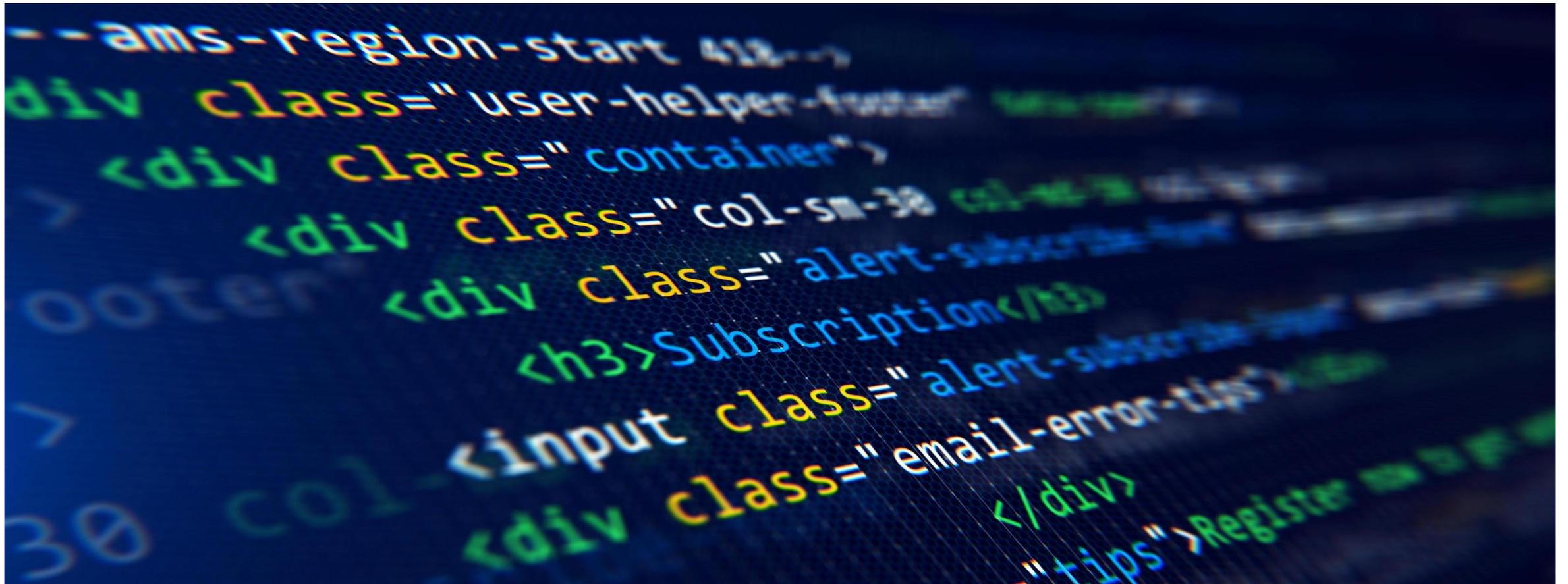


РАЗРАБОТКА КЛИЕНТСКИХ ЧАСТЕЙ ИНТЕРНЕТ-РЕСУРСОВ

ВВЕДЕНИЕ В HTML, ОЗНАКОМЛЕНИЕ СО СТРУКТУРОЙ ЯЗЫКА, РАБОТА С ТЕГАМИ



Ассистент кафедры ИиППО Русляков Алексей Александрович

Почта:

ruslyakov@mirea.ru

ВВЕДЕНИЕ

«язык гипертекстовой разметки» – «скелет», каркас веб-сайта

язык программирования, который показывает как будет вести себя веб-страница по отношению к действию пользователя

