Профессиональная верстка

Содержание

Введение

HTML

CSS

Javascript

Введение

Дипломный проект — большая практическая работа. В ней вы сможете использовать все полученные знания и закрепить теорию практикой, а результат положите в портфолио. Защита вашего дипломного проекта является обязательной итоговой аттестацией, после прохождения технологических специализаций и для получения диплома после обучения

Тему проекта вы выбираете сами. Это шанс сделать проект мечты: объединить знания, полученные на курсе, ваши интересы или опыт работы. Подумайте, какого проекта не хватает на рынке, и попробуйте его воплотить.  
  
Ваша задача — найти интересную для вас область и применить полученные знания. Важное условие: тема должна быть актуальной, реализуемой и практичной. Здорово, если вы уже работаете в компании и сможете внедрить свои наработки. Поэтому подумайте, какие решения вы могли бы создать для компании, в которой работаете, и какие процессы могли бы улучшить. Если идея не приходит в голову, посмотрите [список победителей в конкурсе ИТ-проектов](https://gb.ru/posts/itogi-konkursa-proektov). Повторять чужие темы не надо, но переосмысливать и вдохновляться — вполне.

Вы должны подготовить текстовую часть диплома. Основные требования к ней:

* Объём не менее 50 страниц. Сюда входит не только текст: учитываются примеры кода, схемы, графики и иллюстрации.
* Оригинальность. Работы будем проверять на плагиат.
* Структурированный документ. В нём должны быть введение, план, основная часть, выводы и список используемых материалов. В основной части расскажите о своей работе над проектом: что делали, какие инструменты использовали, какие были полезными, а какие нет, каких результатов добились.

Если ваш проект предполагает и другой результат, например, готовую программу или сайт, его тоже нужно приложить к текстовой части. Если нет, достаточно только текстовой работы.

Создайте репозиторий на GitHub и соберите простой файл Markdown, как делали на практикуме по контролю версий. Затем составьте из заголовков структуру документа с описанием проекта. Представьте свою работу с точки зрения читателя.

Опишите:

* О чём будет ваш проект.
* Какие технологии вы будете использовать.
* Начните готовить инструкцию. Сперва достаточно просто набросать план, а по мере работы — наполнять диплом. ⠀ # Как набрать нужный объём? Чтобы набрать нужный объём, построить работу можно так:

1. Сперва опишите объект — то, с чем вы будете работать в рамках проекта. Например, язык программирования, фреймворк или конкретный проект, которым вы будете заниматься.
2. Затем начните документировать всё, что делаете и узнаёте: для чего нужен выбранный язык программирования, как его установить, какие задачи он поможет решить в рамках вашего проекта и так далее. Не нужно сразу сдавать готовый проект. Работайте итерациями: начинайте с малого, отправляйте документ на проверку и продолжайте работу с учётом обратной связи. ⠀ # Моя специализация — тестирование. О чём мне писать? Вы можете взять готовый сервис и протестировать его. Или объединиться в команду с программистом и сделать дипломный проект вместе: он напишет код, а вы его протестируете.

При наборе текста используйте шрифт Times New Roman, кегль — 14. Для некоторых структурных единиц работы можно выбрать полужирное выделение.  
  
Имена собственные — наименования компаний/изделий, фамилии учёных и т. д., указывайте на языке оригинала.  
  
Установите межстрочный интервал — 1,5, отступ первой строки — 1,25 мм, выравнивание — по ширине.  
  
Настройте поля страниц, где размещается текст: нижнее и верхнее — 20 мм, правое — 10 мм, левое (для последующей прошивки) — 30 мм.  
  
Все листы дипломного проекта нумеруйте по порядку, начиная с введения (используется сквозная нумерация). Номер указывайте в центре нижней части страницы без точки.  
  
После текста перед следующим заголовком установите одиночный интервал.  
  
После заголовка перед подзаголовком проставьте двойной интервал.  
  
Заголовок раздела с одним подразделом или пунктом не нумеруется.  
  
Каждый структурный элемент начинайте с нового листа.  
  
Вся работа должна составлять 50+ страниц формата А4. Это текст, графики, диаграммы, схемы и прочее.  
  
При печати текст проекта оформляйте только с одной стороны листа формата А4. Объёмные таблицы/иллюстрации можно размещать на листах А3, А2 и выносить в приложения.  
⠀

# Структура дипломного проекта

Дипломный проект должен включать в себя:

* Титульный лист дипломного проекта;
* Содержание дипломного проекта;
* Введение;
* Основную часть;
* Заключение;
* Список использованной литературы и ресурсов;
* Приложения;
* Презентацию к дипломному проекту.

Разделы расположите именно в таком порядке.  
  
Посмотреть пример структуры в Git можно в [здесь](https://github.com/MorgunovaAO/Diploma_project/blob/main/MorgunovaA_group3189.md).  
⠀

# Титульный лист

Титульный лист — это самая первая страница работы. Укажите на ней:

* Название учебного заведения — Geekbrains;
* Специальность;
* Тему дипломного проекта;
* ФИО студента;
* Место и год написания. ⠀

# Содержание

В содержании дипломного проекта перечислите основные блоки и подпункты работы с указанием страниц.

# Введение

Введение раскрывает обоснование необходимости исследования выбранной студентом задачи и представляет схему проведения дипломного исследования.  
  
В этой части в свободной форме опишите:

* Что из себя представляет проект;
* Обоснование темы проекта;
* Цель проекта;
* План работы;
* Какую проблему будет решать ваш проект;
* По какой специализации вы будете делать дипломный проект;
* Есть ли у вас полезный опыт для решения этой задачи. Если да, опишите его;
* Какими инструментами вы будете пользоваться;
* Какие технологии вы планируете использовать при выполнении проекта;
* Состав команды. Даже если делали все сами, напишите, какие роли вы выполняли на проекте. Хороший пример: Участвовала в командном проекте в роли Java-разработчика. Также на проекте были задействованы: проджект-менеджер, дизайнер и тестировщик. ⠀

# Основная часть

Основная часть составляет 80% от всего объёма работы. Содержит 2 главы — теоретическую и практическую, где автор выражает свою позицию относительно гипотез.  
  
Пошагово опишите, как вы делали проект, какие инструменты использовали в работе, на каком этапе использовали инструменты и как они помогли в достижении цели работы.  
  
Расскажите, как вы решили вопросы, перечисленные в введении, какие цели достигли и с каким результатом. Укажите практические примеры, результаты и выводы после каждого элемента исследования.  
⠀

# Заключение

В заключении укажите:

* Краткие и ёмкие теоретические и практические выводы, которые получили во время анализа теоретической базы и практического исследования;
* Оценку проведённого исследования, описание его результатов;
* Практическую значимость работы, рекомендации и планы на дальнейшие исследования;
* Общий итог — достижение цели, выполнение задач, доказательство гипотезы;
* Предложения по совершенствованию объекта исследования. ⠀

# Список литературы

Укажите литературу и ссылки на все ресурсы, которые использовали для создания проектной работы.

## Основные правила оформления использованной литературы и ресурсов

Каждый источник упоминается единожды, независимо от того, насколько часто на него ссылаются.  
  
При использовании нескольких работ одного автора их очерёдность определяется алфавитным порядком названий трудов или годами публикации (в прямой хронологии).  
  
Иностранная литература образует отдельный алфавитный ряд, который размещается после русскоязычных источников.  
  
Список литературы оформляется в алфавитном порядке по фамилии автора, сначала русскоязычная литература, затем иностранная, далее интернет-сайты.  
  
Название города пишется полностью, за исключением следующих общепринятых сокращений для городов Санкт-Петербург (Ленинград), Ростов-на-Дону и Москва: СПб., Л., Ростов н/Д, М.  
  
Список литературы набирается тем же шрифтом, что и основной текст, с соблюдением межстрочного интервала и кегля.  
  
Библиографическая запись обязательно включает:

* Фамилию автора или фамилии их группы, инициалы (при наличии);
* Название статьи, книги, справочника, закона, иного документа;
* Населённый пункт, в котором был издан источник, наименование издательства;
* Год публикации;
* Число страниц. ⠀

# Приложения

В приложения обычно входят:

* Объёмные графики и таблицы, которые не помещаются на лист А4;
* Длинные математические формулы и расчёты по ним;
* Характеристики аппаратуры, которая использовалась для проведения исследования;
* Авторские методики;
* Вспомогательный материал: тесты, карточки, схемы, рисунки;
* Материалы, полученные на предприятии: отчёты, прочие документы.
* Презентация к дипломному проекту
* Презентация вам будет нужна, чтобы при устройстве на работу ярко показать основное из вашего проекта, как портфолио.

Примеры презентаций можно посмотреть в [конкурсных работах на «Лучший ИТ-проект»](https://cloud.mail.ru/public/yQ13/ZJKGagBKj).  
  
Используйте [инструкцию по подготовке презентации](https://docs.google.com/presentation/d/1EAQ8Jz0U89Iz_eLpCG6LSlySuXNqKl-H/edit#slide=id.p1). Удалите лишние слайды, оставьте только те, что подходят для вашей проектной работы.

HTML

**1.1. Основы HTML**

Основы HTML содержат основные правила языка HTML, описание структуры HTML-страницы, отношения в структуре HTML-документа между HTML-элементами.

Гипертекст – это система текстовых страниц, соединенных между собой ссылками. Практически все сайты в интернете представляют собой гипертексты. Этот термин также означает текст, созданный с помощью языка разметки и рассчитанный на использование гиперссылок

HTML-документ — это обычный текстовый документ, может быть создан как в обычном текстовом редакторе (Блокнот), так и в специализированном, с подсветкой кода (Notepad++, Visual Studio Code и т.п.). HTML-документ имеет расширение .html.

HTML-документ состоит из дерева HTML-элементов и текста. Каждый элемент обозначается в исходном документе начальным (открывающим) и конечным (закрывающим) тегом (за редким исключением).

**Начальный тег** показывает, где начинается элемент, конечный — где заканчивается. **Закрывающий тег** образуется путем добавления слэша / перед именем элемента: <имя элемента>…</имя элемента>. Между начальным и закрывающим тегами находится содержимое элемента — контент.

Элементы, представленные одиночными тегами, не могут хранить в себе содержимого напрямую, оно прописывается как значение атрибута, например, элемент <input type="button" value="Кнопка"> создаст кнопку с текстом **Кнопка** внутри.

Элементы могут вкладываться друг в друга, например, <p><i>Текст</i></p>. При вложении следует соблюдать порядок их закрытия **(принцип «матрёшки»)**, например, следующая запись будет неверной: <p><i>Текст</p></i>.

HTML-элементы могут иметь атрибуты (глобальные, применяемые для всех HTML-элементов, и собственные). Атрибуты прописываются в открывающем теге элемента и содержат имя и значение, указываемые в формате имя атрибута="значение". Атрибуты позволяют изменять свойства и поведение элемента, для которого они заданы.

Каждому элементу можно присвоить несколько значений class и только одно значение id. Множественные значения class записываются через пробел, <div class="nav top">. Значения class и id должны состоять только из букв, цифр, дефисов и нижних подчеркиваний и должны начинаться только с букв или цифр.

Браузер просматривает (интерпретирует) HTML-документ, выстраивая его структуру (DOM) и отображая ее в соответствии с инструкциями, включенными в этот файл (таблицы стилей, скрипты). Если разметка правильная, то в окне браузера будет отображена HTML-страница, содержащая HTML-элементы — заголовки, таблицы, изображения и т.д.

Процесс интерпретации **(парсинг)** начинается прежде, чем веб-страница полностью загружена в браузер. Браузеры обрабатывают HTML-документы последовательно, с самого начала, при этом обрабатывая CSS и соотнося таблицы стилей с элементами страницы.

HTML-документ состоит из двух разделов — заголовка — содержимое элемента <head> и содержательной части — содержимое <body>.

**Начальный тег** показывает, где начинается элемент, конечный — где заканчивается. **Закрывающий тег** образуется путем добавления слэша / перед именем элемента: <имя элемента>…</имя элемента>. Между начальным и закрывающим тегами находится содержимое элемента — контент.

Элементы, представленные одиночными тегами, не могут хранить в себе содержимого напрямую, оно прописывается как значение атрибута, например, элемент <input type="button" value="Кнопка"> создаст кнопку с текстом **Кнопка** внутри.

Элементы могут вкладываться друг в друга, например, <p><i>Текст</i></p>. При вложении следует соблюдать порядок их закрытия **(принцип «матрёшки»)**, например, следующая запись будет неверной: <p><i>Текст</p></i>.

HTML-элементы могут иметь атрибуты (глобальные, применяемые для всех HTML-элементов, и собственные). Атрибуты прописываются в открывающем теге элемента и содержат имя и значение, указываемые в формате имя атрибута="значение". Атрибуты позволяют изменять свойства и поведение элемента, для которого они заданы.

Каждому элементу можно присвоить несколько значений class и только одно значение id. Множественные значения class записываются через пробел, <div class="nav top">. Значения class и id должны состоять только из букв, цифр, дефисов и нижних подчеркиваний и должны начинаться только с букв или цифр.

Браузер просматривает (интерпретирует) HTML-документ, выстраивая его структуру (DOM) и отображая ее в соответствии с инструкциями, включенными в этот файл (таблицы стилей, скрипты). Если разметка правильная, то в окне браузера будет отображена HTML-страница, содержащая HTML-элементы — заголовки, таблицы, изображения и т.д.

