запрос выполняется с передачей учетных данных. Синтаксис: <code>crossorigin="anonymous"</code> .
height	Атрибут <code>height</code> задает высоту изображения в px. Синтаксис: <code>height="300"</code> .
ismap	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>&lt;img&gt;</code> является потомком элемента <code>&lt;a&gt;</code> с действительным атрибутом <code>href</code> . Синтаксис: <code>ismap</code> .

longdesc	URL расширенного описания изображения, дополняющее атрибут alt. Синтаксис: longdesc="http://www.example.com/description.txt".
src	Атрибут src задает путь к изображению. Синтаксис: src="flower.jpg".
sizes	Задаёт размер изображения в зависимости от параметров отображения. Работает только при заданном атрибуте srcset. Значением атрибута является одна или несколько строк, указанных через запятую.
srcset	Создаёт список источников для изображения, которые будут выбраны, исходя из разрешения экрана. Может использоваться вместе или вместо атрибута src. Значением атрибута является одна или несколько строк, разделённых запятой. <pre>&lt;img src="flower.jpg" srcset=" img/flower-mobile.jpg 320w, img/flower-wide-mobile.jpg 480w, img/flower-tablet.jpg 768w, img/flower-desktop.jpg 1024w, img/flower-hires.jpg 1280w" sizes=" (max-width: 20em) 30vw, (max-width: 30em) 60vw, (max-width: 40em) 90vw" alt="Роза в моём саду"&gt;</pre>
usemap	Атрибут usemap определяет изображение в качестве карты-изображения. Значение обязательно должно начинаться с символа #. Значение ассоциируется со значением атрибута name или id элемента <map> и создаёт связь между элементами <img> и <map>. Атрибут нельзя использовать, если элемент <map> является потомком элемента <a> или <button>. Синтаксис: usemap="#mymap".
width	Атрибут width задает ширину изображения в px. Синтаксис: width="500".

### Адрес изображения

Адрес изображения может быть указан полностью (абсолютный URL), например: url(http://anysite.ru/images/anyphoto.png).

Или же через относительный путь от документа или корневого каталога сайта:

- url(../images/anyphoto.png) – относительный путь от документа,
- url(/images/anyphoto.png) – относительный путь от корневого каталога.

Это интерпретируется следующим образом:

- ../ – означает подняться вверх на один уровень, к корневому каталогу,
- images/ – перейти к папке с изображениями,
- anyphoto.png – указывает на файл изображения.

### Размеры изображения

Без задания размеров изображения рисунок отображается на странице в реальном размере. Отредактировать размеры изображения можно с помощью атрибутов width и height. Если будет задан только один из атрибутов, то второй будет вычисляться автоматически для сохранения пропорций рисунка.

## Форматы графических файлов

**Формат JPEG (Joint Photographic Experts Group).** Изображения JPEG идеальны для фотографий, они могут содержать миллионы различных цветов. Сжимают изображения лучше GIF, но текст и большие площади со сплошным цветом могут покрываться пятнами.

**Формат GIF (Graphics Interchange Format).** Идеален для сжатия изображений, в которых есть области со сплошным цветом, например, логотипов. GIF-файлы позволяют установить один из цветов прозрачным, благодаря чему фон веб-страницы может проявляться через часть изображения. Также GIF-файлы могут включать в себя простую анимацию. GIF-изображения содержат всего лишь 256 оттенков, из-за чего изображения выглядят пятнистыми и нереалистичного цвета, как плакаты.

**Формат PNG (Portable Network Graphics).** Включает в себя лучшие черты GIF- и JPEG-форматов. Содержит 256 цветов и дает возможность сделать один из цветов прозрачным, при этом сжимает изображения в меньший размер, чем GIF-файл.

**Формат APNG (Animated Portable Network Graphics).** Формат изображения, основанный на формате PNG. Позволяет хранить анимацию, а также поддерживает прозрачность.

**SVG (Scalable Vector Graphics).** SVG-рисунок состоит из набора геометрических фигур, описанных в формате XML: линия, эллипс, многоугольник и т.п. Поддерживается как статичная, так и анимированная графика. Набор функций включает в себя различные преобразования, альфа-маски, эффекты фильтров, возможность использования шаблонов. Изображения в формате SVG могут изменяться в размере без снижения качества.

**Формат BMP (Bitmap Picture).** Представляет собой несжатое (оригинальное) растровое изображение, шаблоном которого является прямоугольная сетка пикселей. Bitmap-файл состоит из заголовка, палитры и графических данных. В заголовке хранится информация о графическом изображении и файле (глубина пикселей, высота, ширина и количество цветов). Палитра указывается только в тех Bitmap-файлах, которые содержат палитровые изображения (8 и менее бит) и состоят не более чем из 256 элементов. Графические данные представляют саму картинку. Глубина цвета в данном формате может быть 1, 2, 4, 8, 16, 24, 32, 48 бит на пиксель.

**Формат WebP (WebP Image).** Формат файла, разработанный компанией Google в 2010 году. Его особенностью является продвинутый алгоритм сжатия, позволяющий сократить размер изображения без видимых потерь в качестве. В среднем вес изображения сокращается на 25–35%.

**Формат ICO (Windows icon).** Формат хранения значков файлов в Microsoft Windows. Также, Windows icon, используется как иконка на сайтах в интернете. Именно картинка такого формата отображается рядом с адресом сайта или закладкой в браузере. Один ICO-файл содержит один или несколько значков, размер и цветность каждого из которых задается отдельно. Размер значка может быть любым, но наиболее употребимы квадратные значки со сторонами 16, 32 и 48 пикселей.

## Элемент <map>

Элемент `<map>` служит для представления графического изображения в виде карты с активными областями. Активные области определяются по изменению вида курсора мыши при наведении. Щелкая мышью на активных областях, пользователь может переходить к связанным документам.



Для элемента доступен атрибут `name`, который задает имя карты. Значение атрибута `name` для элемента `<map>` должно соответствовать имени в атрибуте `usemap` элемента `<img>`:

```

<map name="имя_карты">
...
</map>
```

Элемент `<map>` содержит ряд элементов `<area>`, определяющих интерактивные области в изображении карты.