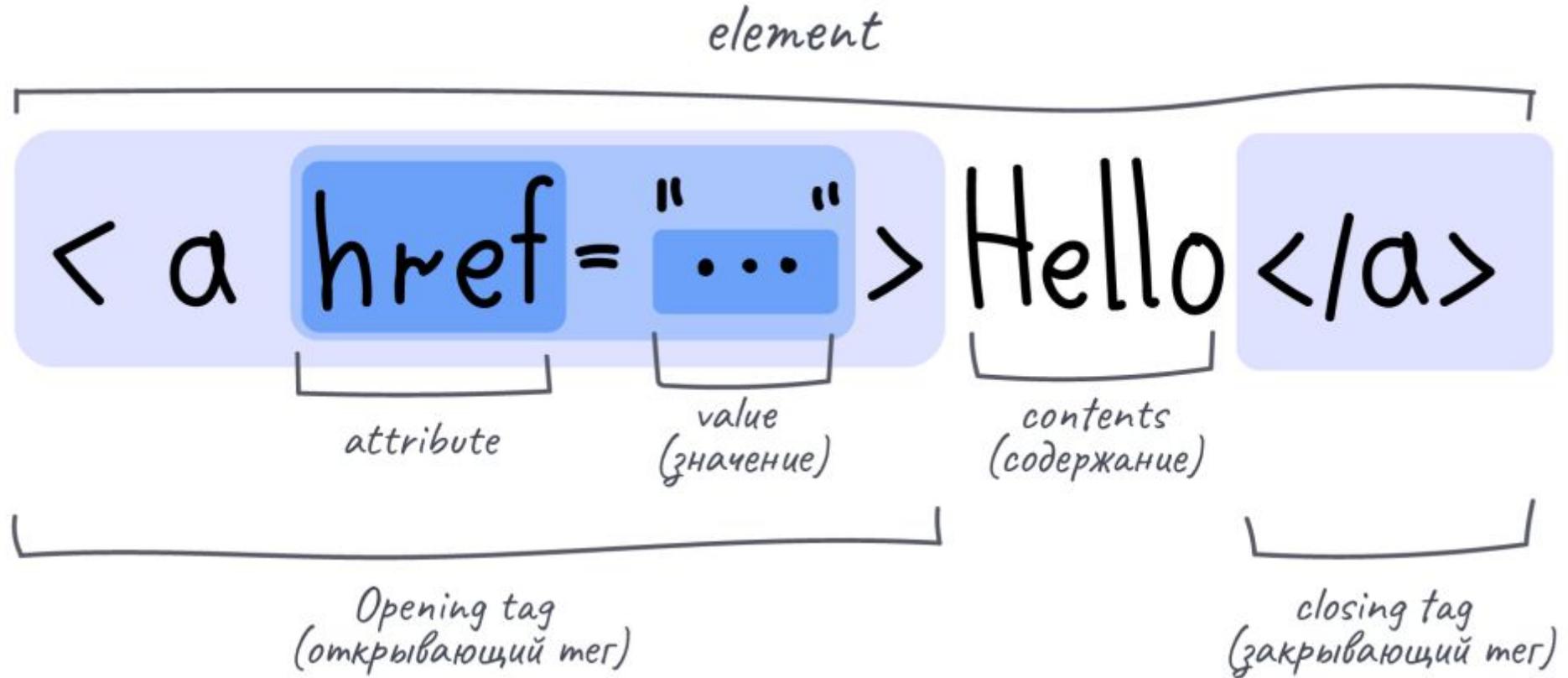


«язык стилей» – это код, который используется для стилизации вашей веб-страницы

РЕДАКТОРЫ КОДА

- **Visual Studio**
(<https://code.visualstudio.com/>)
- **Atom** (<https://atom.io/>)
- **Sublime Text** (<https://www.sublimetext.com/>)
- **Code**
 - **Brackets** <https://brackets.io/>
 - **Notepad++**
(<https://notepad-plus-plus.org/downloads/>)

СТРУКТУРА ЯЗЫКА HTML



ЧТО ТАКОЕ «ТЕГ» (TAG) И «ЭЛЕМЕНТ» (ELEMENT)?

```
<element_name>...содержимое (content)...</element_name>
```

```
<p><em>Тег</em> придает особый акцент тексту.</p>
```

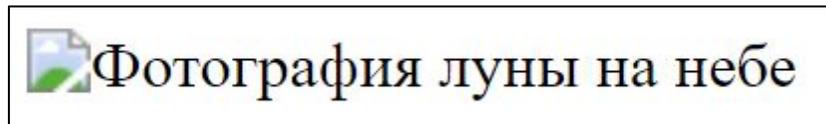
ОДИНОЧНЫЕ ТЕГИ

```
<link>  
<br>  
<meta>  
<hr>  
<img>  
<input>
```