Процесс интерпретации **(парсинг)** начинается прежде, чем веб-страница полностью загружена в браузер. Браузеры обрабатывают HTML-документы последовательно, с самого начала, при этом обрабатывая CSS и соотнося таблицы стилей с элементами страницы.

HTML-документ состоит из двух разделов — заголовка — содержимое элемента <head> и содержательной части — содержимое <body>.

Тут надо вставить код из ВСКОД

<!DOCTYPE html> <!-- Объявление формата документа -->

<html>

<head> <!-- Техническая информация о документе -->

<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->

<title>...</title> <!-- Задаем заголовок документа -->

<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю таблицу стилей -->

<script src="script.js"></script> <!-- Подключаем сценарии -->

</head>

<body> <!-- Основная часть документа -->

</body>

</html>

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

Чтобы разобраться во взаимодействии элементов веб-страницы, необходимо рассмотреть так называемые **«родственные отношения»** между элементами. Отношения между множественными вложенными элементами подразделяются на родительские, дочерние и сестринские.

* **Предок** — элемент, который заключает в себе другие элементы. На рисунке 1 предком для всех элементов является <html>. В то же время элемент <body> является предком для всех содержащихся в нем элементов: <h1>, <p>, <span>, <nav> и т.д.
* **Потомок** — элемент, расположенный внутри одного или более типов элементов. Например, <body> является потомком <html>, а элемент <p> является потомком одновременно для <body> и <html>.
* **Родительский элемент** — элемент, связанный с другими элементами более низкого уровня, и находящийся на дереве выше их. На рисунке 1 <html> является родительским только для <head> и <body>. Элемент <p> является родительским только для <span>.
* **Дочерний элемент** — элемент, непосредственно подчиненный другому элементу более высокого уровня. На рисунке 1 только элементы <h1>, <h2>, <p> и <nav> являются дочерними по отношению к <body>.
* **Сестринский элемент** — элемент, имеющий общий родительский элемент с рассматриваемым, так называемые элементы одного уровня. На рисунке 1 <head> и <body> — элементы одного уровня, так же как и элементы <h1>, <h2> и <p> являются между собой сестринскими.
* **1.1. Элемент <html>**
* Является корневым элементом документа. Все остальные элементы содержатся внутри <html>...</html>. Все, что находится за пределами элемента, не воспринимается браузером как HTML-код и никак им не обрабатывается.
* Для элемента доступен атрибут, описанный в таблице ниже, а также [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

|  |  |
| --- | --- |
| ТАБЛИЦА 1. АТРИБУТ ЭЛЕМЕНТА <HTML> | |
| **Атрибут** | **Описание, принимаемое значение** |
| manifest | С помощью значения атрибута указывается путь к документу кэша манифеста, например: <html manifest="about\_company.appcache"> |

**1.1. Элемент <html>**

Является корневым элементом документа. Все остальные элементы содержатся внутри <html>...</html>. Все, что находится за пределами элемента, не воспринимается браузером как HTML-код и никак им не обрабатывается.

Для элемента доступен атрибут, описанный в таблице ниже, а также [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

|  |  |
| --- | --- |
| ТАБЛИЦА 1. АТРИБУТ ЭЛЕМЕНТА <HTML> | |
| **Атрибут** | **Описание, принимаемое значение** |
| manifest | С помощью значения атрибута указывается путь к документу кэша манифеста, например: <html manifest="about\_company.appcache"> |

**1.2. HTML-элементы**

**HTML-элементы** — основа языка HTML. 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 элемента.

Все HTML-элементы делятся на пять типов:

* **пустые элементы** — <area>, <base>, <br>, <col>, <embed>, <hr>, <img>, <input>, <link>, <menuitem>, <meta>, <param>, <source>, <track>, <wbr>;
* **элементы с неформатированным текстом** — <script>, <style>;
* **элементы, выводящие неформатированный текст** — <textarea>, <title>;
* **элементы из другого пространства имён** — MathML и SVG;
* **обычные элементы** — все остальные элементы.

В таблице приведен полный список элементов, поддерживаемых HTML4 и HTML5. Экспериментальные и устаревшие элементы исключены. Элементы, добавленные в спецификацию HTML5, выделены бирюзовым цветом.

**Полный список HTML-элементов**

|  |  |
| --- | --- |
| ТАБЛИЦА 1. HTML-ЭЛЕМЕНТЫ | |
| **Элемент** | **Описание** |
| <!--...--> | Используется для добавления комментариев. |
| <!DOCTYPE> | Объявляет тип документа и предоставляет основную информацию для браузера — его язык и версия. |
| <a> | Создаёт гипертекстовые ссылки. |
| <abbr> | Определяет текст как аббревиатуру или акроним. Поясняющий текст задаётся с помощью атрибута title. |
| <address> | Задает контактные данные автора/владельца документа или статьи. Отображается в браузере курсивом. |
| <area> | Представляет собой гиперссылку с текстом, соответствующей определенной области на карте-изображении или активную область внутри карты-изображения. Всегда вложен внутрь элемента <map>. |
| <article> | Раздел контента, который образует независимую часть документа или сайта, например, статья в журнале, запись в блоге, комментарий. |
| <aside> | Представляет контент страницы, который имеет косвенное отношение к основному контенту страницы/сайта. |
| <audio> | Загружает звуковой контент на веб-страницу. |
| <b> | Задает полужирное начертание отрывка текста, не придавая акцент или важность выделенному. |
| <base> | Задает базовый адрес (URL), относительно которого вычисляются все относительные адреса. Это поможет избежать проблем при переносе страницы в другое место, так как все ссылки будут работать, как и прежде. |
| <bdi> | Изолирует отрывок текста, написанный на языке, в котором чтение текста происходит справа налево, от остального текста. |
| <bdo> | Отображает текст в направлении, указанном в атрибуте dir, переопределяя текущее направление написания текста. |
| <blockquote> | Выделяет текст как цитату, применяется для описания больших цитат. |
| <body> | Представляет тело документа (содержимое, не относящееся к метаданным документа). |
| <br> | Перенос текста на новую строку. |
| <button> | Создает интерактивную кнопку. Элемент может содержать текст или изображение. |
| <canvas> | Холст-контейнер для динамического отображения изображений, таких как простые изображения, диаграммы, графики и т.п. Для рисования используется скриптовый язык JavaScript. |
| <caption> | Добавляет подпись к таблице. Вставляется сразу после открывающего тега <table>. |
| <cite> | Используется для указания источника цитирования. Отображается курсивом. |
| <code> | Представляет фрагмент программного кода, отображается шрифтом семейства monospace. |
| <col> | Выбирает для форматирования один или несколько столбцов таблицы, не содержащих информацию одного типа. |
| <colgroup> | Создает структурную группу столбцов, выделяющую множество логически однородных ячеек. |
| <data> | Элемент используется для связывания значения атрибута value, которое представлено в машиночитаемом формате и может быть обработано компьютером, с содержимым элемента. |
| <datalist> | Элемент-контейнер для выпадающего списка элемента <input>. Варианты значений помещаются в элементы <option>. |
| <dd> | Используется для описания термина из элемента <dt>. |
| <del> | Помечает текст как удаленный, перечёркивая его. |
| <details> | Создаёт интерактивный виджет, который пользователь может открыть или закрыть. Представляет собой контейнер для контента, видимый заголовок виджета помещается в элемент <summary>. |
| <dfn> | Определяет слово как термин, выделяя его курсивом. Текст, идущий следом, должен содержать расшифровку этого термина. |
| <dialog> | Интерактивный элемент, с которым взаимодействует пользователь для выполнения задачи, например, диалоговое окно, инспектор или окно. Без атрибута open не виден для пользователя. |
| <div> | Элемент-контейнер для разделов HTML-документа. Используется для группировки блочных элементов с целью форматирования стилями. |
| <dl> | Элемент-контейнер, внутри которого находятся термин и его описание. |
| <dt> | Используется для задания термина. |
| <em> | Выделяет важные фрагменты текста, отображая их курсивом. |
| <embed> | Элемент-контейнер для встраивания внешнего интерактивного контента или плагина. |
| <fieldset> | Группирует связанные элементы в форме, рисуя рамку вокруг них. |
| <figcaption> | Заголовок/подпись для элемента <figure>. |
| <figure> | Самодостаточный элемент-контейнер для такого контента как иллюстрации, диаграммы, фотографии, примеры кода, обычно с подписью. |
| <footer> | Определяет завершающую область (нижний колонтитул) документа или раздела. |
| <form> | Форма для сбора и отправки на сервер информации от пользователей. Не работает без атрибута action. |
| <h1-h6> | Создают заголовки шести уровней для связанных с ними разделов. |
| <head> | Элемент-контейнер для метаданных HTML-документа, таких как <title>, <meta>, <script>, <link>, <style>. |
| <header> | Секция для вводной информации сайта или группы навигационных ссылок. Может содержать один или несколько заголовков, логотип, информацию об авторе. |
| <hr> | Горизонтальная линия для тематического разделения параграфов. |
| <html> | Корневой элемент HTML-документа. Сообщает браузеру, что это HTML-документ. Является контейнером для всех остальных html-элементов. |
| <i> | Выделяет отрывок текста курсивом, не придавая ему дополнительный акцент. |
| <iframe> | Создает встроенный фрейм, загружая в текущий HTML-документ другой документ. |
| <img> | Встраивает изображения в HTML-документ с помощью атрибута src, значением которого является адрес встраиваемого изображения. |
| <input> | Создает многофункциональные поля формы, в которые пользователь может вводить данные. |
| <ins> | Выделяет текст подчеркиванием. Применяется для выделения изменений, вносимых в документ. |
| <kbd> | Выделяет текст, который должен быть введён пользователем с клавиатуры, шрифтом семейства monospace. |
| <label> | Добавляет текстовую метку для элемента <input>. |
| <legend> | Заголовок элементов формы, сгруппированных с помощью элемента <fieldset>. |
| <li> | Элемент маркированного или нумерованного списка. |
| <link> | Определяет отношения между документом и внешним ресурсом. Также используется для подключения внешних таблиц стилей. |
| <main> | Контейнер для основного уникального содержимого документа. На одной странице должно быть не более одного элемента <main>. |
| <map> | Создаёт активные области на карте-изображении. Является контейнером для элементов <area>. |
| <mark> | Выделяет фрагменты текста, помечая их желтым фоном. |
| <meta> | Используется для хранения дополнительной информации о странице. Эту информацию используют браузеры для обработки страницы, а поисковые системы — для ее индексации. В блоке <head> может быть несколько элементов <meta>, так как в зависимости от используемых атрибутов они несут разную информацию. |
| <meter> | Индикатор измерения в заданном диапазоне. |
| <nav> | Раздел документа, содержащий навигационные ссылки по сайту. |
| <noscript> | Определяет секцию, не поддерживающую сценарий (скрипт). |
| <object> | Контейнер для встраивания мультимедиа (например, аудио, видео, Java-апплеты, ActiveX, PDF и Flash). Также можно вставить другую веб-страницу в текущий HTML-документ. Для передачи параметров встраиваемого плагина используется элемент <param>. |
| <ol> | Упорядоченный нумерованный список. Нумерация может быть числовая или алфавитная. |
| <optgroup> | Контейнер с заголовком для группы элементов <option>. |
| <option> | Определяет вариант/опцию для выбора в раскрывающемся списке <select>, <optgroup> или <datalist>. |
| <output> | Поле для вывода результата вычисления, рассчитанного с помощью скрипта. |
| <p> | Параграфы в тексте. |
| <param> | Определяет параметры для плагинов, встраиваемых с помощью элемента <object>. |
| <picture> | Элемент-контейнер, содержащий один элемент <img> и ноль или несколько элементов <source>. Сам по себе ничего не отображает. Дает возможность браузеру выбирать наиболее подходящее изображение. |
| <pre> | Выводит текст без форматирования, с сохранением пробелов и переносов текста. Может быть использован для отображения компьютерного кода, сообщения электронной почты и т.д. |
| <progress> | Индикатор выполнения задачи любого рода. |
| <q> | Определяет краткую цитату. |
| <ruby> | Контейнер для Восточно-Азиатских символов и их расшифровки. |
| <rb> | Определяет вложенный в него текст как базовый компонент аннотации. |
| <rt> | Добавляет краткую характеристику сверху или снизу от символов, заключенных в элементе <ruby>, выводится уменьшенным шрифтом. |
| <rtc> | Отмечает вложенный в него текст как дополнительную аннотацию. |
| <rp> | Выводит альтернативный текст в случае если браузер не поддерживает элемент <ruby>. |
| <s> | Отображает текст, не являющийся актуальным, перечеркнутым. |
| <samp> | Используется для вывода текста, представляющего результат выполнения программного кода или скрипта, а также системные сообщения. Отображается моноширинным шрифтом. |
| <script> | Используется для определения сценария на стороне клиента (обычно JavaScript). Содержит либо текст скрипта, либо указывает на внешний файл сценария с помощью атрибута src. |
| <section> | Определяет логическую область (раздел) страницы, обычно с заголовком. |
| <select> | Элемент управления, позволяющий выбирать значения из предложенного множества. Варианты значений помещаются в <option>. |
| <small> | Отображает текст шрифтом меньшего размера. |
| <source> | Указывает местоположение и тип альтернативных медиаресурсов для элементов <picture>, <video>, <audio>. |
| <span> | Контейнер для строчных элементов. Можно использовать для форматирования отрывков текста, например, выделения цветом отдельных слов. |
| <strong> | Расставляет акценты в тексте, выделяя полужирным. |
| <style> | Подключает встраиваемые таблицы стилей. |
| <sub> | Задает подстрочное написание символов, например, индекса элемента в химических формулах. |
| <summary> | Создаёт видимый заголовок для элемента <details>. Отображается с закрашенным треугольником, кликнув по которому можно просмотреть подробности заголовка. |
| <sup> | Задает надстрочное написание символов. |
| <table> | Элемент для создания таблицы. |
| <tbody> | Определяет тело таблицы. |
| <td> | Создает ячейку таблицы. |
| <template> | Используется для объявления фрагментов HTML-кода, которые могут быть клонированы и вставлены в документ скриптом. Содержимое элемента не является его дочерним элементом. |
| <textarea> | Создает большие поля для ввода текста. |
| <tfoot> | Определяет нижний колонтитул таблицы. |
| <th> | Создает заголовок ячейки таблицы. |
| <thead> | Определяет заголовок таблицы. |
| <time> | Определяет дату/время. |
| <title> | Заголовок HTML-документа, отображаемый в верхней части строки заголовка браузера. Также может отображаться в результатах поиска, поэтому это следует принимать во внимание предоставление названия. |
| <tr> | Создает строку таблицы. |
| <track> | Добавляет субтитры для элементов <audio> и <video>. |
| <u> | Выделяет отрывок текста подчёркиванием, без дополнительного акцента. |
| <ul> | Создает маркированный список. |
| <var> | Выделяет переменные из программ, отображая их курсивом. |
| <video> | Добавляет на страницу видео-файлы. Поддерживает 3 видео формата: MP4, WebM, Ogg. |
| <wbr> | Указывает браузеру возможное место разрыва длинной строки. |

