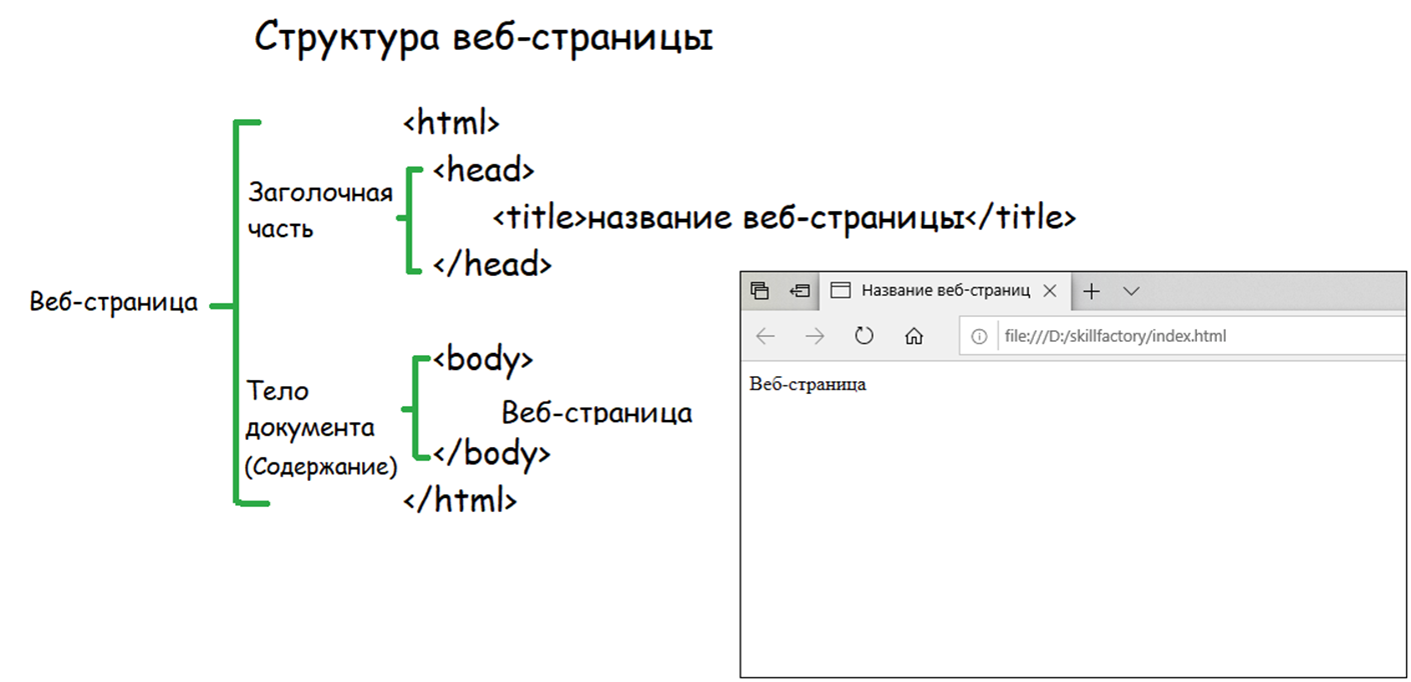
**4.2. Структура веб-страницы**

Основными компонентами *HTML* являются:

* + **Элемент** — конструкция языка *HTML*. Это контейнер, содержащий данные и позволяющий отформатировать их определённым образом. Любая веб-страница представляет собой набор элементов. Одна из основных идей гипертекста (понятие гипертекста разберём в следующем юните) — возможность вложения элементов.

Типичная структура веб-страницы:

Структура веб-страницы

* + **Тег** — начальный или конечный маркеры элемента. Теги определяют границы действия элементов и отделяют элементы друг от друга. В тексте веб-страницы теги заключаются в угловые скобки, а конечный тег всегда снабжается косой чертой.

Пример: <p> тег абзаца </p>

Теги представляют собой зарезервированные стандартом *HTML* последовательности символов, начинающиеся c < (знака меньше) и заканчивающиеся > (знаком больше).

Закрытие тега отличается от открытия наличием символа косой черты /.

Предположим, у нас есть гипотетический атрибут форматирования текста, управляемый кодом <p>, и мы хотим применить его к словам «Это мой текст».

*HTML*-последовательность кодов и, собственно, текста будет выглядеть так:

<p>Это мой текст</p>

Теги могут вкладываться друг в друга иерархически, но без пересечений. То есть допустимо вложение вида:

<tag1>

<tag2>

</tag2>

</tag1>

Но не в таком виде:

<tag1>

<tag2>

</tag1>

</tag2>

Действие вложенных тегов может объединяться. Например, если внутрь тега, создающего жирное начертание шрифта, вложен тег курсива, то в результате получится жирный курсив.

Ниже приведена структура типичного веб-документа:

<html>

<head>

<title>Название web-страницы</title>

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

<meta name="author" content="ivanov ivan">

<meta name="keywords" content="www, html, document, страничка, структура">

</head>

<body bgcolor="blue">

<h2>заголовок вашей страницы</h2>

<p>текст первого абзаца вашей страницы</p>

<p>текст второго абзаца вашей страницы</p>

</body>

</html>

|  |  |
| --- | --- |
| <html> | Этот тег указывает на начало *HTML*-документа |
| <head> | Этот тег указывает на начало области заголовка веб-страницы. Служит для формирования общей структуры документа. |
| <title>Название web-страницы</title> | Элемент для размещения заголовка веб-страницы. Строка отображается в заголовке окна браузера. |
| <meta http-equiv="Content-Type" content="text/html; charset=utf-8"> | Этот тег со служебной информацией о кодировке веб-страницы. В данном примере указана кодировка *UTF*-8, наиболее распространённая и универсальная. |
| <meta name="Author" content="Ivanov Ivan"> | Имя автора веб-страницы. |
| <meta name="Keywords" content="WWW, HTML, document, страничка, структура"> | Набор ключевых слов для поисковых машин. Для поисковых машин полезно оставлять информацию, в которой указаны ключевые слова, отражающие содержание вашего сайта. |
| </head> | Конец области заголовка веб-страницы. |
| <body bgcolor="blue"> | Начало содержимого веб-страницы. Тег <body> включает в себя атрибут bgcolor, который задаёт цвет вашей страницы. В данном случае голубой. Если не использовать этот атрибут, то по умолчанию цвет страницы будет белым. |
| <h2> Здесь расположен заголовок вашей страницы </h2> | <h2></h2> — элемент заголовка. |
| <p> Здесь расположен текст первого абзаца вашей страницы</p> | <p></p> — элемент абзаца. |
| <p> Здесь расположен текст второго абзаца вашей страницы</p> | <p></p> — элемент абзаца. |
| </body> | Конец содержимого веб-страницы. |
| </html> | Конец *HTML*-документа. |

Теги можно записывать как строчными, так и заглавными латинскими символами, но принято писать строчными буквами для удобства.

**Основные правила**

* 1. Веб-страница начинается с блока <html>.
  2. Взаимное расположение элементов <html>, <head>, <body> должно быть стандартным на любой странице. Блок <head> всегда на одном уровне с блоком <body>, блок <body> идёт после блока <head>.

Оба блока <head> и <body> внутри блока <html>:

<html>

<head>

<title>.....</title>

</head>

<body>

..................

</body>

</html>

* 1. Необходимо всегда использовать теги для расположения элементов (не забывать <p>, <h1>, <table>, <div> и другие).
  2. Не нарушать правила вложения тегов.

**Правильно**:

<p>

Текст

<em> Вложенный </em>

</p>

**Неправильно**:

<p>

Текст

<em> Вложенный

</p>

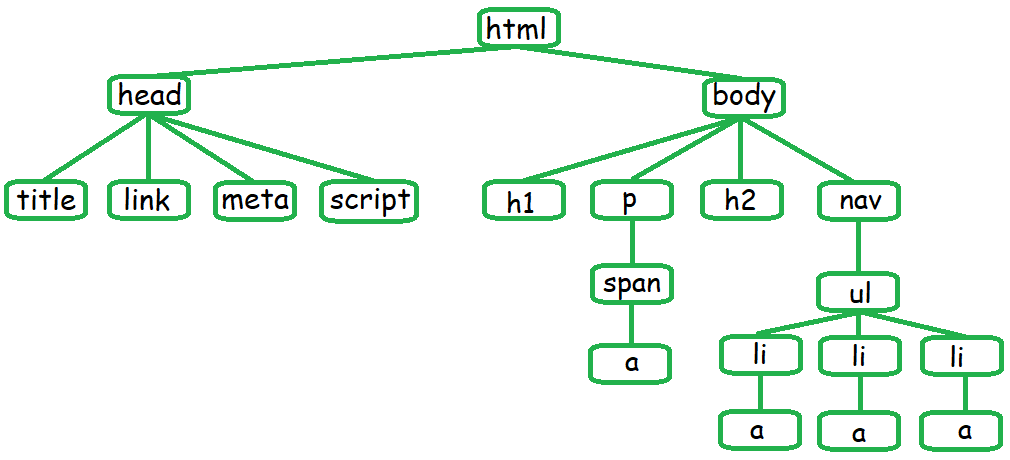
</em>

* 1. Любая полезная информация должна находится между начальным и конечным тегами.
  2. Все атрибуты (подробнее об атрибутах мы поговорим позже) и их значения располагаются в начальном теге.