### Элемент `<area>`

Элемент `<area>` описывает только одну активную область в составе карты изображений на стороне клиента.

Если одна активная область перекрывает другую, то будет реализована первая ссылка из списка областей.

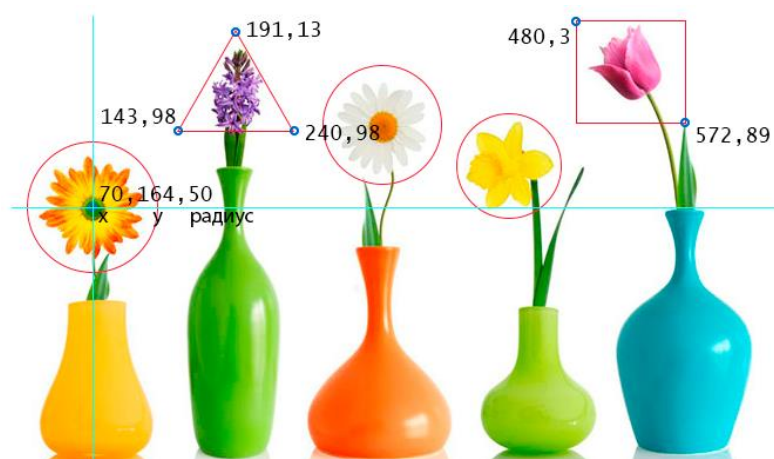
```
<map name="имя_карты">
<area атрибуты>
</map>
```

Атрибут	Описание, принимаемое значение
<code>alt</code>	Задаёт альтернативный текст для активной области карты.
<code>coords</code>	Определяет позицию области на экране. Координаты задаются в единицах длины и разделяются запятыми: для <b>круга</b> – координаты центра и радиус круга; для <b>прямоугольника</b> – координаты верхнего левого и правого нижнего углов; для <b>многоугольника</b> – координаты вершин многоугольника в нужном порядке, также рекомендуется указывать последние координаты, равные первым, для логического завершения фигуры.
<code>download</code>	Дополняет атрибут <code>href</code> 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.
<code>href</code>	Указывает URL-адрес для ссылки. Может быть указан абсолютный или относительный адрес ссылки.
<code>hreflang</code>	Определяет язык связанного веб-документа. Используется только вместе с атрибутом <code>href</code> . Принимаемые значения – аббревиатура, состоящая из пары букв, обозначающих код языка.
<code>media</code>	Определяет, для каких типов устройств файл будет оптимизирован. Значением может быть любой медиа-запрос.
<code>rel</code>	Дополняет атрибут <code>href</code> информацией об отношении между текущим и связанным документом. Принимаемые значения: <code>alternate</code> – ссылка на альтернативную версию документа (например, печатную форму страницы, перевод или зеркало). <code>author</code> – ссылка на автора документа. <code>bookmark</code> – постоянный URL-адрес, используемый для закладок.

	<p><code>help</code> – ссылка на справку.</p> <p><code>license</code> – ссылка на информацию об авторских правах на данный веб-документ.</p> <p><code>next/prev</code> – указывает связь между отдельными URL. Благодаря этой разметке Google может определить, что содержание данных страниц связано в логической последовательности.</p> <p><code>nofollow</code> – запрещает поисковой системе переходить по ссылкам на данной странице или по конкретной ссылке.</p> <p><code>noreferrer</code> – 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.</p> <p><code>prefetch</code> – указывает, что целевой документ должен быть кэширован, т.е. браузер в фоновом режиме загружает содержимое страницы к себе в кэш.</p> <p><code>search</code> – указывает, что целевой документ содержит инструмент для поиска.</p> <p><code>tag</code> – указывает ключевое слово для текущего документа.</p>
shape	<p>Задаёт форму активной области на карте и её координаты. Может принимать следующие значения:</p> <p><code>rect</code> – активная область прямоугольной формы;</p> <p><code>circle</code> – активная область в форме круга;</p> <p><code>poly</code> – активная область в форме многоугольника;</p> <p><code>default</code> – активная область занимает всю площадь изображения.</p>
target	<p>Указывает, куда будет загружен документ при переходе по ссылке. Принимает следующие значения:</p> <p><code>_self</code> – страница загружается в текущее окно;</p> <p><code>_blank</code> – страница открывается в новом окне браузера;</p> <p><code>_parent</code> – страница загружается во фрейм-родитель;</p> <p><code>_top</code> – страница загружается в полное окно браузера.</p>
type	<p>Указывает MIME-тип файлов ссылки, т.е. расширение файла.</p>

### Пример создания карты-изображения

1. Размечаем исходное изображение на активные области нужной формы. Координаты областей можно вычислить с помощью программы для обработки фотографий, например, **Adobe Photoshop** или **Paint**.



2. Задаём имя карты, добавив её в элемент `<map>` с помощью атрибута `name`. Это же значение присваиваем атрибуту `usemap` элемента `<img>`.

3. Добавляем ссылки на веб-страницы или части веб-документа для каждой активной области, по которым пользователь будет переходить при нажатии курсором мыши на активную область изображения.

```

<map name="flowers">
  <area shape="circle" coords="70,164,50" href="https://ru.wikipedia.org/wiki/Гербера" alt="gerbera" target="_blank">
  <area shape="poly" coords="191,13,240,98,143,98,191,13" href="https://ru.wikipedia.org/wiki/%C3%E8%E0%F6%E8%E0%F2" alt="hyacinth" target="_blank">
  <area shape="circle" coords="318,93,50" href="https://ru.wikipedia.org/wiki/Ромашка" alt="camomiles" target="_blank">
  <area shape="circle" coords="425,129,45" href="https://ru.wikipedia.org/wiki/Нарцисс_(растение)" alt="narcissus" target="_blank">
  <area shape="rect" coords="480,3,572,89" href="https://ru.wikipedia.org/wiki/Тюльпан" alt="tulip" target="_blank">
</map>
```

## HTML-таблицы

**HTML-таблицы** упорядочивают и выводят на экран данные с помощью строк или столбцов. Таблицы состоят из ячеек, образующихся при пересечении строк и столбцов.