**Таблица-шпаргалка с элементами**

Для удобства пользования я сгруппировала HTML-элементы по тематическим разделам, добавив значения свойства display. Чтобы перейти к таблице, кликните по рисунку.

**1.3. HTML-атрибуты**

**HTML-атрибуты** это специальные слова, которые управляют поведением HTML-элемента. Они добавляют дополнительную функциональность, либо меняют поведение элемента по умолчанию. Атрибуты элемента выражаются внутри начального тега элемента.

Атрибут имеет имя и значение. Имена атрибутов должны состоять из одного или нескольких символов, кроме управляющих, таких как пробел, ", ', >, / и =. Имена и значения атрибутов не чувствительны к регистру, но, тем не менее, рекомендуется набирать их в нижнем регистре.

Некоторые атрибуты не требуют значение, потому что у них есть только одна опция. Они называются **логическими атрибутами**.

Атрибуты могут быть указаны четырьмя различными способами:

* имя атрибута, например, disabled
* значение атрибута без кавычек, например, autocapitalize=sentences
* значение атрибута в одинарных кавычках, например, type='checkbox'
* значение атрибута в двойных кавычках, например, class="external icon-link"

**Глобальные атрибуты**

Глобальные атрибуты, приведенные в таблице ниже, могут быть использованы для любого HTML-элемента, хотя некоторые из них могут не оказывать на элементы никакого влияния.

|  |  |
| --- | --- |
| ТАБЛИЦА 1. ГЛОБАЛЬНЫЕ HTML-АТРИБУТЫ | |
| **Атрибут** | **Описание, принимаемое значение** |
| accesskey | Значение атрибута используется браузером в качестве руководства для создания сочетания клавиш, которое активирует или фокусирует элемент. Состоит из разделенного пробелами списка символов. Браузер в первую очередь выбирает те клавиши, которые существуют на раскладке клавиатуры.  **Синтаксис:** accesskey="A" accesskey="N @ 1" |
| autocapitalize | Дает подсказку браузеру, каким образом вводимый текст будет автоматически писаться с заглавной буквы при вводе/редактировании пользователем. Атрибут не влияет на поведение при наборе текста на физической клавиатуре, только на поведение других механизмов ввода, таких как виртуальные клавиатуры на мобильных устройствах и голосовой ввод. Атрибут autocapitalize никогда не приводит к автоматическому включению заглавных букв для элемента <input> типа url, email или password.  **Разрешенные значения:** off или none — все буквы по умолчанию в нижнем регистре.  on или sentences — первая буква каждого предложения по умолчанию заглавная; все остальные буквы по умолчанию в нижнем регистре.  words — первая буква каждого слова по умолчанию заглавная; все остальные буквы по умолчанию в нижнем регистре.  characters — все буквы по умолчанию должны быть в верхнем регистре.  **Синтаксис:** autocapitalize="sentences" |
| autofocus | Логический атрибут, указывающий, что элемент должен быть сфокусирован при загрузке страницы или при отображении <dialog>, частью которого он является. Не более одного элемента в документе или диалоговом окне может иметь атрибут autofocus. Если применить к нескольким элементам, то фокус получит первый из них.  **Синтаксис:** autofocus |
| class | Представляет собой разделенный пробелом список классов элемента с учетом регистра. Классы позволяют CSS и Javascript выбирать и получать доступ к элементам с помощью селекторов классов или функций, таких как метод DOM document.getElementsByClassName.  **Синтаксис:** class="toc"  class="external icon-link" |
| contenteditable | Указывает, может ли элемент быть доступным для редактирования пользователем. Цвет курсора в области редактирования можно изменить с помощью CSS-свойства caret-color.  **Разрешенные значения:** true или пустая строка — элемент доступен для редактирования.  false — элемент недоступен для редактирования.  **Синтаксис:** contenteditable="false" |
| data-\* | Создает пользовательские атрибуты данных, которые позволяют сценариям обмениваться информацией между HTML и его представлением DOM. Доступ к ним дает свойство HTMLElement.dataset.  Символ \* может быть заменен любым именем в соответствии с правилом создания имен XML со следующими ограничениями: имя не должно начинаться с xml (без учета регистра). имя не должно содержать двоеточие :. имя не должно содержать заглавных букв.  **Синтаксис:** <img class="spaceship cruiserX3" src="shipX3.png" data-ship-id="324" data-weapons="laserI laserII" data-shields="72%" data-x="414354" data-y="85160" data-z="31940" onclick="spaceships[this.dataset.shipId].blasted()"> |
| dir | Указывает направление текста элемента. Является обязательным для элемента <bdo>.  **Разрешенные значения:** ltr — слева направо и должно использоваться для языков, которые пишутся слева направо (например, английский).  rtl — справа налево и должно использоваться для языков, которые пишут справа налево (например, арабский).  auto — позволяет браузеру использовать базовый алгоритм, который анализирует символы внутри элемента, пока не найдет символ с сильной направленностью, а затем применяет эту направленность ко всему элементу.  **Синтаксис:** dir="auto" |
| draggable | Указывает, можно ли перетаскивать элемент либо с помощью собственного поведения браузера, либо с помощью HTML Drag and Drop API.  **Разрешенные значения:** true — элемент можно перетаскивать.  false — элемент нельзя перетащить.  auto — значение по умолчанию, означает, что поведение перетаскивания является поведением браузера по умолчанию: перетаскивать можно только выделенный текст, изображения и ссылки.  **Синтаксис:** draggable="true" |
| enterkeyhint | Указывает, какой текст или значок должен отображаться на клавише ввода виртуальной клавиатуры.  **Разрешенные значения:** enter — символ новой строки.  done — текст, означающий, что больше нечего вводить, и редактор метода ввода (IME) будет закрыт.  go — текст, означающий, что пользователь должен перейти к цели введенного им текста.  next — текст, означающий, что пользователь должен перейти к следующему полю, которое будет принимать текст.  previous — текст, означающий, что пользователь должен перейти в предыдущее поле, которое будет принимать текст.  search — текст, означающий, что пользователь должен перейти к результатам поиска введенного им текста.  send — текст, означающий, что пользователь должен отправить введенный текст.  **Синтаксис:** enterkeyhint="go" |
| hidden | Указывает браузеру, что элемент должен быть скрыт.  **Синтаксис:** hidden |
| id | Определяет уникальный идентификатор элемента. Его цель — идентифицировать элемент при связывании (используя идентификатор фрагмента — якорь), сценарии или стилизации с помощью CSS. Значение id не должно содержать пробелов. В отличие от атрибута class, элементы могут иметь только одно единственное значение идентификатора.  **Синтаксис:** id="content" |
| inputmode | Указывает, какой механизм ввода будет наиболее полезен для пользователей, что позволяет браузеру отображать соответствующую виртуальную клавиатуру. Браузеры могут поддерживать данный атрибут в элементах управления формы, например, <textarea>, или в элементах, для которых задан атрибут contenteditable.  **Разрешенные значения:** none — виртуальная клавиатура не отображается.  text — значение по умолчанию, отображается виртуальная клавиатура для ввода текста.  decimal — отображается клавиатура для ввода дробных чисел, содержащая цифры и десятичный разделитель для языкового стандарта пользователя (обычно . или ,). Устройства могут отображать или не отображать клавишу -.  numeric — отображается цифровая клавиатура ввода. Устройства могут отображать или не отображать клавишу -. Это ключевое слово полезно для ввода PIN-кода.  tel — отображается клавиатура с возможностью ввода номера телефона. Она включает клавиши для цифр от 0 до 9, символ # и \*. Поля, для которых требуется номер телефона, обычно должны использовать вместо этого <input type="tel">.  search — отображается клавиатура, оптимизированная для поиска. Например, клавиша возврата/отправки может быть помечена как «Поиск» вместе с возможными другими оптимизациями. Поля, требующие поискового запроса, должны использовать вместо этого <input type="search">.  email — отображается клавиатура, оптимизированная для ввода адресов электронной почты, например, с символами @ и .. Для полей, которым требуются адреса электронной почты, следует использовать <input type="email">.  url — отображается клавиатура, оптимизированная для ввода URL-адресов, например, с символами / и . и строками www. или .com. Для полей, которым требуется URL-адрес, вместо этого обычно следует использовать <input type="url">.  **Синтаксис:** inputmode="numeric" |
| is | Атрибут является частью [спецификации пользовательских элементов](https://html.spec.whatwg.org/multipage/custom-elements.html), он указывает, что существующий стандартный элемент на самом деле является пользовательским элементом, а значение атрибута задает его имя.  **Синтаксис:** <p is="word-count"> |
| itemid | Предоставляет микроданные в виде уникального глобального идентификатора элемента. Например, книги можно идентифицировать по номеру ISBN. Словари, определяемые атрибутом itemtype, могут быть разработаны таким образом, чтобы элементы однозначно ассоциировались со своим глобальным идентификатором, выражая глобальные идентификаторы в виде URL-адресов, заданных в атрибуте itemid. Точное значение URL-адресов, указанных в itemid, зависит от используемого словаря.  Атрибут может быть указан только для элемента, для которого заданы атрибуты itemscope и itemtype.  **Синтаксис:** itemid="urn:isbn:0-330-34032-8" |
| itemprop | Описывает свойства сущности.  **Синтаксис:** itemprop="name" |
| itemref | Свойства, которые не являются потомками элемента с атрибутом itemscope, могут быть связаны с элементом с помощью глобального атрибута itemref. itemref предоставляет список идентификаторов элементов (не itemid) в другом месте документа с дополнительными свойствами. Атрибут может быть указан только для элементов, для которых указан атрибут itemscope.  Атрибут itemref не является частью модели данных микроданных. Это просто синтаксическая конструкция, помогающая авторам добавлять аннотации к страницам, где аннотируемые данные не соответствуют древовидной структуре. Например, он позволяет авторам размечать данные в таблице, чтобы каждый столбец определял отдельный элемент, сохраняя свойства в ячейках.  **Синтаксис:** itemref="a b" |
| itemscope | Используется для установки области, указывая, что HTML-код, содержащийся в данном блоке, описывает некоторую сущность, заданную itemtype.  **Синтаксис:** itemscope |
| itemtype | Указывает тип сущности, описанной в словаре по данному url.  **Синтаксис:** itemtype="http://schema.org/Movie" |
| lang | Указывает основной язык для содержимого элемента и для любого из атрибутов элемента, содержащих текст. Его значение должно быть допустимым языковым тегом [BCP 47](https://datatracker.ietf.org/doc/html/rfc5646) или пустой строкой. Установка для атрибута пустой строки означает, что основной язык неизвестен.  **Синтаксис:** lang="en-GB" |
| nonce | Определяет криптографический одноразовый номер, который может использоваться политикой безопасности контента для определения того, будет ли разрешена данная выборка для данного элемента. Атрибут полезен для включения элементов, таких как встроенный <script> или <style> в список разрешенных, что поможет избежать использования директивы CSP unsafe-inline.  **Синтаксис:** nonce="8IBTHwOdqNKAWeKl7plt8g==" |
| slot | Используется для назначения [слота](https://html5book.ru/skripty/#slot) элементу.  **Синтаксис:** slot="title" |
| spellcheck | Определяет, может ли элемент быть проверен на наличие орфографических ошибок, является просто подсказкой для браузера. Если этот атрибут не установлен, его значение по умолчанию определяется типом элемента и браузером.  **Разрешенные значения:** true — элемент должен быть по возможности проверен на наличие орфографических ошибок.  false — элемент не следует проверять на орфографические ошибки.  **Синтаксис:** spellcheck="false" |
| style | Содержит объявления стилей CSS, которые следует применить к элементу. Рекомендуется определять стили в отдельном файле или файлах. Этот атрибут и элемент <style> в основном предназначены для быстрой настройки стилей, например, для целей тестирования.  **Синтаксис:** style="color: blue; background: transparent" |
| tabindex | Указывает, что элемент может быть сфокусирован для перемещения вперед через последовательно фокусируемые области (обычно с помощью клавиши Tab). Атрибут tabindex, если он указан, должен иметь целое значение.  Отрицательное значение означает, что элемент недоступен с помощью последовательной навигации с клавиатуры, но его можно сфокусировать с помощью JavaScript или визуально, щелкнув мышью.  Нулевое значение означает, что элемент должен находиться в фокусе при последовательной навигации с помощью клавиатуры после любых положительных значений tabindex, а его порядок определяется исходным порядком документа.  Положительное значение означает, что элемент должен находиться в фокусе при последовательной навигации с помощью клавиатуры, а его порядок определяется значением числа.  **Синтаксис:** tabindex="-1" |
| title | Содержит дополнительную информацию об элементе, задавая всплывающую подсказку для страницы. Атрибут title может содержать несколько строк, каждый символ перевода строки добавляет ее разрыв. Если элемент не имеет атрибута title, то он наследует его от своего родительского элемента, который, в свою очередь, может наследовать его от своего родительского элемента и так далее. Если для этого атрибута задана пустая строка, это означает, что заголовки его родительских элементов не будут отображаться для этого элемента.  **Синтаксис:** title="Hypertext Transport Protocol" |
| translate | Указывает, следует ли переводить переводимые значения атрибута элемента и его дочерние текстовые узлы при переводе страницы. Хотя не все браузеры распознают этот атрибут, он учитывается системами автоматического перевода, такими как Google Translate, а также может учитываться инструментами, используемыми людьми-переводчиками. Поэтому важно, чтобы веб-авторы использовали этот атрибут, чтобы пометить контент, который не следует переводить.  **Разрешенные значения:** yes или пустая строка — элемент должен быть переведен при переводе страницы.  no — элемент не должен быть переведен.  **Синтаксис:** translate="no" |

