

HTML Basic knowledges

Vira Huskova



HTML Basic knowledges:

- syntax
- doctype
- metadatas
- basic charecter entities

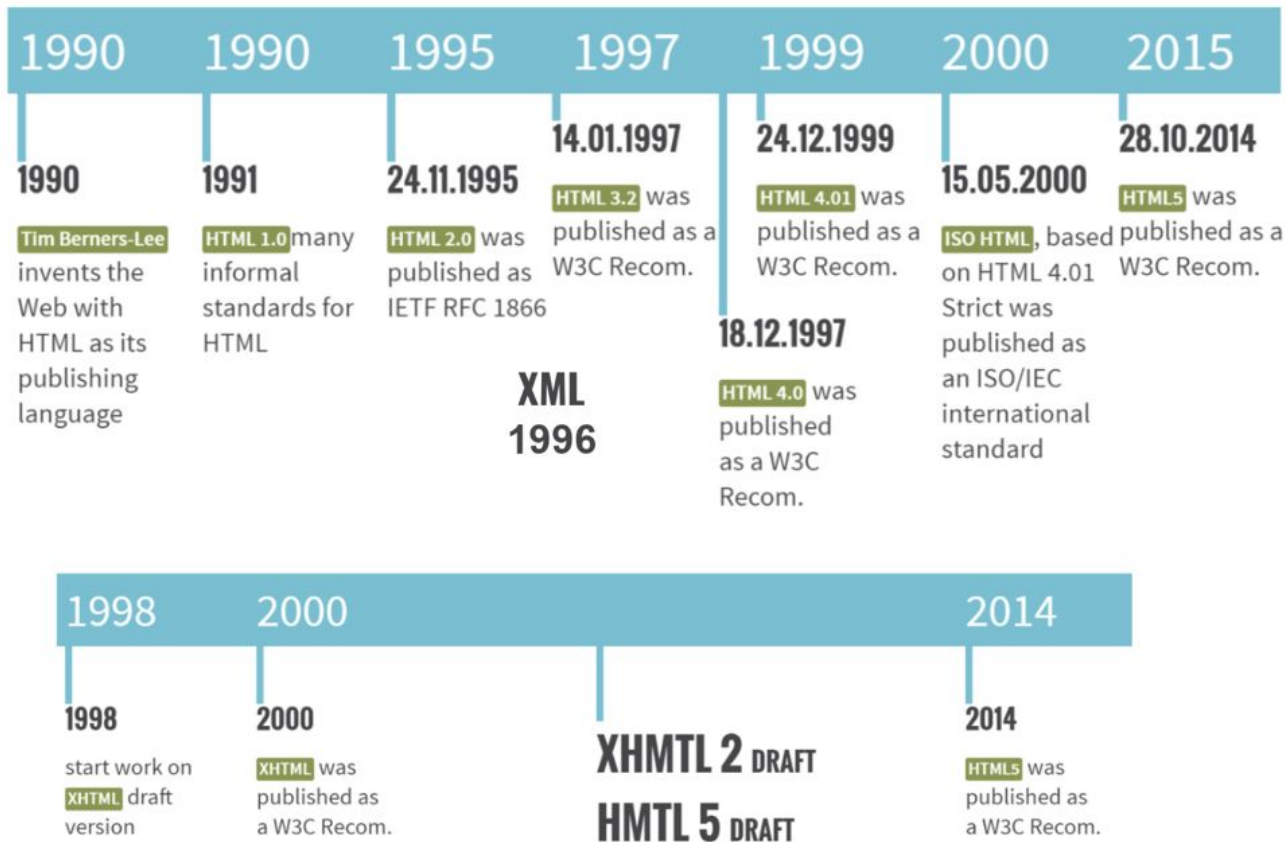
- HTML all(include HTML5):
- tags
- attributes

