

HTML

HTML (HyperText Markup Language, язык разметки гипертекста) — это система верстки, которая определяет, как и какие элементы должны располагаться на веб-странице. Информация на сайте, способ ее представления и оформления зависят исключительно от разработчика и тех целей, которые он перед собой ставит.

Проделаем следующие шаги.

1. В Windows откройте программу Блокнот (Пуск > Выполнить > набрать «notepad» или Пуск > Программы > Стандартные > Блокнот).

2. Наберите код в Блокноте:

```
<html>
```

```
  <head>
```

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

```
  <title>Моя первая веб-страница</title>
```

```
</head>
```

```
<body>
```

```
  <h1>Заголовок страницы</h1>
```

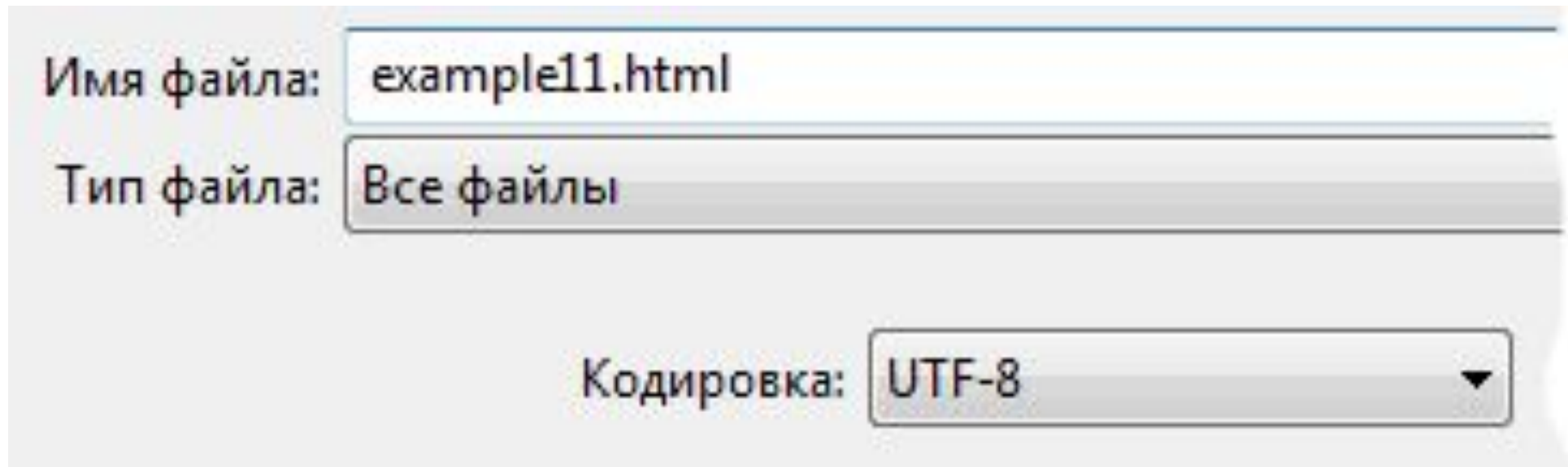
```
  <p>Основной текст группы Имя Вашей Группы.</p>
```

```
  </body>
```

```
</html>
```