**1.4. HTML-текст**

**HTML-текст** представлен в спецификации элементами для форматирования и группировки текста. Данные элементы являются контейнерами для текста и не имеют визуального отображения.

Элементы для форматирования текста несут смысловую нагрузку и обычно задают для текста, заключенного внутрь, стилевое оформление, например, выделяют текст жирным начертанием или отображают его шрифтом другого семейства (свойство font-family).

Грамотно отформатированный текст дает понять поисковым системам, какие слова несут важную смысловую нагрузку, по каким из них предпочтительно ранжировать веб-страницу в поисковой выдаче. Вся текстовая информация, отображаемая на сайте, размещается внутри элемента <body>.

**HTML-элементы для текста**

* **СОДЕРЖАНИЕ:**
* [1. Заголовки: <h1...h6>](https://html5book.ru/html-text/#part1)
* [2. Форматирование текста: <b>, <em>, <i>, <small>, <strong>, <sub>, <sup>, <ins>, <del>, <mark>](https://html5book.ru/html-text/#part2)
* [3. Ввод «компьютерного» текста: <code>, <kbd>, <samp>, <var>, <pre>](https://html5book.ru/html-text/#part3)
* [4. Оформление цитат и определений: <abbr>, <bdo>, <blockquote>, <q>, <cite>, <dfn>](https://html5book.ru/html-text/#part4)
* [5. Абзацы, средства переноса текста: <p>, <br>, <hr>](https://html5book.ru/html-text/#part5)

### 1. HTML-элементы для заголовков

* Заголовки являются важными элементами веб-страницы, они упорядочивают текст, формируя его визуальную структуру. Элементы <h1>...<h6> должны использоваться только для выделения заголовков нового раздела или подраздела.
* При использовании заголовков необходимо учитывать их иерархию, т.е. за <h1> должен следовать <h2> и т.д. Также не рекомендуется вкладывать в заголовки другие элементы.

# Заголовок 1-го уровня

## Заголовок 2-го уровня

### Заголовок 3-го уровня

#### Заголовок 4-го уровня

##### Заголовок 5-го уровня

###### Заголовок 6-го уровня

* ФИГУРА. 1. ЭЛЕМЕНТЫ ДЛЯ ЗАГОЛОВКОВ HTML-ДОКУМЕНТА

#### 1.1. Элемент <h1>

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

#### 1.2. Элемент <h2>

* Представляет подзаголовки элемента <h1>. Размер шрифта в браузере равен **1.5em**, верхний и нижний отступ по умолчанию **0.83em**.

#### 1.3. Элемент <h3>

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

#### 1.4. Элементы <h4>, <h5>, <h6>

* Обозначают подзаголовки четвёртого, пятого и шестого уровня. Размер шрифта в браузере равен **1em** / **0.83em** / **0.67em**, верхний и нижний отступ по умолчанию **1.33em** / **1.67em** / **2.33em** соответственно.
* Для всех элементов доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

### 2. Элементы для форматирования текста

#### 2.1. Элемент <b>

* Задаёт полужирное начертания шрифта. Выделяет текст без акцента на его важность.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 2.2. Элемент <em>

* Отображает шрифт курсивом, придавая тексту значимость.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 2.3. Элемент <i>

* Отображает шрифт курсивом.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 2.4. Элемент <small>

* Уменьшает размер шрифта на единицу по отношению к обычному тексту.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 2.5. Элемент <strong>

* Задаёт полужирное начертание шрифта, относится к элементам логической разметки, указывая браузеру на важность текста.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 2.6. Элемент <sub>

* Используется для создания нижних индексов. Сдвигает текст ниже уровня строки, уменьшая его размер.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 2.7. Элемент <sup>

* Используется для создания степеней. Сдвигает текст выше уровня строки, уменьшая его размер.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 2.8. Элемент <ins>

* Выделяет текст в новой версии документа, подчёркивая его.
* Для элемента доступны атрибуты cite и datetime.

#### 2.9. Элемент <del>

* Перечёркивает текст. Используется для выделения текста, удаленного из документа.
* Для элемента доступны атрибуты cite и datetime.

#### 2.10. Элемент <mark>

* Применяется для выделения фрагментов текста в справочных целях, окрашивая блок символов желтым цветом.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

### 3. Элементы для ввода «компьютерного» текста

#### 3.1. Элемент <code>

* Служит для выделения фрагментов программного кода. Отображает текст моноширинным шрифтом.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 3.2. Элемент <kbd>

* Отмечает фрагмент как вводимый пользователем с клавиатуры. Отображает текст моноширинным шрифтом.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 3.3. Элемент <samp>

* Применяется для выделения результата, полученного в ходе выполнения программы. Отображает текст моноширинным шрифтом.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 3.4. Элемент <var>

* Выделяет имена переменных, отображая текст курсивом.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 3.5. Элемент <pre>

* Позволяет вывести текст на экран, сохранив изначальное форматирование. Пробелы и переносы строк при этом не удаляются.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

### 4. Элементы для оформления цитат и определений

#### 4.1. Элемент <abbr>

* Применяется для форматирования аббревиатур. Браузером обычно подчеркивается пунктирной линией. Расшифровка сокращения осуществляется с помощью атрибута title, она появляется при наведении курсора мыши на текст.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 4.2. Элемент <bdo>

* Используется для изменения текущего направления текста.
* Для элемента доступен атрибут dir.

#### 4.3. Элемент <blockquote>

* Выделяет цитаты внутри документа, выделяя его отступами и переносами строк.
* Для элемента доступен атрибут cite.

#### 4.4. Элемент <q>

* Используется для выделения коротких цитат. Браузерами заключается в кавычки.
* Для элемента доступен атрибут cite.

#### 4.5. Элемент <cite>

* Применяется для выделения цитат, названий произведений, сносок на другие документы.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 4.6. Элемент <dfn>

* Позволяет выделить текст как определение. Несмотря на наличие данного элемента, рекомендуется выделять текст силами CSS.
* Для элемента доступен атрибут title.

### 5. Абзацы, средства переноса текста

#### 5.1. Элемент <p>

* Разбивает текст на отдельные абзацы, отделяя друг от друга пустой строкой. Браузер автоматически добавляет верхнее и нижнее внешнее поле margin, равное 1em, при этом поля соседних абзацев «схлопываются».
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 5.2. Элемент <br>

* Переносит текст на следующую строку, создавая разрыв строки.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

#### 5.3. Элемент <hr>

* Используется для разделения контента на веб-странице. Отображается в виде горизонтальной линии.
* Для элемента доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

**1.5. HTML-ссылки**

**HTML-ссылки** создаются с помощью элементов <a>, <area> и <link>. Ссылки представляют собой связь между двумя ресурсами, одним из которых является текущий документ.

Ссылки можно поделить на две категории:

* **ссылки на внешние ресурсы** — создаются с помощью элемента <link> и используются для расширения возможностей текущего документа при обработке браузером;
* **гиперссылки** — ссылки на другие ресурсы, которые пользователь может посетить или загрузить.

## Как сделать гиперссылки на сайте

* **СОДЕРЖАНИЕ:**
* [1. Структура ссылки](https://html5book.ru/hyperlinks-in-html/#part1)
* [2. Абсолютный и относительный путь](https://html5book.ru/hyperlinks-in-html/#part2)
* [3. Якорь](https://html5book.ru/hyperlinks-in-html/#part3)
* [4. Как сделать изображение-ссылку](https://html5book.ru/hyperlinks-in-html/#part4)
* [5. Как сделать ссылку на телефонный номер, скайп или адрес электронной почты](https://html5book.ru/hyperlinks-in-html/#part5)
* [6. Атрибуты ссылок](https://html5book.ru/hyperlinks-in-html/#part6)

### 1. Структура ссылки

Гиперссылки создаются с помощью элемента <a></a>. Внутрь помещается текст, который будет отображаться на веб-странице. Текст ссылки отображается в браузере с подчёркиванием, цвет шрифта — синий, при наведении на ссылку курсор мыши меняет вид.

Обязательным параметром элемента <a> является атрибут href, который задает URl-адрес веб-страницы.

<a href="http://site.ru">указатель ссылки</a>

HTML

Ссылка состоит из двух частей — **указателя** и **адресной части**. **Указатель ссылки** представляет собой фрагмент текста или изображение, видимые для пользователя. **Адресная часть** ссылки пользователю не видна, она представляет собой адрес ресурса, к которому необходимо перейти.

Адресная часть ссылки состоит из URl. **URl** (Uniform Resource Locator) — унифицированный адрес ресурса. При создании адресов для разделения слов между собой рекомендуется использовать дефис, а не символ подчёркивания. В общем виде URl имеющий следующий формат:

метод доступа://имя сервера:порт/путь

HTML

**Метод доступа**, или протокол, осуществляет обмен данными между рабочими станциями в разных сетях. Наиболее распространенные протоколы передачи данных:

file обеспечивает чтение файла с локального диска:

file:/gallery/pictures/summer.html

HTML

http предоставляет доступ к веб-странице по протоколу HTTP:

http://site.ru/

HTML

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

https://site.ru/

HTML

ftp осуществляет запрос к FTP-серверу на получение файла:

ftp://pgu/directory/library

HTML

mailto запускает сеанс почтовой связи с указанным адресатом и хостом:

mailto: nika@gmail.com

HTML

**Имя сервера** описывает полное имя машины в сети, например, site.ru. Если имя сервера не указано, то ссылка считается локальной, т.е. она относится к той же машине, на которой находится HTML-документ, содержащий ссылку.

**Номер порта ТСР**, на котором функционирует веб-сервер. Представляет собой число, которое необходимо указывать, если метод требует номер порта (отдельные сервера могут иметь свой отличительный номер порта). Если порт не указан, по умолчанию используется порт 80. Стандартными портами являются:

* 21 — FTP
* 23 — Telnet
* 70 — Gopher
* 80 — HTTP

**Путь** содержит имя папки, в которой находится файл.

### 2. Абсолютный и относительный путь

Когда в ссылке указывается только имя файла, браузер предполагает, что файл находится в той же папке, что и документ, содержащий гиперссылку. На практике веб-сайты содержат сотни документов, которые размещают в отдельные папки, чтобы ими было легче управлять. Чтобы создать ссылку на файл, находящийся вне папки, содержащей текущий документ, необходимо указать расположение файла или путь. HTML поддерживает два вида пути: абсолютный и относительный.

#### 2.1. Абсолютный путь

**Абсолютный путь** указывает точное местоположение файла в пределах всей структуры папок на компьютере (сервере). Абсолютный путь к файлу даёт доступ к файлу со сторонних ресурсов и содержит следующие компоненты:

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

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

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

//site.ru/pages/tips/tips1.html

HTML

Когда вы ссылаетесь на страницу на другом сайте, вы можете использовать только полный абсолютный путь.