HTML

HyperText Markup Language



История



Инструменты для работы

Редакторы



Отладка кода



Другие



<https://www.w3.org/standards/webdesign/htmlcss>

https://www.w3schools.com/html/html5_syntax.asp

<https://google.github.io/styleguide/htmlcssguide.html>

Online сервисы



JSFiddle

The Web Playground



JS Bin

*A JavaScript, HTML
and CSS playground*



CodePen

*A front-end
playground*



ICEcoder

Online IDE



Divshot

*The Interface Builder
for Web Apps*



Codio

*Web IDE for web
development, teaching
and learning*



Codeanywhere

*Multi-platform cloud
editor*



SourceLair
















*In-browser IDE for
web technologies*

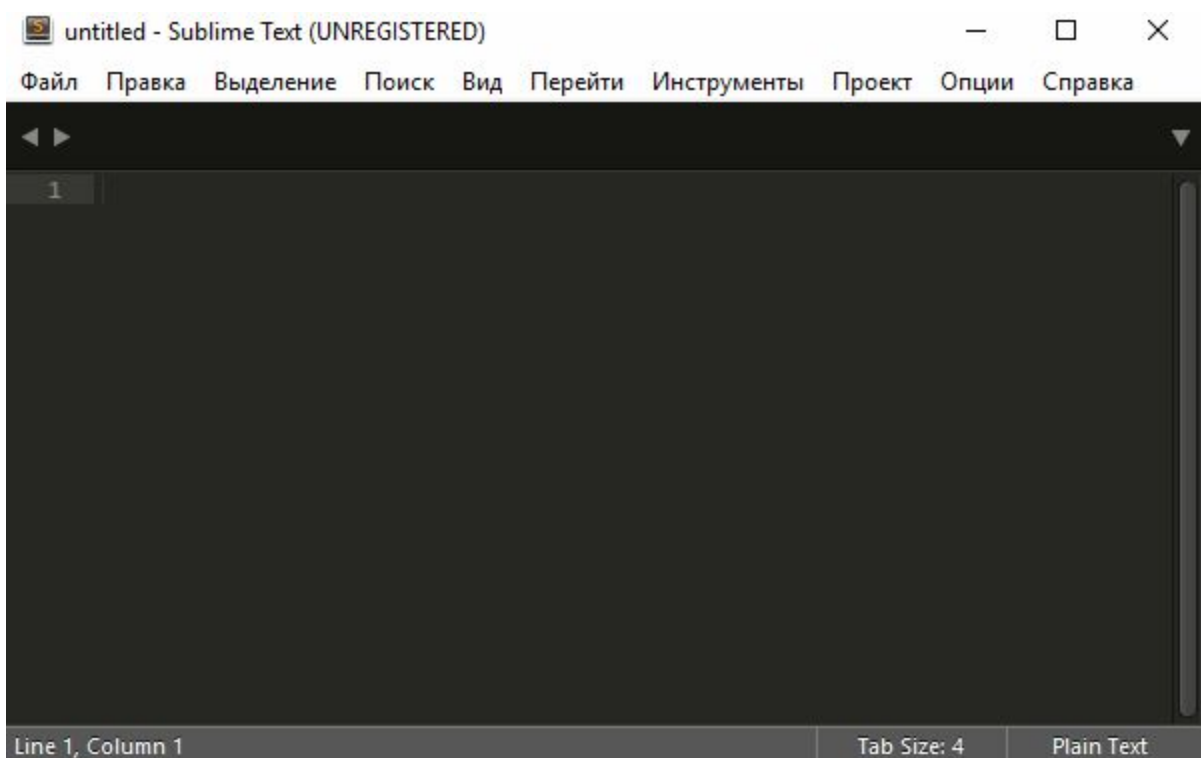


Litmus

*The code editor built to
streamline email
development workflow*

Emmet

 Sublime Text <i>cross-platform</i>	 Atom <i>cross-platform</i>	 Coda <i>OS X</i>
 Eclipse/Aptana <i>cross-platform</i>	 TextMate 2 <i>OS X</i>	 Espresso <i>OS X</i>
 Chocolat <i>OS X</i>	 Komodo Edit <i>cross-platform</i>	 Notepad++ <i>Windows</i>
 PSPad <i>Windows</i>	 textarea <i>browser based, cross-platform</i>	 CodeMirror <i>browser based, cross-platform</i>
 Brackets <i>cross-platform</i>	 NetBeans <i>cross-platform</i>	 Adobe Dreamweaver <i>Windows, OS X</i>



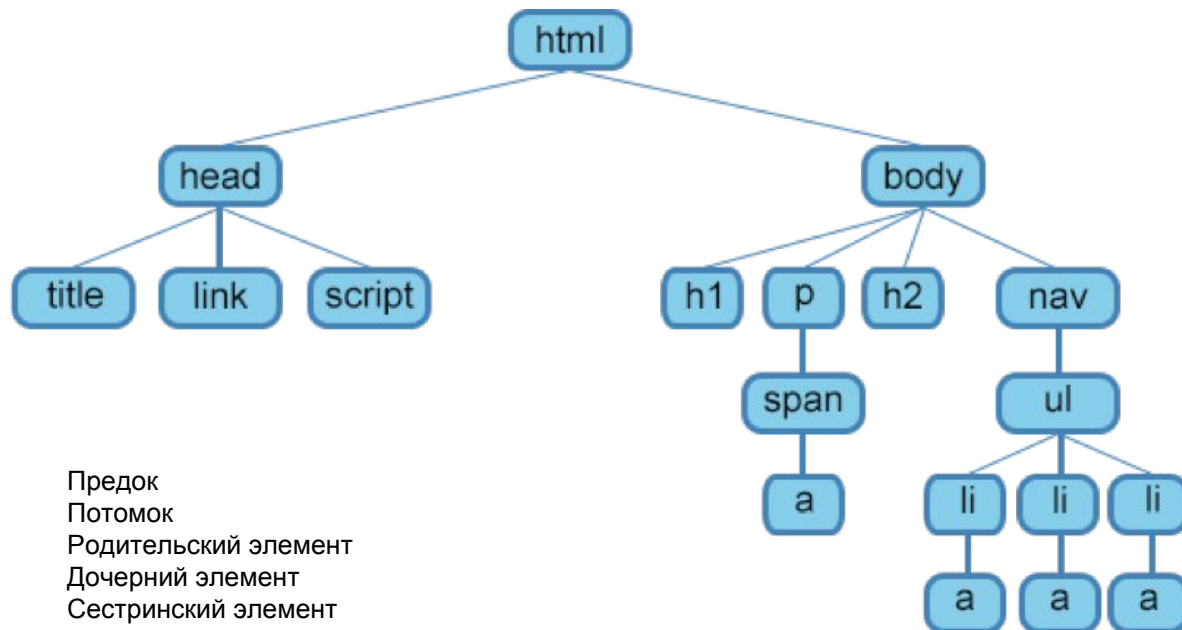

```
1  <!DOCTYPE html>
   <html>
   <head>
     <meta charset="UTF-8">
     <meta name="viewport" content="width=device-width, initial-s...
     <meta http-equiv="X-UA-Compatible" content="ie=edge">
     <title>Document</title>
   </head>
   <body lang="en">

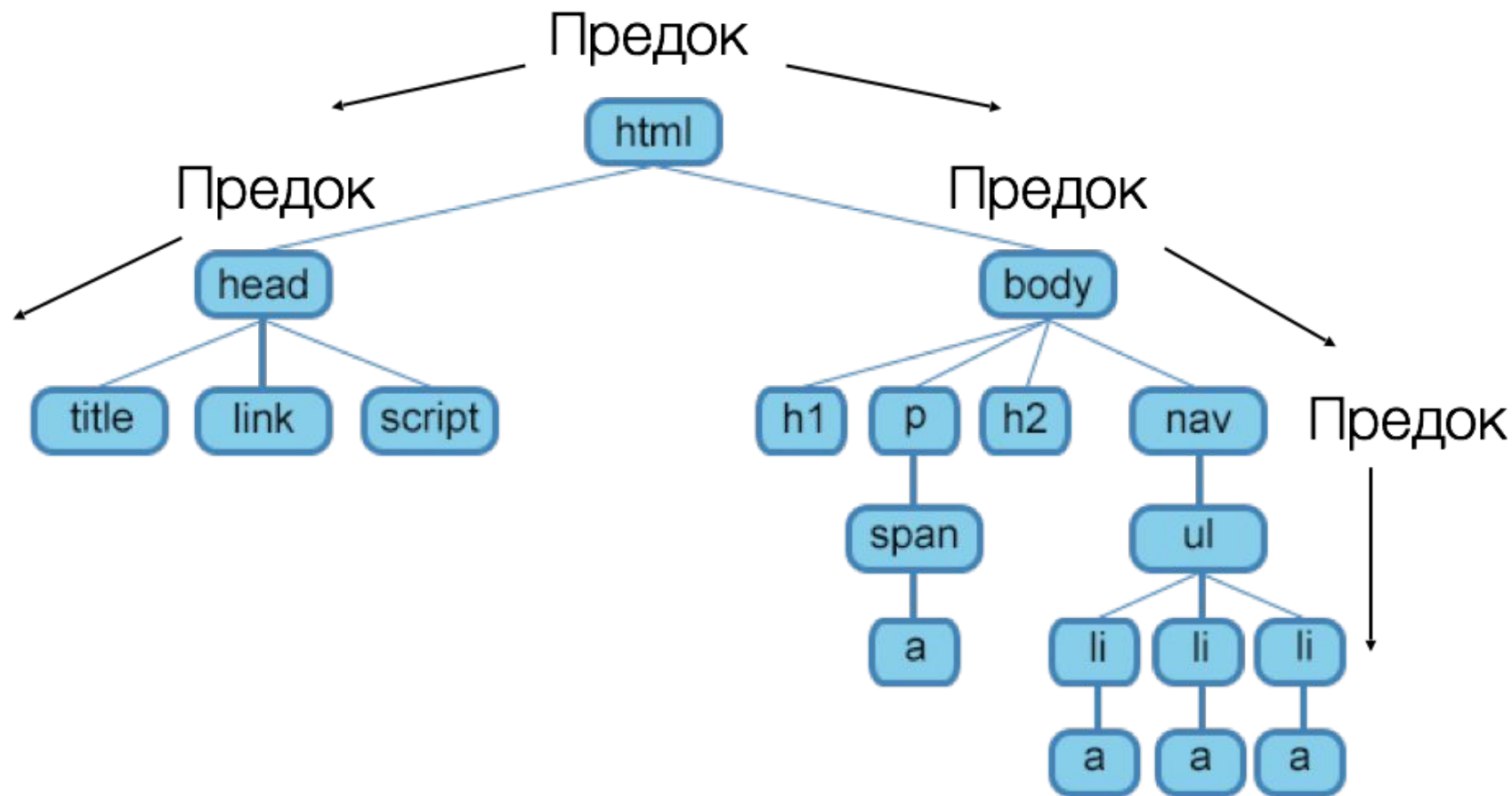
   </body>
   </html>
```