**Ячейки таблиц** могут содержать любые HTML-элементы, такие как заголовки, списки, текст, изображения, элементы форм, а также другие таблицы. Каждой таблице можно добавить связанный с ней заголовок, расположив его перед таблицей или после нее.

Таблицы больше не используются для верстки веб-страниц и компоновки отдельных элементов, потому что такой прием не обеспечивает гибкость структуры и адаптивность сайта, существенно увеличивая HTML-разметку.

Для всех элементов таблицы доступны [глобальные атрибуты](#), а также собственные атрибуты.

## Как создать таблицу

Таблица создается при помощи элемента `<table>...</table>`, который является контейнером для элементов таблицы и все элементы должны находиться внутри него.

Например, с помощью данной разметки можно создать таблицу, состоящую из двух столбцов и двух строк:

```
<table>
<tr><th>текст заголовка</th><th>текст заголовка</th></tr> <!--ряд с ячейками заголовков-->
<tr><td>данные</td><td>данные</td></tr> <!--ряд с ячейками тела таблицы-->
</table>
```

текст заголовка	текст заголовка
данные	данные

По умолчанию таблица и ячейки не имеют видимых границ. Границы задаются с помощью свойства `border`:

```
table {
  border: 1px solid grey;
}
/* границы ячеек первого ряда таблицы */
th {
  border: 1px solid grey;
}
/* границы ячеек тела таблицы */
td {
  border: 1px solid grey;
}
```

Промежутки между ячейками таблицы убираются с помощью свойства `table {border-collapse: collapse;}`.

Ширина таблицы по умолчанию равна ширине ее внутреннего содержимого. Чтобы установить ширину, нужно задать значение для свойства `width`:

```

/* сделает ширину таблицы равной ширине блока контейнера, в котором она находится */
table {
    width: 100%;
}
/* задаст фиксированную ширину для таблицы */
table {
    width: 600px;
}

```

Если для ячеек таблицы заданы внутренние отступы и границы, то ширина таблицы будет включать в себя следующие значения: `padding-left` и `padding-right`, ширина `border-left` плюс ширина `border-right` последней ячейки в ряду.

Если заданы ширина и границы ячеек, то ширина таблицы будет складываться из ширины ячеек плюс ширина `border-left` и ширина `border-right` последней ячейки в ряду.

### Как создать строки (ряды) таблицы

Элемент `<th>` создает заголовок столбца – специальную ячейку, текст в которой выделяется полужирным. Количество ячеек заголовка определяется количеством элементов `<th>... </th>`. Для элемента доступны атрибуты `colspan`, `rowspan`, `headers`.

### Как сделать ячейку заголовка столбца таблицы

Элемент `<th>` создает заголовок столбца – специальную ячейку, текст в которой выделяется полужирным. Количество ячеек заголовка определяется количеством элементов `<th>... </th>`. Для элемента доступны атрибуты `colspan`, `rowspan`, `headers`.

```

<table>
<tr><th>ячейка заголовка</th><th>ячейка заголовка</th></tr>
</table>

```

### Как сделать ячейку тела таблицы

Элемент `<td>` создает ячейки таблицы, внутрь которых помещаются данные таблицы. Элементы `<td>... </td>`, расположенные в одном ряду, определяют количество ячеек в строке таблицы.

Количество пар ячеек `<td>` должно быть равно количеству пар ячеек `<th>`. Для элемента доступны атрибуты `colspan`, `rowspan`, `headers`.

```

<table>
<tr><th>ячейка заголовка</th><th>ячейка заголовка</th></tr>
<tr><td>ячейка тела таблицы</td><td>ячейка тела таблицы</td></tr>
</table>

```

### Как добавить подпись (заголовок) к таблице

Элемент `<caption>` создает подпись таблицы. Добавляется непосредственно после тега `<table>`, вне строки или ячейки.

```

<table>
<caption>Перечень продуктов</caption>
<tr>
<th>№ п/п</th>
<th>Наименование товара</th>
<th>Ед. изм.</th>
<th>Количество</th>
<th>Цена за ед. изм., руб.</th>
<th>Стоимость, руб.</th>
</tr>
<tr>
<td>1.</td>
<td>Томаты свежие</td>
<td>шт</td>
<td>20</td>
<td>0,50</td>
<td>10,00</td>
</tr>
<tr>
<td>2.</td>
<td>Огурцы свежие</td>
<td>шт</td>
<td>10</td>
<td>0,38</td>
<td>3,80</td>
</tr>
<tr>
<td colspan="5">ИТОГО:</td>
<td>13,80</td>
</tr>
</table>

```

## Группирование строк и столбцов таблицы

Элемент `<colgroup>` создает структурную группу столбцов, выделяя логически однородные ячейки. Группирует один или более столбцов для единого форматирования, позволяя применить стили к столбцам вместо того, чтобы повторять стили для каждой ячейки и для каждой строки.

Добавляется непосредственно после тегов `<table>` и/или `<caption>`.

Элемент `<col>` формирует группы столбцов, которые делят таблицу на разделы, не относящиеся к общей структуре, т.е. не содержащие информацию одного типа. Позволяет задавать свойства столбцов для каждого столбца в пределах элемента `<colgroup>`.

С помощью атрибута `style` можно изменить основной цвет фона ячеек. Для элемента `<col>` доступен атрибут `span`, задающий количество столбцов для объединения.

```
<table>
  <colgroup>
    <col span="2" style="background:Khaki"><!-- С помощью этой конструкции задаем цвет фона для первых двух столбцов таблицы-->
    <col style="background-color:LightCyan"><!-- Задаем цвет фона для следующего (одного) столбца таблицы-->
  </colgroup>
  <tr>
    <th>№</th>
    <th>Наименование</th>
    <th>Цена, руб.</th>
  </tr>
  <tr>
    <td>1</td>
    <td>Карандаш цветной</td>
    <td>2,00</td>
  </tr>
  <tr>
    <td>2</td>
    <td>Линейка 20 см</td>
    <td>3,00</td>
  </tr>
</table>
```

№	Наименование	Цена, руб.
1	Карандаш цветной	2,00
2	Линейка 20 см	3,00

## Группировка разделов таблицы

Элемент `<thead>` создает группу заголовков для строк таблицы с целью задания единого оформления. Используется в сочетании с элементами `<tbody>` и `<tfoot>` для указания каждой части таблицы.