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

http://site.ru/index.html

HTML

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

http://site.ru/

HTML

Обычно в качестве индексного файла выступает документ с именем index.html. Наличие завершающего слеша / означает, что обращение идет к папке, если его нет — напрямую к файлу.

#### 2.2. Относительный путь

**Относительный путь** описывает путь к указанному документу относительно текущего. Путь определяется с учётом местоположения веб-страницы, на которой находится ссылка. Относительные ссылки используются при создании ссылок на другие документы на одном и том же сайте. Когда браузер не находит в ссылке протокол http://, он выполняет поиск указанного документа на том же сервере.

Относительный путь содержит следующие компоненты:

* папка (имя папки, указывающей путь к файлу);
* файл (имя файла).

Путь для относительных ссылок имеет три специальных обозначения:

* / указывает на корневую директорию и говорит о том, что нужно начать путь от корневого каталога документов и идти вниз до следующей папки
* ./ указывает на текущую папку
* ../ подняться на одну папку (директорию) выше

Главное отличие относительного пути от абсолютного в том, что относительный путь не содержит имени корневой папки и родительских папок, что делает адрес короче, и в случае переезда с одного домена на другой не нужно прописывать новый абсолютный адрес. Но если сторонний ресурс будет ссылаться например, на ваши изображения с относительными адресами, то они не будут отображаться на другом сайте.

### 3. Якоря

**Якоря**, или внутренние ссылки, создают переходы на различные разделы текущей веб-страницы, позволяя быстро перемещаться между разделами. Это оказывается очень удобным в случае, когда на странице слишком много текста. Внутренние ссылки также создаются при помощи элемента <a> с разницей в том, что атрибут href содержит имя указателя — так называемый **якорь**, а не URl-адрес. Перед именем указателя всегда ставится знак #.

Следующая разметка создаст оглавление с быстрыми переходами на соответствующие разделы:

<h1>Времена года</h1>

<h2>Оглавление</h2>

<a href="#p1">Лето</a> <!--создаём якорь, указав #id элемента-->

<a href="#p2">Осень</a>

<a href="#p3">Зима</a>

<a href="#p4">Весна</a>

<p id="p1">...</p> <!--добавляем соответствующий id элементу-->

<p id="p2">...</p>

<p id="p3">...</p>

<p id="p4">...</p>

HTML

Если нужно сделать ссылку **с одной страницы** сайта на определенный раздел **другой страницы**, то необходимо задать id для этого раздела страницы, а затем добавить его к абсолютному адресу ссылки:

<th id="about-color">color</th>

<a href="https://html5book.ru/css-shrifty/#about-color" class="site" target="\_blank">color</a>

HTML

### 4. Как сделать изображение-ссылку

Чтобы сделать кликабельное изображение, необходимо поместить элемент <img> внутрь элемента <a>. Чтобы ссылка открывалась в другом окне, нужно добавить атрибут target="\_blank" для ссылки.

<a href="http://www.fast-torrent.ru/film/gran-za-granyu-tv.html" target="\_blank"><img src="//htm5book.ru/wp-content/uploads/2014/07/Fringe.jpg" alt="Fringe"></a>

**5. Как сделать ссылку на телефонный номер, скайп или адрес электронной почты**

У ссылок появились новые возможности — по клику можно не только переходить на другие страницы и скачивать файлы, но и совершать звонки на телефоны, отправлять сообщения или звонить по скайпу.

ссылка на телефонный номер

<a href="tel:+74951234567">+7 (495) 123-45-67</a>

ссылка на адрес электронной почты

<a href="mailto:example@mail.ru">example@mail.ru</a>

ссылка на скайп (позвонить)

<a href="skype:имя-пользователя?call">Skype</a>

ссылка на скайп (открыть чат)

<a href="skype:имя-пользователя?chat">Skype</a>

ссылка на скайп (добавить в список контактов)

<a href="skype:имя-пользователя?add">Skype</a>

ссылка на скайп (отправить файл)

<a href="skype:имя-пользователя?sendfile">Skype</a>

HTML

**6. Атрибуты ссылок**

Элемент <a> поддерживает [глобальные атрибуты](https://html5book.ru/html-attributes/) и собственные.

|  |  |
| --- | --- |
| ТАБЛИЦА 1. АТРИБУТЫ ЭЛЕМЕНТА <A> | |
| **Атрибут** | **Описание, принимаемое значение** |
| download | Дополняет атрибут href и сообщает браузеру, что ресурс должен быть загружен в момент, когда пользователь щелкает по ссылке, вместо того, чтобы, например, предварительно открыть его (как PDF-файл). Задавая имя для атрибута, мы таким образом задаем имя загружаемому объекту. Разрешается использовать атрибут без указания его значения:  <a href="/images/logo.png" download><img src="/images/logo.png" alt="logo"></a>  <a href="/images/logo.png" download="logo"><img src="/images/logo.png" alt="logo"></a>  <a href="files/20022014.pdf" download="Отчет Февраль 2014.pdf">Загрузить отчет за Февраль 2014</a> |
| href | Значением атрибута является URL-адрес документа, на который указывается ссылка. |
| hreflang | Определяет язык связанного веб-документа. Используется только вместе с атрибутом href. Принимаемые значения — аббревиатура, состоящая из пары букв, обозначающих [код языка](http://www.w3schools.com/tags/ref_language_codes.asp), например:  <a href="http://www.anysite.ru" hreflang="en">Anysite</a> |
| media | Определяет, для каких типов устройств файл будет оптимизирован. Значением может быть любой медиа-запрос. |
| ping | Отслеживает URL-адреса ресурсов, по которым переходил пользователь. |
| rel | Дополняет атрибут href информацией об отношении между текущим и связанным документом. Принимаемые значения: alternate — ссылка на альтернативную версию документа (например, печатную форму страницы, перевод или зеркало). author — ссылка на автора документа. bookmark — постоянный URL-адрес, используемый для закладок. help — ссылка на справку. license — ссылка на информацию об авторских правах на данный веб-документ. next/prev — указывает связь между отдельными URL. Благодаря этой разметке Google может определить, что содержание данных страниц связано в логической последовательности. nofollow — запрещает поисковой системе переходить по ссылкам на данной странице или по конкретной ссылке. noreferrer — указывает, что переходе по ссылке браузер не должен посылать заголовок HTTP-запроса (Referrer), в который записывается информация о том, с какой страницы пришел посетитель сайта. prefetch — указывает, что целевой документ должен быть кэширован, т.е. браузер в фоновом режиме загружает содержимое страницы к себе в кэш. search — указывает, что целевой документ содержит инструмент для поиска. tag — указывает ключевое слово для текущего документа. |
| target | Указывает на то, в каком окне должен открываться документ, к которому ведет ссылка. Принимает следующие значения: \_self — страница загружается в текущее окно; \_blank — страница открывается в новом окне браузера; \_parent — страница загружается во фрейм-родитель; \_top — страница загружается в полное окно браузера. |
| type | Указывает [MIME-тип](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_MIME-%D1%82%D0%B8%D0%BF%D0%BE%D0%B2) файлов ссылки, т.е. расширение файла, носит больше справочную информацию. |

**1.6. HTML-изображения**

**HTML-изображения** добавляются на веб-страницы с помощью элемента <img>. Использование графики делает веб-страницы визуально привлекательнее. Изображения помогают лучше передать суть и содержание веб-документа.

Элементы <map> и <area> позволяют создавать **карты-изображения** с активными областями.

## Вставка изображений в HTML-документ

* **СОДЕРЖАНИЕ:**
* [1. Элемент <img>](https://html5book.ru/images-in-html/#part1)
* [1.1. Адрес изображения](https://html5book.ru/images-in-html/#part2)
* [1.2. Размеры изображения](https://html5book.ru/images-in-html/#part3)
* [1.3. Форматы графических файлов](https://html5book.ru/images-in-html/#part4)
* [2. Элемент <map>](https://html5book.ru/images-in-html/#part5)
* [3. Элемент <area>](https://html5book.ru/images-in-html/#part6)
* [4. Пример создания карты-изображения](https://html5book.ru/images-in-html/#part7)

### 1. Элемент <img>

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

Элемент <img> имеет обязательный атрибут src, значением которого является абсолютный или относительный путь к изображению:

<img src="image.png" alt="Пример кода">

HTML

Для элемента <img> доступны следующие атрибуты:

|  |  |
| --- | --- |
| ТАБЛИЦА 1. АТРИБУТЫ ЭЛЕМЕНТА <IMG> | |
| **Атрибут** | **Описание, принимаемое значение** |
| alt | Атрибут alt добавляет альтернативный текст для изображения. Выводится на месте появления изображения до его загрузки или при отключенной графике. Синтаксис: alt="описание изображения". |
| crossorigin | Атрибут crossorigin позволяет загружать изображения с ресурсов другого домена с помощью CORS-запросов. Изображения, загруженные в холст с помощью CORS-запросов, могут быть использованы повторно. Допускаемые значения: anonymous — Cross-origin запрос выполняется с помощью HTTP-заголовка, при этом учетные данные не передаются. Если сервер не даёт учетные данные серверу, с которого запрашивается контент, то изображение будет испорчено и его использование будет ограничено. use-credentials — Cross-origin запрос выполняется с передачей учетных данных. Синтаксис: crossorigin="anonymous". |
| height | Атрибут height задает высоту изображения в px. Синтаксис: height="300". |
| ismap | Атрибутismap указывает на то, что картинка является частью изображения-карты, расположенного на сервере (изображение-карта — изображение с интерактивными областями). При нажатии на изображение-карту координаты передаются на сервер в виде строки запроса URL-адреса. Атрибут ismap допускается только в случае, если элемент <img> является потомком элемента <a> с действительным атрибутом href. Синтаксис: ismap. |
| longdesc | URL расширенного описания изображения, дополняющее атрибут alt. Синтаксис: longdesc="http://www.example.com/description.txt". |
| src | Атрибут src задает путь к изображению. Синтаксис: src="flower.jpg". |
| sizes | Задаёт размер изображения в зависимости от параметров отображения. Работает только при заданном атрибуте srcset. Значением атрибута является одна или несколько строк, указанных через запятую. |
| srcset | Создаёт список источников для изображения, которые будут выбраны, исходя из разрешения экрана. Может использоваться вместе или вместо атрибута src. Значением атрибута является одна или несколько строк, разделенных запятой.  <img src="flower.jpg"  srcset="  img/flower-mobile.jpg 320w,  img/flower-wide-mobile.jpg 480w,  img/flower-tablet.jpg 768w,  img/flower-desktop.jpg 1024w,  img/flower-hires.jpg 1280w"  sizes="  (max-width: 20em) 30vw,  (max-width: 30em) 60vw,  (max-width: 40em) 90vw"  alt="Роза в моём саду">  HTML |
| usemap | Атрибут usemap определяет изображение в качестве карты-изображения. Значение обязательно должно начинаться с символа #. Значение ассоциируется со значением атрибута name или id элемента <map> и создает связь между элементами <img> и <map>. Атрибут нельзя использовать, если элемент <map> является потомком элемента <a> или <button>. Синтаксис: usemap="#mymap". |
| width | Атрибут width задает ширину изображения в px. Синтаксис: width="500". |

#### 1.1. Адрес изображения

Адрес изображения может быть указан полностью (абсолютный URL), например: url(http://anysite.ru/images/anyphoto.png)

Или же через относительный путь от **документа** или **корневого каталога** сайта:

* url(../images/anyphoto.png) — относительный путь от документа,
* url(/images/anyphoto.png) — относительный путь от корневого каталога.

Это интерпретируется следующим образом:

* ../ — означает подняться вверх на один уровень, к корневому каталогу,
* images/ — перейти к папке с изображениями,
* anyphoto.png — указывает на файл изображения.

#### 1.2. Размеры изображения

Без задания размеров изображения рисунок отображается на странице в реальном размере. Отредактировать размеры изображения можно с помощью атрибутов width и height. Если будет задан только один из атрибутов, то второй будет вычисляться автоматически для сохранения пропорций рисунка.