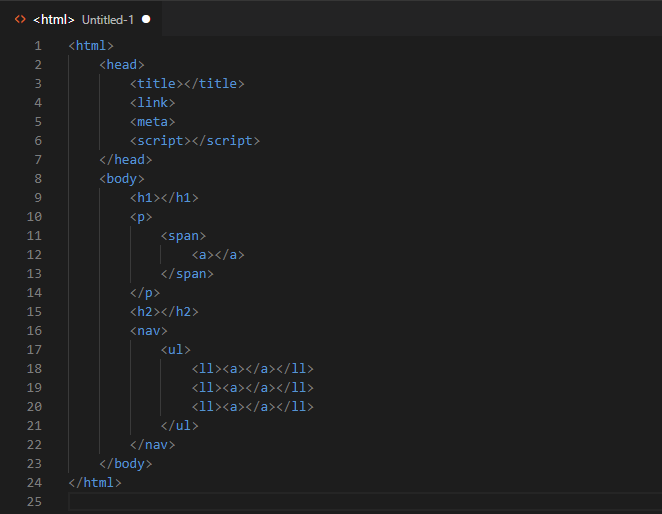
**Структура HTML-документа**

Элементы, находящиеся внутри тега <html>, образуют дерево документа. При этом элемент <html> является корневым элементом.

Простейшая структура веб-страницы:

Структура веб-страницы

Это её *HTML*-код:

*HTML*-код

Для того чтобы разобраться во взаимодействии элементов веб-страницы, необходимо рассмотреть так называемые «родственные отношения» между элементами.

Отношения между множественными вложенными элементами подразделяются на:

* + Родительские,
  + Дочерние
  + Сестринские.

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

**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 <span>. Родительским элементом является непосредственный предковый элемент.

**Дочерний элемент** — элемент, непосредственно подчинённый 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 соседние элементы одного уровня. На рисунке <head> и <body> — элементы одного уровня, так же как и элементы <h1>, <h2> и <p> являются между собой сестринскими.

## 4.3. Основы HTML

Подробнее про *index.html* можно прочитать [здесь](https://www.lifewire.com/index-html-page-3466505).

**Правила синтаксиса**

Каждый *HTML*-документ должен начинаться с декларации типа документа или «доктайпа» (<!DOCTYPE html>), он означает, что код в документе будет написан на языке *HTML*. Затем идёт сам *HTML*-код, который начинается и заканчивается тегами.

Между тегами располагаются два основных блока:

* + Первый блок — это заголовочная часть *HTML*-документа, который начинается и заканчивается тегами;
  + Второй блок — это тело *HTML*-документа, который начинается и заканчивается тегами.

В заголовочной части *HTML*-документа содержится различная служебная информация, которую пользователь не видит (кроме тега title), там находятся следующие теги:

**<title> </title>** — название *HTML*-страницы,

**<meta>** — мета-теги, в них содержится служебная информация о странице,

**<link>** — тег, ссылающийся на внешние файлы, например, .css, .ico и так далее,

**<script> </script>** — теги могут содержать *JavaScript*-код или ссылаться на внешний файл .js.

В теле *HTML*-документа обычно содержится основная информация, которую мы видим на странице, там могут находиться следующие теги:

**<h1> </h1>** — заголовок статьи первого уровня,

**<img>** — изображение,

**<p> </p>** — абзац,

**<a> </a>** — ссылка,

**</br>** — перевод строки,

**<table> </table>** — таблица,

**<form> </form>** — форма ввода данных.

Познакомиться с другими тегами HTML можно [здесь](http://htmlbook.ru/html).

Комментарий в HTML-коде задаётся так:

<!-- любой текст -->

Текст внутри комментария не отображается браузером на странице. Комментарии обычно используются для комментирования кода или временного отключения кода. Их можно использовать в любом месте страницы, кроме тега — title внутри него они не работают.

## 4.4. Основные атрибуты

Вы уже знакомы с несколькими HTML-тегами и их использованием, например, тегами заголовков <h1>, <h2>, тегом абзаца <p> и другими. До сих пор мы использовали их в простейшей форме, но большинство тегов в HTML могут иметь атрибуты.

**Атрибут** используется для определения характеристик HTML-элемента и помещается внутри открытого тега элемента. Все атрибуты состоят из двух частей — это имя и значение:

* + **Имя** — это свойство, которое вы хотите установить. Например, элемент абзаца <p> в примере ниже содержит атрибут align, который вы можете использовать для указания выравнивания абзаца на странице.
  + **Значение** — это значение, которое вы хотите установить для свойства. Значение атрибута всегда помещается в кавычки. В приведённом ниже примере показаны три возможных значения атрибута align: left, center и right.

Имена и значения атрибутов в HTML не зависят от регистра. Однако консорциум World Wide Web (W3C) рекомендует использовать их в нижнем регистре.

**Атрибут** — это переменная, которая имеет стандартное имя и к которой может присваиваться определенный набор значений: стандартный или произвольный. Атрибуты располагаются внутри начального тега и отделяются друг от друга пробелами.

Пример задания атрибута для элемента страницы:



 **Пример**

<!DOCTYPE html>

<html>

<head>

<title>Пример атрибута align</title>

</head>

<body>

<p align="left">Выровнено по левому краю.</p>

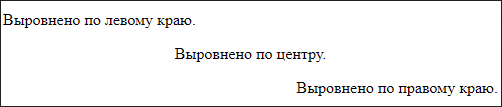
<p align="center">Выровнено по центру.</p>

<p align="right">Выровнено по правому краю.</p>

</body>

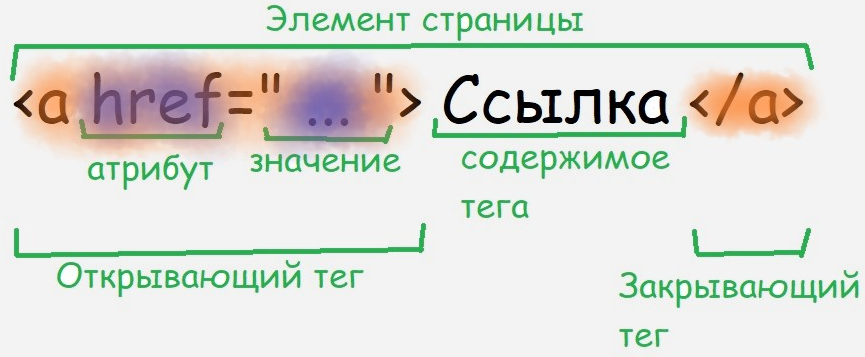
</html>

Получим следующий результат:



## рис ****Важно знать:****

* 1. Страница состоит из **элементов**.
  2. Элемент — это набор из **тегов** и **содержания**.
  3. Теги чаще всего идут парами: **открывающий** (<tag>) и **закрывающий** (</tag>). Между ними находится то, что попадает под действие этого элемента.
  4. Открывающий тег может содержать дополнительную информацию — **атрибуты** и **значения** атрибутов. Атрибуты позволяют «настроить» тег.



Это ссылка, и у неё есть атрибут href со значением https://ru.skillfactory.io. Между открывающим и закрывающим тегами находится содержание — слово «SkillFactory». Именно оно становится ссылкой.

<a href="https://skillfactory.ru/">SkillFactory</a>

А ниже есть картинка. Это пример одиночного тега, элемента, в котором не нужен закрывающий тег. Их ещё называют **self-closing element** — «самозакрывающийся элемент».

<img src="https://static.tildacdn.com/tild6539-3361-4761-a434-353330643930/skillfactory\_logo\_bl.svg">

Как видите, мы указали атрибут src (от английского «source» — источник) со значением https://static.tildacdn.com/tild6539-3361-4761-a434-353330643930/skillfactory\_logo\_bl.svg. Это адрес файла изображения.

Существует четыре основных атрибута в HTML, которые могут использоваться для большинства HTML-элементов (хотя и не для всех):

* 1. id
  2. title
  3. class
  4. style

## ****Атрибут id****

Атрибут id HTML-тега может быть использован для однозначной идентификации любого элемента внутри HTML-страницы.

Существуют две основные причины, по которым вы можете использовать атрибут id для элемента:

* + Если элемент содержит атрибут id как уникальный идентификатор, то можно идентифицировать только этот элемент и его содержимое.
  + Если на веб-странице (или в таблице стилей) есть два элемента с одним и тем же именем, вы можете использовать атрибут id для различения этих элементов.

Таблицу стилей мы обсудим в отдельном модуле. Пока же давайте использовать атрибут id для различения двух элементов абзаца, как показано ниже.

<p id="html">Этот абзац объясняет, что такое HTML.</p>

<p id="css">Этот абзац объясняет, что такое каскадная таблица стилей (CSS).</p>

## ****Атрибут title****

Атрибут title — описывает содержимое элемента. Синтаксис для атрибута title аналогичен атрибуту id.