Элемент должен быть использован в следующем порядке: как дочерний элемент `<table>`, после `<caption>` и `<colgroup>`, и перед `<tbody>`, `<tfoot>` и `<tr>` элементами. В пределах одной таблицы можно использовать один раз.

Элемент `<tbody>` группирует основное содержимое таблицы. Используется в сочетании с элементами `<thead>` и `<tfoot>`.

Элемент `<tfoot>` создает группу строк для представления информации о суммах или итогах, расположенную в нижней части таблицы. Используется в таблице один раз. Располагается после элемента `<thead>`, перед элементами `<tbody>` и `<tr>`.

```

<table>
<thead>
<tr>
<th>№ п/п</th>
<th>Наименование товара</th>
<th>Ед. изм.</th>
<th>Количество</th>
<th>Цена за ед. изм., руб.</th>
<th>Стоимость, руб.</th>
</tr>
</thead>
<tfoot>
<tr>
<td colspan="5" style="text-align:right">ИТОГО:</td><td>24,80</td>
</tr>
</tfoot>
<tbody>
<tr>
<td>1.</td>
<td>Томаты свежие</td><td>кг</td><td>2,00</td><td>5,20</td><td>10,40</td>
</tr>
<tr>
<td>2.</td>
<td>Огурцы свежие</td><td>кг</td><td>3,00</td><td>4,80</td><td>14,40</td>
</tr>
</tbody>
</table>

```

### Как объединить ячейки таблицы

Атрибуты `colspan` и `rowspan` объединяют ячейки таблицы. Атрибут `colspan` задает количество ячеек, объединенных по горизонтали, а `rowspan` – по вертикали.

```

<table>
<colgroup>
<col span="5" style="background:Khaki"><!-- С помощью этой конструкции задаем цвет фона для первых двух столбцов таблицы-->
<col style="background-color:LightCyan"><!-- Задаем цвет фона для следующего (одного) столбца таблицы-->
</colgroup>
<tr>
<th>№ п/п</th>
<th>Наименование товара</th>
<th>Ед. изм.</th>
<th>Количество</th>
<th>Цена за ед. изм., руб.</th>
<th>Стоимость, руб.</th>
</tr>
<tr>
<td>1.</td>
<td>Томаты свежие</td><td>кг</td><td>2,00</td><td>5,20</td><td>10,40</td>
</tr>
<tr>
<td>2.</td>
<td>Огурцы свежие</td><td>кг</td><td>3,00</td><td>4,80</td><td>14,40</td>
</tr>
<tr>
<td colspan="5" style="text-align:right">ИТОГО:</td><td>24,80</td><!-- Задаем количество ячеек по горизонтали для объединения-->
</tr>
</table>

```

№ п/п	Наименование товара	Ед. изм.	Количество	Цена за ед. изм., руб.	Стоимость, руб.
1.	Томаты свежие	кг	2,00	5,20	10,40
2.	Огурцы свежие	кг	3,00	4,80	14,40
ИТОГО:					24,80

### Атрибуты элементов таблицы

Атрибут	Описание, принимаемое значение
<code>colspan</code>	Количество ячеек в строке для объединения по горизонтали. <code>&lt;td colspan="3"&gt;</code> Возможные значения: число от 1 до 999.
<code>headers</code>	Задаёт список ячеек заголовка, содержащих информацию о заголовке текущей ячейки данных. Предназначен для речевых браузеров. <code>&lt;th id="идентификатор"&gt;...&lt;/th&gt;</code> <code>&lt;th headers="идентификатор"&gt;...&lt;/th&gt;</code>

	Принимаемые значения: список имен ячеек, разделенных пробелами; эти имена должны быть присвоены ячейкам через их атрибут <code>id</code> .
<code>rowspan</code>	Количество ячеек в столбце для объединения по вертикали. <code>&lt;td rowspan="2"&gt;</code> Возможные значения: число от 1 до 999.
<code>span</code>	Количество колонок, объединяемых для задания единого стиля, по умолчанию равно 1. <code>&lt;col span="2"&gt;</code> Принимаемые значения: любое целое положительное число.

## HTML-списки

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 описаний – `<dl>` – служит контейнером для списка пар терминов (определяемых элементом `<dt>`) и их описаний (определяемых элементами `<dd>`).

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.

## Маркированный список

Маркированный список представляет собой неупорядоченный список (от англ. Unordered List). Создается 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;
}
```

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

<pre>&lt;ul&gt;   &lt;li&gt;Microsoft&lt;/li&gt;   &lt;li&gt;Google&lt;/li&gt;   &lt;li&gt;Apple&lt;/li&gt;   &lt;li&gt;IBM&lt;/li&gt; &lt;/ul&gt;</pre>	<ul style="list-style-type: none"> <li>• Microsoft</li> <li>• Google</li> <li>• Apple</li> <li>• IBM</li> </ul>
--	---

Для элемента `<ul>` доступны [глобальные атрибуты](#).

## Нумерованный список

Нумерованный список создается с помощью элемента `<ol>`. Каждый пункт списка также создается 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;
}
```

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:

Атрибут	Описание, принимаемое значение
reversed	Атрибут <code>reversed</code> задает отображение списка в обратном порядке (например, 9, 8, 7...).
start	Атрибут <code>start</code> задает начальное значение, от которого пойдет отсчет нумерации, например, конструкция <code>&lt;ol start="10"&gt;</code> первому пункту присвоит порядковый номер «10». Также можно одновременно задавать тип нумерации, например, <code>&lt;ol type="I" start="10"&gt;</code> .
type	Атрибут <code>type</code> 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>
```

1. Microsoft
2. Google
3. Apple
4. IBM

### Список описаний

Списки описаний создаются с помощью элемента `<dl>`. Для добавления термина применяется элемент `<dt>`, а для вставки определения – элемент `<dd>`.

Блок списка описаний имеет следующие стили браузера по умолчанию:

```
dl {
  margin-top: 1em;
  margin-bottom: 1em;
}
```

Для элементов `<dl>`, `<dt>` и `<dd>` доступны [глобальные атрибуты](#).