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Моя первая веб-страница</title>
  </head>
  <body>
    <h1>Заголовок страницы</h1>
    <p>Основной текст группы СЭЗ11Д.</p>
  </body>
</html>
```

3. Сохраните готовый документ (Файл > Сохранить как...) под именем `c:\www\example11.html`, при этом обязательно поставьте в диалоговом окне сохранения тип файла: Все файлы и кодировку UTF-8 (рис. ниже). Обратите внимание, что расширение у файла должно быть именно `html`.

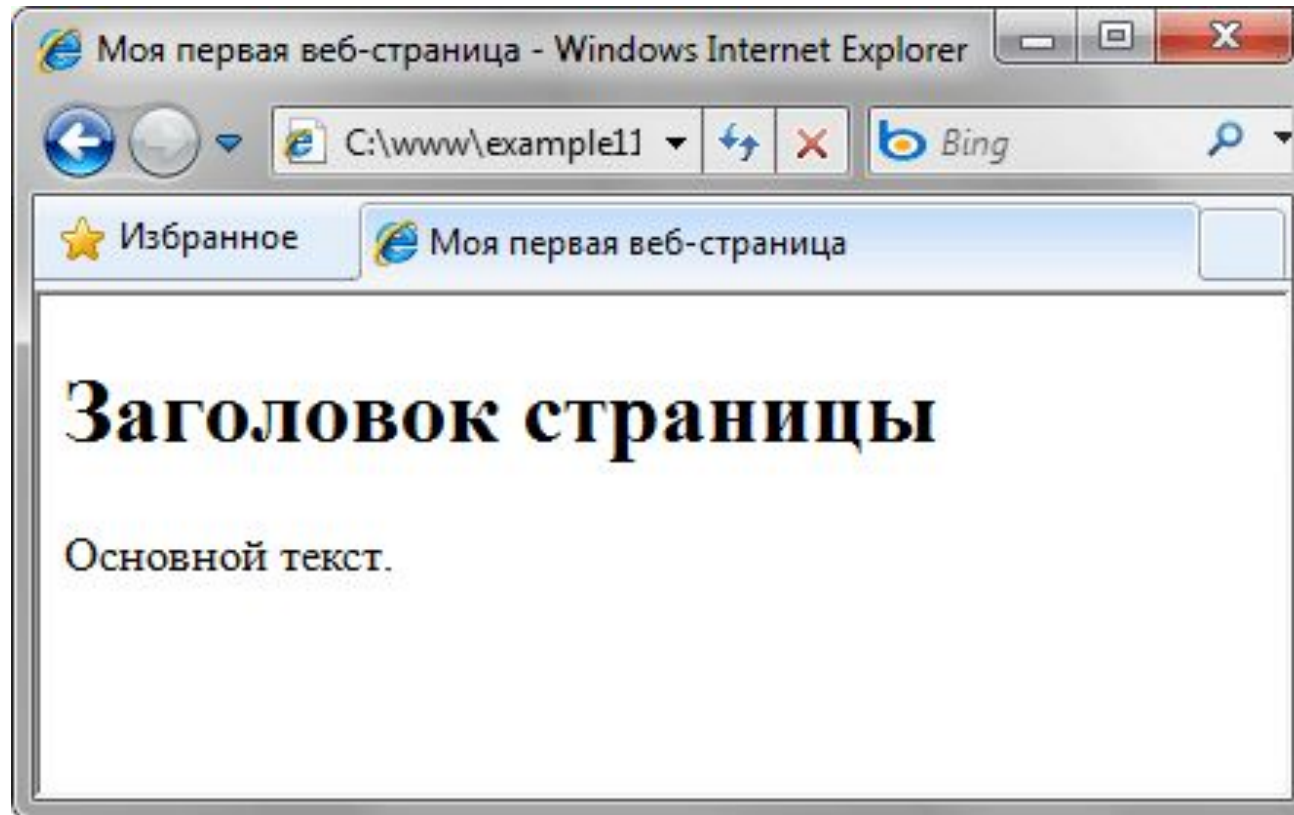


Имя файла:

Тип файла:

Кодировка:

4. Запустите браузер (Пуск > Выполнить > набрать «iexplore» или Пуск > Программы > Internet Explorer).
5. В браузере выберите пункт меню Файл > Открыть и укажите путь к вашему файлу.
6. Если все сделано правильно, то в браузере вы увидите результат, как показано на рисунке.



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).

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 тега — контент.

Одиночные теги не могут хранить в себе содержимого напрямую, оно прописывается как значение атрибута, например, тег `<input type="button" value="Кнопка">` создаст кнопку с текстом Кнопка внутри.

Теги 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 — между тегами <head>...</head> и содержательной части — между тегами <body>...</body>.

1. Структура HTML-документа

Язык HTML следует правилам, которые содержатся в файле объявления типа документа (*Document Type Definition*, или *DTD*).

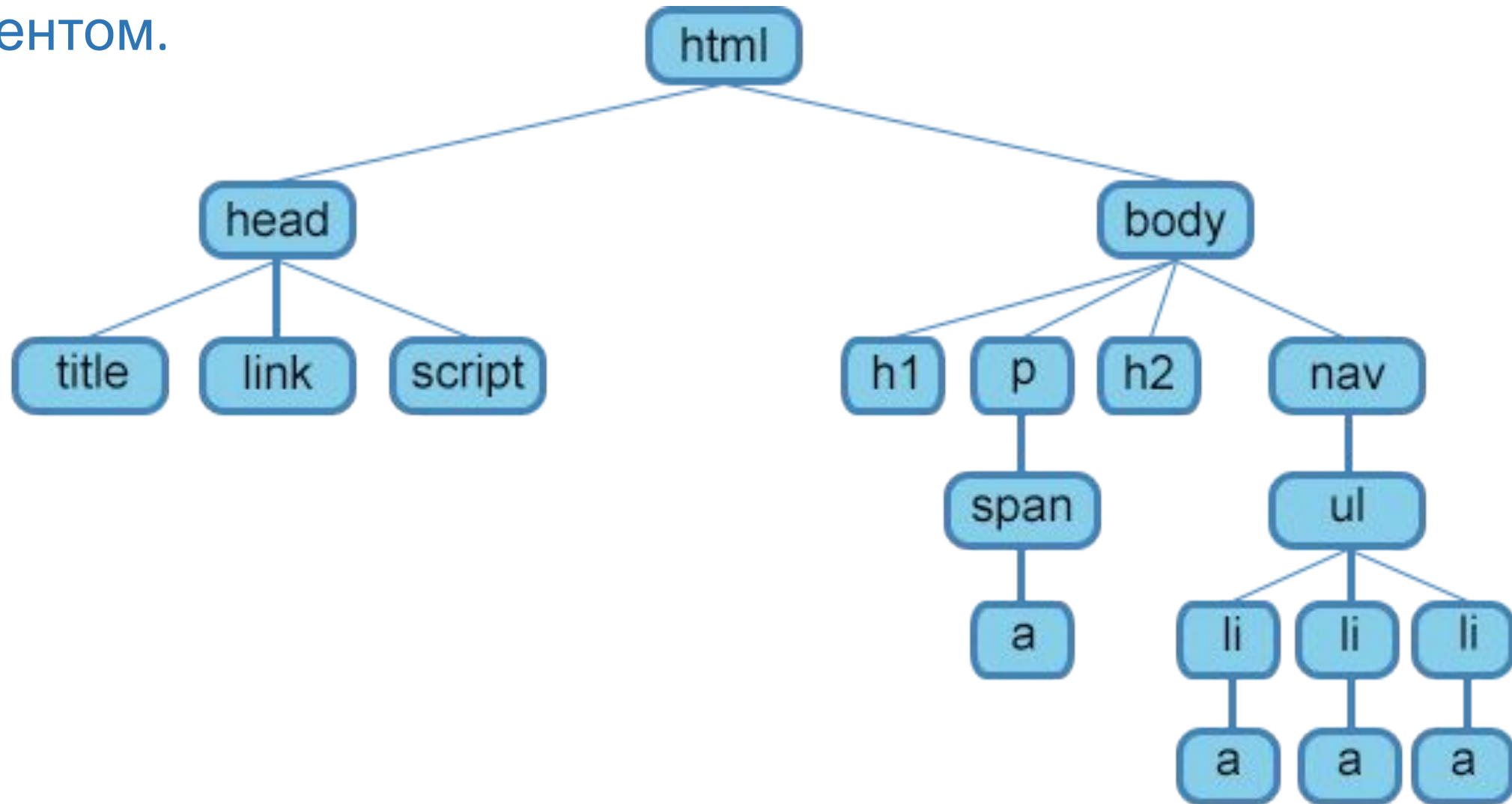
DTD представляет собой XML-документ, определяющий, какие теги, атрибуты и их значения действительны для конкретного типа HTML.

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 внешние
<script src="script.js"></script> <!-- Подключаем сценарии -->
</head>
<body> <!-- Основная часть документа -->
</body>
</html>
```

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



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 включает 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 тегов: `<h1>`, `<p>`, ``, `<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>`. Тег `<p>` является родительским только для ``.

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 тегов `<html>...</html>`. Все, что находится за пределами тегов, не воспринимается браузером как код HTML и никак им не обрабатывается. Для элемента доступны атрибуты `manifest` и `xmlns`, а также глобальные атрибуты.

```
<!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 внешние
<script src="script.js"></script> <!-- Подключаем сценарии -->
</head>
<body> <!-- Основная часть документа -->
</body>
</html>
```

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](#).

```
<!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 внешние
<script src="script.js"></script> <!-- Подключаем сценарии -->
</head>
<body> <!-- Основная часть документа -->
</body>
</html>
```

Глобальные 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.