ЧТО ТАКОЕ «АТТРИБУТ» (ATTRIBUTE)?

```

```

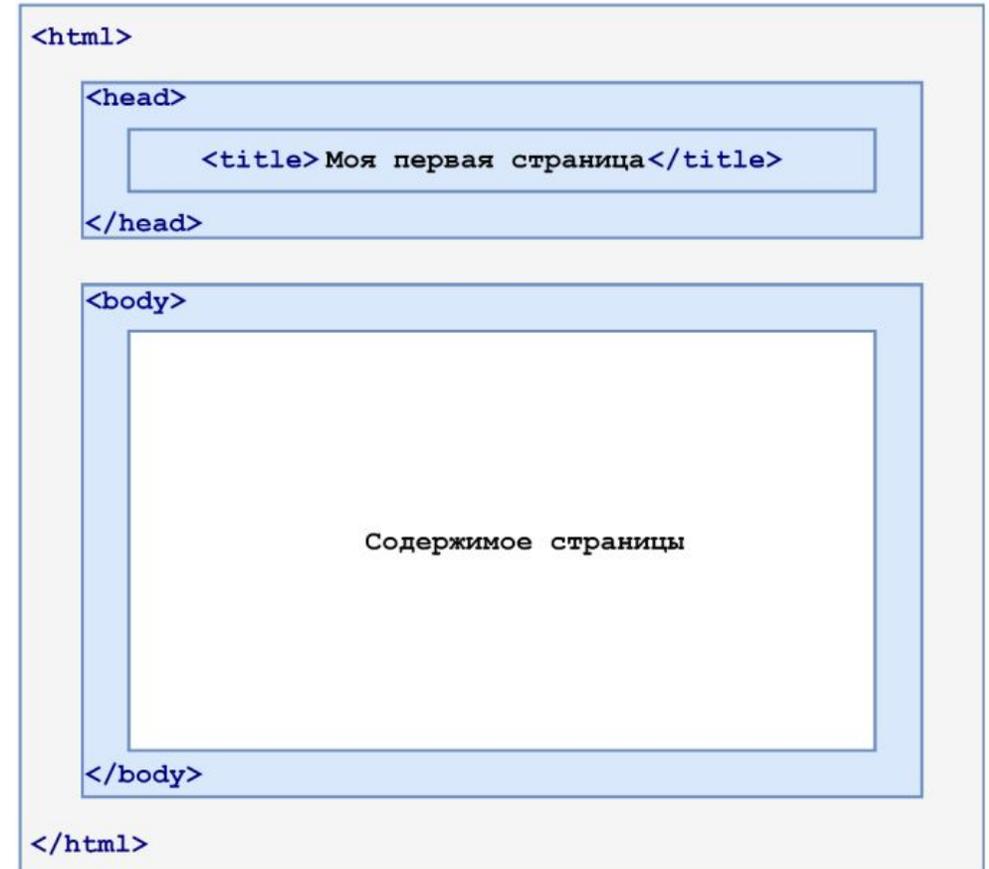


В данном случае источник изображения *source* (`src`), замещающий (*alternative*) изображение текст (`alt`) и всплывающая подсказка с текстом, которая появляется при наведении курсора на элемент – изображение, – являются атрибутами тега ``.

БАЗОВАЯ СТРУКТУРА HTML-ДОКУМЕНТА

```
<!DOCTYPE html> <!-- Тип документа для браузера -->
<html lang="ru"> <!-- Самый главный тег -->
<head> <!-- Вся техническая информация о странице -->
  <meta charset="UTF-8"> <!-- Кодировка страницы-->
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <!-- Для совместимости с IE-->
  <meta name="viewport" content="width=device-width, initial-s
cale=1.0">
  <!-- Для красивого отображения на мобильных устройствах-->
  <title>Document</title> <!-- Заголовок страницы -->
</head>
<body> <!-- Тело веб-страницы -->

</body>
</html>
```



КОММЕНТАРИИ В КОДЕ HTML

```
<!-- Это HTML-комментарий -->
```

```
<h1>Для комментария <!-- Это HTML-комментарий между тегами --></h1>.
```

Но комментарии **не могут** появляться внутри самого HTML-тега, например:

```
<h1 <!-- Это HTML-комментарий внутри тега -->>Для комментария</h1>
```