Поведение этого атрибута будет зависеть от элемента, который его несёт, хотя он часто отображается как подсказка, когда курсор наводится на элемент или когда элемент загружается.

**Пример:**

<!DOCTYPE html>

<html>

<head>

<title>Пример атрибута title</title>

</head>

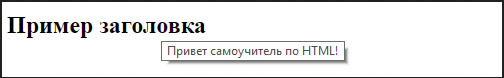
<body>

<h2 title="Привет самоучитель по HTML!">Пример заголовка</h2>

</body>

</html>

Получим следующий результат:



Теперь попробуйте навести курсор на «Пример заголовка», и вы увидите, что title, который вы использовали в вашем коде, выйдет как подсказка курсора.

## ****Атрибут class****

Атрибут class — используется для связывания элемента со списком стилей и задаёт класс элементу. Вы узнаете больше об использовании атрибута class, когда изучите CSS (каскадные таблицы стилей).

Значением атрибута может быть также список имён классов, разделённых пробелами.

**Пример**:

class="названиеКласса1 названиеКласса2 названиеКласса3"

## ****Атрибут style****

Атрибут style — позволяет указывать правила каскадной таблицы стилей (CSS) внутри элемента.

<!DOCTYPE html>

<html>

<head>

<title>Пример атрибута style</title>

</head>

<body>

<p style="text-align: center; color: #008cff;">Немного текста... </p>

</body>

</html>

Получим следующий результат:

img

На данный момент мы не изучаем CSS, так что давайте продолжим, не беспокоясь о стилях. В первую очередь вам необходимо понять, какие существуют HTML-атрибуты и как их можно использовать при форматировании содержимого.

## ****Общие атрибуты****

Ниже представлена таблица некоторых других атрибутов, которые можно легко использовать со многими HTML-тегами.

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Значение** | **Функция** |
| align | right, left, center | Горизонтальное выравнивание тегов. |
| valign | top, middle, bottom | Вертикальное выравнивание тегов внутри HTML-элемента. |
| bgcolor | числовые, шестнадцатеричные, RGB-значения | Помещает фоновый цвет за элемент. |
| background | URL | Помещает фоновое изображение за элемент. |
| id | определяется пользователем | Именование элемента для использования с каскадными таблицами стилей. |
| class | определяется пользователем | Классифицирует элемент для использования с каскадными таблицами стилей. |
| width | числовое или процентное значение | Задаёт ширину таблиц, изображений или ячеек таблицы. |
| height | числовое или процентное значение | Задаёт высоту таблиц, изображений или ячеек таблицы. |
| title | определяется пользователем | «Всплывающий» заголовок элементов. |

### 

## 4.5. Элементы веб-страницы

## ****Специальные символы****

Текст на веб-странице можно оборачивать в разные теги в зависимости от задачи. Рассмотрим несколько редких тегов:

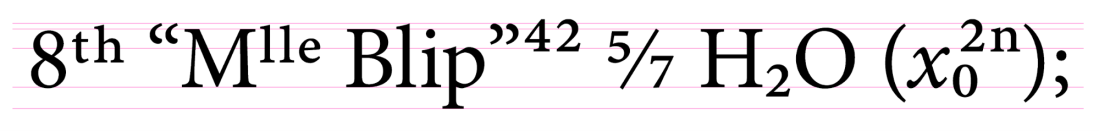
[**<del>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/del) — вычёркивает текст, показывает, что информация устарела, но важна для сравнения, например, цена товара в магазине.



<span>7 999 **&#8381;**</span>

<del>9 999 **&#8381;**</del>

[**<sup>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/sup) и [**<sub>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/sub) — верхний и нижний регистры, полезны для сносок и формул. Пример: 8th, H2O.



<p>Теорему Пифагора можно выразить с помощью уравнения: a<sup>2</sup> + b<sup>2</sup> = c<sup>2</sup>.</p>

<p>Любимая молекула разработчиков C<sub>8</sub>H<sub>10</sub>N<sub>4</sub>O<sub>2</sub>, также известная как "кофеин".</p>

[**<q>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/q) и [**<blockquote>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/blockquote) — элементы для цитат, в первом случае внутри текста, во втором — для отдельного блока текста.

Вот пример содержимого, для которого можно использовать элемент **<q>**, чтобы вставить цитату в текст пункта списка:

"Русский писатель, философ и общественный деятель XIX века Владимир Одоевский в незаконченном утопическом романе «4338-й год», написанном в 1837 году, похоже, первым предсказал появление современных блогов и интернета: в тексте романа есть строки: "между знакомыми домами устроены магнетические телеграфы, посредством которых живущие на далёком расстоянии общаются друг с другом".

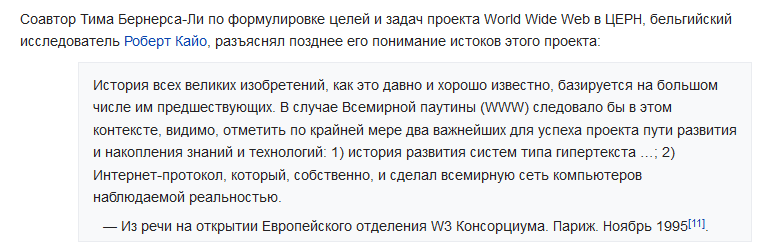
<ul>

<li>Русский писатель, философ и общественный деятель XIX века Владимир Одоевский в незаконченном утопическом романе «4338-й год», написанном в 1837 году, похоже, первым предсказал появление современных блогов и интернета: в тексте романа есть строки: <q>между знакомыми домами устроены магнетические телеграфы, посредством которых живущие на далёком расстоянии общаются друг с другом</q>.

</li>

</ul>

Пример **цитаты** как отдельного блока на странице. Здесь весь текст цитаты — это содержимое элемента <blockquote>:



<p>В 1991 году Всемирная паутина стала общедоступна в интернете, а в 1993 году появился знаменитый веб-браузер NCSA Mosaic. Всемирная паутина набирала популярность.</p>

<blockquote>Можно считать, что существуют две ясно различимые эры в истории Web: [до браузера Mosaic] Марка Андриссена и после. Именно сочетание веб-протокола от Тима Бернерс-Ли, который обеспечивал коммуникацию, и браузера (Mosaic) от Марка Андриссена, который предоставил функционально совершенный пользовательский интерфейс, создало условия для наблюдаемого взрыва (интереса к Веб). За первые 24 месяца, истёкшие после появления браузера Моsaic, Web прошёл стадию от полной неизвестности (за пределами считанного числа людей внутри узкой группы учёных и специалистов лишь одного мало кому известного профиля деятельности) до полной и абсолютно везде в мире его распространённости.</blockquote>

По умолчанию браузеры оформляют текст, обернутый **<q>** , кавычками, а **<blockquote>** — отодвигают от левого края. Эти стили оформления можно будет менять.

Некоторые символы нельзя просто так написать в текстовом содержимом HTML-элемента, потому что браузер подумает, что это часть кода, например, угловые скобки, кавычки. Чтобы показать их пользователю нужно использовать **специальные символы**, которые браузер расшифрует в нужный вам знак. Например:

* + **&lt;** — левая угловая скобка, начало тега (<),
  + **&gt;**— угловая скобка, конец тега (>),
  + **&quot;** — двойная кавычка (“). Зачастую в качестве двойных кавычек также используют **&laquo;** («) — открывающая двойная кавычка и **&raquo;** (») — закрывающая двойная кавычка
  + **&amp;** — знак амперсанда (&)
  + **&ndash**; — среднее тире (–)
  + **&mdash;** — длинное тире (—),
  + **&nbsp;**— неразрывный пробел. «Неразрывный» означает, что подобный пробел будет игнорироваться переносом строки документа, и словосочетание будет всегда на одной строке.

Остальные спецсимволы вы найдете самостоятельно в интернете (можно начать [отсюда](https://html5book.ru/specsimvoly-html/)).

## ****Другие элементы форматирования****

Есть ещё несколько полезных элементов для форматирования текста: **перенос** строки и линия **разделения**. Оба элемента состоят из одиночного тега, без закрывающего.

[**<br>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/br) — перенос строки, после него начнется новая строка, не стоит использовать для оформления блоков на странице. Перенос строки стоит использовать на уровне текста внутри параграфа, например, для стихов или строчек с адресом компании.

<address>

190000, Санкт-Петербург,<br>

Невский проспект, д.1<br>

555-55-55<br>

</address>

[**<hr>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/hr) — создаёт черту, которая отделяет содержимое на уровне параграфов. Можно использовать, чтобы обозначить изменение темы. Пример с MDN:

<p>§1: The first rule of Fight Club is: You do not talk about Fight Club.</p>

<hr>

<p>§2: The second rule of Fight Club is: Always bring cupcakes.</p>

[**<ol>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol) — **нумерованный список**, здесь важен порядок, последовательность действий/пунктов, которые нельзя менять местами, нумеруется числами.

[**<li>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/li) — **пункт списка**, используется в списках любого типа.