```
<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>
```

Режиссер:  
Петр Точилин  
В ролях:  
Андрей Гайдулян  
Алексей Гаврилов  
Виталий Гогунский  
Мария Кожевникова

### 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>
```

- Пункт 1.
- Пункт 2.
  - Подпункт 2.1.
  - Подпункт 2.2.
    - Подпункт 2.2.1.
    - Подпункт 2.2.2.
  - Подпункт 2.3.
- Пункт 3.

### 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>
```

1. пункт
2. пункт
  - 2.1. пункт
  - 2.2. пункт
  - 2.3. пункт
    - 2.3.1. пункт
    - 2.3.2. пункт
    - 2.3.3. пункт
  - 2.4. пункт
3. пункт
4. пункт

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 счетчиков, задавая значение для сброса;
- `counter-increment` задает значение приращения счетчика, 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, вставленное 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, ".") ". ";
}

```

## Спецсимволы HTML

**Спецсимволы HTML**, или символы-мнемоники, представляют собой конструкцию SGML (англ. **Standard Generalized Markup Language** – стандартный обобщенный язык разметки), ссылающуюся на определенные символы из символьного набора документа. В основном они используются для указания символов, которых нет в стандартной компьютерной клавиатуре, либо которые не поддерживает кодировка HTML-страницы (Windows-1251, UTF-8 и т.д.).

Чтобы разместить символ на веб-странице, необходимо указать HTML-код или мнемонику. [Популярные спецсимволы HTML](#).

Спецсимволы чувствительны к регистру, поэтому их необходимо прописывать точно так, как указано в таблице. Спецсимволы, не имеющие мнемоники, могут не отображаться вовсе или же некорректно отображаться в тех или иных браузерах.

Для вставки символов внутрь тегов воспользуйтесь HTML-кодом символа, а для использования символов в таблицах стилей, например, в качестве значения свойства `content` – CSS-код.

Спецсимвол наследует цвет от цвета текста родительского элемента. Чтобы изменить цвет спецсимвола, можно поместить HTML-код внутрь тега `<span>` или задать нужное значение свойства `color` (при вставке спецсимволов через свойство `content`).

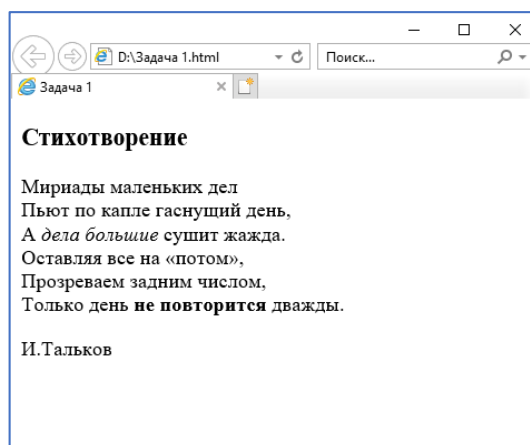
## Задание к лабораторной работе №1

Для выполнения лабораторной работы необходимо установить и настроить редактор кода.

Задание к лабораторной работе №1 состоит из задач разного уровня сложности. **Выполнение задач 1-6 оценивается максимально в 5 баллов, задач 1-12 в 10 баллов.**

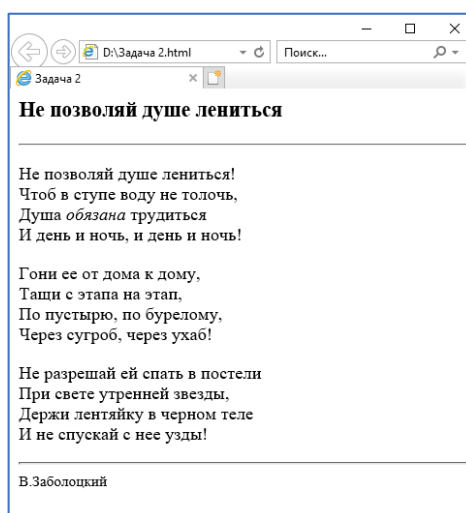
### Задача 1

**Условие:** создайте html-файл, результат которого показан на рисунке. Используйте подходящий элемент заголовков (<h1>...<h6>), теги <br>, <p>, <i>, <em>, <strong>, <b> (необязательно все перечисленные). Выясните различие тегов <b> и <strong>, тегов <i> и <em>. В html-код добавьте комментарии: дату решения данной задачи и ФИО.