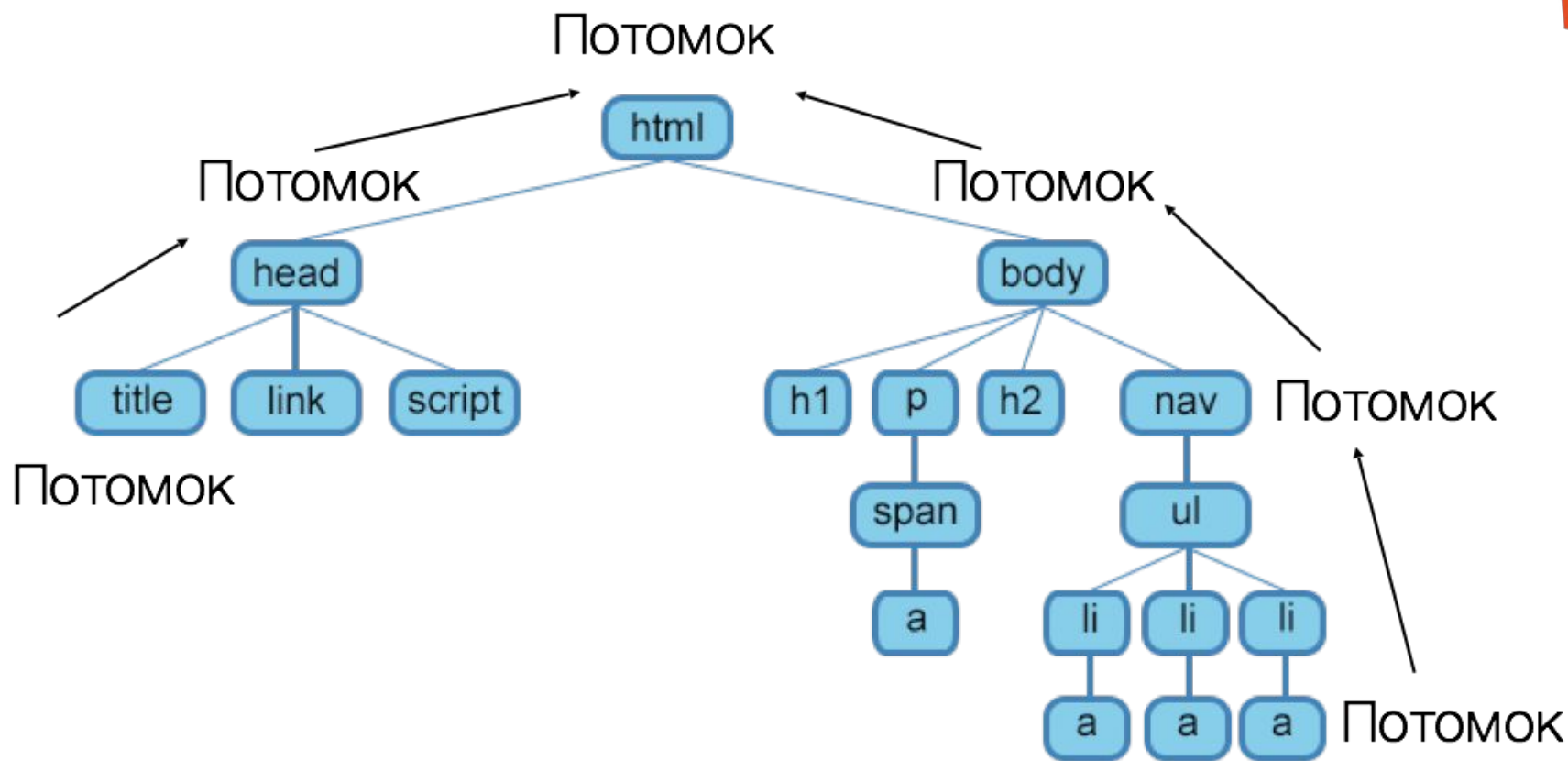
HTML-документ



Каждый элемент обозначается в исходном документе начальным (открывающим) и конечным (закрывающим) тегом (за редким исключением).









```
<!DOCTYPE html>
<html>
  <head>
    <title>!DOCTYPE</title>
    <meta charset="utf-8">
  </head>
  <body>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et
dolore magna aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris nisi ut aliquip ex
ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu
fugiat nulla pariatur. Excepteur sint occaecat cupidatat
non proident, sunt in culpa qui officia deserunt mollit
anim id est laborum.
    </p>
  </body>
</html>
```

<!DOCTYPE>

document type definition



Предназначен для указания типа текущего документа, необходимо, чтобы браузер понимал, как следует интерпретировать текущую веб-страницу.

```
<!DOCTYPE [Элемент верхнего уровня] [Публичность] "[Регистрация]//[Организация]//[Тип]
[Имя]//[Язык]" "[URL]">
```

```
<!DOCTYPE html>
<html xmlns="
<head>
<meta http
htt
```

```
<!DOCTYPE [Элемент верхнего уровня] [Публичность] "[Регистрация]/[Организация]/[Тип] [Имя]/[Язык]"
"[URL]">
```

Элемент верхнего уровня — указывает элемент верхнего уровня в документе, для HTML это тег **<html>**.

Публичность — объект является публичным (значение PUBLIC) или системным ресурсом (значение SYSTEM), например, таким как локальный файл.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

ExportString[Cell["Some text", "Text"], "XML"]

```
<?xml version='1.0'?>
<!DOCTYPE Cell SYSTEM 'http://www.wolfram.com/XML/notebookxml1.d
<Cell class='Text'
  CreatedBy='Mathematica 4.2'
  xmlns:xhtml='http://www.w3.org/1999/xhtml'
  xmlns:mathematica='http://www.wolfram.com/XML/'
  xmlns='http://www.wolfram.com/XML/'>
  <String>Some text</String>
  <Style>
    <String>Text</String>
  </Style>
</Cell>
```

```
<!DOCTYPE [Элемент верхнего уровня] [Публичность] "[Регистрация]//[Организация]//[Тип] [Имя]//[Язык]"
"[URL]">
```

Регистрация — сообщает, что разработчик DTD(**Document Type Definition**) зарегистрирован в международной организации по стандартизации (International Organization for Standardization, ISO). Принимает одно из двух значений: плюс (+) — разработчик зарегистрирован в ISO и - (минус) — разработчик не зарегистрирован. Для W3C значение ставится «-».