<h3>Занятия первого модуля курса HTML/CSS</h3>

<ol>

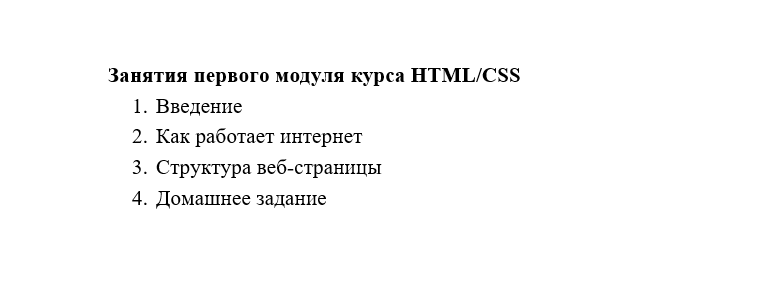
<li>Введение</li>

<li>Как работает интернет</li>

<li>Структура веб-страницы</li>

<li>Домашнее задание</li>

</ol>



[**<ul>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul) — **маркированный список**, здесь последовательность не важна, главное, чтобы пункты отвечали на один вопрос. По умолчанию ставятся чёрные кружки, но их можно изменить на любые символы или вовсе скрыть (например, чтобы сделать с помощью такого списка меню на сайте).

<h3>Список покупок</h3>

<ul>

<li>молоко</li>

<li>хлеб</li>

<li>яблоки</li>

<li>сыр</li>

</ul>



Нумерованные и маркированные списки могут находится друг в друге, вложенность при этом не ограничена, например:

<ul>

<li>Пункт первый</li>

<li>

Пункт второй

<ol>

<li>Подпункт первый</li>

<li>Подпункт второй</li>

</ol>

</li>

<li>Пункт третий</li>

</ul>

[**<pre>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/pre) — показывает отформатированное содержимое с любым количеством **пробелов**, например, когда автору сайта нужно на странице показать код с отступами. Если вы захотите отобразить на странице HTML-код, то нужно заменить все скобки и некоторые знаки спецсимволами.

<pre>

Асфальт — стекло.

Иду и звеню.

Леса и травинки —

сбриты.

На север

с юга

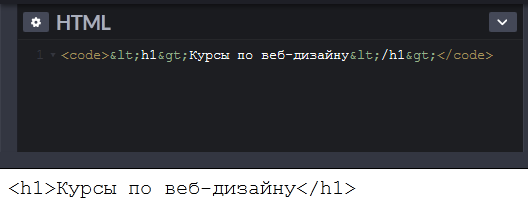
идут авеню,

на запад с востока —

стриты.

</pre>

Также для отображения кода можно использовать отдельный тег **code**, но он не будет показывать все пробелы и их придётся отмечать вручную:



## ****Изображения на сайте****

Мы уже познакомились с разными HTML-элементами, и вы знаете, что они состоят из тегов и текстового содержимого. Но иногда нужно сообщить браузеру какую-то дополнительную информацию, связанную с тегом. Чтобы понять, зачем это нужно, давайте рассмотрим два очень важных для любого сайта элемента: **ссылку и изображение**.

[**<img>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img)— элемент, который вставляет на страницу **изображения**, используется для картинок, являющихся **содержимым**, а не оформлением. Иллюстрация к статье — содержимое, её нельзя убрать без потери смысла. А вот узор на фоне — оформление, его мы научимся делать позже с помощью CSS. Понять разницу не всегда просто, но мы научимся делать и то, и другое.

Для того, чтобы вставить на страницу картинку, нужно указать браузеру, где взять изображение. Это можно сделать с помощью дополнительных настроек тега, или **атрибутов**, которые пишутся внутри открывающего тега.

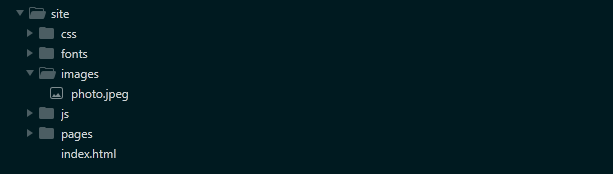
<img src="">

У атрибута обычно есть некое значение, которое пишется в кавычках после пробела. У элемента **img** есть два **обязательных атрибута**— **src** и **alt**. Атрибут **src** указывает путь до графического файла, который браузер вставит на страницу.

### ****Путь к файлу****

Существует два типа пути: абсолютный и относительный. **Абсолютный** представляет собой полный адрес (URL) файла или страницы.

**Относительный путь** указывает место расположения картинки относительно файла, в котором он написан. Например, если файл картинки **image.jpg** находится в той же папке, что и html-файл, то вы указываете "**image.jpg**". Если он находится во вложенной папке images, то есть уровнем ниже, то мы начинаем путь с её названия: "**images/image.jpg**". Если уровнем выше, то начинаем с символа "../", который поднимает вас на один уровень: "**../images/image.jpg**". Относительный путь обычно используется для файлов на сервере.



Предположим, у нас есть **папка сайта** со стандартным набором папок внутри: CSS, fonts, images, JS и файл index.html. В папке images есть картинка **photo.jpeg**. Если её вставить на страницу index.html с помощью тега, то путь к ней начнётся на уровне файла, в котором пишем код. Идём в папку images, находим там файл:

<img src="images/photo.jpeg">

А если мы указываем путь к этой же картинке, находясь в файле CSS (который, соответственно, расположен в папке CSS), то сначала пришлось бы подняться на уровень выше:

<img src="../images/photo.jpeg">

### https://lms.skillfactory.ru/assets/courseware/v1/55179c1c222ed792bd54ccf578a092cd/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-image-file-50.png****Описание изображения****

Второй обязательный атрибут у тега img  — **alt**, это текстовое **описание** изображения. Оно нужно, если картинка не загрузилась, а также будет полезно для людей с плохим зрением, потому что браузер на месте незагруженной картинки напишет содержимое этого атрибута, если оно есть. Например:

<img src="../images/photo.jpeg" alt="Рабочее место с компьютером">

### https://lms.skillfactory.ru/assets/courseware/v1/55179c1c222ed792bd54ccf578a092cd/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-image-file-50.png****Другие атрибуты изображения****

Есть еще два атрибута, которые принято указывать для изображений: **width** (ширина) и **height** (высота), указываются оригинальные или желаемые размеры. Внешний вид и размеры элементов указываются через **CSS**, но если стили не загрузятся, стоит к этому подготовиться.

<img src="../images/photo.jpeg" alt="Рабочее место с компьютером" width="300" height="200">

### https://lms.skillfactory.ru/assets/courseware/v1/7ce0ce2ba6f91320d81359a31c7f5c5e/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-create-50.png****Важные моменты при использовании img****

* + Все атрибуты отделяются от названия тега и друг от друга **пробелами,** не пропускайте их.
  + У атрибутов ставятся кавычки и знак равно.
  + Проверяйте опечатки в пути, названиях папок и файлов. Чтобы этого избежать, стоит копировать путь из проводника/вашего файлового менеджера.
  + Убедитесь, что у вас в редакторе и браузере открыт один и тот же файл.

### https://lms.skillfactory.ru/assets/courseware/v1/55179c1c222ed792bd54ccf578a092cd/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-image-file-50.png****Вес изображения****

Место, которое картинки занимают на диске, называют **весом**. Его важно **контролировать**, потому что не у всех есть быстрый интернет, у кого-то стоит ограниченный трафик, большие картинки могут тормозить работу сайта, из-за чего вы потеряете посетителей. На вес влияют тип файла, сжатие, размеры (ширина, высота), содержимое.

|  |  |  |  |
| --- | --- | --- | --- |
| **Типы графического файла** | **Область применения** | **Вес** | **Когда использовать** |
| .jpg, .jpeg — растровое изображение со сжатием | Фото, иллюстрации. Самый популярный тип изображений для фото в сети. | Зависит от сжатия и размеров. От десятков килобайт до нескольких мегабайт. | Если фото на странице много, то лучше их сжать и уменьшить. Если важно качество цвета и деталей, не сжимать сильнее 80-90% качества. |
| .png — растровое изображение без сжатия | Скриншоты с текстом, картинки, на которых очень важны детали | Зависит от содержимого. Белый фон с текстом — сотня килобайт, фото — десятки мегабайт | Если важны маленькие детали |
| .svg — векторное изображение (может содержать только формы из линий с заливкой цветом) | Логотипы, иконки, некоторые иллюстрации | Почти ничего не весит (внутри код с координатами). Обязательно указывать ширину/высоту в html или в svg коде. | Если нужно изменять размеры без потери качества |

## ****Ссылки****

[**<a>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a) — ссылка, **интерактивный текст**, который переносит пользователя на другую страницу, сайт или часть страницы. Содержимое элемента становится текстом ссылки, которую видит пользователь, адрес ссылки указывается в атрибуте **href**.

<a href=""></a>

Путь ссылок может быть **относительным** и **абсолютным.** Относительный используют для ссылок на другие страницы того же сайта. Они могут находиться в отдельной папке, например, pages или там же, где и главная страница index.html. В примере, когда страница catalog.html находится в другой папке pages, путь к этой странице будет таким:

<a href="pages/catalog.html">Каталог товаров</a>

Абсолютный путь полезен, если вы ссылаетесь на другой сайт, у которого есть постоянное доменное имя и URL-адрес. Странные символы в адресе — кириллица, которая кодируется таким образом.