#### 1.3. Форматы графических файлов

* Формат **JPEG** (Joint Photographic Experts Group). Изображения JPEG идеальны для фотографий, они могут содержать миллионы различных цветов. Сжимают изображения лучше GIF, но текст и большие площади со сплошным цветом могут покрыться пятнами.
* Формат **GIF** (Graphics Interchange Format). Идеален для сжатия изображений, в которых есть области со сплошным цветом, например, логотипов. GIF-файлы позволяют установить один из цветов прозрачным, благодаря чему фон веб-страницы может проявляться через часть изображения. Также GIF-файлы могут включать в себя простую анимацию. GIF-изображения содержат всего лишь 256 оттенков, из-за чего изображения выглядят пятнистыми и нереалистичного цвета, как плакаты.
* Формат **PNG** (Portable Network Graphics). Включает в себя лучшие черты GIF- и JPEG-форматов. Содержит 256 цветов и дает возможность сделать один из цветов прозрачным, при этом сжимает изображения в меньший размер, чем GIF-файл.
* Формат **APNG** (Animated Portable Network Graphics). Формат изображения, основанный на формате PNG. Позволяет хранить анимацию, а также поддерживает прозрачность.
* **SVG** (Scalable Vector Graphics). SVG-рисунок состоит из набора геометрических фигур, описанных в формате XML: линия, эллипс, многоугольник и т.п. Поддерживается как статичная, так и анимированная графика. Набор функций включает в себя различные преобразования, альфа-маски, эффекты фильтров, возможность использования шаблонов. Изображения в формате SVG могут изменяться в размере без снижения качества.
* Формат **BMP** (Bitmap Picture). Представляет собой несжатое (оригинальное) растровое изображение, шаблоном которого является прямоугольная сетка пикселей. Bitmap-файл состоит из заголовка, палитры и графических данных. В заголовке хранится информация о графическом изображении и файле (глубина пикселей, высота, ширина и количество цветов). Палитра указывается только в тех Bitmap-файлах, которые содержат палитровые изображения (8 и менее бит) и состоят не более чем из 256 элементов. Графические данные представляют саму картинку. Глубина цвета в данном формате может быть 1, 2, 4, 8, 16, 24, 32, 48 бит на пиксель.
* Формат **ICO** (Windows icon). Формат хранения значков файлов в Microsoft Windows. Также, Windows icon, используется как иконка на сайтах в интернете. Именно картинка такого формата отображается рядом с адресом сайта или закладкой в браузере. Один ICO-файл содержит один или несколько значков, размер и цветность каждого из которых задаётся отдельно. Размер значка может быть любым, но наиболее употребимы квадратные значки со сторонами 16, 32 и 48 пикселей.

### 2. Элемент <map>

Элемент <map> служит для представления графического изображения в виде карты с активными областями. Активные области определяются по изменению вида курсора мыши при наведении. Щелкая мышью на активных областях, пользователь может переходить к связанным документам.

Для элемента доступен атрибут name, который задает имя карты. Значение атрибут name для элемента <map> должно соответствовать имени в атрибуте usemap элемента <img>:

<img src="url" usemap="#имя\_карты">

<map name="имя\_карты">

...

</map>

HTML

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

### 3. Элемент <area>

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

<map name="имя\_карты">

<area атрибуты>

</map>

HTML

|  |  |
| --- | --- |
| ТАБЛИЦА 2. АТРИБУТЫ ЭЛЕМЕНТА <AREA> | |
| **Атрибут** | **Краткое описание** |
| alt | Задает альтернативный текст для активной области карты. |
| coords | Определяет позицию области на экране. Координаты задаются в единицах длины и разделяются запятыми: для **круга** — координаты центра и радиус круга; для **прямоугольника** — координаты верхнего левого и правого нижнего углов; для **многоугольника** — координаты вершин многоугольника в нужном порядке, также рекомендуется указывать последние координаты, равные первым, для логического завершения фигуры. |
| download | Дополняет атрибут href и сообщает браузеру, что ресурс должен быть загружен в момент, когда пользователь щелкает по ссылке, вместо того, чтобы, например, предварительно открыть его (как PDF-файл). Задавая имя для атрибута, мы таким образом задаем имя загружаемому объекту. Разрешается использовать атрибут без указания его значения. |
| href | Указывает URL-адрес для ссылки. Может быть указан абсолютный или относительный адрес ссылки. |
| hreflang | Определяет язык связанного веб-документа. Используется только вместе с атрибутом href. Принимаемые значения — аббревиатура, состоящая из пары букв, обозначающих код языка. |
| media | Определяет, для каких типов устройств файл будет оптимизирован. Значением может быть любой медиа-запрос. |
| rel | Дополняет атрибут href информацией об отношении между текущим и связанным документом. Принимаемые значения: alternate — ссылка на альтернативную версию документа (например, печатную форму страницы, перевод или зеркало). author — ссылка на автора документа. bookmark — постоянный URL-адрес, используемый для закладок. help — ссылка на справку. license — ссылка на информацию об авторских правах на данный веб-документ. next/prev — указывает связь между отдельными URL. Благодаря этой разметке Google может определить, что содержание данных страниц связано в логической последовательности. nofollow — запрещает поисковой системе переходить по ссылкам на данной странице или по конкретной ссылке. noreferrer — указывает, что переходе по ссылке браузер не должен посылать заголовок HTTP-запроса (Referrer), в который записывается информация о том, с какой страницы пришел посетитель сайта. prefetch — указывает, что целевой документ должен быть кэширован, т.е. браузер в фоновом режиме загружает содержимое страницы к себе в кэш. search — указывает, что целевой документ содержит инструмент для поиска. tag — указывает ключевое слово для текущего документа. |
| shape | Задает форму активной области на карте и ее координаты. Может принимать следующие значения: rect — активная область прямоугольной формы; circle — активная область в форме круга; poly — активная область в форме многоугольника; default — активная область занимает всю площадь изображения. |
| target | Указывает, куда будет загружен документ при переходе по ссылке. Принимает следующие значения: \_self — страница загружается в текущее окно; \_blank — страница открывается в новом окне браузера; \_parent — страница загружается во фрейм-родитель; \_top — страница загружается в полное окно браузера. |
| type | Указывает MIME-тип файлов ссылки, т.е. расширение файла. |

1. Размечаем исходное изображение на активные области нужной формы. Координаты областей можно вычислить с помощью программы для обработки фотографий, например, **Adobe Photoshop** или **Paint**.
2. Задаем имя карты, добавив ее в элемент <map> с помощью атрибута name. Это же значение присваиваем атрибуту usemap элемента <img>.
3. Добавляем ссылки на веб-страницы или части веб-документа для каждой активной области, по которым пользователь будет переходить при нажатии курсором мыши на активную область изображения.

<img src="https://html5book.ru/wp-content/uploads/2014/12/flowers\_foto.jpg" alt="flowers\_foto" width="680" height="383" usemap="#flowers">

<map name="flowers">

<area shape="circle" coords="70,164,50" href="https://ru.wikipedia.org/wiki/Гербера" alt="gerbera" target="\_blank">

<area shape="poly" coords="191,13,240,98,143,98,191,13" href="https://ru.wikipedia.org/wiki/%C3%E8%E0%F6%E8%ED%F2" alt="hyacinth" target="\_blank">

<area shape="circle" coords="318,93,50" href="https://ru.wikipedia.org/wiki/Ромашка" alt="camomiles" target="\_blank">

<area shape="circle" coords="425,129,45" href="https://ru.wikipedia.org/wiki/Нарцисс\_(растение)" alt="narcissus" target="\_blank">

<area shape="rect" coords="480,3,572,89" href="https://ru.wikipedia.org/wiki/Тюльпан" alt="tulip" target="\_blank">

</map>

# 1.7. HTML-таблицы

**HTML-таблицы** упорядочивают и выводят на экран данные с помощью строк или столбцов. Таблицы состоят из ячеек, образующихся при пересечении строк и столбцов.

**Ячейки таблиц** могут содержать любые HTML-элементы, такие как заголовки, списки, текст, изображения, элементы форм, а также другие таблицы. Каждой таблице можно добавить связанный с ней заголовок, расположив его перед таблицей или после неё.

Таблицы больше не используются для вёрстки веб-страниц и компоновки отдельных элементов, потому что такой приём не обеспечивает гибкость структуры и адаптивность сайта, существенно увеличивая HTML-разметку.

Для всех элементов таблицы доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/), а также собственные атрибуты.

## Создание таблиц в HTML

* **СОДЕРЖАНИЕ:**
* [1. Как создать таблицу](https://html5book.ru/html-table/#table)
* [2. Как создать строки (ряды) таблицы](https://html5book.ru/html-table/#tr)
* [3. Как сделать ячейку заголовка столбца таблицы](https://html5book.ru/html-table/#th)
* [4. Как сделать ячейку тела таблицы](https://html5book.ru/html-table/#td)
* [5. Как добавить подпись (заголовок) к таблице](https://html5book.ru/html-table/#caption)
* [6. Группирование строк и столбцов таблицы <colgroup> и <col>](https://html5book.ru/html-table/#colgroup-col)
* [7. Группировка разделов таблицы <thead>, <tbody> и <tfoot>](https://html5book.ru/html-table/#grouping)
* [8. Как объединить ячейки таблицы](https://html5book.ru/html-table/#grouping-cell)
* [9. Атрибуты элементов таблицы](https://html5book.ru/html-table/#table-attributes)
* [10. Пример создания таблицы](https://html5book.ru/html-table/#table-example)

### 1. Как создать таблицу

Таблица создаётся при помощи элемента <table></table>, который является контейнером для элементов таблицы и все элементы должны находиться внутри него.

Например, с помощью данной разметки можно создать таблицу, состоящую из **двух столбцов и двух строк**:

<table>

<tr><th>текст заголовка</th><th>текст заголовка</th></tr> <!--ряд с ячейками заголовков-->

<tr><td>данные</td><td>данные</td></tr> <!--ряд с ячейками тела таблицы-->

</table>

HTML

|  |  |
| --- | --- |
| **текст заголовка** | **текст заголовка** |
| данные | данные |

ФИГУРА 1. ВНЕШНИЙ ВИД ТАБЛИЦЫ БЕЗ ФОРМАТИРОВАНИЯ CSS-СВОЙСТВАМИ

По умолчанию таблица и ячейки не имеют видимых границ. **Границы** задаются с помощью свойства border:

/\* внешние границы таблицы серого цвета толщиной 1px \*/

table {

border: 1px solid grey;

}

/\* границы ячеек первого ряда таблицы \*/

th {

border: 1px solid grey;

}

/\* границы ячеек тела таблицы \*/

td {

border: 1px solid grey;

}

CSS

**Промежутки между ячейками таблицы** убираются с помощью свойства table {border-collapse: collapse;}.

**Ширина** таблицы по умолчанию равна ширине её внутреннего содержимого. Чтобы установить ширину, нужно задать значение для свойства width:

/\* сделает ширину таблицы равной ширине блока контейнера, в котором она находится \*/

table {

width: 100%;

}

/\* задаст фиксированную ширину для таблицы \*/

table {

width: 600px;

}

CSS

Если для ячеек таблицы заданы внутренние отступы и границы, то ширина таблицы будет включать в себя следующие значения: padding-left и padding-right, ширина border-left плюс ширина border-right последней ячейки в ряду.

Если заданы ширина и границы ячеек, то ширина таблицы будет складываться из ширины ячеек плюс ширина border-left и ширина border-right последней ячейки в ряду.

### 2. Как создать строки (ряды) таблицы

Строки или ряды таблицы создаются с помощью элемента <tr>. Количество горизонтальных строк таблицы определяется количеством элементов <tr></tr>.

### 3. Как сделать ячейку заголовка столбца таблицы

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

<table>

<tr><th>ячейка заголовка</th><th>ячейка заголовка</th></tr>

</table>

HTML

### 4. Как сделать ячейку тела таблицы

Элемент <td> создаёт ячейки таблицы, внутрь которых помещаются данные таблицы. Элементы <td></td>, расположенные в одном ряду, определяют количество ячеек в строке таблицы.

Количество пар ячеек <td> должно быть равно количеству пар ячеек <th>. Для элемента доступны атрибуты colspan, rowspan, headers.

<table>

<tr><th>ячейка заголовка</th><th>ячейка заголовка</th></tr>

<tr><td>ячейка тела таблицы</td><td>ячейка тела таблицы</td></tr>

</table>

HTML

### 5. Как добавить подпись (заголовок) к таблице

Элемент <caption> создает подпись таблицы. Добавляется непосредственно после тега <table>, вне строки или ячейки.

<table>

<caption>Перечень продуктов</caption>

<tr>

<th>№ п/п</th>

<th>Наименование товара</th>

<th>Ед. изм.</th>

<th>Количество</th>

<th>Цена за ед. изм., руб.</th>

<th>Стоимость, руб.</th>

</tr>

<tr>

<td>1.</td>

<td>Томаты свежие</td><td>кг</td><td>15,20</td><td>69,00</td><td>1048,80</td>

</tr>

<tr>

<td>2.</td>

<td>Огурцы свежие</td><td>кг</td><td>2,50</td><td>48,00</td><td>120,00</td>

</tr>

<tr>

<td colspan="5" style="text-align:right">ИТОГО:</td><td>1168,80</td>

</tr>

</table>

HTML

### 6. Группирование строк и столбцов таблицы

Элемент <colgroup> создает структурную группу столбцов, выделяя логически однородные ячейки. Группирует один или более столбцов для единого форматирования, позволяя применить стили к столбцам вместо того, чтобы повторять стили для каждой ячейки и для каждой строки.

Добавляется непосредственно после тегов <table> и/или <caption>.

Элемент <col> формирует группы столбцов, которые делят таблицу на разделы, не относящиеся к общей структуре, т.е. не содержащие информацию одного типа. Позволяет задавать свойства столбцов для каждого столбца в пределах элемента <colgroup>.

С помощью атрибута style можно изменить основной цвет фона ячеек. Для элемента <col> доступен атрибут span, задающий количество столбцов для объединения.

<table>

<colgroup>

<col span="2" style="background:Khaki"><!-- С помощью этой конструкции задаем цвет фона для первых двух столбцов таблицы-->

<col style="background-color:LightCyan"><!-- Задаем цвет фона для следующего (одного) столбца таблицы-->

</colgroup>

<tr>

<th>№ п/п</th>

<th>Наименование</th>

<th>Цена, руб.</th>

</tr>

<tr>

<td>1</td>

<td>Карандаш цветной</td>

<td>20,00</td>

</tr>

<tr>

<td>2</td>

<td>Линейка 20 см</td>

<td>30,00</td>

</tr>

</table>

**7. Группировка разделов таблицы**

Элемент <thead> создает группу заголовков для строк таблицы с целью задания единого оформления. Используется в сочетании с элементами <tbody> и <tfoot> для указания каждой части таблицы.