ВЛОЖЕННЫЕ ЭЛЕМЕНТЫ HTML

```
<body>
  <p>Этот абзац - дочерний элемент body</p>
</body>
```

```
<body>
  <div>
    <h1>Родственник p и потомок body</h1>
    <p>Родственник h1 и потомок body</p>
  </div>
</body>
```

ЗАГОЛОВКИ В HTML

```
<h1>Один</h1><!-- Используется для основных заголовков.  
Все остальные заголовки используются для подзаголовков. -->  
<h2>Два</h2>  
<h3>Три</h3>  
<h4>Четыре</h4>  
<h5>Пять</h5>  
<h6>Шесть</h6>
```

Один

Два

Три

Четыре

Пять

Шесть

ФОРМАТИРОВАНИЕ ТЕКСТА В HTML

```
<body>
  <p>
    Серые глаза – рассвет,
    Пароходная сирена,
    Дождь, разлука, серый след
    За винтом бегущей пены.
    ...
  </p>
</body>
```

<p>

Серые глаза — рассвет, Пароходная сирена, Дождь, разлука, серый след За винтом бегущей пены. ...

<pre>

```
<body>
  <pre>
    Серые глаза – рассвет,
    Пароходная сирена,
    Дождь, разлука, серый след
    За винтом бегущей пены.
    ...
  </pre>
</body>
```

```
Серые глаза – рассвет,
Пароходная сирена,
Дождь, разлука, серый след
За винтом бегущей пены.
...
```

Тег (Tag)	Описание
	Bold (жирный шрифт)
	Strong («сильный/серьезный» шрифт)
<i>	<i>Italic</i> (курсив)
	<i>Emphasised Text</i> (акцентированный текст)
<mark>	Marked Text (выделенный текст)

<small>	Small Text (мелкий шрифт)
<s>	Strikethrough (зачеркнутый)
	Delete (удаленный)
<u>	<u>Unarticulated Text</u> (ранее Underlined – до HTML4) (неясный/невнятный текст)
<ins>	<u>Inserted Text</u> (вставленный текст)

<code><sub></code>	Subscript Text (подстрочный текст)
<code><sup></code>	Superscript Text (надстрочный текст)
<code><q></code>	«Quotation» (цитирование)
<code><cite></code>	<i>Cite</i> (ссылка на источник цитаты)
<code><blockquote></code>	Blockquote (цитирование)

<code></code>	Span (диапазон)
<code><abbr></code>	Abbreviation (Аббревиатура)

<DIV>

<div> – **division** – раздел, секция

В HTML5 и в материалах «живого стандарта» HTML, который актуален в настоящий момент, описание элемента `<div>` стало таким:

Элемент `div` совершенно не несёт в себе какого-то особого смысла. Он представляет свои дочерние элементы. Он может использоваться с атрибутами `class`, `lang`, `title` для обозначения семантики, общей для некоей последовательности элементов. Ещё его можно использовать в элементе `dl`, в качестве контейнера для групп элементов `dt` и `dd`.

Далее в стандарте идёт речь о следующем:

Авторам документов настоятельно рекомендуется рассматривать элемент `div` как элемент, к которому прибегают в самом крайнем случае, когда для решения некоей задачи не подходит ни один другой элемент. Использование более подходящих элементов вместо `div` ведёт к улучшению доступности документов для пользователей и к упрощению поддержки этих документов для их авторов.

ОШИБОЧНО

О

```
<div>Hi there, I'm a paragraph of content!</div>
```

```
<div class=button onclick=foo()>  
  Click me!  
</div>
```

ПРАВИЛЬНО

Для начала — пусть имеется набор абзацев или других элементов, содержащих текст на языке, который отличается от основного языка документа (веб-страницы). Тут `<div>` можно использовать с атрибутом `lang`, содержащим подходящий языковой тег. В этот `<div>` можно включить весь контент на другом языке, это позволит указать на то, что этот язык использован в документе совершенно сознательно.

```
<div lang=fr>  
  <h3><font color="#3AC1EF"> ...</font></h3>  
  <p>...</p>  
  <ul>  
    <li>...  
    <li>...  
    <li>...  
    ...  
  </ul>  
  ...  
</div>
```

Здесь применение контейнера `<div>` с атрибутом `lang` гораздо легче применения этого атрибута к каждому элементу, включённому в этот контейнер. Кроме того, если исходить из предположения о том, что этот контент представляет собой нечто особенное, используемое только на конкретной странице, окажется, что применение `<div>` тут допустимо без всяких вопросов.