<a href="https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82">Статья про интернет на Википедии</a>

### *https://lms.skillfactory.ru/assets/courseware/v1/5025ba830d023e7f5ba6a1e67a4eeee5/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-html-50.png*****Навигация по странице****

Также этот тег можно использовать как якорь для навигации, т. е. ссылка может вести на любой элемент на странице. Для этого в атрибуте href нужно указать ID элемента, на который ссылается якорь.

Ссылка с якорем содержит символ #, после которого идёт идентификатор.

**Идентификатор** — уникальное имя элемента, которое вы задаете самостоятельно. Есть определенные критерии для его создания:

* + Идентификатор должен обязательно начинаться с латинского символа (A–Z, a–z).
  + Может содержать в себе цифры (0–9), символ дефиса (-) и подчеркивания (\_).
  + Использование русских букв в именах идентификатора недопустимо.

В итоге это выглядит следующим образом:

<a href="#target\_element">Ссылка-якорь</a>

где "target\_element" - это ID элемента (например <p id="target\_element">Текст параграфа</p>).

### *https://lms.skillfactory.ru/assets/courseware/v1/5025ba830d023e7f5ba6a1e67a4eeee5/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-html-50.png*****Атрибут title****

Атрибут **title**—необязательный, но довольно часто встречающийся элемент. Он добавляет поясняющий текст к ссылке в виде всплывающей подсказки. Она отображается, когда курсор мыши задерживается на ссылке.

Его синтаксис (комбинация символов) выглядит следующим образом:

<a title="текст">...</a>

Допустим, вам нужно создать ссылку на страницу "contact.html" c текстом "Контакты" и всплывающей подсказкой "свяжитесь с нами".

В итоге у вас должно получиться:

<a href="contact.html" title="свяжитесь с

нами">Контакты</a>

### *https://lms.skillfactory.ru/assets/courseware/v1/5025ba830d023e7f5ba6a1e67a4eeee5/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-html-50.png*****Атрибут target****

Есть еще один необязательный атрибут **target**. Его используют, когда хотят, чтобы ссылка открылась в новом окне. В качестве значения используется имя окна или фрейма, заданное атрибутом **name** (имя).

Используйте имя\_blank для открытия страницы в новом окне (с текстом “Открыть в новом окне”):

<a href="new.html" target="\_blank">Открыть в новом окне</a>

Но и это ещё не всё! Ссылкой может быть практически любой элемент, например, изображение. С ним вы уже учились работать в предыдущем разделе.

Сейчас нужно обернуть тег  **<img>** в тег **<a>**. Давайте посмотрим, что будет, если поместить одно в другое:

<a href="main.html"><img src="images/logo.svg" alt="logo"></a>

## 4.6. Поток и дерево документа

В HTML-документе есть правила отображения элементов относительно документа в целом и друг друга. Некоторые из них интуитивно понятны. Например, элементы отображаются в том же порядке, что указан в коде: чем раньше элемент находится в коде, тем выше он размещён на странице. Но, конечно, встречается и менее очевидное поведение, которое связано с позиционированием элемента (подробнее об этом вы узнаете в модулях, посвященных CSS).

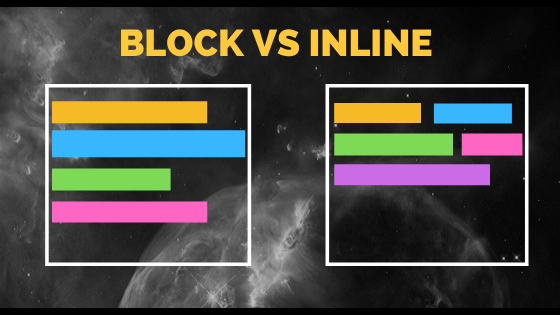
**Поток документа** — порядок отображения элементов на странице. Поток называют прямым, если у элементов нет какого-то заданного позиционирования.

Поток документа ещё называют **flow**.

По типу поведения в структуре документа выделяют две основные категории HTML-элементов:

* 1. **Блочные** (block),
  2. **Строчные**(inline).

Также существуют и комбинированные элементы — строчно-блочные.



## ****Блочные элементы****

* + Располагаются один под другим.
  + Такие элементы имеют ширину, высоту, отступы.
  + Как правило, должны иметь закрывающий тег. Исключение — <hr>.
  + Блоки занимают всю ширину родительского элемента (100%). Например, если задать им фон, то фон будет виден по всей ширине родительского элемента.
  + Так как блочный элемент занимает всю ширину, его внутренние элементы могут быть выровнены горизонтально, то есть по левому, правому краю, посередине и по всей ширине.
  + Блочный элемент может быть внутри другого блочного элемента.
  + В блочные элементы можно вставлять строчные и строчно-блочные элементы.

К самым распространённым **блочным** элементам можно отнести следующие:

[<html>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/html) — документ HTML;

[<body>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/body) — тело документа;

[<div>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/div) —  блок;

[<p>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/p) —  параграф;

[<h1–h6>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Heading_Elements) —  заголовки от самого большого до самого малого;

<[ul](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul" \t "_blank), [ol](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol" \t "_blank), [dl](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/dl" \t "_blank)> —  маркированный, нумерованный список и список определений;

[<form>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/form) —  форма для обмена информацией между клиентом и сервером.

## ****Строчные элементы****

* + В строчный элемент можно вставлять текст и другие строчные элементы.
  + Как правило, в строчные элементы запрещено вставлять блочные. Пример исключения — <a>, которым можно обернуть блок, подробности [тут](http://htmlbook.ru/blog/mozhno-li-oborachivat-ssylkoy-blok).
  + Высота строчных элементов не контролируется свойством height.
  + Ширина пропорциональна ширине контента.
  + Строчные элементы расположены друг за другом в строке, могут переноситься на следующую строку при необходимости.
  + Строчным элементам можно задать вертикальное выравнивание (vertical-align).

К самым распространённым строчным элементам можно отнести следующие:

[<a>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/A) —  гиперссылка;

[<span>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/span) —  строчный элемент;

[<code>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/code) —  код программы.

## ****Строчно-блочные элементы****



* + Можно размещать текст или блочные элементы.
  + Высота элемента рассчитывается автоматически браузером на основе контента.
  + Ширина равна ширине контента с учётом отступов и границ.
  + Inline-block элементы расположены в одной строке и переносятся на следующую при необходимости.
  + Могут быть выровнены по вертикали.
  + Допускается установка ширины и высоты.

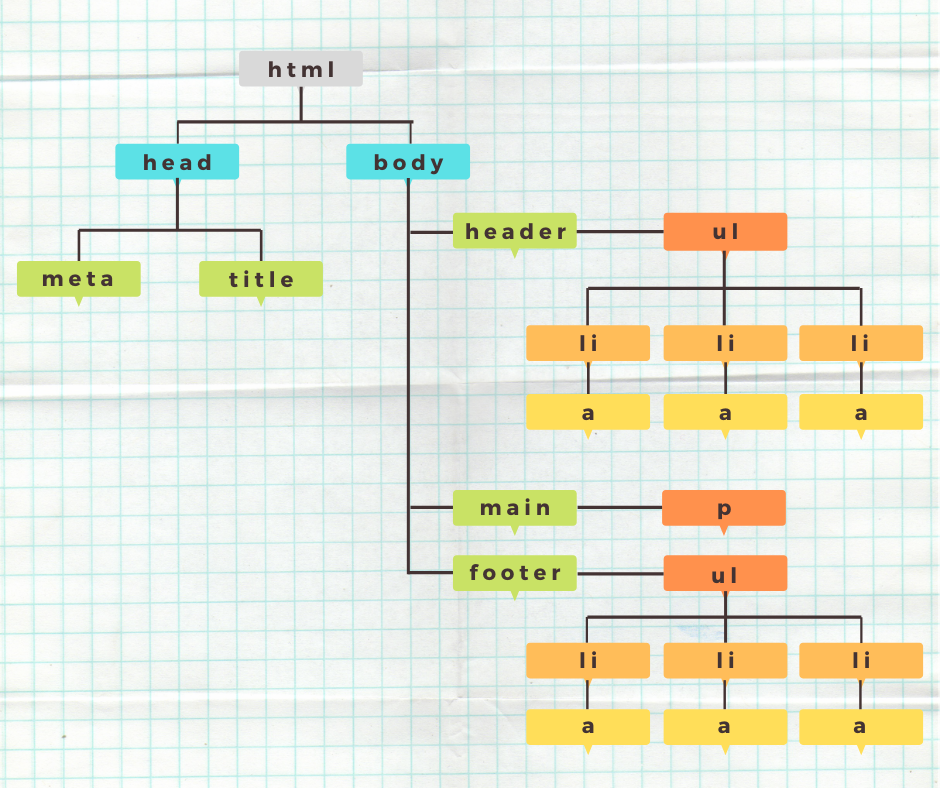
К самым распространённым строчно-блочным элементам можно отнести следующие:

[<button>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/A) —  кнопка;

[<input>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/input) —  интерактивный элемент управления в веб-формах для принятия данных от пользователя;

[<textarea>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/textarea) —  элемент формы, представляющий собой многострочное поле ввода.