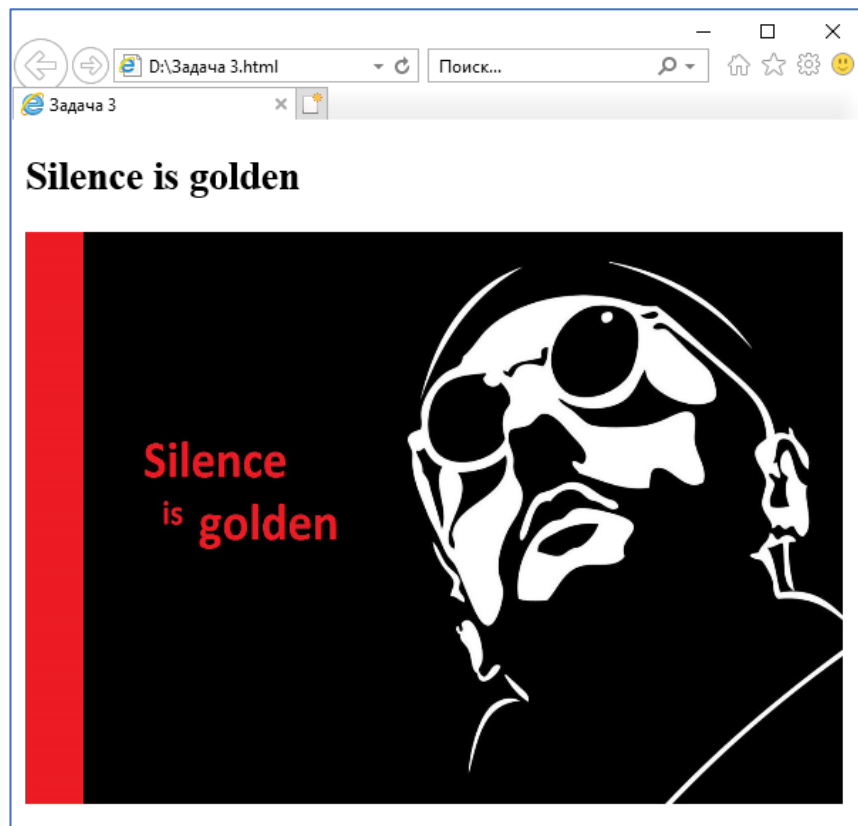
### Задача 2

**Условие:** создайте html-файл, результат которого показан на рисунке. Используйте подходящий элемент заголовков (<h1>...<h6>), теги <br>, <p>, <i>, <em>, <strong>, <b>, <hr>, <small> (необязательно все перечисленные). Обратите внимание, что шрифт фамилии автора меньшего размера. Заметим, что тег <hr> в разных браузерах может отображаться по-разному. В html-код добавьте условие данной задачи в виде комментариев.



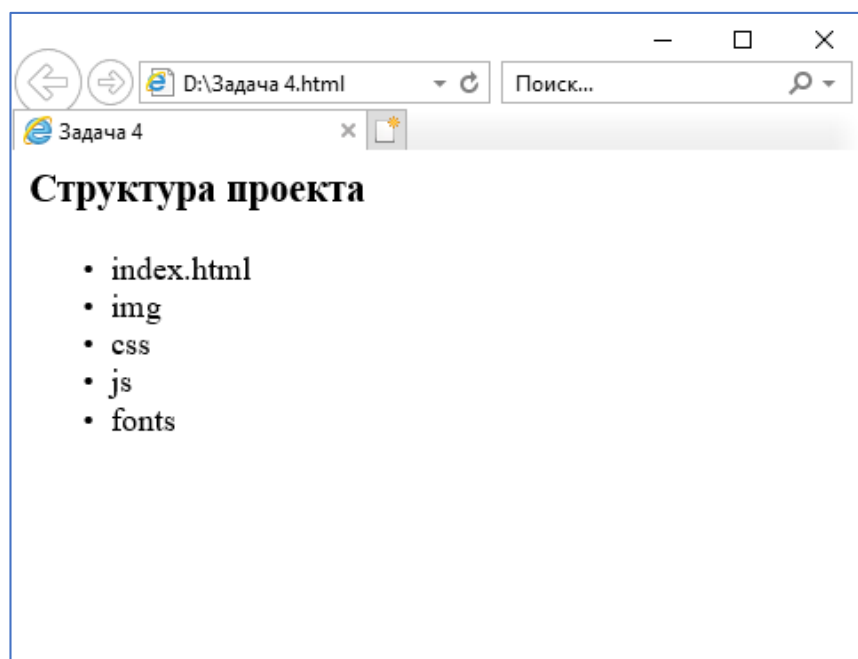
### Задача 3

**Условие:** создайте html-файл, результат которого показан на рисунке. Изображение можно [загрузить по ссылке](#). Использовать элемент `<img>` с заданием альтернативного текста и всплывающей подсказки. Размер самого файла изображения не менять.