ЯКОРЯ И ГИПЕРССЫЛКИ В HTML

Атрибут (Attribute)	Описание
<code>href</code>	Указывает адрес назначения. Это может быть абсолютный или относительный URL-адрес или имя (<code>name</code>) якоря. Абсолютный URL — это полный URL-адрес веб-сайта, например, <code>http://example.com/</code> . Относительный URL-адрес указывает на другой каталог и/или документ на том же веб-сайте, например, <code>/about-us/</code> указывает на каталог «о нас» внутри корневого каталога (<code>/</code>). При указании на другой каталог без явного указания документа веб-серверы обычно возвращают документ « <code>index.html</code> » внутри этого каталога.
<code>hreflang</code>	Идентифицирует язык текста по ссылке.
<code>rel</code>	Указывает связь между текущим документом и связанным документом.
<code>target</code>	Указывает, где открыть ссылку, например, в новой вкладке или окне. Возможные значения: <code>_blank</code> , <code>_self</code> , <code>_parent</code> , <code>_top</code> .
<code>title</code>	Указывает дополнительную информацию о ссылке. Информация чаще всего отображается в виде всплывающей подсказки при наведении курсора на ссылку. Этот атрибут не ограничивается ссылками, его можно использовать почти во всех тегах HTML.
<code>download</code>	Указывает, что цель будет загружена, когда пользователь щелкнет гиперссылку. Значением атрибута будет имя загружаемого файла. Ограничений на допустимые значения нет, а браузер автоматически определит правильное расширение файла и добавит его к файлу (<code>.img</code> , <code>.pdf</code> и т.д.). Если значение опущено, используется исходное имя файла.

<a>

ССЫЛКА НА ДРУГОЙ САЙТ

<a> – anchor – якорь, ссылка

href – hypertext reference – гипертекстовая ссылка

rel – relationship – отношение

Основное простое использование элемента **<a>**:

```
<a href="https://www.google.ru/">Пример ссылки</a>
```

Пример ссылки с атрибутами **href** и **rel**:

```
<a href="https://www.google.ru/" rel="external">Пример ссылки</a>
```

ССЫЛКА-ЯКОРЬ

1. Создание страницы **index.html**

2. Создание тем на странице:

```
<h2>Первая тема</h2>  
<p>Описание первой темы</p>  
<h2>Вторая тема</h2>  
<p>Описание второй темы</p>
```

3. Присвоение темам атрибута **id**:

```
<h2 id="Topic1">Первая тема</h2>  
<p>Описание первой темы</p>  
<h2 id="Topic2">Вторая тема</h2>  
<p>Описание второй темы</p>
```

ССЫЛКА-ЯКОРЬ

4. Создание оглавления через якоря (символы #):

```
<h1>Содержание страницы</h1>  
  
<a href="#Topic1">Нажать для перехода на первую тему</a>  
<a href="#Topic2">Нажать для перехода на вторую тему</a>
```

5. Ссылка на тему/заголовок на другой странице:

```
<a href="index.html#Topic1">Вернуться к первой теме на главной  
странице</a>
```

ССЫЛКА НА НОМЕР ТЕЛЕФОНА

Если значение атрибута **href** начинается с **tel:**, тогда ваше устройство будет набирать номер при нажатии на него:

```
<a href="tel:01234567890">Позвонить по номеру</a>
```

ССЫЛКА НА ПОЧТОВЫЙ КЛИЕНТ

Если значение атрибута **href** начинается с **mailto:** браузер попытается открыть почтовый клиент по клику:

```
<a href="mailto:sendemail@gmail.com">Написать письмо</a>
```

ССЫЛКА НА ПОЧТОВЫЙ КЛИЕНТ: КОПИЯ (СС) И СКРЫТАЯ КОПИЯ (ВСС)

Адреса для получателей копии (**cc** - carbon copy) или скрытой копии (**bcc** - blind carbon copy):

```
<a href="mailto:sendemail@gmail.com?cc=copy@mail.ru&bcc=blindcopy@mail.ru">  
    Написать письмо</a>
```