Организация — уникальное название организации, разработавшей DTD. Официально HTML/XHTML публикует W3C, это название и пишется в **<!DOCTYPE>**.

Тип — тип описываемого документа. Для HTML/XHTML значение указывается DTD.

Имя — уникальное имя документа для описания DTD.

Язык — язык, на котором написан текст для описания объекта. Содержит две буквы, пишется в верхнем регистре. Для документа HTML/XHTML указывается английский язык (EN).

URL — адрес документа с DTD.

DOCTYPE	Описание
HTML 4.01	
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">	Строгий синтаксис HTML.
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">	Переходный синтаксис HTML.
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">	В HTML-документе применяются фреймы.
HTML 5	
<!DOCTYPE html>	Для всех документов.
XHTML 1.0	
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">	Строгий синтаксис XHTML.
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">	Переходный синтаксис XHTML.
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">	Документ написан на XHTML и содержит фреймы.
XHTML 1.1	
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">	Разработчики XHTML 1.1 предполагают, что он постепенно вытеснит HTML. Никакого деления на виды это определение не имеет, синтаксис один и подчиняется четким правилам.

<html></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.

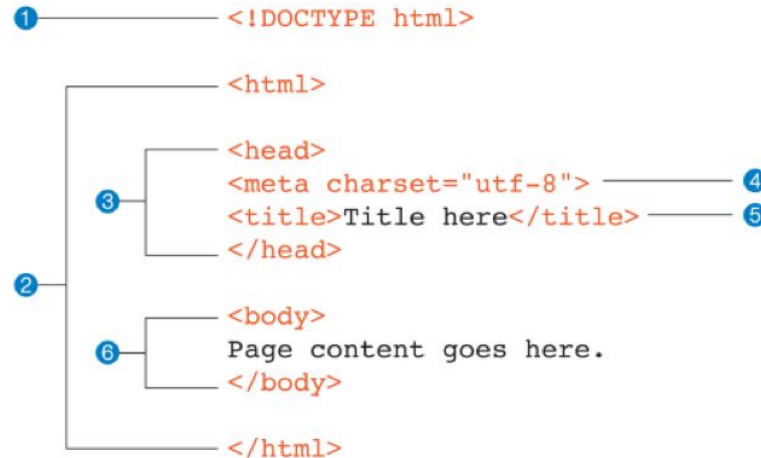
Для элемента доступны атрибуты manifest и xmlns, а также глобальные атрибуты.



<head></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.



<title></title>

Обязательным тегом раздела <head> является тег <title>.

Текст 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.

```
<!DOCTYPE html>

<html>

<head>

  <title>Title of the document</title>

</head>

<body>

The content of the document.....

</body>

</html>
```

simple.html

```
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Level-One Header</h1>
    <p id="RipVanWinkle">
      RipVanWinkle paragraph.
    </p>
  </body>
</html>
```

A Browser Window



<meta>



Задается описание содержимого страницы и ключевые слова для поисковых машин

```
<meta charset="UTF-8">
```

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

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

```
<meta name="author" content="John Doe">
```

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

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
{% block meta %}
<!--Facebook-->
<meta property="og:url" content="{{ request.scheme }}//{{ request.get_host }}{{ request.path }}" />
<meta property="og:type" content="product" />
<meta property="og:title" content="{{ object.style.human_name }} - {{ object.choreographer }} - {{ object.song }}" />
<meta property="og:description" content="{{ object.description|mark2|striptags }}" />
{% thumbnail object.poster '1024x768' as im %}
  <meta property="og:image" content="{{ request.scheme }}//{{ request.get_host }}{{ im.url }}" />
{% endthumbnail %}

<meta name="description" content="{{ seo_object.description }}">
<meta name="keywords" content="{{ seo_object.keywords }}">
<meta name="robots" content="{{ seo_object.robots_meta_tag }}" />

{% endblock meta %}
```

<link>

Задать стили для документа можно записав их в отдельный файл с расширением .css, например, style.css.

```
<!DOCTYPE HTML>

<html>

  <head>

    <meta charset="utf-8">

    <title>Ter LINK</title>

    <link rel="stylesheet" href="ie.css">

    <link rel="alternate" type="application/rss+xml"
      title="Статьи с сайта htmlbook.ru" href="http://htmlbook.ru/rss.xml">

    <link rel="shortcut icon" href="http://htmlbook.ru/favicon.ico">








  </head>

  <body>

    <p>...</p>

  </body>

</html>
```

 Internet Explorer	 Chrome	 Opera	 Safari	 Firefox	 Android	 iOS
✓ 3.0+	✓ 1.0+	✓ 4.0+	✓ 1.0+	✓ 1.0+	✓ 1.0+	✓ 1.0+

HTML	<pre><head> <link атрибуты> </head></pre>
XHTML	<pre><head> <link атрибуты /> </head></pre>

Атрибуты

charset

Кодировка связываемого документа.

href

Путь к связываемому файлу.

media

Определяет устройство, для которого следует применять стилевое оформление.

rel

Определяет отношения между текущим документом и файлом, на который делается ссылка.

sizes

Указывает размер иконок для визуального отображения.

type

MIME-тип данных подключаемого файла.

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

<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.

<script> может располагаться в заголовке или теле HTML-документа в неограниченном количестве.

Однако скрипты, которые должны выполняться в первую очередь, обычно помещают в заголовок документа.

```
<script type="тип">
```

```
...
```

```
</script>
```

```
<script type="тип" src="URL"></script>
```


Including a Dart app

Dart app source file

Identifies a Dart app

Bootstrap script

```
<!DOCTYPE html>

<html>
  <head>
    <title>A Minimalist App</title>
  </head>

  <body>
    <p id="RipVanWinkle">
      RipVanWinkle paragraph.
    </p>
    <script type="application/dart" src="main.dart"></script>
    <script src="packages/browser/dart.js"></script>
  </body>
</html>
```

async

Загружает скрипт асинхронно.

defer

Откладывает выполнение скрипта до тех пор, пока вся страница не будет загружена полностью.

language

Устанавливает язык программирования на котором написан скрипт.

src

Адрес скрипта из внешнего файла для импорта в текущий документ.

type

Определяет тип содержимого тега **<script>**.

<body></body>

HTML тег <body> определяет содержимое (контент) HTML-документа, которое отображается в окне браузера (текст, ссылки, картинки (изображения), таблицы, списки и т.д.).

Он должен быть единственным в документе и всегда должен располагаться внутри элемента <html>, сразу после элемента <head>.

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

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

Универсальные атрибуты применяются практически ко всем тегам, поэтому выделены в отдельную группу, чтобы не повторять их для всех тегов.

accesskey Позволяет получить доступ к элементу с помощью заданного сочетания клавиш.

Internet Explorer: Alt + S

Chrome: Alt + S

Opera: Shift + Esc, S

Safari: Alt + S

Firefox: Shift + Alt + S

class Определяет имя класса, которое позволяет связать тег со стилевым оформлением.

contenteditable Сообщает, что элемент доступен для редактирования пользователем.