Атрибут	Описание, принимаемое значение
<code>accesskey</code>	<p>Генерирует сочетания клавиш для доступа к текущему элементу. Состоит из разделенного пробелами списка символов. Браузер в первую очередь выбирает те клавиши, которые существуют на раскладке клавиатуры.</p> <p>Применяется к следующим элементам: <code><a></code>, <code><area></code>, <code><button></code>, <code><input></code>, <code><label></code>, <code><legend></code>, <code><textarea></code>.</p> <p>Принимаемые значения: перечень названий клавиш.</p>
<code>class</code>	<p>Определяет имя класса для элемента (используется для определения класса в таблице стилей).</p> <p>Принимаемые значения: имя класса.</p>
<code>contenteditable</code>	<p>Определяет, может ли пользователь редактировать содержимое (контент). Позволяет преобразовать любое поле HTML в редактируемый элемент.</p> <p>Принимаемые значения: <code>true/false</code>.</p>

<code>contextmenu</code>	Добавляет к элементу контекстное меню, заданное тегом <code><menu></code> . Принимаемые значения: значение атрибута <code>id</code> элемента <code><menu></code> .
<code>dir</code>	Определяет направление текста контента в элементах <code><bdo></code> и <code><bdi></code> . Принимаемые значения: <code>ltr/rtl/auto</code> .
<code>draggable</code>	Определяет, может ли пользователь перетащить элемент. Принимаемые значения: <code>true/false/auto</code> .
<code>dropzone</code>	Определяет область для приема перемещаемых элементов, сообщая браузеру пользователя, какие действия совершить при перемещении. Принимаемые значения: <code>copy</code> — содержимое перемещаемого элемента будет скопировано в область. <code>move</code> — содержимое перемещаемого элемента будет перемещено в новую область. <code>link</code> — при перемещении будет создана ссылка на первоначальные данные элемента.
<code>hidden</code>	Указывает на то, что элемент должен быть скрыт. Принимаемые значения: <code>hidden</code> .
<code>id</code>	Определяет уникальный идентификатор элемента. Принимаемые значения: <code>id</code> — идентификатор элемента.

<code>lang</code>	Определяет код языка содержимого (контента) в элементе. Принимаемые значения: код языка.
<code>spellcheck</code>	Указывает, подлежит ли содержимое элемента проверке орфографии и грамматики. Принимаемые значения: <code>true/false</code> .
<code>style</code>	Указывает на код CSS, применяемую для оформления элемента. Принимаемые значения: код CSS.
<code>tabindex</code>	Определяет порядок перехода к элементу при помощи клавиши TAB. Принимаемые значения: порядковый номер.
<code>title</code>	Определяет дополнительную информацию об элементе, задавая всплывающую подсказку для страницы. Принимаемые значения: текст.
<code>translate</code>	Разрешает или запрещает перевод текста внутри элемента. Принимаемые значения: <code>yes/no</code> .

1.2.1. Элемент <title>

Обязательным тегом раздела `<head>` является тег `<title>`. Текст, размещенный внутри этого тега, 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](#).

```
<!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 внешние
<script src="script.js"></script> <!-- Подключаем сценарии -->
</head>
<body> <!-- Основная часть документа -->
</body>
</html>
```

1.2.2. Элемент <meta>

Необязательным тегом раздела `<head>` является одинарный тег `<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">
```


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

Индексация и переход по ссылкам разрешены:

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

Индексация разрешена, переход по ссылкам запрещен:

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

Индексация и переход по ссылкам запрещены:

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

Для автоматической перезагрузки страницы через заданный промежуток времени нужно воспользоваться значением `refresh` :

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

Страница будет перезагружена через 30 секунд. Чтобы перебросить посетителя на другую страницу, нужно указать URL-адрес в параметре `url` :

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

Для элемента `<meta>` доступны атрибуты `charset`, `content`, `http-equiv`, `name`, а также глобальные атрибуты.

Атрибут	Описание, принимаемое значение
<code>charset</code>	Указывает кодировку символов для текущего HTML-документа: <code><meta charset="UTF-8"></code>
<code>content</code>	Содержит произвольный текст, который определяет значение, ассоциируемое с атрибутом <code>http-equiv</code> или <code>name</code> , в зависимости от их значения.
<code>http-equiv</code>	<p>Контролирует действия браузера на данной веб-странице (эквивалент HTTP заголовков). При отображении страницы браузер будет следовать инструкциям, заданным в атрибуте:</p> <p><code>default-style</code> указывает предпочтительный стиль для использования на странице. Атрибут <code>content</code> должен содержать идентификатор элемента <code><link></code>, который ссылается на таблицу стилей CSS, или идентификатор элемента <code><style></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:</p> <pre><meta http-equiv="refresh" content="30"></pre> <p>Если необходимо сразу перебросить посетителя на другую страницу, то можно указать URL-адрес в параметре <code>url</code>:</p> <pre><meta http-equiv="refresh" content="0; url=http://mail.ru/"></pre>

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 `media`, `scoped`, `type`, а также глобальные атрибуты.

Внутри этого элемента можно записывать код форматирования как самих элементов веб-страницы, так и веб-страницы целиком.

```
<style type="text/css">
.paper {
width: 200px;
height: 300px;
background-color: #ef4444;
color: #666666;
}
```

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`