Схематичное расположение элементов в документе HTML называют **деревом документа**. Вы уже имели возможность посмотреть на него, когда разбирались с работой браузера.



Между элементами в дереве существуют отношения, которые помогают обратиться к конкретным элементам или узлам. Например, тег htmlназывают**корневым элементом**, так как он связан со всеми остальными ветками и листочками дерева. При этом <!Doctype> элементом не считается.

Другие возможные варианты отношений:

**Предок** — элемент, внутри которого располагаются другие элементы с произвольным уровнем вложения. Если вы посмотрите на картинку, то предками header будут body и html.

**Потомок** — элемент, расположенный внутри одного или нескольких предков. На картинке потомками header будут ul, li и a.

**Родитель** — непосредственный предок элемента, который располагается выше на один уровень в дереве документа. Для main родителем будет body.

**Дочерний элемент** — непосредственный потомок элемента, который располагается ниже на один уровень в дереве документа. Для main дочерним элементом будет p.

**Братья** (сиблинги) —элементы, имеющие общего родителя. Посмотрите на схему, здесь очевидно родство всех li.

**Смежные элементы** — братские элементы, расположенные рядом друг с другом. Кажется, что свойство дублирует предыдущее, но это не совсем так. Например, в нумерованном списке некоторые «братья» точно не смогут увидеться друг с другом.

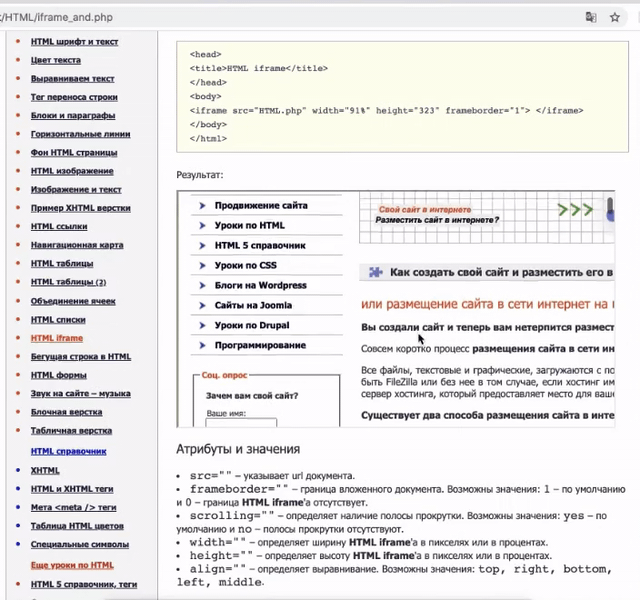
## 4.7. iframe

**iframe**— сокращение от Inline Frame. Frame — это отдельный, завершённый HTML-документ, который может отображаться в браузере вместе с другими HTML-документами.

iframe — мощный элемент в веб-дизайне, с помощью которого можно встраивать в сайт сторонние ресурсы. Но использование данного тега зачастую рискованно: сайт перестаёт полностью контролировать содержимое документа, так как его часть подгружается с другого сайта через iframe.

Поскольку владелец сайта перестаёт контролировать содержимое, то через iframe возможна подгрузка нежелательного контента (18+, вирусы и прочее), что, в свою очередь, может привести к неблагоприятным последствиям. Рассматривать iFrame нужно как часть контента, но не как часть приложения.

Данные элементы используются для отображения веб-страницы внутри веб-страницы (проще говоря, принцип «сайт внутри сайта»).

Источник: [ab-w.net](https://ab-w.net/HTML/frame_and.php)

**Синтаксис**:

<iframe src="URL" width="200" height="200"></iframe>

Тег <iframe> является контейнером, содержание которого игнорируется браузерами, не поддерживающими данный тег. Для таких браузеров можно указать альтернативный текст, который увидят пользователи. Он должен располагаться между элементами <iframe> и </iframe>.

Также тег <iframe>…</iframe> используется для размещения видео во фрейме.

Источник src является источником контента с внешнего или внутреннего сервера. Не забудьте добавить встроенный код в URL.

Width и height — это соотношение сторон кадра. Можно вставить фиксированные размеры, такие как 680 × 480 пикселей (px). Или можно использовать метод на основе процента (10% -100%) для автоматической настройки iframe.

**Важно!** Данный тег поддерживается большинством браузеров, но некоторые параметры могут отличаться или вовсе не запускаться. Например, фреймы не запускаются в браузерах на основе аудио, используемых в автомобильном транспорте, или для людей со слабым зрением, а также на очень старых версиях браузеров.

Если необходимо поддерживать доступность на сайте, стоит задуматься об отказе от использования iframe.

## ****Практика****

Попробуйте воспроизвести следующий пример в собственном .html файле или песочнице:

<ul>

<li><a href="https://www.google.ru" target="myframe">Google</a></li>

<li><a href="http://www.yandex.ru/" target="myframe">Yandex</a></li>

<li><a href="https://duckduckgo.com/" target="myframe">DuckDuckGo</a></li>

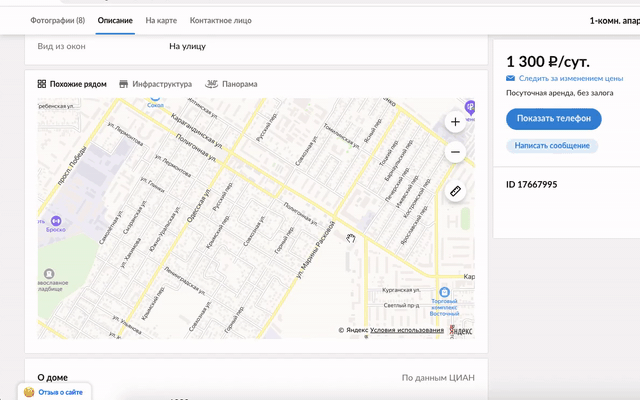
</ul>

<iframe src="http://example.com/" name="myframe"></iframe>

## ****Где используют фреймы****

* + Встроенные видео, презентации и прочий визуальный контент.
  + Для оптимизации внешнего вида приложений служб поддержки.
  + Для создания фонового трафика через партнёрские программы. Применение тега обусловлено накруткой посещений, которые фиксируются системами аналитики и завышают результаты трафика.
  + Web-дизайн.

До появления плагинов iframe использовали повсеместно для встраивания карт, например:

Источник: [orenburg.cian.ru](https://orenburg.cian.ru/rent/flat/193406136/" \t "_blank)

Кроме того, использование iframe может быть обусловлено анимацией, управляемой JavaScript, которая не так медленно работает в отдельном окне.

Из примера выше можно заметить, что фрейм можно открыть, используя тег <a>.

В качестве значения используется имя окна или фрейма, заданное атрибутом name. Если установлено несуществующее имя, то будет открыто новое окно. В качестве зарезервированных имён используются следующие:

**\_blank** — загружает страницу в новое окно браузера.

**\_self** — загружает страницу в текущее окно.

**\_parent** — загружает страницу во фрейм-родитель, если фреймов нет, то это значение работает как \_self.

**\_top** — отменяет все фреймы и загружает страницу в полном окне браузера, если фреймов нет, то это значение работает как \_self.

**Пример вставки в ссылку**:

<a href="https://duckduckgo.com/" target="myframe">DuckDuckGo</a>

<iframe src="http://example.com/" name="myframe"></iframe>

Многие сайты блокируют использование <iframe>…</iframe> и запрещают их самостоятельное использование в качестве фреймов.

## https://lms.skillfactory.ru/assets/courseware/v1/1c671c7793b447ba1c8db2b3b6254c26/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/bulb_1.png****Совет****

Если ваша задача встроить независимые, уже существующие HTML-документы в текущий, используйте <iframe>.

Если же вы пишете всё с нуля, то нет смысла разбивать дизайн страницы на несколько самостоятельных документов. Особенно если они не являются независимыми частями контента.

## 4.8. Семантическая верстка

Начнём с версий *HTML*, а если быть точнее, с *HTML*5. Кратко цели *HTML*5 можно обозначить так:

* + Ликвидация плагинов, таких как *Flash*, для общих функций, которые необходимы каждому. Построить собственную поддержку для таких вещей, как аудио, видео и так далее.
  + Снижение потребности в*JavaScript* и дополнительном коде, благодаря использованию новых *HTML*5 элементов.
  + Обеспечение согласованности между браузерами и устройствами.
  + Обеспечение максимально прозрачной и понятной структуры страницы.

Семантическая вёрстка подразумевает использование специальных *HTML*-элементов с определённой ролью. Эти теги описывают смысл или назначение как для браузеров, так и для разработчиков. Их мы рассмотрим в видео ниже.

Подробнее о семантических элементах [здесь](https://html5book.ru/html5-semantic-elements/) и [здесь](https://www.w3schools.com/html/html5_semantic_elements.asp).

