**Основные элементы веб-страницы**

 Добавить страницу в мои закладки

**https://lms.skillfactory.ru/assets/courseware/v1/0a7a03593b2af75dff2c87733bfcb0f1/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-view-50.pngОсновные элементы веб-страницы**

HTML состоит из **элементов**, которые обозначаются**тегами** — открывающая и закрывающая скобка**<>**. У любого элемента есть как минимум один тег, но чаще их два — открывающий <> и закрывающий </> тег. Между тегами находится текстовое содержимое, чтобы всё работало, необходимо закрывать тег там, где это нужно. Необязательно запоминать сразу все теги, можно пользоваться подсказками из сети.  
  
Вот так выглядит элемент параграфа:

<p>Текст параграфа, который увидит пользователь</p>

У элементов могут быть **атрибуты** — дополнительные настройки, которые пишутся внутри открывающего тега. Есть элементы, без которых ваша страница может неправильно отображаться в браузерах у пользователей. С этих элементов должно начинаться создание любого HTML-документа.

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

У некоторых элементов в коде есть **отступы слева**, они показывают уровни **вложенности** элементов: HTML-элементы могут быть вложены друг в друга, и тогда внутренние элементы называют **потомками** или дочерними, а те, в которые они вложены — **родителями**. Это делается для оформления с помощью **CSS**, для семантики или для использования одного элемента в другом. Отступы упрощают чтение кода и не влияют на результат, но их стоит ставить, чтобы упростить работу с кодом. Отступы состоят из пробелов и табов, пробелов может быть 2 или 4, это зависит от принятых в компании правил оформления. Примеры отступов на картинке:



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

Создание любой HTML страницы, как правило, начинается с объявления версии HTML и трех элементов, которые содержат все остальные:

[**<!DOCTYPE html>**](https://developer.mozilla.org/ru/docs/%D0%A1%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8C/Doctype)— тег говорит браузеру, что документ написан на современной версии HTML5.

[**<html>**](https://developer.mozilla.org/ru/docs/Web/HTML/Element/html) — элемент, который содержит остальные элементы HTML, как правило, у него всегда два непосредственных потомка: **<head>** и **<body>**. Для тега HTML с помощью атрибута **lang** можно указать **язык** текста на странице: ru — русский, en — английский, если используется несколько элементов, то для каждого задаётся свой атрибут.

[**<head>**](https://developer.mozilla.org/ru/docs/Web/HTML/Element/head) — здесь находятся элементы, отвечающие за работу и настройку страницы, большинство этих настроек пользователь не увидит.

[**<body>**](https://developer.mozilla.org/ru/docs/Web/HTML/Element/body) — здесь нужно создавать все элементы, содержимое которых увидит пользователь, это контент страницы.

**https://lms.skillfactory.ru/assets/courseware/v1/5025ba830d023e7f5ba6a1e67a4eeee5/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-html-50.pngДругие важные элементы**

[**<meta>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meta) — универсальный элемент, который может содержать настройки в атрибутах, **meta charset** указывает кодировку страницы. Мы будем указывать здесь значение **UTF-8**, это необходимо, чтобы символы у всех пользователей мира отображались правильно.

[**<title>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/title) — заголовок (название или титул) документа, который посетитель увидит на вкладке браузера и в закладках.

[**<h1>...</h1>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Heading_Elements) — это заголовок первого уровня, у любой страницы стоит сделать главный заголовок. Такой заголовок на странице может быть только один.

[**<p>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p) — параграф, самостоятельный блок информации, абзац.

Так выглядит HTML-код внутри <head> на странице [учебника MDN](https://developer.mozilla.org/ru/docs/Web/HTML):



Содержимое **заголовка первого уровня** может совпадать с содержимым элемента **title** в **head**. При этом структура страницы может быть сложной. Если документ состоит из частей, то стоит обозначить каждую часть своим заголовком. Существует шесть уровней заголовков, обычно используется до четырёх уровней на странице. Чем меньше цифра, тем больше размер заголовка. Пример страниц с множеством уровней можно увидеть на Википедии:

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

В коде может содержаться очень много разной информации, но мы начнём с самого необходимого. Перед вами пример минимально работающей страницы с тегами:

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

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

</head>

<body>

<h1>Моя первая веб-страница</h1>

<p>Текст параграфа, который увидит пользователь</p>

</body>

</html>

Создайте содержимое вашей страницы с рассказом о себе и чему вы хотите научиться. Заголовки отвечают на вопрос, о чём пойдет речь в каждой части, и разбивают страницу на части.