Элемент должен быть использован в следующем порядке: как дочерний элемент <table>, после <caption> и <colgroup>, и перед <tbody>, <tfoot> и <tr> элементами. В пределах одной таблицы можно использовать один раз.

Элемент <tbody> группирует основное содержимое таблицы. Используется в сочетании с элементами <thead> и <tfoot>.

Элемент <tfoot> создает группу строк для представления информации о суммах или итогах, расположенную в нижней части таблицы. Используется в таблице один раз. Располагается после элемента <thead>, перед элементами <tbody> и <tr>.

<table>

<thead>

<tr>

<th>№ п/п</th>

<th>Наименование товара</th>

<th>Ед. изм.</th>

<th>Количество</th>

<th>Цена за ед. изм., руб.</th>

<th>Стоимость, руб.</th>

</tr>

</thead>

<tfoot>

<tr>

<td colspan="5" style="text-align:right">ИТОГО:</td><td>1168,80</td>

</tr>

</tfoot>

<tbody>

<tr>

<td>1.</td>

<td>Томаты свежие</td><td>кг</td><td>15,20</td><td>69,00</td><td>1048,80</td>

</tr>

<tr>

<td>2.</td>

<td>Огурцы свежие</td><td>кг</td><td>2,50</td><td>48,00</td><td>120,00</td>

</tr>

</tbody>

</table>

HTML

*Такая группировка строк была заложена в стандарте в расчете на то, что обозреватели при отображении длинных таблиц обеспечат прокрутку строк данных при сохранении надзаголовка и подзаголовка неподвижными, а при их выводе на принтер смогут использовать надзаголовок и подзаголовок в качестве колонтитулов страницы. Однако, современные обозреватели этого не делают и либо просто отображают <thead> и <tfoot> как строки данных, либо, в лучшем случае, просто помещают соответствующие строки в начало и конец таблицы.*

**8. Как объединить ячейки таблицы**

Атрибуты colspan и rowspan объединяют ячейки таблицы. Атрибут colspan задает количество ячеек, объединенных по горизонтали, а rowspan — по вертикали.

<table>

<tr>

<th>№ п/п</th>

<th>Наименование товара</th>

<th>Ед. изм.</th>

<th>Количество</th>

<th>Цена за ед. изм., руб.</th>

<th>Стоимость, руб.</th>

</tr>

<tr>

<td>1.</td>

<td>Томаты свежие</td><td>кг</td><td>15,20</td><td>69,00</td><td>1048,80</td>

</tr>

<tr>

<td>2.</td>

<td>Огурцы свежие</td><td>кг</td><td>2,50</td><td>48,00</td><td>120,00</td>

</tr>

<tr>

<td colspan="5" style="text-align:right">ИТОГО:</td><td>1168,80</td><!-- Задаем количество ячеек по горизонтали для объединения-->

</tr>

</table>

**9. Атрибуты элементов таблицы**

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

<table>

<caption>Меню ресторана "Ромашка"</caption>

<tr>

<th rowspan="2" class="first">Кухня</th>

<th colspan="2">Холодные блюда</th>

<th colspan="2">Горячие блюда</th>

<th rowspan="2">Десерты</th>

</tr>

<tr>

<td class="first">Салаты</td>

<td class="first">Закуски</td>

<td class="first">Первые блюда</td>

<td class="first">Вторые блюда</td>

</tr>

<tr>

<td rowspan="3" class="first">Русская</td>

<td>Винегрет</td>

<td>Язык с хреном</td>

<td>Щи с квашеной капустой</td>

<td>Вареники с картошкой</td>

<td>Печеные яблоки с медом</td>

</tr>

<tr>

<td>Оливье</td>

<td>Студень говяжий</td>

<td>Рассольник домашний</td>

<td>Караси запеченые в сметане</td>

<td>Блинчатый пирог</td>

</tr>

<tr>

<td>Сельдь под "шубой"</td>

<td>Судак заливной</td>

<td>Мясная солянка</td>

<td>Котлеты "Пожарские"</td>

<td>Пирожное "Картошка"</td>

</tr>

<tr>

<td rowspan="3" class="first">Испанская</td>

<td>Севиче из гребешков</td>

<td>Эмпанадас</td>

<td>Хлебный суп с чесноком</td>

<td>Паэлья с морепродуктами</td>

<td>Чуррос</td>

</tr>

<tr>

<td>Тимбал из авокадо и тунца</td>

<td>Ахотомате</td>

<td>Астурийская фабада</td>

<td>Свиное раксо</td>

<td>Альмойшавена</td>

</tr>

<tr>

<td>Фасоль с ветчиной</td>

<td>Чанфайна</td>

<td>Рыбный суп с манными клецками </td>

<td>Тортилья картофельная</td>

<td>Бунуэлос</td>

</tr>

<tr>

<td rowspan="3" class="first">Французская</td>

<td>Вогезский салат</td>

<td>Рийет из курицы</td>

<td>Баклажанный крем-суп "Ренуар"</td>

<td>Картофель огратен</td>

<td>Бриоши</td>

</tr>

<tr>

<td>Салат "Панзанелла"</td>

<td>Делисьез из сыра</td>

<td>Французский тыквенный суп</td>

<td>Гратин из птицы</td>

<td>Лигурийский лимонный пирог</td>

</tr>

<tr>

<td>Тар-тар</td>

<td>Маринованный лосось</td>

<td>Суп "Конти"</td>

<td>Тартифлетт</td>

<td>Саварен "Триумф"</td>

</tr>

</table>

HTML

body {

margin: 0;

background: #F4F1F8;

}

table {

border-collapse: collapse;

line-height: 1.1;

font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;

background: radial-gradient(farthest-corner at 50% 50%, white, #DCECF8);

color: #0C213B;

}

caption {

font-family: annabelle, cursive;

font-weight: bold;

font-size: 2em;

padding: 10px;

color: #F3CD26;

text-shadow: 1px 1px 0 rgba(0,0,0,.3);

}

caption:before, caption:after {

content: "\274B";

color: #A9E2CC;

margin: 0 10px;

}

th {

padding: 10px;

border: 1px solid #A9E2CC;

}

td {

font-size: 0.8em;

padding: 5px 7px;

border: 1px solid #A9E2CC;

}

.first {

font-size: 1em;

font-weight: bold;

text-align: center;

}

# 1.8. HTML-списки

**HTML-списки** используются для группировки связанных между собой фрагментов информации. Существует три вида списков:

* **маркированный список** — <ul> — каждый элемент списка <li> отмечается маркером,
* **нумерованный список** — <ol> — каждый элемент списка <li> отмечается цифрой,
* **список определений** — <dl> — состоит из пар термин <dt> — <dd> определение.

Каждый список представляет собой контейнер, внутри которого располагаются элементы списка или пары термин-определение.

Элементы списка ведут себя как блочные элементы, располагаясь друг под другом и занимая всю ширину блока-контейнера. Каждый элемент списка имеет дополнительный блок, расположенный сбоку, который не участвует в компоновке.

## Создание HTML-списков

* **СОДЕРЖАНИЕ:**
* [1. Маркированный список <ul>](https://html5book.ru/html-lists/#part1)
* [2. Нумерованный список <ol>](https://html5book.ru/html-lists/#part2)
* [3. Список определений <dl>](https://html5book.ru/html-lists/#part3)
* [4. Вложенный список](https://html5book.ru/html-lists/#part4)
* [5. Многоуровневый нумерованный список](https://html5book.ru/html-lists/#part5)

### 1. Маркированный список

**Маркированный список** представляет собой неупорядоченный список (от англ. Unordered List). Создаётся с помощью элемента <ul>. В качестве маркера элемента списка выступает метка, например, закрашенный кружок.

Браузеры по умолчанию добавляют следующее форматирование блоку списка:

ul {

padding-left: 40px;

margin-top: 1em;

margin-bottom: 1em;

}

CSS

Каждый элемент списка создаётся с помощью элемента <li> (от англ. List Item).

Для элемента <ul> доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

<ul>

<li>Microsoft</li>

<li>Google</li>

<li>Apple</li>

<li>IBM</li>

</ul>

HTML

* Microsoft
* Google
* Apple
* IBM

ФИГУРА. 1. МАРКИРОВАННЫЙ СПИСОК

### 2. Нумерованный список

**Нумерованный список** создаётся с помощью элемента <ol>. Каждый пункт списка также создаётся с помощью элемента <li>. Браузер нумерует элементы по порядку автоматически и если удалить один или несколько элементов такого списка, то остальные номера будут автоматически пересчитаны.

Блок списка также имеет стили браузера по умолчанию:

ol {

padding-left: 40px;

margin-top: 1em;

margin-bottom: 1em;

}

CSS

Для элемента <li> доступен атрибут value, который позволяет изменить номер по умолчанию для выбранного элемента списка. Например, если для первого пункта списка задать <li value="10">, то остальная нумерация будет пересчитана относительно нового значения.

Для элемента <ol> доступны следующие атрибуты:

|  |  |
| --- | --- |
| ТАБЛИЦА 1. АТРИБУТЫ ЭЛЕМЕНТА <OL> | |
| **Атрибут** | **Описание, принимаемое значение** |
| reversed | Атрибут reversed задает отображение списка в обратном порядке (например, 9, 8, 7…). |
| start | Атрибут start задает начальное значение, от которого пойдет отсчет нумерации, например, конструкция <ol start="10"> первому пункту присвоит порядковый номер «10». Также можно одновременно задавать тип нумерации, например, <ol type="I" start="10">. |
| type | Атрибут type задает вид маркера для использования в списке (в виде букв или цифр). Принимаемые значения: 1 — значение по умолчанию, десятичная нумерация. A — нумерация списка в алфавитном порядке, заглавные буквы (A, B, C, D). a — нумерация списка в алфавитном порядке, строчные буквы (a, b, c, d). I — нумерация римскими заглавными цифрами (I, II, III, IV). i — нумерация римскими строчными цифрами (i, ii, iii, iv). |

<ol>

<li>Microsoft</li>

<li>Google</li>

<li>Apple</li>

<li>IBM</li>

</ol>

HTML

1. Microsoft
2. Google
3. Apple
4. IBM

ФИГУРА. 2. НУМЕРОВАННЫЙ СПИСОК

### 3. Список определений

**Списки определений** создаются с помощью элемента <dl>. Для добавления термина применяется элемент <dt>, а для вставки определения — элемент <dd>.

Блок списка определений имеет следующие стили браузера по умолчанию:

dl {

margin-top: 1em;

margin-bottom: 1em;

}

CSS

Для элементов <dl>, <dt> и <dd> доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

<dl>

<dt>Режиссер:</dt>

<dd>Петр Точилин</dd>

<dt>В ролях:</dt>

<dd>Андрей Гайдулян</dd>

<dd>Алексей Гаврилов</dd>

<dd>Виталий Гогунский</dd>

<dd>Мария Кожевникова</dd>

</dl>

HTML

Режиссер:

Петр Точилин

В ролях:

Андрей Гайдулян

Алексей Гаврилов

Виталий Гогунский

Мария Кожевникова

ФИГУРА. 3. СПИСОК ОПРЕДЕЛЕНИЙ

### 4. Как создать вложенный список

Зачастую возможностей простых списков не хватает, например, при создании оглавления никак не обойтись без **вложенных пунктов**. Разметка для вложенного списка будет следующей:

<ul>

<li>Пункт 1.</li>

<li>Пункт 2.

<ul>

<li>Подпункт 2.1.</li>

<li>Подпункт 2.2.

<ul>

<li>Подпункт 2.2.1.</li>

<li>Подпункт 2.2.2.</li>

</ul>

</li>

<li>Подпункт 2.3.</li>

</ul>

</li>

<li>Пункт 3.</li>

</ul>

HTML

* Пункт 1.
* Пункт 2.
  + Подпункт 2.1.
  + Подпункт 2.2.
    - Подпункт 2.2.1.
    - Подпункт 2.2.2.
  + Подпункт 2.3.
* Пункт 3.