ТЕМА И ОСНОВНОЙ ТЕКСТ

Тема (**subject**) и тело (**body**) нового письма:

```
<a href="mailto:sendemail@mail.ru?subject=Example+subject&body=Message+text">  
    Отправить письмо</a>
```

СПИСКИ В HTML

Три способа задания списков:

1. Упорядоченные (нумерованные) списки (**O**rdered **L**ist)

```
<ol> ... </ol>
```

2. Неупорядоченные (маркированные) списки (**U**nordered **L**ist)

```
<ul> ... </ul>
```

3. Списки описаний (**D**escription **L**ist)

```
<dl> ... </dl>
```

СПИСКИ В HTML: УПОРЯДОЧЕННЫЙ СПИСОК (ORDERED LIST)

`` – ordered list

`` – list item – элемент (пункт) списка

```
<ol>  
  <li>Жарко</li>  
  <li>Тепло</li>  
  <li>Прохладно</li>  
  <li>Холодно</li>  
</ol>
```

1. Жарко
2. Тепло
3. Прохладно
4. Холодно

СПИСКИ В HTML: УПОРЯДОЧЕННЫЙ СПИСОК (ORDERED LIST)

Атрибут `start`

```
<ol start="5">  
  <li>Жарко</li>  
  <li>Тепло</li>  
  <li>Прохладно</li>  
  <li>Холодно</li>  
</ol>
```

5. Жарко
6. Тепло
7. Прохладно
8. Холодно

Атрибуты `start` и `value`

```
<ol start="5">  
  <li>Жарко</li>  
  <li value="9">Тепло</li>  
  <li>Прохладно</li>  
  <li>Холодно</li>  
</ol>
```

5. Жарко
9. Тепло
10. Прохладно
11. Холодно

СПИСКИ В HTML: УПОРЯДОЧЕННЫЙ СПИСОК (ORDERED LIST)

Атрибуты **type**, **start** и **value** (со списком в алфавитном порядке)

```
<ol type="A" start="5">  
  <li>Жарко</li>  
  <li value="9">Тепло</li>  
  <li>Прохладно</li>  
  <li>Холодно</li>  
</ol>
```

Е. Жарко
I. Тепло
J. Прохладно
K. Холодно

Атрибут **type**

Тип (type)	Описание	Примеры
1	Десятичные числа (по умолчанию)	1, 2, 3, 4, 5
a	Строчные буквы в алфавитном порядке	a, b, c, d, e
A	Заглавные буквы в алфавитном порядке	A, B, C, D, E
i	Строчные римские цифры	i, ii, iii, iv, v
I	Заглавные римские цифры	I, II, III, IV, V

СПИСКИ В HTML: УПОРЯДОЧЕННЫЙ СПИСОК (ORDERED LIST)

Атрибут `reversed`

```
<ol reversed>  
  <li>Жарко</li>  
  <li value="9">Тепло</li>  
  <li>Прохладно</li>  
  <li>Холодно</li>  
</ol>
```

4. Жарко
9. Тепло
8. Прохладно
7. Холодно

Примечание: использование `` для отображения списка элементов необходимо в тех случаях, когда **важен порядок элементов и его следует подчеркнуть**. Если изменение порядка элементов не делает список неправильным, следует использовать ``.

СПИСКИ В HTML: НЕУПОРЯДОЧЕННЫЙ СПИСОК (UNORDERED LIST)

`` – unordered list

`` – list item – элемент (пункт) списка

```
<ul>  
  <li>Жарко</li>  
  <li>Тепло</li>  
  <li>Прохладно</li>  
  <li>Холодно</li>  
</ul>
```

- Жарко
- Тепло
- Прохладно
- Холодно

СПИСКИ В HTML: НЕУПОРЯДОЧЕННЫЙ СПИСОК (UNORDERED LIST)

Атрибут `type`

Тип (<code>type</code>)		Описание
●	<code>disc</code>	Список с маркерами в виде круга
○	<code>circle</code>	Список с маркерами в виде окружности
■	<code>square</code>	Список с квадратными маркерами

Примечание: использование `` для отображения списка элементов необходимо в тех случаях, когда **порядок элементов не важен**. Если изменение порядка элементов делает список неправильным, следует использовать ``.