```
<h1 contenteditable="true">Тортюга</h1>
<p contenteditable="true">А вслед за этим на Тортюге наступили
дни кипучей деятельности: суда оснащались в дальний поход,
заготавливалось мясо и другие продукты, грузились различные
запасы, необходимые для военных действий.</p>
```

contextmenu Устанавливает контекстное меню для элемента.

dir Задаёт направление и отображение текста — слева направо или справа налево.

```
dir={ltr | rtl}
```

hidden Скрывает содержимое элемента от просмотра.

id Указывает имя стилевого идентификатора.

lang Браузер использует значение параметра для правильного отображения некоторых национальных символов.

```
<p>Цитата на французском языке: <q lang="fr">Ce que femme veut, Dieu le veut</q>.</p>  
<p>Цитата на немецком: <q lang="de">Der Mensch, versuche die Gotter nicht</q>.</p>  
<p>Цитата на английском: <q lang="en">To be or not to be</q>.</p>
```

spellcheck Указывает браузеру проверять или нет правописание и грамматику в тексте.

style Применяется для определения стиля элемента с помощью правил CSS.

tabindex Устанавливает порядок получения фокуса при переходе между элементами с помощью клавиши Tab.

```
<a tabindex="число">...</a>
```

title Описывает содержимое элемента в виде всплывающей подсказки.

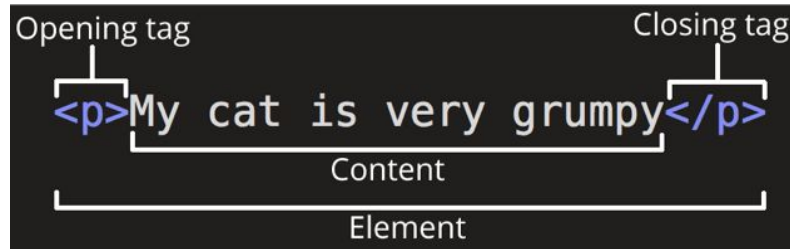
xml:lang Этот атрибут по своему действию похож на lang, но применяется только в XHTML-документах и указывает язык всего текста или его фрагмента.

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.



```
<тег атрибут1="значение" атрибут2="значение">
```

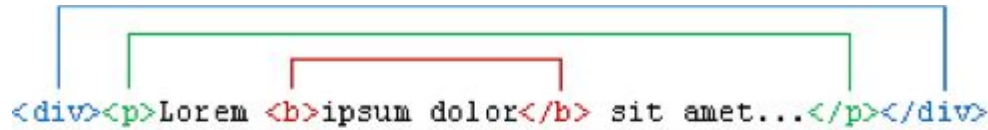
```
<тег атрибут1="значение" атрибут2="значение">...</тег>
```

Парные

Парные теги состоят из двух частей — открывающий и закрывающий тег.

Открывающий тег обозначается как и одиночный — `<тег>`, а в закрывающем используется слэш — `</тег>`.

Допускается вкладывать в контейнер другие теги, однако следует соблюдать их порядок.



```
<div><p>Lorem <b>ipsum dolor</b> sit amet...</p></div>
```



```
<div><p>Lorem <b>ipsum <i>dolor</b> sit</i> amet...</div></p>
```

```

<div> ... </div>

<div><p> ... </p></div>

<div><p><b> ... </b></p></div>

<div><p><b> TEXT </b></p></div>

```

```

<!DOCTYPE html>
<html>
  <head>
    <title>
      Заголовок сайта
    </title>
  </head>
  <body>
    <h1>Заголовок 1 уровень</h1>
    <h2>Заголовок 2 уровень</h2>
    <h3>Заголовок 3 уровень</h3>
    <h4>Заголовок 4 уровень</h4>
    <h5>Заголовок 5 уровень</h5>
    <h6>Заголовок 6 уровень</h6>
    <p>Абзац</p>
    <ul>
      <li>Маркированный список 1</li>
      <li>Маркированный список 2</li>
    </ul>
    <ol>
      <li>Нумерованный список 1</li>
      <li>Нумерованный список 2</li>
    </ol>
  </body>
</html>

```

Одиночные HTML тэги не имеют закрывающегося тега и содержимого

Предназначены одиночные HTML тэги для формирования какого-либо графического HTML элемента или символа на странице.

Одиночные теги НЕ имеют закрывающего тега. <имяТега>

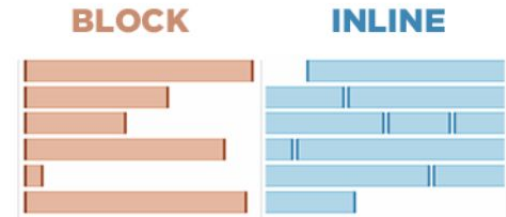
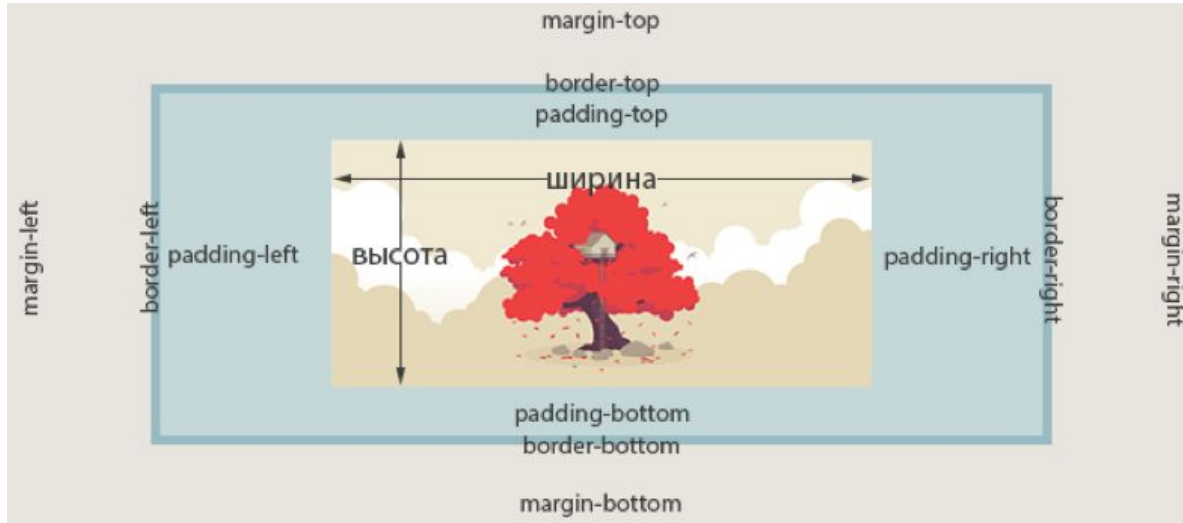




`<area>` – часть карты изображений,
`<base>` – базовый адрес страницы,
`<basefont>` – форматирует текст для всей страницы, устаревший тег,
`<bgsound>` – проигрывает музыку при открытии страницы, нестандартный тег,
`
` – перевод строки,
`<col>` – форматирует одну/несколько колонок таблицы,
`<command>` – команда-переключатель, тег HTML5,
`<embed>` – внедряет плагины, тег HTML5,
`<hr>` – горизонтальная линия,
`` – изображение (фотография, картинка, рисунок и т.д.),
`<input>` – элемент формы, может быть кнопкой, текстовым полем, переключателем и др.,
`<isindex>` – поисковая строка, устаревший тег,
`<keygen>` – шифровка расшифровка данных, тег HTML5,
`<link>` – загружает внешние файлы .css, .xml, .ico,
`<meta>` – мета-теги,
`<source>` – внедряет аудио/видео файлы, с помощью тегов audio и video, тег HTML5,
`<track>` – внедряет текстовую информацию (субтитры, характеристики медиафайлов и т.д.) в аудио/видео файлы, через теги audio и video, тег HTML5,
`<wbr>` – указывает, где нужно перенести длинное слово, если оно не помещается в окно браузера, тег HTML5.

Блочные/строчные элементы

В блочной модели элемент рассматривается как прямоугольный контейнер, имеющий область содержимого и необязательные рамки и отступы (внутренние и внешние). Свойство `display` определяет тип контейнера элемента. Для каждого элемента существует значение браузера по умолчанию.



Блочные элементы

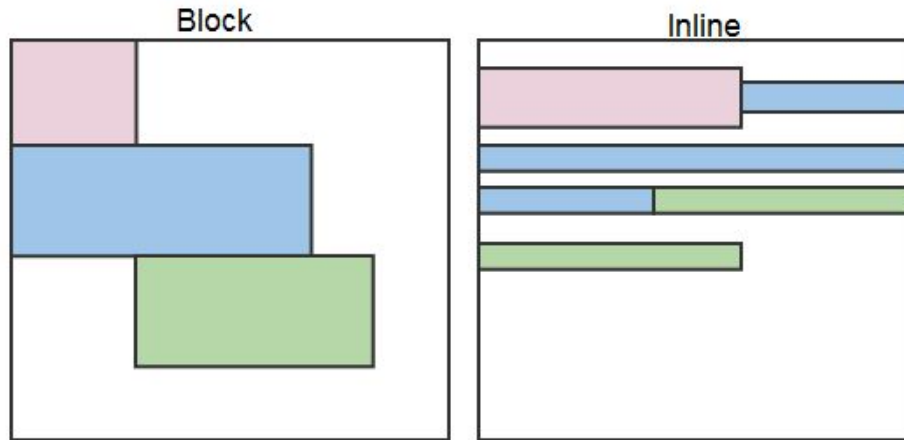
Блочные элементы — элементы высшего уровня, которые форматируются визуально как блоки, располагаясь на странице в окне браузера вертикально.

Значения свойства `display`, такие как `block`, `list-item` и `table` делают элементы блочными.

Блочные элементы генерируют основной блок, который содержит только блок элемента. Элементы со значением `display: list-item` генерируют дополнительные блоки для маркеров, которые позиционируются относительно основного блока.