**HEADER** или шапка сайта — верхняя область на сайте, предназначенная для облегчения навигации по странице, первый элемент, который привлекает внимание пользователя. Здесь указывается информация, которая поможет пользователю понять, где он находится и что есть на сайте (например, меню, регистрация, вход в личный кабинет, контакты и так далее).

2

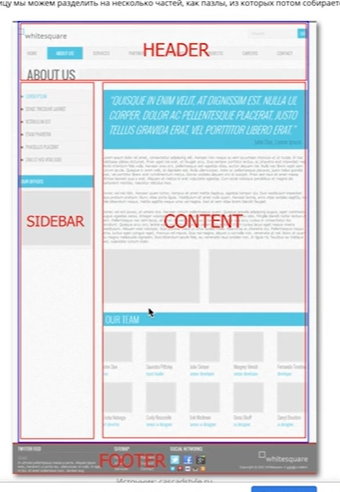
**CONTENT** — это информация (статьи, аудио, видео, изображение и так далее) — всё то, ради чего посетитель приходит на сайт. Контент — основа любого интернет-ресурса, и от его качества зависит посещаемость и заработок на сайте.

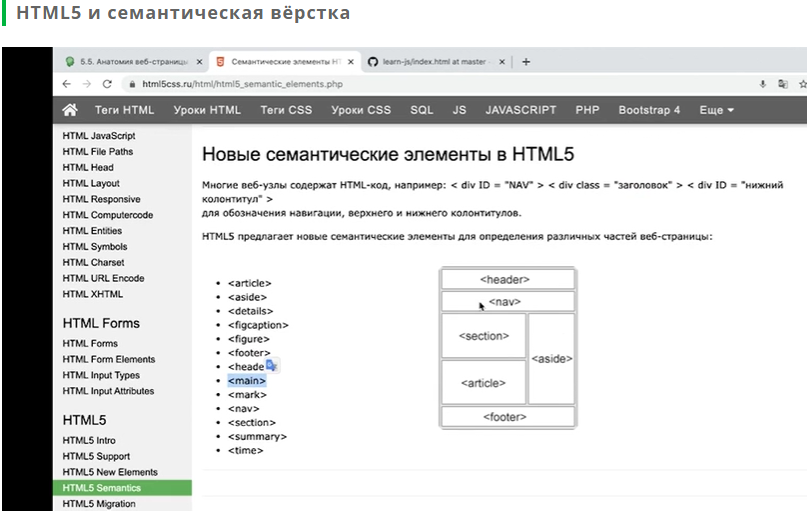
3

**FOOTER** или подвал сайта — нижняя область на сайте, предназначена для логического завершения страницы, облегчения навигации и размещения дополнительных страниц. Там обычно также дублируется навигация по сайту в виде списка с названием разделов, контакты, ссылки на другие ресурсы компании и так далее.

4

**SIDEBAR** — это закреплённая боковая панель ресурса, область навигации или вспомогательной информации, графически отделенная от основной области контента. Сайдбар может находиться как с одной стороны сайта, например, слева, так и с двух. Здесь можно расположить навигационное меню; информационные блоки (например, популярные публикации); функциональные элементы (например, форму поиска, корзину); объявления с рекламой; предложения товаров и услуг; дополнительные виджеты.

****



## ****Дополнительно****

1. [HTML5 Семантические элементы](https://html5css.ru/html/html5_semantic_elements.php),

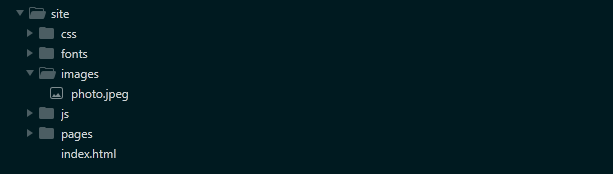
2. [HTML5 Новые элементы](https://html5css.ru/html/html5_new_elements.php).

**4.9. Создание сайта**

Воспользуемся полученными знаниями и **создадим сайт визитку** со структурой посложнее. Можете считать это своим первым ТЗ (то есть техническим заданием, планом работы), но пока делать мы его будем вместе.

* 1. Сайт должен содержать три страницы: главную, «О нас» и «Контакты».
  2. На всех страницах должны быть шапка, основное содержимое и подвал.
  3. Все страницы должны быть связаны.

Начнем с **файловой структуры**: в корневой папке лежат папки CSS, fonts, images, JS, pages.



Разметим главную страницу index:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

</head>

<body>

<header></header>

<main></main>

<footer></footer>

</body>

</html>

Содержимое страницы в **body** часто разделяют на смысловые части с помощью специальных **семантических элементов**:

* **<header>** — шапка сайта, где обычно помещают название, логотип, главное меню, регистрацию и вход для пользователей. Повторяется на всех страницах.
* **<main>** — основное содержимое страницы, уникальное для каждой.
* **<footer>** — подвал или футер, сюда помещают информацию о компании, контакты, ссылки, копирайт, авторство, даты создания сайта.

Создадим в шапке ссылки на остальные страницы. Не забудьте про относительные ссылки, которые меняются в зависимости от файла, в котором пишите код:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

</head>

<body>

<header>

<a href="#">Главная</a>

<a href="pages/about.html">О нас</a>

<a href="pages/contacts.html">Контакты</a>

</header>

<main></main>

<footer></footer>

</body>

</html>

Добавим в main информацию о себе, а в футер - о копирайте, вспомним про использование спецсимволов:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

</head>

<body>

<header>

<a href="#">Главная</a>

<a href="pages/about.html">О нас</a>

<a href="pages/contacts.html">Контакты</a>

</header>

<main>

<h1>Моя первая веб-страница</h1>

<h2>Про меня</h2>

<p>Меня зовут Вася. Я работаю <strong>маркетологом</strong>. Решил получше узнать смежную профессию, чтобы разобраться как устроен веб.</p>

<p>Живу в Питере. Люблю кататься на велосипеде.</p>

<h2>Чему я хочу научиться</h2>

<p>На <em>этом</em> курсе я хотел бы научиться создавать сайты с нуля.</p>

</main>

<footer>

<span>**&copy;&nbsp;**Skillfactory</span>

</footer>

</body>

</html>

На других страницах создайте такую же структуру, но с другим основным содержимым и правильными путями у ссылок. Для **about.html**:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>О нас - Моя первая веб-страница</title>

</head>

<body>

<header>

<a href="../index.html">Главная</a>

<a href="#">О нас</a>

<a href="contacts.html">Контакты</a>

</header>

<main>

<h1>О нас</h1>

<p>Мы лучшие в своем деле. Покупайте слонов у нас!</p>

</main>

<footer>

<span>**&copy;&nbsp;**Skillfactory</span>

</footer>

</body>

</html>

Для **contacts.html**:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

</head>

<body>

<header>

<a href="../index.html">Главная</a>

<a href="about.html">О нас</a>

<a href="#">Контакты</a>

</header>

<main>

<h1>Как нас найти</h1>

<address>

190000, Санкт-Петербург,<br>

Невский проспект, д.1<br>

555-55-55<br>

</address>

</main>

<footer>

<span>**&copy;&nbsp;**Skillfactory</span>

</footer>

</body>

</html>

### https://lms.skillfactory.ru/assets/courseware/v1/7ce0ce2ba6f91320d81359a31c7f5c5e/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-create-50.png****Не забудьте про комментарии****

В коде можно писать **комментарии**, чтобы оставить пометки для себя или разработчиков, а ещё с помощью них можно **отключать** часть кода. Любой код, заключенный в комментарий **<!-- Текст -->** не будет выводиться на странице и пользователь не увидит элемент. Но будет виден в инструментах разработчика. Во многих редакторах кода есть сочетания клавиш, которые позволяют сделать код комментарием, например, в  Sublime — **Ctrl + /**.

Пример кода, отключенного комментарием:

<!-- <span>&copy;&nbsp;Skillfactory</span> -->

Попробуйте временно отключить разные элементы на вашей странице.

## ****Критерии оценки выполненного задания****

**0 баллов**

* + Задание не выполнено.

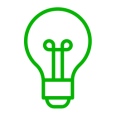
**5 баллов**

* + Заполнена главная веб-страница index.html.
  + Заполнена страница about.html.
  + Заполнена страница contacts.html.

**10 баллов**

* + Текстовое содержимое заключено в теги.
  + Добавлены изображения тегом <img>.
  + Сайт проверен через [Validator W3C](https://validator.w3.org/" \l "validate_by_uri" \t "_blank) и не имеет ошибок.
  + Сайт содержит пояснительные комментарии.

Результат практического задания для проверки вы можете оформить в виде HTML-файлов и, применив навыки полученные в модуле GIT, загрузить их в ваш репозиторий.

** Обратите внимание!**

Вы создаёте публичный репозиторий, поэтому рекомендуется не размещать на странице-резюме те данные, которые вы не хотите обнародовать, например, номер телефона или дату рождения. Вы можете указать "фейковые" (недействительные) данные.