СПИСКИ В HTML: СПИСОК ОПИСАНИЙ (DESCRIPTION LIST)

`<dl>` – description list

`<dt>` – description list term – термин

`<dd>` – description list description – значение (описание)

```
<dl>
  <dt>Термин 1</dt>
  <dd>Описание термина 1</dd>

  <dt>Термин 2</dt>
  <dd>Описание термина 2</dd>
  <dd>Описание термина 2</dd>

  <dt>Термин 3</dt>
  <dd>Описание термина 3</dd>
</dl>
```

Термин 1
 Описание термина 1
Термин 2
 Описание термина 2
 Описание термина 2
Термин 3
 Описание термина 3

Примечание: группа «имя-значение» может иметь более одного имени и/или более одного значения, которые представляют альтернативы.

ИЗОБРАЖЕНИЯ В HTML

`` – **image** – GIF, JPEG, SVG или PNG

Пример синтаксиса с локальной ссылкой (значения **width** и **height** проставлены в px):

```

```

Пример синтаксиса с URL-адресом:

```

```

ИЗОБРАЖЕНИЯ В HTML: АТТРИБУТЫ ТЕГА ``

Атрибут (Attribute)	Описание
<code>src</code>	Указывает локальный или URL-адрес изображения (<i>обязательный</i>).
<code>srcset</code>	Изображения для использования в различных ситуациях, например, на дисплеях с высоким разрешением, небольших мониторах и т.д. <i>Рекомендуется использовать с адаптивными изображениями.</i>
<code>alt</code>	Заменяющий текст для использования, когда изображения недоступны (<i>обязательный; при необходимости его значение можно оставлять пустым – для программ чтения с экрана, чтобы содержание страницы оставалось логичным в случае выполнения значения атрибута на странице</i>).
<code>width</code>	Ширина изображения. Используется в случае, когда нужно задать пространство, занимаемое изображением, чтобы обеспечить стабильность макета страницы перед его загрузкой
<code>height</code>	Высота изображения. Используется в случае, когда нужно задать пространство, занимаемое изображением, чтобы обеспечить стабильность макета страницы перед его загрузкой
<code>ismap</code>	Используется для указания изображения в качестве карты изображения на стороне сервера.
<code>usemap</code>	Используется для указания изображения в качестве карты изображения на стороне клиента.

ИЗОБРАЖЕНИЯ В HTML: АТРИБУТЫ ТЕГА ``

<code>loading</code>	Используется, чтобы указать, должен ли браузер откладывать загрузку изображений до тех пор, пока не будут выполнены некоторые условия, или загрузить изображение немедленно.
<code>decoding</code>	Представляет собой подсказку браузеру о том, как он должен декодировать изображение (синхронно, асинхронно, по умолчанию).
<code>referrerpolicy</code>	Он используется для указания того, какую информацию о реферере использовать при извлечении изображения, т.е. <code>no-referrer</code> , <code>no-referrer-when-downgrade</code> , <code>origin</code> , <code>origin-when-cross-origin</code> , <code>unsafe-url</code> .
<code>sizes</code>	Размеры изображений для различных макетов страниц. <i>Рекомендуется использовать с адаптивными изображениями.</i>
<code>crossorigin</code>	Обработка перекрестных запросов элементом.

Примечание: альтернативный текст (`alt`) используется программами чтения с экрана для слабовидящих пользователей и поисковыми системами. Поэтому важно написать хороший альтернативный текст для ваших изображений.