```
<!DOCTYPE html>
<html>
<head>
<style>
@import url(style.css);
</style>
<meta>
<title> </title>
</head>
```

с использованием элемента `<link>`. Элемент не требует закрывающего тега. Данный элемент определяет отношение между текущей страницей и другими документами. Таких элементов на странице может быть несколько. Запись будет иметь следующий вид:

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

1.2.5. Элемент `<script>`

Элемент `<script>` позволяет присоединять к документу различные сценарии. Закрывающий тег обязателен, при этом 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 `async`, `charset`, `defer`, `src`, `type`, а также глобальные атрибуты.

1.3. Элемент `<body>`

В этом разделе располагается все содержимое документа. Для элемента доступны глобальные атрибуты.

1.2. 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>`, `
`, `<col>`, `<embed>`, `<hr>`, ``, `<input>`, `<link>`, `<menuitem>`, `<meta>`, `<param>`, `<source>`, `<track>`, `<wbr>` ;
- **элементы с неформатированным текстом** - `<script>`, `<style>` ;
- **элементы, выводющие неформатированный текст** - `<textarea>`, `<title>` ;
- **элементы из другого пространства имён** - MathML и SVG;
- **обычные элементы** - все остальные элементы.

Тег	Описание
<code><!--...--></code>	Используется для добавления комментариев.
<code><!DOCTYPE></code>	Объявляет тип документа и предоставляет основную информацию для браузера - его язык и версия.
<code><a></code>	Создаёт гипертекстовые ссылки.
<code><abbr></code>	Определяет текст как аббревиатуру или акроним. Поясняющий текст задаётся с помощью атрибута <code>title</code> .
<code><address></code>	Задаёт контактные данные автора/владельца документа или статьи. Отображается в браузере курсивом.
<code><area></code>	Представляет собой гиперссылку с текстом, соответствующей определенной области на карте-изображении или активную область внутри карты-изображения. Всегда вложен внутри тега <code><map></code> .

<code><article></code>	Раздел контента, который образует независимую часть документа или сайта, например, статья в журнале, запись в блоге, комментарий.
<code><aside></code>	Представляет контент страницы, который имеет косвенное отношение к основному контенту страницы/сайта.
<code><audio></code>	Загружает звуковой контент на веб-страницу.
<code></code>	Задаёт полужирное начертание отрывка текста, не придавая акцент или важность выделенному.
<code><base></code>	Задаёт базовый адрес (URL), относительно которого вычисляются все относительные адреса. Это поможет избежать проблем при переносе страницы в другое место, так как все ссылки будут работать, как и прежде.
<code><bdi></code>	Изолирует отрывок текста, написанный на языке, в котором чтение текста происходит справа налево, от остального текста.

<code><bdo></code>	атрибуте <code>dir</code> , переопределяя текущее направление написания текста.
<code><blockquote></code>	Выделяет текст как цитату, применяется для описания больших цитат.
<code><body></code>	Представляет тело документа (содержимое, не относящееся к метаданным документа).
<code>
</code>	Перенос текста на новую строку.
<code><button></code>	Создает интерактивную кнопку. Внутри тега можно поместить содержимое - текст или изображение.
<code><canvas></code>	Холст-контейнер для динамического отображения изображений, таких как простые изображения, диаграммы, графики и т.п. Для рисования используется скриптовый язык JavaScript.
<code><caption></code>	Добавляет подпись к таблице. Вставляется сразу после тега <code><table></code> .
<code><cite></code>	Используется для указания источника цитирования. Отображается курсивом.

<code><col></code>	Выбирает для форматирования один или несколько столбцов таблицы, не содержащих информацию одного типа.
<code><colgroup></code>	Создает структурную группу столбцов, выделяющую множество логически однородных ячеек.
<code><data></code>	Элемент используется для связывания значения атрибута <code>value</code> , которое представлено в машиночитаемом формате и может быть обработано компьютером, с содержимым тега.
<code><datalist></code>	Элемент-контейнер для выпадающего списка элемента <code><input></code> . Варианты значений помещаются в элементы <code><option></code> .
<code><dd></code>	Используется для описания термина из тега <code><dt></code> .
<code></code>	Помечает текст как удаленный, перечёркивая его.

`<details>`

Создаёт интерактивный виджет, который пользователь может открыть или закрыть. Представляет собой контейнер для контента, видимый заголовок виджета помещается в тег `<summary>`.

`<dfn>`

Определяет слово как термин, выделяя его курсивом. Текст, идущий следом, должен содержать расшифровку этого термина.

`<dialog>`

Интерактивный элемент, с которым взаимодействует пользователь для выполнения задачи, например, диалоговое окно, инспектор или окно. Без атрибута `open` не виден для пользователя.

`<div>`

Тег-контейнер для разделов HTML-документа. Используется для группировки блочных элементов с целью форматирования стилями.

`<dl>`

Тег-контейнер, внутри которого находятся термин и его описание.

`<dt>`

Используется для задания термина.

<code></code>	Выделяет важные фрагменты текста, отображая их курсивом.
<code><embed></code>	Тег-контейнер для встраивания внешнего интерактивного контента или плагина.
<code><fieldset></code>	Группирует связанные элементы в форме, рисуя рамку вокруг них.
<code><figcaption></code>	Заголовок/подпись для элемента <code><figure></code> .
<code><figure></code>	Самодостаточный тег-контейнер для такого контента как иллюстрации, диаграммы, фотографии, примеры кода, обычно с подписью.
<code><footer></code>	Определяет завершающую область (нижний колонтитул) документа или раздела.
<code><form></code>	Форма для сбора и отправки на сервер информации от пользователей. Не работает без атрибута <code>action</code> .
<code><h1-h6></code>	Создают заголовки шести уровней для связанных с ними разделов.

<code><header></code>	Секция для вводной информации сайта или группы навигационных ссылок. Может содержать один или несколько заголовков, логотип, информацию об авторе.
<code><hr></code>	Горизонтальная линия для тематического разделения параграфов.
<code><html></code>	Корневой элемент HTML-документа. Сообщает браузеру, что это HTML-документ. Является контейнером для всех остальных html-элементов.
<code><i></code>	Выделяет отрывок текста курсивом, не придавая ему дополнительный акцент.
<code><iframe></code>	Создает встроенный фрейм, загружая в текущий HTML-документ другой документ.
<code></code>	Встраивает изображения в HTML-документ с помощью атрибута <code>src</code> , значением которого является адрес встраиваемого изображения.
<code><input></code>	Создает многофункциональные поля формы, в которые пользователь может вводить данные.

<code><ins></code>	Выделяет текст подчеркиванием. Применяется для выделения изменений, вносимых в документ.
<code><kbd></code>	Выделяет текст, который должен быть введен пользователем с клавиатуры, шрифтом семейства monospace.
<code><label></code>	Добавляет текстовую метку для элемента <code><input></code> .
<code><legend></code>	Заголовок элементов формы, сгруппированных с помощью элемента <code><fieldset></code> .
<code></code>	Элемент маркированного или нумерованного списка.
<code><link></code>	Определяет отношения между документом и внешним ресурсом. Также используется для подключения внешних таблиц стилей.
<code><main></code>	Контейнер для основного уникального содержимого документа. На одной странице должно быть не более одного элемента

<code><map></code>	Создаёт активные области на карте-изображении. Является контейнером для элементов <code><area></code> .
<code><mark></code>	Выделяет фрагменты текста, помечая их желтым фоном.
<code><meta></code>	Используется для хранения дополнительной информации о странице. Эту информацию используют браузеры для обработки страницы, а поисковые системы – для ее индексации. В блоке <code><head></code> может быть несколько тегов <code><meta></code> , так как в зависимости от используемых атрибутов они несут разную информацию.
<code><meter></code>	Индикатор измерения в заданном диапазоне.
<code><nav></code>	Раздел документа, содержащий навигационные ссылки по сайту.
<code><noscript></code>	Определяет секцию, не поддерживающую сценарий (скрипт).

<code><object></code>	Контейнер для встраивания мультимедиа (например, аудио, видео, Java-апплеты, ActiveX, PDF и Flash). Также можно вставить другую веб-страницу в текущий HTML-документ. Для передачи параметров встраиваемого плагина используется тег <code><param></code> .
<code></code>	Упорядоченный нумерованный список. Нумерация может быть числовая или алфавитная.
<code><optgroup></code>	Контейнер с заголовком для группы элементов <code><option></code> .
<code><option></code>	Определяет вариант/опцию для выбора в раскрывающемся списке <code><select></code> , <code><optgroup></code> или <code><datalist></code> .
<code><output></code>	Поле для вывода результата вычисления, рассчитанного с помощью скрипта.