<body>

<h1>Моя первая веб-страница</h1>

<h2>Про меня</h2>

<p>Меня зовут Вася. Я работаю маркетологом. 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 этом курсе я хотел бы научиться создавать сайты с нуля.</p>

</body>

**https://lms.skillfactory.ru/assets/courseware/v1/2cad8756a0831f1e384379cd05cdb6c5/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-txt-50.pngЭлементы для форматирования текста:**

[**<strong>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/strong) — оборачивает важный текст, по умолчанию выделяет слово, фразу или весь текст полужирным.

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

<p><strong>Внимание!</strong>. Сайт находится на ремонте. Скоро мы вернемся.</p>

Стоит упомянуть о близком родственнике этого тега — теге [**<b>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/b). Несмотря на визуальную идентичность, данный тег не придает тексту особой важности. С его помощью выделяют часть, которая не более важна, чем основной текст, и просто по каким-либо причинам должна быть выделена (например, ключевые слова в документе).

[**<em>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/em) — ставит ударение на слове или фразе, по умолчанию курсивное написание выделенных элементов.

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

Ударение на слове “этом” подразумевает, что был еще какой-то другой курс.

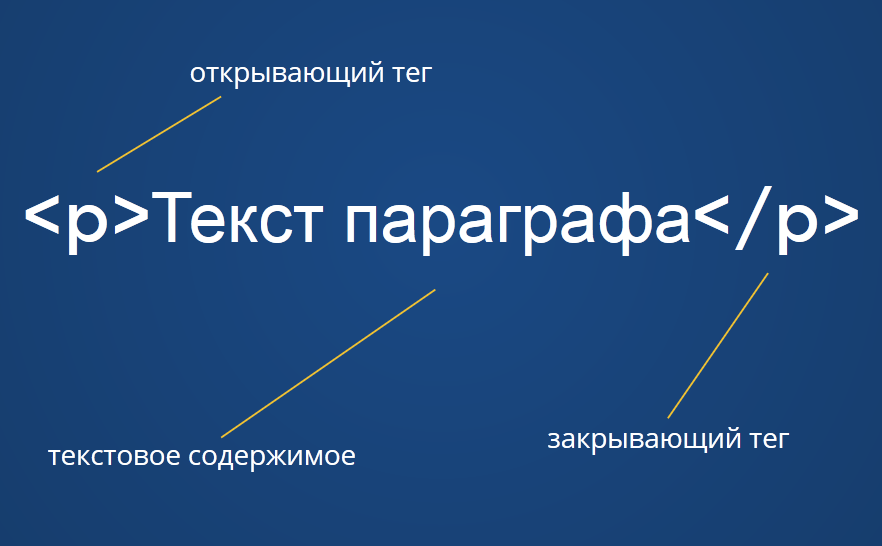
И у этого тега есть близкая родня. Тег [**<i>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/i) используется, когда акцент не меняет смысла предложения, а просто привлекает внимание читателя.

По умолчанию браузеры будут оформлять разные элементы определенным образом. Как это происходит, вы узнаете позже, когда мы будем проходить язык **CSS**. Сейчас вам достаточно понять, что многие элементы передают смысл (семантику) и при этом могут иметь какой угодно внешний вид.

**https://lms.skillfactory.ru/assets/courseware/v1/0a7a03593b2af75dff2c87733bfcb0f1/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-view-50.pngА если что-то не работает?**

Если у вас что-то не работает, отображается не так, как вы хотите, или не отображается вовсе, не спешите менять разметку на другие теги. Сначала проверьте код на популярные **ошибки** или опечатки:

* + Не хватает открывающей или **закрывающей скобки** у тега.
  + Не хватает **косой черты** в закрывающем теге.
  + Не хватает открывающего/закрывающего тега у элемента или в названии тега **опечатка**.
  + Ошибки **вложенности**: закрыли родительский элемент до потомка, поэтому лучше оформлять код отступами.