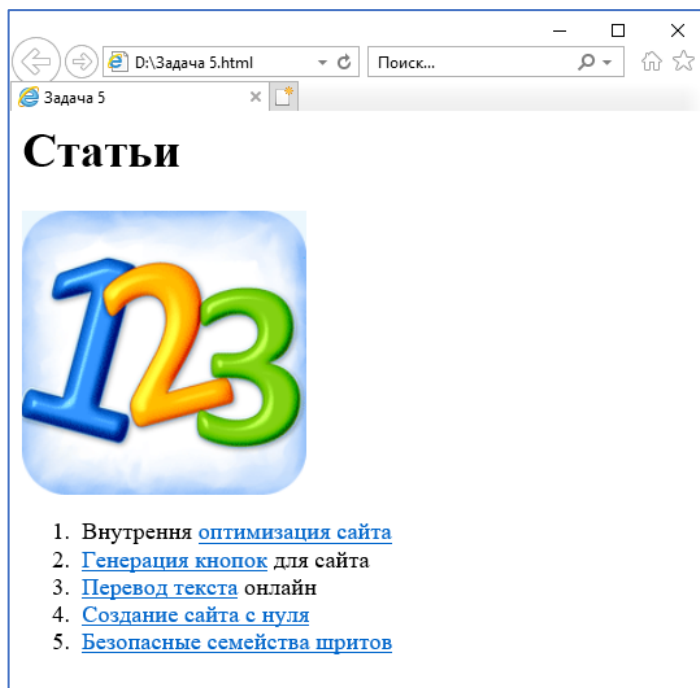
### Задача 4

**Условие:** создайте html-файл, результат которого показан на рисунке.



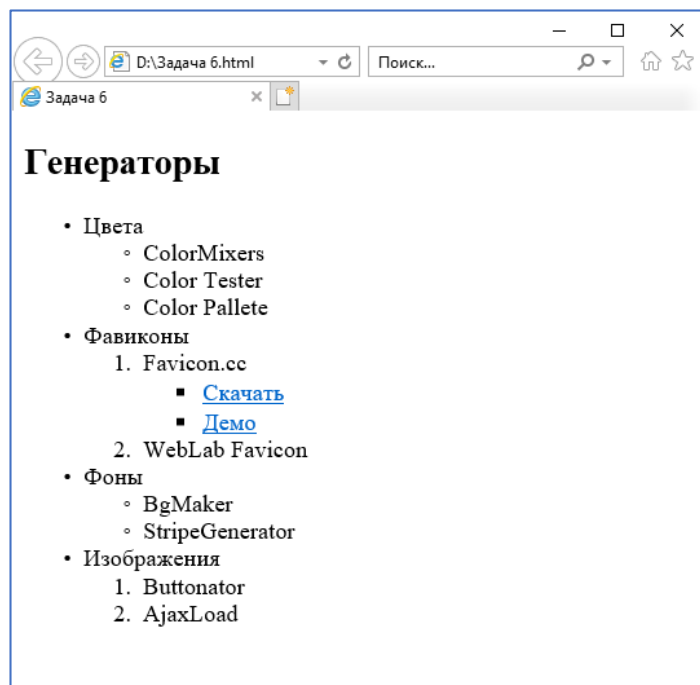
## Задача 5

**Условие:** создайте html-файл, результат которого показан на рисунке. Изображение можно [загрузить по ссылке](#).



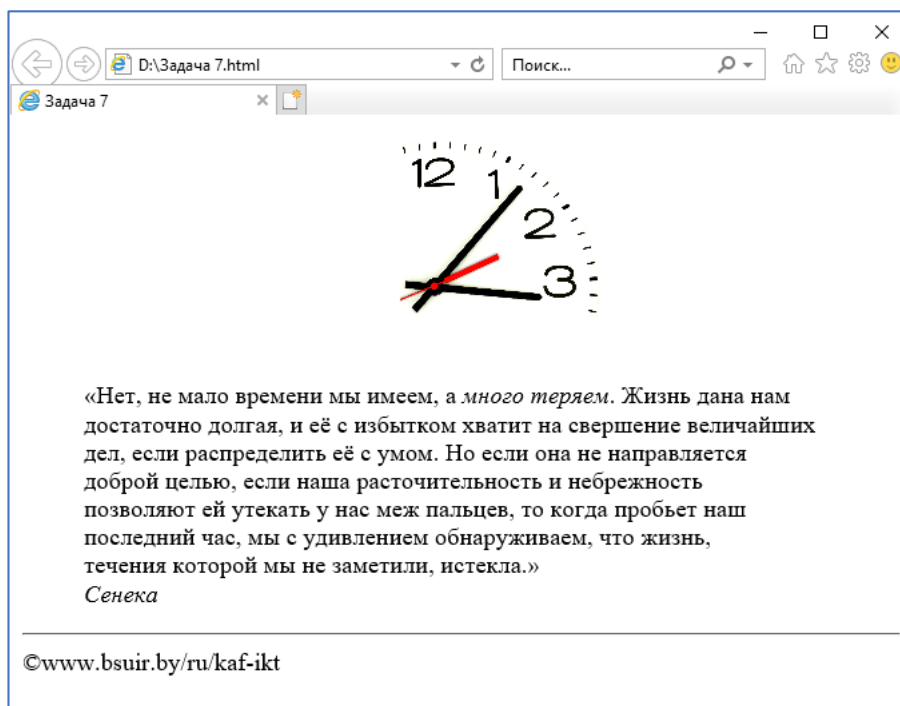
## Задача 6

**Условие:** создайте html-файл, результат которого показан на рисунке.



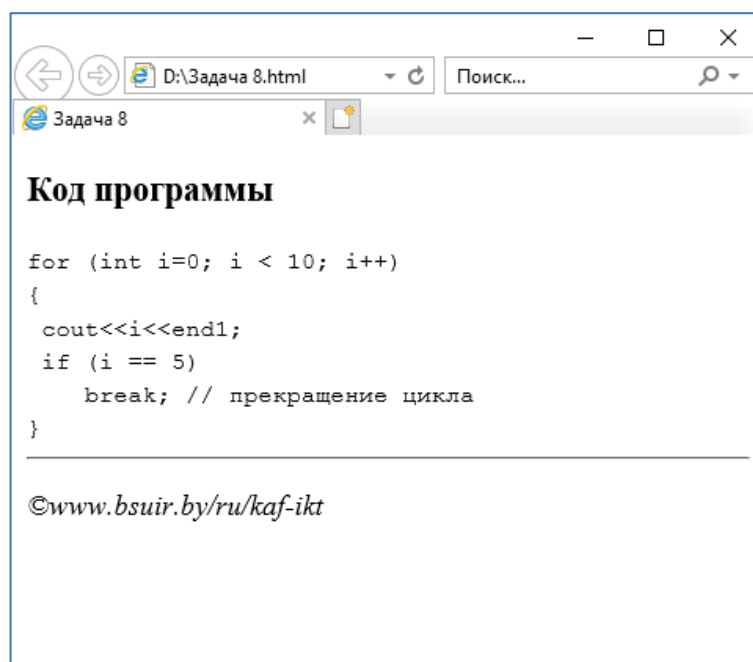
## Задача 7

**Условие:** создайте html-файл, результат которого показан на рисунке. Используйте элементы `<q>`, `<cite>`, `<blockquote>`, `<hr>`. При вставке символа авторского права не используйте мнемоническую запись. Изображение можно [загрузить по ссылке](#).



## Задача 8

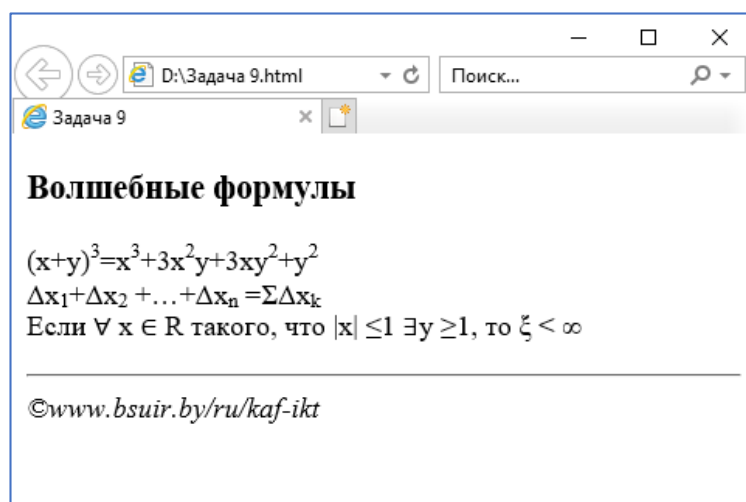
**Условие:** создайте html-файл, результат которого показан на рисунке. Используйте элементы `<code>` и `&nbsp;`.