<code><p></code>	Параграфы в тексте.

<code><picture></code>	Элемент-контейнер, содержащий один элемент <code></code> и ноль или несколько элементов <code><source></code> . Сам по себе ничего не отображает. Дает возможность браузеру выбирать наиболее подходящее изображение.
<code><pre></code>	Выводит текст без форматирования, с сохранением пробелов и переносов текста. Может быть использован для отображения компьютерного кода, сообщения электронной почты и т.д.
<code><progress></code>	Индикатор выполнения задачи любого рода.
<code><q></code>	Определяет краткую цитату.
<code><ruby></code>	Контейнер для Восточно-Азиатских символов и их расшифровки.
<code><rb></code>	Определяет вложенный в него текст как базовый компонент аннотации.

<code><rt></code>	Добавляет краткую характеристику сверху или снизу от символов, заключенных в элементе <code><ruby></code> , выводится уменьшенным шрифтом.
<code><rtc></code>	Отмечает вложенный в него текст как дополнительную аннотацию.
<code><rp></code>	Выводит альтернативный текст в случае если браузер не поддерживает элемент <code><ruby></code> .
<code><s></code>	Отображает текст, не являющийся актуальным, перечеркнутым.
<code><samp></code>	Используется для вывода текста, представляющего результат выполнения программного кода или скрипта, а также системные сообщения. Отображается моноширинным шрифтом.
<code><script></code>	Используется для определения сценария на стороне клиента (обычно JavaScript). Содержит либо текст скрипта, либо указывает на внешний файл сценария с помощью атрибута <code>src</code> .

<code><section></code>	Определяет логическую область (раздел) страницы, обычно с заголовком.
<code><select></code>	Элемент управления, позволяющий выбирать значения из предложенного множества. Варианты значений помещаются в <code><option></code> .
<code><small></code>	Отображает текст шрифтом меньшего размера.
<code><source></code>	Указывает местоположение и тип альтернативных медиаресурсов для элементов <code><picture></code> , <code><video></code> , <code><audio></code> .
<code></code>	Контейнер для строчных элементов. Можно использовать для форматирования отрывков текста, например, выделения цветом отдельных слов.
<code></code>	Расставляет акценты в тексте, выделяя полужирным.
<code><style></code>	Подключает встраиваемые таблицы стилей.

<code><sub></code>	Задаёт подстрочное написание символов, например, индекса элемента в химических формулах.
<code><summary></code>	Создаёт видимый заголовок для тега <code><details></code> . Отображается с закрашенным треугольником, кликнув по которому можно просмотреть подробности заголовка.
<code><sup></code>	Задаёт надстрочное написание символов.
<code><table></code>	Тег для создания таблицы.
<code><tbody></code>	Определяет тело таблицы.
<code><td></code>	Создает ячейку таблицы.
<code><template></code>	Используется для объявления фрагментов HTML-кода, которые могут быть клонированы и вставлены в документ скриптом. Содержимое тега не является его дочерним элементом.
<code><textarea></code>	Создает большие поля для ввода текста.

<code><tfoot></code>	Определяет нижний колонтитул таблицы.
<code><th></code>	Создает заголовок ячейки таблицы.
<code><thead></code>	Определяет заголовок таблицы.
<code><time></code>	Определяет дату/время.
<code><title></code>	Заголовок HTML-документа, отображаемый в верхней части строки заголовка браузера. Также может отображаться в результатах поиска, поэтому это следует принимать во внимание при предоставлении названия.
<code><tr></code>	Создает строку таблицы.
<code><track></code>	Добавляет субтитры для элементов <code><audio></code> и <code><video></code> .

<code><u></code>	Выделяет отрывок текста подчёркиванием, без дополнительного акцента.
<code></code>	Создает маркированный список.
<code><var></code>	Выделяет переменные из программ, отображая их курсивом.
<code><video></code>	Добавляет на страницу видео-файлы. Поддерживает 3 видео формата: MP4, WebM, Ogg.
<code><wbr></code>	Указывает браузеру возможное место разрыва длинной строки.

1.4. HTML-текст

HTML- текст представлен в спецификации тегами для форматирования и группировки текста. Теги представляют собой контейнеры для текста и не имеют визуального отображения.

Теги 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 тега `<body>`.

1. Теги заголовков

Заголовок первого уровня

Заголовок второго уровня

Заголовок третьего уровня

Заголовок четвертого уровня

Заголовок пятого уровня

Заголовок шестого уровня

Заголовки являются важными элементами веб-страницы, они упорядочивают текст, формируя его визуальную структуру. Теги `<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 допускается вложение других тегов в теги `<h1>...<h6>`.

1.1. Тег <h1>

Заголовок самого верхнего уровня, на странице рекомендуется использовать только один раз, по возможности частично дублируя заглавие страницы. Тег `<h1>` должен быть уникальным для каждой страницы сайта. Рекомендуется прописывать тег в начале статьи, используя ключевое слово в тексте заголовка. Размер шрифта в браузере равен **2em**, верхний и нижний отступ по умолчанию **0.67em**.

1.2. Тег <h2>

Им обозначаются подзаголовки тега `<h1>`. Размер шрифта в браузере равен **1.5em**, верхний и нижний отступ по умолчанию **0.83em**.

1.3. Тег <h3>

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

1.4. Теги <h4>, <h5>, <h6>

Обозначают подзаголовки четвёртого, 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.

Относительное измерение высоты шрифта в em пространстве. Поскольку единица em эквивалентна размеру данного шрифта, то, если Вы укажете шрифт 12pt, каждый элемент «em» будет 12pt. Таким образом, 2em будет 24pt.

2. Теги для форматирования текста

2.1. Тег

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

Для тега доступны глобальные атрибуты.

2.2. Тег

Отображает шрифт курсивом, придавая тексту значимость.

Для тега доступны глобальные атрибуты.

2.3. Тег <i>

Отображает шрифт курсивом.

Для тега доступны глобальные атрибуты.

2.4. Тег <small>

Уменьшает размер шрифта на единицу по отношению к обычному тексту.

Для тега доступны глобальные атрибуты.

2.5. Тег

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

2.6. Тег <sub>

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

Для тега доступны глобальные атрибуты.

2.7. Тег <sup>

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

Для тега доступны глобальные атрибуты.

2.8. Тег <ins>

Выделяет текст в новой версии документа, подчёркивая его.

Для тега доступны следующие атрибуты: `cite`, `datetime`.

2.9. Тег

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

3. Теги для ввода «компьютерного» текста

3.1. Тег <code>

Служит для выделения фрагментов программного кода. Отображается моноширинным шрифтом.

Для тега доступны [глобальные атрибуты](#).

3.2. Тег <kbd>

Отмечает фрагмент как вводимый пользователем с клавиатуры. Отображается моноширинным шрифтом.

Для тега доступны [глобальные атрибуты](#).

3.3. Тег <samp>

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

Для тега доступны [глобальные атрибуты](#).

3.4. Тег <var>

Выделяет имена переменных, отображая курсивом.

Для тега доступны [глобальные атрибуты](#).

3.5. Тег <pre>

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

4. Теги для оформления цитат и определений

4.1. Тег <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 тега доступны глобальные атрибуты.

4.2. Тег <bdo>

Используется для замещения текущего направления текста, т.е. текст в теге отображается зеркально.

Для тега доступен атрибут `dir`.

4.3. Тег <blockquote>

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

Для тега доступен атрибут `cite`.

4.4. Тег <q>

Используется для выделения коротких цитат. Браузерами заключается в кавычки.

Для тега доступен атрибут `cite`.

4.5. Тег <cite>

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

Для тега доступны глобальные атрибуты.

4.6. Тег <dfn>

Позволяет выделить текст как определение. Несмотря на наличие специального тега, рекомендуется выделять текст силами CSS.

Для тега доступен атрибут `title`.

5. Абзацы, средства переноса текста

5.1. Тег <p>

Разбивает текст на отдельные абзацы, отделяя друг от друга пустой строкой. Браузер автоматически добавляет верхний и нижний отступ, равный `1em`, при этом отступы соседних абзацев «схлопываются».

Для тега доступны глобальные атрибуты.

5.2. Тег

Переносит текст на следующую строку, создавая разрыв строки.

Для тега доступны глобальные атрибуты.

5.3. Тег <hr>

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

1.5. 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 тега `<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 парного тега `<a>`. Внутри тега 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.

Обязательным параметром тега `<a>` является атрибут `href`, который задает URL-адрес веб-страницы.

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

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