**Специальные символы**

 Добавить страницу в мои закладки

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

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

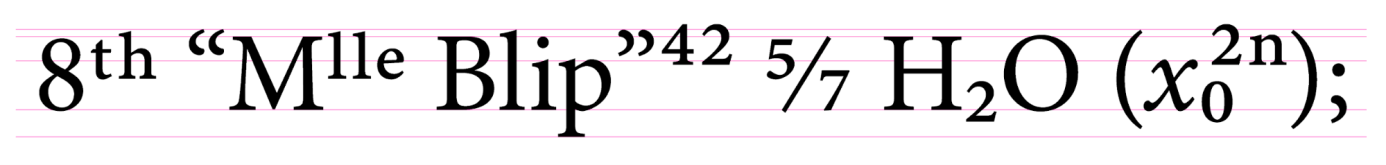
[**<del>**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/del) — вычёркивает текст, показывает, что информация устарела, но важна для сравнения, например, цена товара в магазине.



<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".

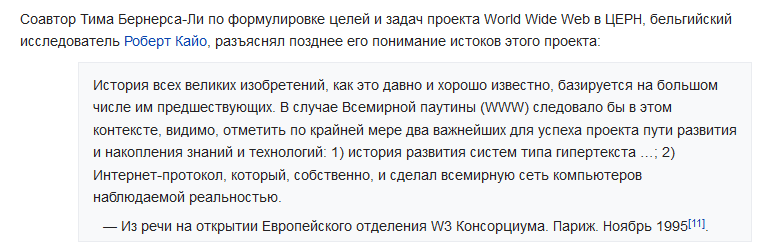
<ul>

<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**

 Добавить страницу в мои закладки

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

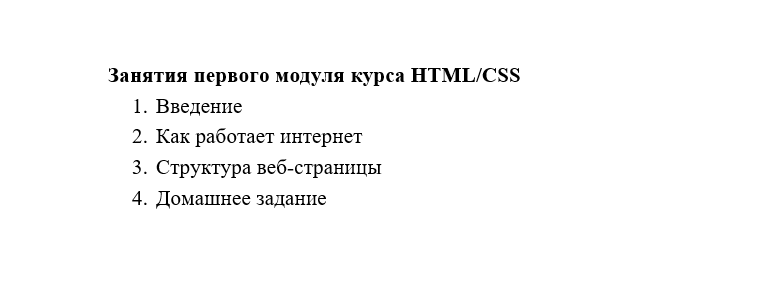
<li>Введение</li>

<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>

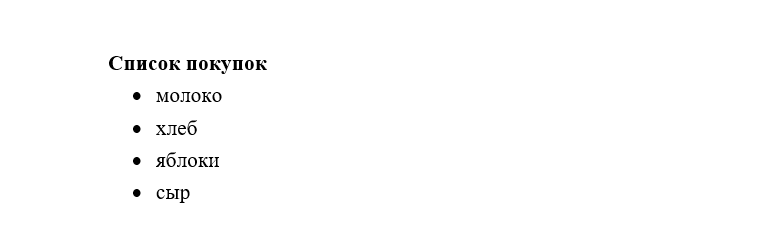
<li>молоко</li>

<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

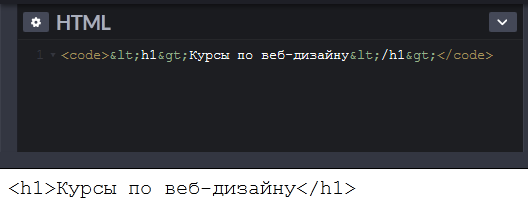
идут авеню,

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**

 Добавить страницу в мои закладки

**https://lms.skillfactory.ru/assets/courseware/v1/55179c1c222ed792bd54ccf578a092cd/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-image-file-50.pngИзображения на сайте**

Мы уже познакомились с разными HTML-элементами, и вы знаете, что они состоят из тегов и текстового содержимого. Но иногда нам 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.

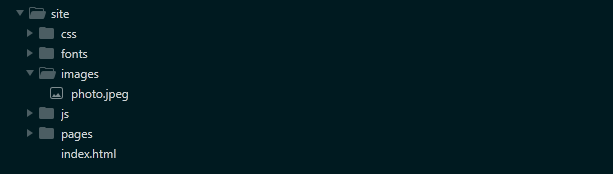
<img src="">

У атрибута обычно есть некое значение, которое пишется в кавычках после пробела. Атрибут **src** указывает путь до графического файла, который браузер вставит на страницу.

**https://lms.skillfactory.ru/assets/courseware/v1/55179c1c222ed792bd54ccf578a092cd/asset-v1:SkillFactory+PHPDEV+2021+type@asset+block/icons8-image-file-50.pngПуть 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.pngОписание 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.pngДругие 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.pngВажные 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.pngВес 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**

 Добавить страницу в мои закладки

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