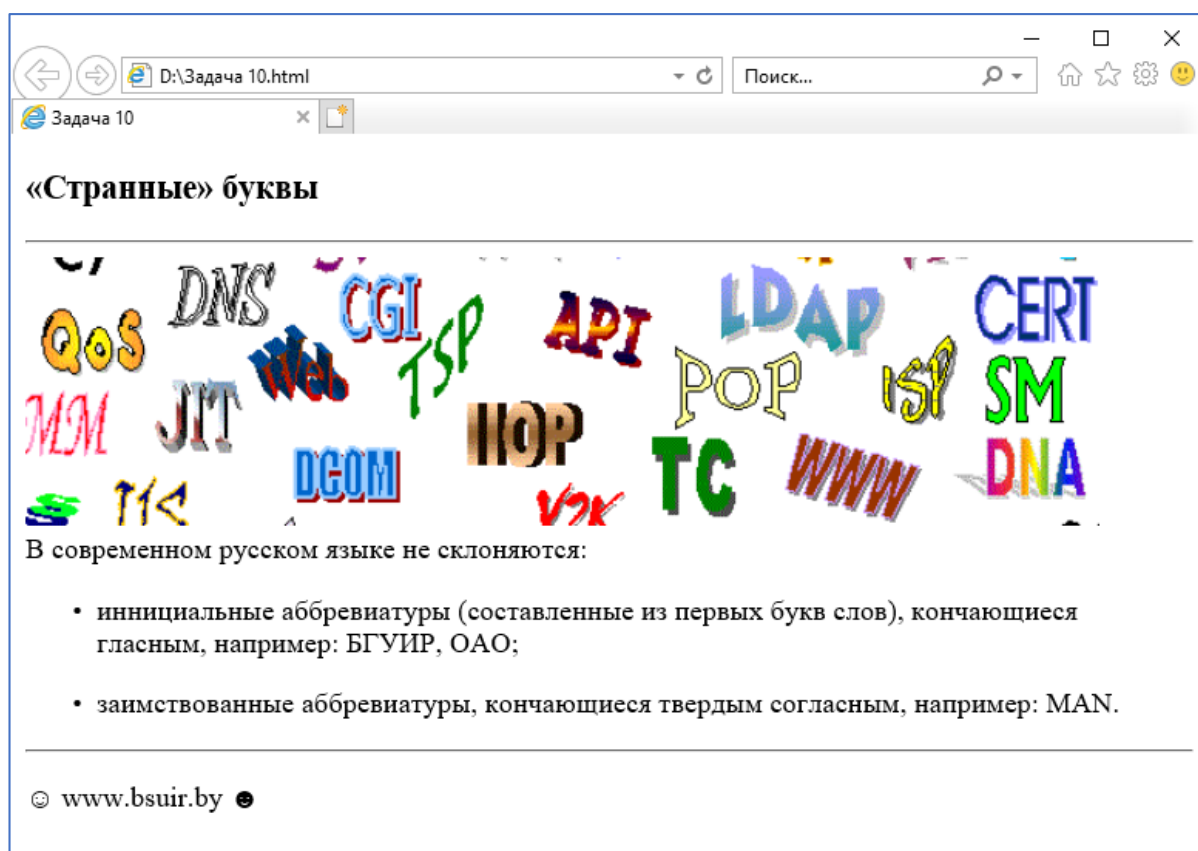
## Задача 9

**Условие:** создайте html-файл, результат которого показан на рисунке.



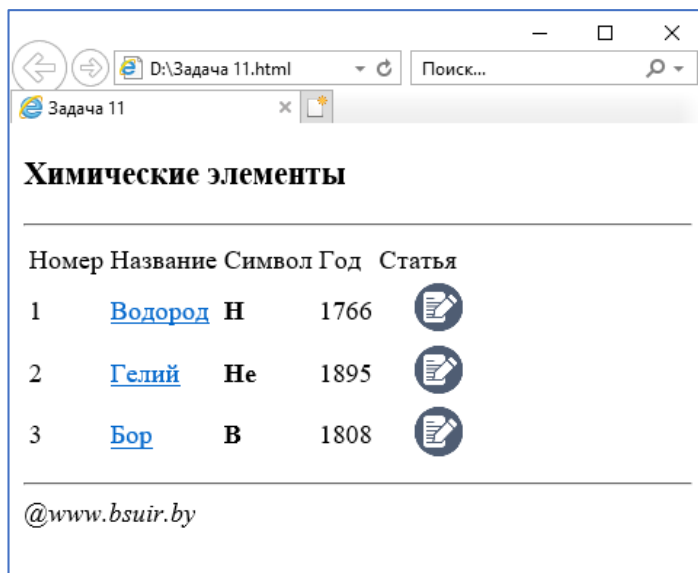
## Задача 10

**Условие:** создайте html-файл, результат которого показан на рисунке. Используйте тег <abbr> для оформления аббревиатур. Обратите внимание, что при наведении на аббревиатуру появляется подсказка. Изображение можно [скачать здесь](#). Последнюю строку на рисунке тоже необходимо повторить, а также кавычки в заголовке.



## Задача 11

**Условие:** создайте html-файл, результат которого показан на рисунке. Используйте элементы html для оформления в виде таблицы. Обратите внимание, что в таблице присутствуют текстовые ссылки и ссылки-изображения. Изображение для ячейки «статья» можно [скачать здесь](#). Последнюю строку на рисунке тоже необходимо повторить.



## Задача 12

**Условие:** создайте html-файл, результат которого показан на рисунке. Для этого добавьте изображение, которое можно [скачать здесь](#). Далее укажите на изображении ссылочные области: клад, акула, осьминог, корабль. При наведении курсора на ссылочную область должна появляться подсказка. Используйте тег `<progress>` для элемента под изображением, наглядно показывающем какая часть пути до клада пройдена.