```
https://site.ru/
```

`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
```

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 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

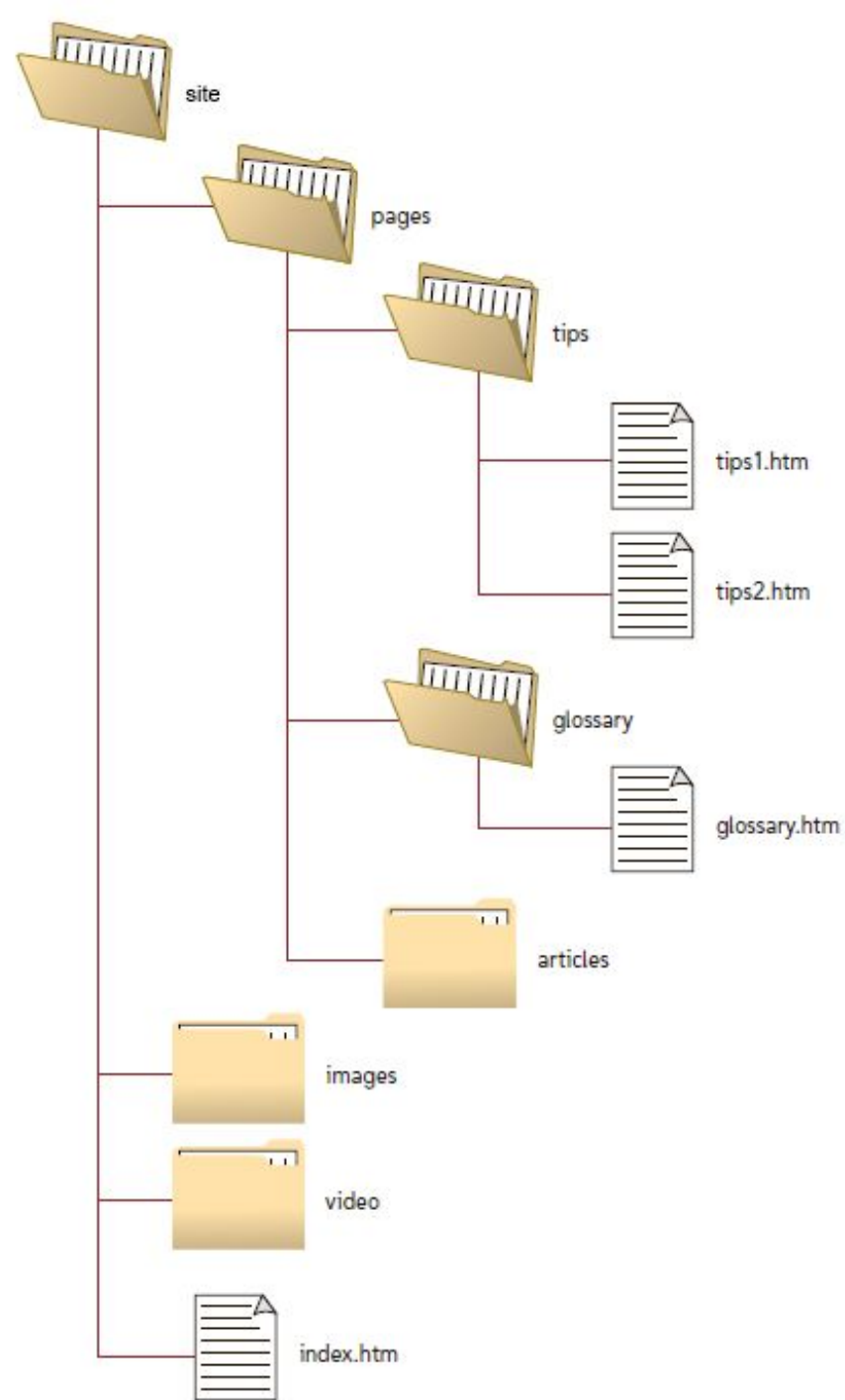
70 – Gopher

80 – HTTP

Путь содержит имя папки, в которой находится файл.

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.



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:

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

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

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

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

```
http://site.ru/index.html
```

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

```
http://site.ru/
```

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:

- 1) папка (имя папки, указывающей путь к файлу);
- 2) файл (имя файла).