```
<address>, <article>, <aside>,  
<blockquote>,  
<dd>, <div>, <dl>, <dt>, <details>,  
<fieldset>, <figcaption>, <figure>, <footer>, <form>,  
<h1>-<h6>, <header>, <hr>,  
<li>, <legend>,  
<nav>, <noscript>,  
<ol>, <output>, <optgroup>, <option>,  
<p>, <pre>,  
<section>, <summary>,  
<table>,  
<ul>
```

Блочные элементы могут размещаться непосредственно внутри элемента `<body>`. Они создают разрыв строки перед элементом и после него, образуя прямоугольную область, по ширине занимающую всю ширину веб-страницы или блока-родителя (если для элемента не задано значение `width`).

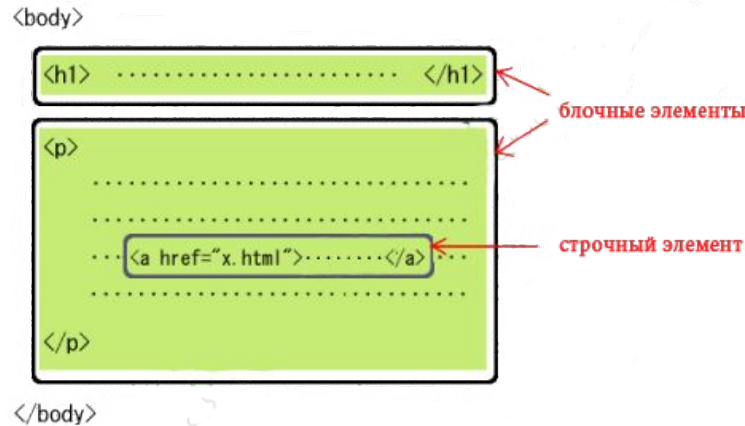


Строчные элементы

Встроенные (строчные) элементы генерируют внутрискрочные контейнеры. Они не формируют новые блоки контента. Значения свойства `display`, такие как `inline` и `inline-table` делают элементы строчными.

```
<a>, <area>,  
<b>, <bdo>, <bdi>,  
<cite>, <code>,  
<dfn>, <del>,  
<em>,  
<i>, <iframe>, <img>, <ins>,  
<kbd>,  
<label>,  
<map>, <mark>,  
<s>, <samp>, <small>, <span>, <strong>, <sub>, <sup>,  
<time>,  
<q>,  
<ruby>,  
<u>,  
<var>
```

Строчные элементы могут содержать только данные и другие строчные элементы. Исключение составляет элемент `<a>`, который согласно спецификации HTML5 может оборачивать целые абзацы, списки, таблицы, заголовки и целые разделы при условии, что они не содержат другие интерактивные элементы — другие ссылки и кнопки.



Строчно-блочные элементы

Существует еще одна группа элементов, которые браузер обрабатывает как строчно-блочные `{display: inline-block;}`. Такие элементы являются встроенным, но для них можно задавать поля, отступы, ширину и высоту.

`<audio>` ,
`<button>` ,
`<canvas>` ,
`<embed>` ,
`<input>` ,
`<keygen>` ,
`<meter>` ,
`<object>` ,
`<progress>` ,
`<select>` ,
`<textarea>` ,
`<video>` .



Семантические элементы HTML5

Семантические элементы четко описывают, что они означают, как браузеру, так и веб-разработчику.

Примеры семантических элементов: **<form>**, **<table>** и **<article>**. Они четко описывают, какого характера контент они содержат.



<article> Определяет статью

<aside> Определяет блок сбоку от основного контента

<details> Определяет дополнительную информацию, которую пользователь может открывать или закрывать

<figcaption> Определяет пояснение для элемента **<figure>**

<figure> Используется для группирования различных самодостаточных элементов - иллюстраций, диаграмм, фотографий, листингов кода и т.д.

<footer> Определяет "подвал" документа или раздела

<header> Определяет "шапку" документа или раздела

<main> Определяет основной контент документа

<mark> Определяет маркированный/подсвеченный текст

<nav> Определяет блок навигационных ссылок

<section> Определяет раздел в документе

<summary> Определяет видимый заголовок элемента **<details>**

<time> Определяет дату/время

Элемент <section>



Элемент <section> определяет раздел в документе. В соответствии со спецификацией W3C по HTML5: "Раздел — это тематически сгруппированный контент, как правило с заголовком." Домашняя страница обычно может быть разбита на следующие разделы: вступление, основной контент и контактная информация.