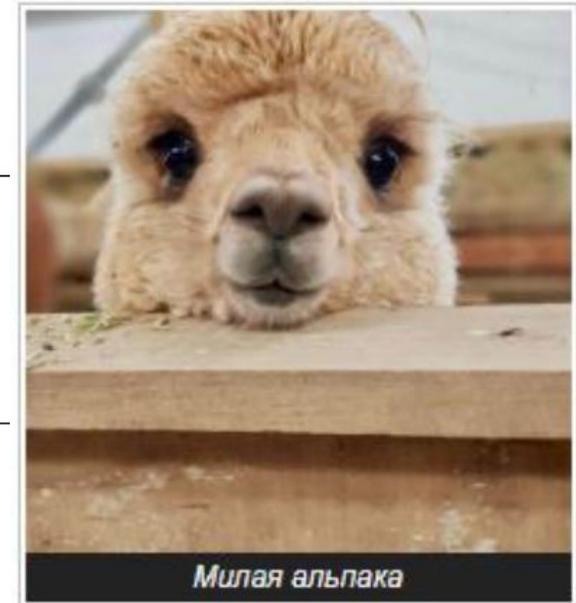
ИЗОБРАЖЕНИЯ В HTML: ИЛЛЮСТРАЦИЯ С НЕОБЯЗАТЕЛЬНОЙ ПОДПИСЬЮ

`<figure>`

`<figure>` – иллюстрации, диаграммы, фотографии или коды

`<figcaption>` – подпись (заголовок) иллюстрации

```
<figure>
  
  <figcaption>Милая альпака</figcaption>
</figure>
```



ИЗОБРАЖЕНИЯ В HTML: КАРТА ИЗОБРАЖЕНИЯ

`` – изображение

`<map>` – карта

`<area>` – кликабельная область на карте

Атрибуты `usemap`, `name` и `alt`

Атрибут	Описание
<code>usemap</code>	Атрибут тега <code></code> . Имя карты с добавленным к нему символом <code>#</code> (решёткой). Например, для карты с <code>name="map"</code> изображение должно иметь <code>usemap="#map"</code> .
<code>name</code>	Атрибут тега <code><map></code> . Название карты для ее идентификации. Взаимодействует с атрибутом <code>usemap</code> изображения.
<code>alt</code>	Атрибут тега <code><area></code> . Альтернативный текст в случае, если изображения не поддерживаются. Это необходимо только в том случае, если <code>href</code> также установлен в <code><area></code> .

ИЗОБРАЖЕНИЯ В HTML: КАРТА ИЗОБРАЖЕНИЯ

Атрибуты `coords`, `href` и `shape`

<code>coords</code>	Атрибут тега <code><area></code> . Координаты, очерчивающие выбираемую область. Для <code>shape="polygon"</code> – координаты: "x, y", разделенных запятыми (<code>shape="polygon" coords="x1, y1, x2, y2, x3, y3, ... "</code>). Для <code>shape="rectangle"</code> – координаты: <code>left, top, right, bottom</code> . Для <code>shape="circle"</code> – координаты: <code>centerX, centerY, radius</code> .
<code>href</code>	URL-адрес гиперссылки, если он указан. Если его опустить, <code><area></code> не будет представлять собой гиперссылку.
<code>shape</code>	Форма <code><area></code> . Позволяет установить значение по умолчанию для выбора всего изображения (атрибут <code>coords</code> не требуется), <code>circle</code> или <code>circ</code> для окружности, <code>rectangle</code> или <code>rect</code> для прямоугольника и <code>polygon</code> или <code>poly</code> для многоугольной области, заданной угловыми точками.

```

<map name="shapes">
  <area shape="polygon" coords="79,6,5,134,153,134">
  <area shape="rectangle" coords="177,6,306,134">
  <area shape="circle" coords="397,71,65">
</map>
```

Примечание: браузер распознает области, когда курсор становится указателем.

ФАВИКОН В HTML

Фавикон (*favicon*) – это крошечный значок, который отображается в адресной строке браузера, вкладках страниц и меню закладок.

Подключается в коде между тегами `<head>` и `</head>` после элемента `<title>`:

- PNG-фавиконка:

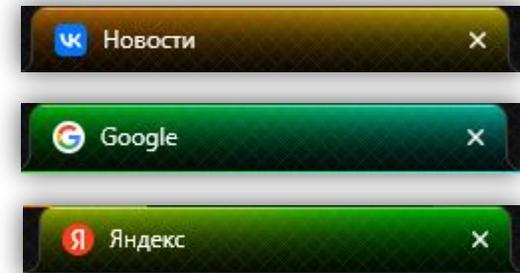
```
<link rel="icon" type="image/png" href="/favicon.png">
```

- icon-фавиконка (*.ico):

```
<link rel="icon" href="/favicon.ico" sizes="any">
```

- SVG-фавиконка:

```
<link rel="icon" type="image/svg+xml"
      href="images/favicons/icon.svg">
```



Примечание: фавиконку в формате ico необходимо класть в корень проекта, так как любой веб-сервер всегда ищет favicon.ico в корне проекта и пытается её подключить к сайту.