Путь для относительных ссылок имеет три специальных обозначения:

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

`./` указывает на текущую папку

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

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

Следующая разметка создаст оглавление с быстрыми переходами на соответствующие разделы:

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

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>
```

4. Как сделать изображение-ссылку

Чтобы сделать кликабельное изображение, необходимо поместить элемент `` внутри тега `<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"></a>
```


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. АТТРИБУТЫ ТЕГА `<A>`

Атрибут	Описание, принимаемое значение
<code>download</code>	<p>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:</p> <pre></pre> <pre></pre> <pre>Загрузить отчет за февраль 2014</pre>
<code>href</code>	<p>Значением атрибута является URL-адрес документа, на который указывается ссылка.</p>

<code>hreflang</code>	<p>Определяет язык связанного веб-документа. Используется только вместе с атрибутом <code>href</code>. Принимаемые значения — аббревиатура, состоящая из пары букв, обозначающих <u>код языка</u>, например:</p> <pre data-bbox="759 211 1625 254">Anysite</pre>
<code>media</code>	<p>Определяет, для каких типов устройств файл будет оптимизирован. Значением может быть любой медиа-запрос.</p>
<code>ping</code>	<p>Отслеживает URL-адреса ресурсов, по которым переходил пользователь.</p>
<code>rel</code>	<p>Дополняет атрибут <code>href</code> информацией об отношении между текущим и связанным документом. Принимаемые значения:</p> <ul data-bbox="759 554 1778 1382" style="list-style-type: none"><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.<code>help</code> — ссылка на справку.<code>license</code> — ссылка на информацию об авторских правах на данный веб-документ.<code>next/prev</code> — указывает связь между отдельными URL. Благодаря этой разметке Google может определить, что содержание данных страниц связано в логической последовательности.<code>nofollow</code> — запрещает поисковой системе переходить по ссылкам на данной странице или по конкретной ссылке.<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.<code>prefetch</code> — указывает, что целевой документ должен быть кэширован, т.е. браузер в фоновом режиме загружает содержимое страницы к себе в кэш.<code>search</code> — указывает, что целевой документ содержит инструмент для поиска.<code>tag</code> — указывает ключевое слово для текущего документа.

<code>target</code>	<p>Указывает на то, в каком окне должен открываться документ, к которому ведет ссылка. Принимает следующие значения:</p> <ul style="list-style-type: none"><code>_self</code> – страница загружается в текущее окно;<code>_blank</code> – страница открывается в новом окне браузера;<code>_parent</code> – страница загружается во фрейм-родитель;<code>_top</code> – страница загружается в полное окно браузера.
<code>type</code>	<p>Указывает MIME-тип файлов ссылки, т.е. расширение файла. На данный момент носит больше справочную информацию.</p>

1.6. HTML-изображения

HTML-изображения добавляются на веб-страницы с помощью тега ``. Использование графики делает веб-страницы визуально привлекательнее.

Изображения помогают лучше передать суть и содержание веб-документа.

С помощью HTML-тегов `<map>` и `<area>` можно создавать карты-изображения с активными областями.

Вставка изображений в HTML-документ

Содержание:

1. Тег ``

1.1. Адрес изображения

1.2. Размеры изображения

1.3. Форматы графических файлов

2. Тег `<map>`

3. Тег `<area>`

4. Пример создания карты-изображения

1. Тег

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

Тег `` имеет обязательный атрибут `src`, значением которого является абсолютный или относительный путь к изображению:

```

```

Для тега `` доступны следующие атрибуты:

Атрибут	Описание, принимаемое значение
<code>alt</code>	<p>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.</p> <p>Синтаксис: <code>alt="описание изображения"</code>.</p>
<code>crossorigin</code>	<p>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:</p> <ul style="list-style-type: none"><code>anonymous</code> — Cross-origin запрос выполняется с помощью HTTP-заголовка, при этом учетные данные не передаются. Если сервер не даёт 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. <p>Синтаксис: <code>crossorigin="anonymous"</code>.</p>
<code>height</code>	<p>Атрибут <code>height</code> задает высоту изображения в <code>px</code>.</p> <p>Синтаксис: <code>height="300"</code>.</p>

<code>ismap</code>	<p>Атрибут <code>ismap</code> указывает на то, что картинка является частью изображения-карты, расположенного на сервере (изображение-карта — изображение с интерактивными областями). При нажатии на изображение-карту координаты передаются на сервер в виде строки запроса URL-адреса. Атрибут <code>ismap</code> допускается только в случае, если элемент <code></code> является потомком элемента <code><a></code> с действительным атрибутом <code>href</code>. Синтаксис: <code>ismap</code>.</p>
<code>longdesc</code>	<p>URL расширенного описания изображения, дополняющее атрибут <code>alt</code>. Синтаксис: <code>longdesc="http://www.example.com/description.txt"</code>.</p>
<code>src</code>	<p>Атрибут <code>src</code> задает путь к изображению. Синтаксис: <code>src="flower.jpg"</code>.</p>
<code>sizes</code>	<p>Задаёт размер изображения в зависимости от параметров отображения. Работает только при заданном атрибуте <code>srcset</code>. Значением атрибута является одна или несколько строк, указанных через запятую.</p>
<code>srcset</code>	<p>Создаёт список источников для изображения, которые будут выбраны, исходя из разрешения экрана. Может использоваться вместе или вместо атрибута <code>src</code>. Значением атрибута является одна или несколько строк, разделенных запятой.</p> <pre></pre>

<code>usemap</code>	<p>Атрибут <code>usemap</code> определяет изображение в качестве карты-изображения. Значение обязательно должно начинаться с символа <code>#</code>. Значение ассоциируется со значением атрибута <code>name</code> или <code>id</code> тега <code><map></code> и создает связь между элементами <code></code> и <code><map></code>. Атрибут нельзя использовать, если элемент <code><map></code> является потомком элемента <code><a></code> или <code><button></code>.</p> <p>Синтаксис: <code>usemap="#mymap"</code>.</p>
<code>width</code>	<p>Атрибут <code>width</code> задает ширину изображения в <code>px</code>.</p> <p>Синтаксис: <code>width="500"</code>.</p>

1.1. 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.
```

1.2. 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.

1.3. 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 покрыться 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.

Формат 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 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 задаётся отдельно. Размер значка может быть любым, но наиболее употребимы квадратные значки со сторонами 16, 32 и 48 пикселей.

2. Тег <map>

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