```
<section>
  <h1>WWF</h1>
  <p>Всемирный фонд дикой природы (WWF) это...</p>
</section>
```

Элемент <article>

Элемент <article> определяет независимый, самодостаточный контент. Контент, помещенный в этот элемент, должен иметь смысл сам по себе, т. е. он должен быть понятен в отрыве от остальных частей веб-сайта.

```
<article>
  <h1>Что делает Всемирный фонд дикой природы?</h1>
  <p>Задача Всемирного фонда дикой природы остановить деградацию окружающей среды на нашей планете и построить будущее, в котором человечество будет жить в гармонии с дикой природой.</p>
</article>
```

Элемент <header>

Элемент <header> предназначен для определения заголовочного блока или "шапки" документа или раздела.

Элемент <header> следует использовать как контейнер для вводной информации.

В одном документе разрешается определять несколько элементов <header>.

```
<article>
  <header>
    <h1>Что делает Всемирный фонд дикой природы (ВФП)?</h1>
    <p>Цель ВФП:</p>
  </header>
  <p>Задача Всемирного фонда дикой природы остановить деградацию окружающей среды
на нашей планете и построить будущее, в котором человечество будет жить в гармонии с
дикой природой.</p>
</article>
```

Элемент <footer>

Элемент <footer> предназначен для определения "подвала" документа или раздела.

Элемент <footer> должен содержать информацию о содержащем его элементе.

Обычно в "подвале" размещают информацию об авторе документа, ссылки на условия использования текста, информация об авторских правах, контактные данные и т.п.

В одном документе разрешается определять несколько элементов <footer>.

```
<footer>
  <p>Автор И.И.Иванов</p>
  <p>Контактная информация: <a
href="mailto:someone@example.com">someone@example.com</a>.</p>
</footer>
```

Элемент <nav>

Элемент <nav> определяет набор ссылок навигации. Обратите внимание, что НЕ ВСЕ ссылки в документе следует размещать внутри элемента <nav>. Элемент <nav> предназначен только для основного блока навигационных ссылок.

```
<nav>

  <a href='/html/'>HTML</a> |

  <a href='/css/'>CSS</a> |

  <a href='/js/'>JavaScript</a> |

  <a href='/jquery/'>jQuery</a>

</nav>
```

Элемент <aside>

Элемент <aside> определяет некий контент, находящийся в стороне от контента, внутри которого он расположен (как боковой блок страницы, "сайдбар").

Контент внутри элемента <aside> должен соотноситься с окружающим контентом.

```
<aside>

  <h4>EPCOT центр</h4>

  <p> EPCOT центр – это тематический парк в развлекательном комплексе
  Уолта Диснея во Флориде.</p>

</aside>
```

Элементы <figure> и <figcaption>

Назначение элемента <figcaption> - добавление визуального пояснения к изображению.

<figure>

```
<img src='img_pulpit.jpg' alt="The Pulpit Rock" width="304" height="228">
```

```
<figcaption>Рис. 1 – Палпит Рок. Гора в Норвегии</figcaption>
```

</figure>



Купеческий клуб



Памятник Св.
Владимиру

html5 & html4 новые теги

На уровне структуры страницы

`<header> </header>` — теги обрамляющие элементы дизайна верхней части сайта, так называемая "шапка страницы".

`<nav> </nav>` — теги обрамляющие основное меню на сайте.

`<article> </article>` — теги обрамляющие основной контент страницы: статью, запись в блоге, новость и т.п.

`<aside> </aside>` — теги обрамляющие сайдбар. Сайдбар — это так называемая "боковая панель", которая обычно содержит блоки рубрик (категорий), облака тегов (меток), список последних записей и т.п.

`<footer> </footer>` — теги обрамляющие "подвал страницы", который может содержать имя автора (компании), контакты (адреса, телефоны), правовую информацию (копирайты) и т.п.

`<section> </section>` — теги обрамляющие повторяющиеся части страницы или основного контента.

На уровне текста

`<mark> </mark>` — текст находящийся между этими тегами становится "выделенным". Одно из предназначений тегов `mark`, обрамлять слова в тексте, которые например совпадают со словом введенным пользователем в поисковую строку.

`<time> </time>` — теги предназначены для создания дат и/или времени в формате ISO вида: YYYY-MM-DDThh:mm:ss. Теги `time` могут обрамлять дату или текст, в него добавляют атрибут `datetime`, значением выступает дата и/или время в формате ISO.

`<details> </details>` — теги предназначены для хранения информации, которая либо скрыта, либо отображена (работает наподобие спойлера).

`<summary> </summary>` — теги обрамляют заголовок, если по нему щелкнуть то появится текст (можно использовать в качестве спойлера), находятся между тегами `details`.

Дополнительные теги языка HTML5

<meter> </meter> — теги предназначены для вывода статического индикатора шкалы, при котором результат измерения не меняется. Для работы нужно минимальное и максимальное значения.

<progress> </progress> — теги предназначены для вывода динамического индикатора шкалы (например шкалы загрузки файла), при котором результат измерения меняется в реальном времени.

<command /> — тег должен находиться между тегами menu, используется при создании сценариев на JavaScript.

<menu> </menu> — между этими тегами помещают тег command.

<output> </output> — теги предназначены для вывода работы скрипта.

<datalist> </datalist> — теги предназначены для создания списка, который будет выводиться при наборе в текстовом поле.

<figcaption> </figcaption> — теги обрамляют описание какого-либо объекта (например изображения), находятся между тегами figure..

<figure> </figure> — теги группируют различные объекты страницы имеющие свои описания, например изображения с описанием, товары с описанием и т.д.

<hgroup> </hgroup> — теги предназначены для группировки заголовков h*

<keygen /> — тег используется для генерации открытых/закрытых пар ключей, шифрования/расшифровки данных, создания/проверки цифровой подписи.

<ruby> </ruby> — теги обрамляют текст и аннотацию к нему.

<rt> </rt> — теги находятся между тегами ruby, предназначены для обрамления аннотации.

<rp> </rp> — теги предназначены для браузеров которые не поддерживают теги ruby.

<wbr /> — тег указывает браузеру в каком месте переносить слово, если это слово не вмещается в окно браузера.

Устаревшие (deprecated tags) теги



`<applet>` — добавляет Java-апплет в документ. Вместо него следует использовать `<embed>` или `<object>`.

`<acronym>` — этот тег вызывал постоянные вопросы, что такое акроним и чем он отличается от аббревиатуры. Для упрощения остался единственный тег `<abbr>`.

`<bgsound>` — определяет музыкальный файл, который будет проигрываться на веб-странице при её открытии. Для воспроизведения музыки используйте новый элемент `<audio>`.

`<dir>` — создает список, содержащий названия директорий, вместо него используйте ``.

`<frame>`, `<frameset>`, `<noframe>` — фреймы более не поддерживаются. Если они вам требуются, используйте другую версию HTML или `<iframe>` совместно со стилями.

`<isindex>` — предназначен для поискового индекса в текущем документе. Комбинация `<form>` и `<input>` лучше справляется с этой задачей.

`<listing>`, `<xmp>` — для вывода листинга программы предназначены `<pre>` и `<code>`.

`<nextid>` — этот тег не предназначен для людей и указывает идентификатор следующего документа для автоматических редакторов HTML. Полностью исключён.

`<noembed>` — предназначен для отображения информации на веб-странице, если браузер не поддерживает работу с плагинами. В качестве альтернативы используйте `<object>`.

`<plaintext>` — отображает содержимое контейнера «как есть», любые теги выводятся как текст. Вместо тега используйте MIME-тип `text/plain`.

`<rb>` — определяет базовый текст внутри `<ruby>`. Этот тег полностью исключён.

`<strike>` — для зачёркнутого текста применяется `<s>`, а для указания редакторской правки ``.

`<basefont>`, `<big>`, `<blink>`, `<center>`, ``, `<marquee>`, `<multicol>`, `<nobr>`, `<spacer>`, `<tt>`, `<u>` — вместо этих тегов управляющих видом текста применяются стили.

Атрибуты тегов

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.

Атрибуты HTML обычно отображаются в виде пар имя-значение, разделяются =и записываются в начальный тег элемента после имени элемента:



```
<p class="editor-note">My cat is very grumpy</p>
```



Атрибуты HTML классифицируются как обязательные атрибуты , необязательные атрибуты , стандартные атрибуты и атрибуты событий . Обычно требуемые и необязательные атрибуты изменяют определенные HTML-элементы, в то время как стандартные атрибуты могут применяться к большинству элементов HTML.

Empty `<input type="text" value="John" disabled>`

Unquoted `<input type="text" value=John>`

Double-quoted `<input type="text" value="John Doe">`

Single-quoted `<input type="text" value='John Doe'>`

Устаревшие, но поддерживаемые атрибуты

Атрибут `http-equiv` тега `<meta>` для указания языка должен заменяться атрибутом `lang`

Было:

```
<meta http-equiv="content-language" content="ru" />
```

Стало:

```
<html lang="ru">
```

Атрибут `name` тега `<a>` должен заменяться атрибутом `id`. Если `name` присутствует, то должен содержать пустую строку или совпадать со значением `id`.

Было:

```
<a name="p12"></a>
```

Стало:

```
<a id="p12"></a>
```

Атрибут `language` тега `<script>` должен быть опущен. Если он присутствует, значение совпадает с JavaScript или другим типом, но в таком случае язык скрипта следует указать через `type`.

Было:

```
<script type="text/javascript" language="JavaScript">
```

Стало:

```
<script>
```

Атрибут `border` тега `` не указывается, а толщина границы задаётся через стили. Если этот атрибут присутствует, его значение должно быть 0.

Было:

```
<img border="1">
```

Стало:

```
<img style="border: 1px solid black">
```

`<a>`

Ссылки являются определяющей особенностью сети, поскольку они позволяют вам перемещаться с одной веб-страницы на другую, что позволяет сама идея просмотра или поиска.

Обычно вы сталкиваетесь со следующими типами ссылок:

- Ссылки с одного сайта на другой
- Ссылки с одной страницы на другую на том же сайте
- Ссылки с одной части веб-страницы на другую часть той же страницы
- Ссылки, открывающиеся в новом окне браузера
- Ссылки, которые запускают вашу почтовую программу и направляют новое электронное письмо кому-то



Абсолютный (точное местоположение файла в пределах всей структуры папок на компьютере (сервере) и относительный путь (путь к указанному документу относительно текущего).



`...`
`...`

accesskey

Активация ссылки с помощью комбинации клавиш.

coords

Устанавливает координаты активной области.

download

Предлагает скачать указанный по ссылке файл.

href

Задаёт адрес документа, на который следует перейти.

hreflang

Идентифицирует язык текста по ссылке.

name

Устанавливает имя якоря внутри документа.

rel

Отношения между ссылаемым и текущим документами.

rev

Отношения между текущим и ссылаемым документами.

shape

Задаёт форму активной области ссылки для изображений.

tabindex

Определяет последовательность перехода между ссылками по Tab.

target

Имя окна или фрейма, куда браузер будет загружать документ.

title

Добавляет всплывающую подсказку к тексту ссылки.

type

Указывает MIME-тип документа, на который ведёт ссылка.

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ter A</title>
  </head>
  <body>
    <p><a href="images/xxx.jpg">Посмотрите на мою фотографию!</a></p>
    <p><a href="tip.html">Как сделать такое же фото?</a></p>
  </body>
</html>
```



HTML5 спроектирован с возможностью расширения данных ассоциированных с каким-либо элементом, но в то же время не обязательно имеющих определённое значение. `data-*` атрибуты позволяют хранить дополнительную информацию в стандартных элементах HTML, без хаков вроде нестандартных атрибутов, лишних DOM-свойств или `Node.setUserData()`.

В HTML5 для любого элемента можно использовать собственные атрибуты, начинающиеся с префикса `data-`. Это позволяет хранить разную информацию, которая может помочь в работе скриптов, а также для оформления элементов через CSS.

Атрибут должен иметь хотя бы один символ в нижнем регистре. Буквы в верхнем регистре хотя и допустимы, но они принудительно будут переведены в нижний регистр, поэтому не дают никакого эффекта. В именах атрибутов можно использовать дополнительные дефисы

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>data-*

```

```
<article
  id="electriccars"
  data-columns="3"
  data-index-number="12314"
  data-parent="cars">
  ...
</article>
```

Доступ в JavaScript

```
var article = document.getElementById('electriccars');
```

```
article.dataset.columns // "3"
article.dataset.indexNumber // "12314"
article.dataset.parent // "cars"
```


Доступ в CSS

```
article::before {
  content: attr(data-parent);
}
article[data-columns='3']{
  width: 400px;
}
article[data-columns='4']{
  width: 600px;
}
```

```
<div class="form-group">
  <label class="col-md-2 control-label" for="Instructor">Instructor</label>
  <div class="col-md-10">
    <input class="form-control" type="text" data-val="true"
      data-val-length="Lenght must be between 3 and 6 characters"
      data-val-length-max="6" data-val-length-min="3"
      data-val-required="This is a custom error message"
      id="Instructor" name="Instructor" value="" />
    <span class="text-danger field-validation-valid"
      data-valmsg-for="Instructor" data-valmsg-replace="true" />
  </div>
</div>
```

Стоить помнить только одно правило, в дата атрибутах нельзя хранить объекты. Хранить можно только строковые данные.



Полезные ссылки:

<https://tinyurl.com/zl7w8bq>

<https://emmet.io/download/>

<http://htmlbook.ru/>

<https://www.w3schools.com/>



Спасибо за внимание!