## img****Дополнительно****

* [Секреты использования семантической вёрстки в HTML5](https://medium.com/@stasonmars/%D1%81%D0%B5%D0%BA%D1%80%D0%B5%D1%82%D1%8B-%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F-%D1%81%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B8%CC%86-%D0%B2%D0%B5%D1%80%D1%81%D1%82%D0%BA%D0%B8-%D0%B2-html5-c7cd5e6f1ebb);
* [Справочник по HTML на русском языке](http://htmlbook.ru/html),
* [Официальный стандарт HTML от консорциума всемирной паутины W3C](https://www.w3.org/TR/2021/SPSD-html52-20210128/),
* [Валидатор HTML от W3C](https://validator.w3.org/).

**4.10. Закрепление знаний по HTML**

Закрепим пройденные HTML-элементы:

|  |  |
| --- | --- |
| **Элемент** | **Описание и где используется** |
| <!DOCTYPE html> | Объявление версии HTML документа. В начале всех страниц на современном HTML |
| <html></html> | Содержит весь HTML код. На всех страницах |
| <head></head> | Содержит настройки страницы. На всех страницах |
| <body></body> | Содержимое страницы, которое видит посетитель. На всех страницах |
| <meta> | Настройки страницы. Функция зависит от атрибутов. Когда нужна конкретная настройка |
| <title></title> | Заголовок всей страницы. Виден на вкладке браузера. На всех страницах |
| <h1></h1>  …  <h6></h6> | Заголовки разных уровней. В начале смысловых частей страницы (в начале каждой article и section, если они есть) |
| <p></p> | Параграф текста. Абзац текста, самостоятельный блок, обёртка для смыслового параграфа не обязательно текста (форма с подписью) |
| <strong></strong> | Важная часть текста. Выделить важную часть текста |
| <em></em> | Ударение на часть текста. Поставить интонационное ударение на слово |
| <del></del> | Удаленное из документа. Показать, что что-то устарело или было зачеркнуто |
| <sub></sub>  и  <sup></sup> | Нижний и верхний индекс. Сноски, степени, формулы |
| <q></q> | Цитата. Цитата в тексте |
| <blockquote></blockquote> | Блок с цитатой. Отдельный блок с цитатой |
| <br> | Разрыв строки. Когда нужно начать новую строчку на уровне текстового содержимого |
| <hr> | Отделяющая линия. Отделить части на уровне абзацев |
| <ol></ol> | Упорядоченный список. Список, в котором важен порядок пунктов |
| <ul></ul> | Неупорядоченный список. Список, в котором не важен порядок пунктов |
| <li></li> | Пункт списка. Может содержать почти всё, что угодно (пункт меню, карточки товара) |
| <pre></pre> | Отформатированный текст. Показать больше одного пробела |
| <img> | Изображение. Картинка, которая является частью содержимого |
| <a></a> | Ссылка. Элемент, который куда-то переносит пользователя. Не обязательно текст, но и картинки |
| <!-- --> | Комментарий. Оставить комментарий себе или другим. Временно выключить часть кода |

### https://lms.skillfactory.ru/assets/courseware/v1/bc0f1f0a06f041afbe5fb1a86d42a318/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-restore-page-50.png****Обязательные и популярные атрибуты у части тегов****

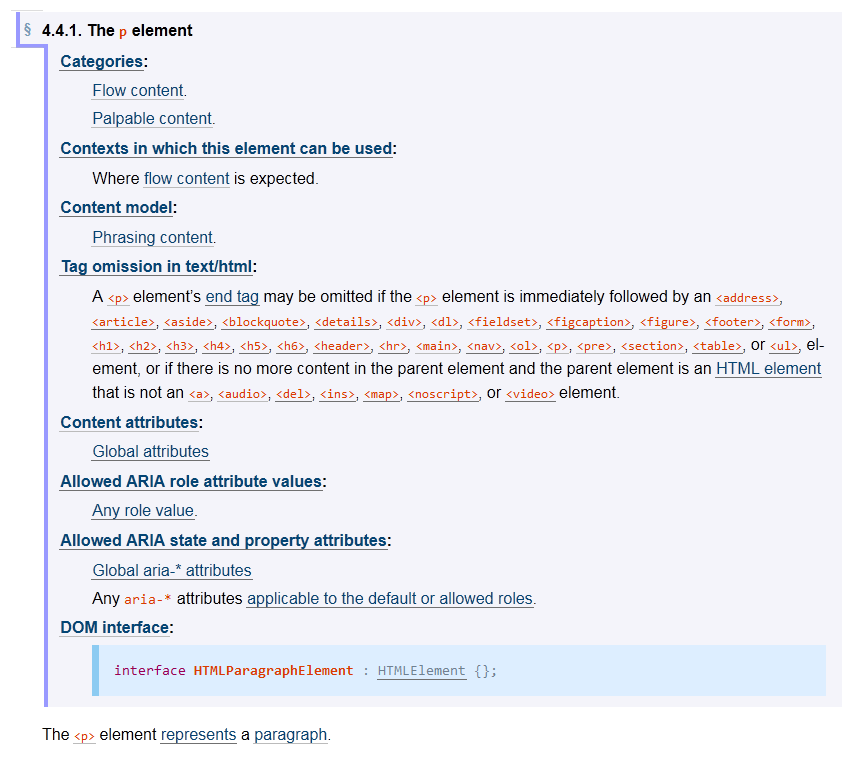
* + <html> —**lang**,
  + <meta> — **charset**, **name**="viewport", name="description",
  + <img> — **src**, **alt**, **width**, **height**,
  + <a> — **href**.

Существует спецификация языка, веб-стандарты, где можно посмотреть список всех правил по элементам. Таких официальных списков два: **W3C** и **WHATWG**, но у них почти нет отличий, так что можно использовать любую из них.

Давайте посмотрим, как пользоваться спецификациями на примере [W3C](https://www.w3.org/TR/html52/). По началу спецификации выглядят пугающе, но на самом деле ими не так сложно пользоваться. Главная страница спецификаций выглядит так:



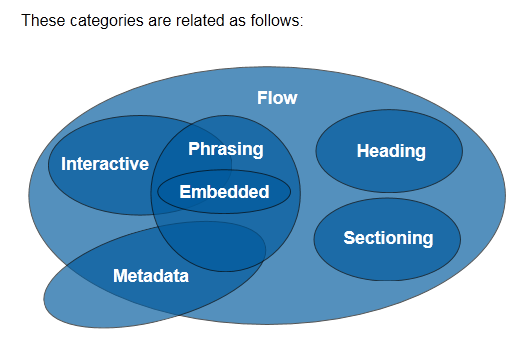
Слева — колонка с содержимым, справа — то, что вы выбрали в оглавлении. В содержимом список всех элементов, искать их удобно с помощью поиска по странице (**Ctrl + F**): наберите нужный элемент и получите информацию. Например, у вас вопрос: «Можно ли вкладывать параграф в параграф». Начинаем вводить **p element**, слева браузер покажет нужный элемент.



### https://lms.skillfactory.ru/assets/courseware/v1/bc0f1f0a06f041afbe5fb1a86d42a318/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-restore-page-50.png****Важные пункты****

* + **Categories** —к какому типу контента относится элемент,
  + **Context in which this element can be used**—куда можно вкладывать элемент,
  + **Content model**—что можно вкладывать в него.

HTML может принадлежать к разным [типам контента](https://www.w3.org/TR/html/dom.html#kinds-of-content). Например, параграф — это **flow** и **palpable content**, внутри него может быть только phrasing. Если не соблюдать спецификации, то браузер может показать не то, что предполагалось, работа с таким кодом усложняется.



В освоении кода помогают и другие сервисы и сайты, например, [документация Mozilla](https://developer.mozilla.org/en-US/docs/Web), где есть информация по HTML, CSS, JS.

### https://lms.skillfactory.ru/assets/courseware/v1/7ce0ce2ba6f91320d81359a31c7f5c5e/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-create-50.png****Важно!****

* + Спецификации могут иметь **разные подходы** и прочтения, поэтому не стоит вслепую копировать код без тестирования.
  + **Браузеры** не всегда одинаково обрабатывают код, особенно при работе с CSS.
  + Одну и ту же задачу можно решить разными способами, **экспериментируйте**, не бойтесь пробовать разные варианты, спрашивать у преподавателей, старших товарищей или в Интернете. .
  + Веб постоянно **развивается**, каждый день появляются новые материалы, технологии и приложения, старайтесь быть в курсе важных изменений.
  + Новые материалы не всегда быстро и качественно переводятся на русский язык, владение **английским** позволит иметь преимущества в этом вопросе.

### https://lms.skillfactory.ru/assets/courseware/v1/0fe7536a55d506add496a6193cf6ad8e/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-file-preview-50.png****Что дальше?****

В следующем модуле мы изучим основы синтаксиса **CSS**, который позволит преображать внешний вид страницы, и приступим к созданию своего личного сайта-визитки. Успехов в учебе!

## img****Дополнительно****

В качестве дополнительного изучения рекомендуем дополнительные ресурсы:

* + [Справочник по HTML на русском языке](http://htmlbook.ru/html),
  + [Официальный стандарт HTML от консорциума всемирной паутины W3C](https://html.spec.whatwg.org/),
  + [Валидатор HTML от W3C](https://validator.w3.org/).