```

<map name="имя_карты">
...
</map>
```

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

3. Тег <area>

Тег `<area>` описывает только одну активную область в составе карты изображений на стороне клиента. Элемент не имеет закрывающего тега. Если одна активная область перекрывает другую, то будет реализована первая ссылка из списка областей.

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

ТАБЛИЦА 2. АТТРИБУТЫ ТЕГА <AREA>

Атрибут	Краткое описание
<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 круга — координаты центра и радиус круга; для прямоугольника — координаты верхнего левого и правого нижнего углов; для многоугольника — 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.

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.

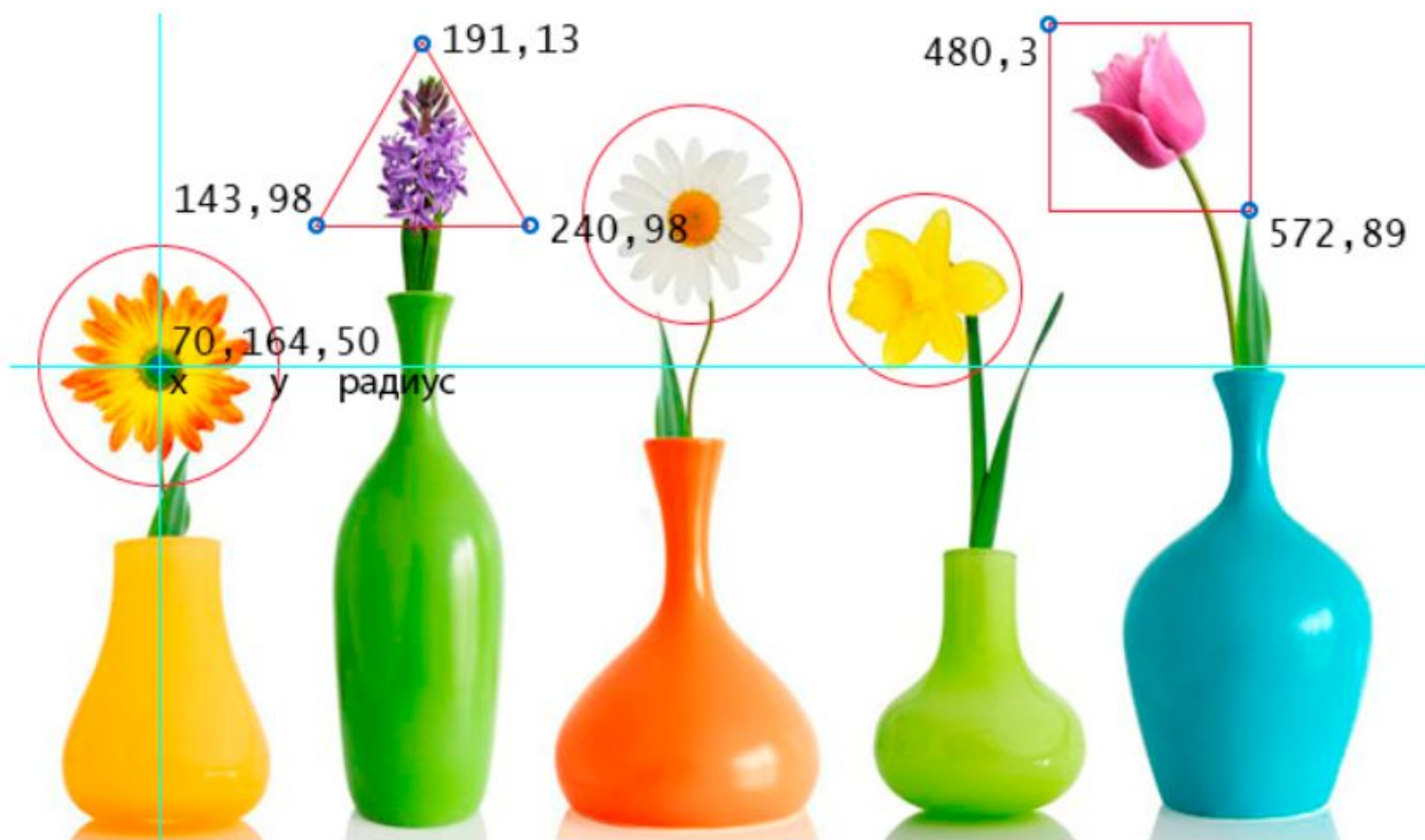
`search` — указывает, что целевой документ содержит инструмент для поиска.

`tag` — указывает ключевое слово для текущего документа.

<code>shape</code>	<p>Задаёт форму активной области на карте и её координаты. Может принимать следующие значения:</p> <ul style="list-style-type: none"><code>rect</code> – активная область прямоугольной формы;<code>circle</code> – активная область в форме круга;<code>poly</code> – активная область в форме многоугольника;<code>default</code> – активная область занимает всю площадь изображения.
<code>target</code>	<p>Указывает, куда будет загружен документ при переходе по ссылке. Принимает следующие значения:</p> <ul style="list-style-type: none"><code>_self</code> – страница загружается в текущее окно;<code>_blank</code> – страница открывается в новом окне браузера;<code>_parent</code> – страница загружается во фрейм-родитель;<code>_top</code> – страница загружается в полное окно браузера.
<code>type</code>	<p>Указывает MIME-тип файлов ссылки, т.е. расширение файла.</p>

4. 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.



📷 РИС. 1. ПРИМЕР РАЗМЕТКИ ИЗОБРАЖЕНИЯ ДЛЯ СОЗДАНИЯ КАРТЫ

2) Задаем имя карты, добавив его в тег `<map>` с помощью атрибута `name`. Это же значение присваиваем атрибуту `usemap` тега ``.

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%ED%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>
```

1.7. HTML-таблицы

1. Как создать таблицу

Таблица создаётся при помощи парного тега `<table></table>`. Данный тег 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:

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

текст заголовка текст заголовка

данные данные

📷 РИС.1. ВНЕШНИЙ ВИД ТАБЛИЦЫ БЕЗ ФОРМАТИРОВАНИЯ CSS-СВОЙСТВАМИ

2. Как создать строки (ряды) таблицы

Строки или ряды таблицы создаются с помощью тега `<tr>`. Количество горизонтальных строк таблицы определяется количеством парных тегов `<tr></tr>`.

3. Как сделать ячейку заголовка столбца таблицы

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

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

4. Как сделать ячейку тела таблицы

Элемент `<td>` создаёт ячейки таблицы, внутрь которых помещаются данные таблицы. Парные теги `<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>
```


5. Атрибуты элементов таблицы

ТАБЛИЦА 1. АТРИБУТЫ ЭЛЕМЕНТОВ ТАБЛИЦЫ

Атрибут	Описание, принимаемое значение
<code>colspan</code>	<p>Количество ячеек в строке для объединения по горизонтали.</p> <pre><td colspan="3"></pre> <p>Возможные значения: число от 1 до 999.</p>
<code>headers</code>	<p>Задаёт список ячеек заголовка, содержащих информацию о заголовке текущей ячейки данных. Предназначен для речевых браузеров.</p> <pre><th id="идентификатор">...</th> <th headers="идентификатор">...</th></pre> <p>Принимаемые значения: список имен ячеек, разделенных пробелами; эти имена должны быть присвоены ячейкам через их атрибут <code>id</code>.</p>
<code>rowspan</code>	<p>Количество ячеек в столбце для объединения по вертикали.</p> <pre><td rowspan="2"></pre> <p>Возможные значения: число от 1 до 999.</p>
<code>span</code>	<p>Количество колонок, объединяемых для задания единого стиля, по умолчанию равно 1.</p> <pre><col span="2"></pre> <p>Принимаемые значения: любое целое положительное число.</p>