ФИГУРА. 4. ВЛОЖЕННЫЙ СПИСОК

### 5. Многоуровневый нумерованный список

Многоуровневый список используется для отображения элементов списка на разных уровнях с различными отступами. Разметка для многоуровневого нумерованного списка будет следующей:

<ol>

<li>пункт</li>

<li>пункт

<ol>

<li>пункт</li>

<li>пункт</li>

<li>пункт

<ol>

<li>пункт</li>

<li>пункт</li>

<li>пункт</li>

</ol>

</li>

<li>пункт</li>

</ol>

</li>

<li>пункт</li>

<li>пункт</li>

</ol>

HTML

Такая разметка по умолчанию создаст для каждого вложенного списка новую нумерацию, начинающуюся с единицы. Чтобы сделать вложенную нумерацию, нужно использовать следующие свойства:

* counter-reset сбрасывает один или несколько счётчиков, задавая значение для сброса;
* counter-increment задаёт значение приращения счётчика, т.е. с каким шагом будет нумероваться каждый последующий пункт;
* content — генерируемое содержимое, в данном случае отвечает за вывод номера перед каждым пунктом списка.

ol {

/\* убираем стандартную нумерацию \*/

list-style: none;

/\* Идентифицируем счетчик и даем ему имя li. Значение счетчика не указано - по умолчанию оно равно 0 \*/

counter-reset: li;

}

li:before {

/\* Определяем элемент, который будет нумероваться — li. Псевдоэлемент before указывает, что содержимое, вставляемое при помощи свойства content, будет располагаться перед пунктами списка. Здесь же устанавливается значение приращения счетчика (по умолчанию равно 1). \*/

counter-increment: li;

/\* С помощью свойства content выводится номер пункта списка. counters() означает, что генерируемый текст представляет собой значения всех счетчиков с таким именем. Точка в кавычках добавляет разделяющую точку между цифрами, а точка с пробелом добавляется перед содержимым каждого пункта списка \*/

content: counters(li, ".") ". ";

}

CSS

1. пункт
2. пункт
   1. пункт
   2. пункт
   3. пункт
      1. пункт
      2. пункт
      3. пункт
   4. пункт
3. пункт
4. пункт

ФИГУРА. 5. МНОГОУРОВНЕВЫЙ НУМЕРОВАННЫЙ СПИСОК

# 1.9. Спецсимволы HTML

**Спецсимволы HTML**, или символы-мнемоники, представляют собой конструкцию SGML (англ. **Standard Generalized Markup Language** — стандартный обобщённый язык разметки), ссылающуюся на определенные символы из символьного набора документа. В основном они используются для указания символов, которых нет в стандартной компьютерной клавиатуре, либо которые не поддерживает кодировка HTML-страницы (Windows-1251, UTF-8 и т.д.).

Чтобы разместить символ на веб-странице, необходимо указать HTML-код или мнемонику.

Спецсимволы чувствительны к регистру, поэтому их необходимо прописывать точно так, как указано в таблице. Спецсимволы, не имеющие мнемоники, могут не отображаться вовсе или же некорректно отображаться в тех или иных браузерах.

Для вставки символов внутрь тегов воспользуйтесь HTML-кодом символа, а для использования символов в таблицах стилей, например, в качестве значения свойства content — CSS-код.

Спецсимвол наследует **цвет** от цвета текста родительского элемента. Чтобы изменить цвет спецсимвола, можно поместить HTML-код внутрь тега <span> или задать нужное значение свойства color (при вставке спецсимволов через свойство content).

## Популярные спецсимволы HTML

* **СОДЕРЖАНИЕ:**
* [1. Полезные знаки и символы](https://html5book.ru/specsimvoly-html/#part1)
* [2. Знаки пунктуации](https://html5book.ru/specsimvoly-html/#part2)
* [3. Стрелки](https://html5book.ru/specsimvoly-html/#part3)
* [4. Карточные масти](https://html5book.ru/specsimvoly-html/#part4)
* [5. Деньги](https://html5book.ru/specsimvoly-html/#part5)
* [6. Знаки зодиака](https://html5book.ru/specsimvoly-html/#part6)

### 1. Полезные знаки и символы

|  |  |  |  |
| --- | --- | --- | --- |
| **Вид** | **HTML-код** | **CSS-код** | **Описание** |
| **🖂** | &#128386; | \1F582 | Обратная сторона конверта |
| **☃** | &#9731; | \2603 | Снеговик |
| **☦** | &#9766; | \2626 | Православный крест |
| **⚓** | &#9875; | \2693 | Якорь |
| **✆** | &#9990; | \2706 | Знак телефона |
| **☎** | &#9742; | \260E | Телефон |
| **☕** | &#9749; | \2615 | Горячие напитки |
| **✎** | &#9998; | \270E | Карандаш, направленный вправо-вниз |
| **✏** | &#9999; | \270F | Карандаш |
| **✐** | &#10000; | \2710 | Карандаш, направленный вправо-вверх |
| **✑** | &#10001; | \2711 | Незакрашенное острие пера |
| **✒** | &#10002; | \2712 | Закрашенное острие пера |
| **⚜** | &#9884; | \269C | Геральдическая лилия |
| **⛑** | &#9937; | \26D1 | Шлем с белым крестом |
| **⚝** | &#9885; | \269D | Начерченная белая звезда |
| **❄** | &#10052; | \2744 | Снежинка |
| **❤** | &#10084; | \2764 | Закрашенное жирное сердце |
| **❅** | &#10053; | \2745 | Зажатая трилистниками снежинка |
| **❆** | &#10054; | \2746 | Жирная остроугольная снежинка |
| **★** | &#9733; | \2605 | Закрашенная звезда |
| **☆** | &#9734; | \2606 | Незакрашенная звезда |
| **✪** | &#10026; | \272A | Незакрашенная звезда в закрашенном круге |
| **✫** | &#10027; | \272B | Закрашенная звезда с незакрашенным кругом внутри |
| **✯** | &#10031; | \272F | Вращающаяся звезда |
| **❉** | &#10057; | \2749 | Звёздочка с шарообразными окончаниями |
| **❋** | &#10059; | \274B | Жирная восьмиконечная каплеобразная звёздочка-пропеллер |
| **✲** | &#10034; | \2732 | Звёздочка с незакрашенным центром |
| **☀** | &#9728; | \2600 | Закрашенное солнце с лучами |
| **☁** | &#9729; | \2601 | Облака |
| **☂** | &#9730; | \2602 | Зонтик |
| **☑** | &#9745; | \2611 | Галочка в квадрате |
| **☒** | &#9746; | \2612 | Крестик в квадрате |
| **☹** | &#9785; | \2639 | Нахмуренный смайлик |
| **☺** | &#9786; | \263A | Улыбающийся смайлик |
| **☻** | &#9787; | \263B | Закрашенный улыбающийся смайлик |
| **☭** | &#9773; | \262D | Серп и молот |
| **⚑** | &#9873; | \2691 | Закрашенный флаг |
| **⚐** | &#9872; | \2690 | Незакрашенный флаг |
| **☰** | &#9776; | \2630 | Триграмма |
| **✿** | &#10047; | \273f | Закрашенный цветок |
| **❀** | &#10048; | \2740 | Незакрашенный цветок |
| **✾** | &#10046; | \273E | Цветок с шестью лепестками |
| **❁** | &#10049; | \2741 | Закрашенный обведённый цветок |
| **❂** | &#10050; | \2742 | Цветок из точек |
| **✉** | &#9993; | \2709 | Конверт |
| **❦** | &#10086; | \2766 | Сердце в виде цветка |
| **❶** | &#10102; | \2776 | Номер 1 |
| **❷** | &#10103; | \2777 | Номер 2 |
| **❸** | &#10104; | \2778 | Номер 3 |
| **❹** | &#10105; | \2779 | Номер 4 |
| **❺** | &#10106; | \277A | Номер 5 |
| **❻** | &#10107; | \277B | Номер 6 |
| **❼** | &#10108; | \277C | Номер 7 |
| **❽** | &#10109; | \277D | Номер 8 |
| **➒** | &#10130; | \2792 | Номер 9 |
| **➓** | &#10131; | \2793 | Номер 10 |
| **✖** | &#10006; | \2716 | Жирный знак умножения |
| **✘** | &#10008; | \2718 | Жирный крестик |
| **✔** | &#10004; | \2714 | Жирная отметка галочкой |
| **✚** | &#10010; | \271A | Жирный крест |
| **⚛** | &#9883; | \269B | Символ атома |
| **♺** | &#9850; | \267A | Символ переработки |
| **❑** | &#10065; | \2751 | Незакрашенный квадрат с правой нижней тенью |
| **❒** | &#10066; | \2752 | Незакрашенный квадрат с правой верхней тенью |
| **◈** | &#9672; | \25C8 | Алмаз в оправе |
| **◐** | &#9680; | \25D0 | Круг с левой закрашенной половиной |
| **◑** | &#9681; | \25D1 | Круг с закрашенной правой половиной |
| **⁂** | &#8258; | \2042 | Три звездочки |

### 2. Знаки пунктуации

|  |  |  |  |
| --- | --- | --- | --- |
| **Вид** | **HTML-код** | **CSS-код** | **Описание** |
|  | &nbsp; | \00A0 | Неразрывный пробел |
| **­** | &shy; | \00AD | Место возможного переноса |
| **<** | &lt; | \003C | Знак "меньше чем" (начало тега) |
| **>** | &gt; | \003E | Знак "больше чем" (конец тега) |
| **«** | &laquo; | \00AB | Левая двойная угловая скобка |
| **»** | &raquo; | \00BB | Правая двойная угловая скобка |
| **‹** | &lsaquo; | \2039 | Левая угловая одиночная кавычка |
| **›** | &rsaquo; | \203A | Правая угловая одиночная кавычка |
| **"** | &quot; | \0022 | Двойная кавычка |
| **′** | &prime; | \2032 | Одиночный штрих |
| **″** | &Prime; | \2033 | Двойной штрих |
| **‘** | &lsquo; | \2018 | Левая одиночная кавычка |
| **’** | &rsquo; | \2019 | Правая одиночная кавычка |
| **‚** | &sbquo; | \201A | Нижняя одиночная кавычка |
| **“** | &ldquo; | \201C | Левая двойная кавычка |
| **”** | &rdquo; | \201D | Правая двойная кавычка |
| **„** | &bdquo; | \201E | Нижняя двойная кавычка |
| **❜** | &#10076; | \275C | Жирная одинарная верхняя запятая |
| **❛** | &#10075; | \275B | Жирная одинарная повёрнутая верхняя запятая |
| **&** | &amp; | \0026 | Амперсанд |
| **'** | &apos; | \0027 | Апостроф (одинарная кавычка) |
| **§** | &sect; | \00A7 | Параграф |
| **©** | &copy; | \00A9 | Знак copyright |
| **¬** | &not; | \00AC | Знак отрицания |
| **®** | &reg; | \00AE | Знак зарегистрированной торговой марки |
| **¯** | &macr; | \00AF | Знак долготы над гласным |
| **°** | &deg; | \00B0 | Градус |
| **±** | &plusmn; | \00B1 | Плюс-минус |
| **¹** | &sup1; | \00B9 | Верхний индекс "1" |
| **²** | &sup2; | \00B2 | Верхний индекс "2" |
| **³** | &sup3; | \00B3 | Верхний индекс "3" |
| **¼** | &frac14; | \00BC | Одна четверть |
| **½** | &frac12; | \00BD | Одна вторая |
| **¾** | &frac34; | \00BE | Три четверти |
| **´** | &acute; | \00B4 | Знак ударения |
| **µ** | &micro; | \00B5 | Микро |
| **¶** | &para; | \00B6 | Знак абзаца |
| **·** | &middot; | \00B7 | Знак умножения |
| **¿** | &iquest; | \00BF | Перевернутый вопросительный знак |
| **ƒ** | &fnof; | \0192 | Знак флорина |
| **™** | &trade; | \2122 | Знак торговой марки |
| **•** | &bull; | \2022 | Маркер списка |
| **…** | &hellip; | \2026 | Многоточие |
| **‾** | &oline; | \203E | Надчеркивание |
| **–** | &ndash; | \2013 | Среднее тире |
| **—** | &mdash; | \2014 | Длинное тире |
| **‰** | &permil; | \2030 | Промилле |
| **}** | &#125; | \007D | Правая фигурная скобка |
| **{** | &#123; | \007B | Левая фигурная скобка |
| **=** | &#61; | \003D | Знак равенства |
| **≠** | &ne; | \2260 | Знак неравенства |
| **≅** | &cong; | \2245 | Конгруэнтность (геометрическое равенство) |
| **≈** | &asymp; | \2248 | Почти равно |
| **≤** | &le; | \2264 | Меньше чем или равно |
| **≥** | &ge; | \2265 | Больше чем или равно |
| **∠** | &ang; | \2220 | Угол |
| **⊥** | &perp; | \22A5 | Перпендикулярно (кнопка вверх) |
| **√** | &radic; | \221A | Квадратный корень |
| **∑** | &sum; | \2211 | N-ичное суммирование |
| **∫** | &int; | \222B | Интеграл |
| **※** | &#8251; | \203B | Знак сноски |
| **÷** | &divide; | \00F7 | Знак деления |
| **∞** | &infin; | \221E | Знак бесконечности |
| **@** | &#64; | \0040 | Символ собака |
| **[** | &#91; | \005B | Левая квадратная скобка |
| **]** | &#93; | \005D | Правая квадратная скобка |

### 3. Стрелки

|  |  |  |  |
| --- | --- | --- | --- |
| **Вид** | **HTML-код** | **CSS-код** | **Описание** |
| **←** | &larr; | \2190 | Стрелка влево |
| **↑** | &uarr; | \2191 | Стрелка вверх |
| **→** | &rarr; | \2192 | Стрелка вправо |
| **↓** | &darr; | \2193 | Стрелка вниз |
| **↔** | &harr; | \2194 | Стрелка влево-вправо |
| **↵** | &crarr; | \21B5 | Стрелка вниз и влево – знак возврата каретки |
| **⇐** | &lArr; | \21D0 | Двойная стрелка налево |
| **⇑** | &uArr; | \21D1 | Двойная стрелка вверх |
| **⇒** | &rArr; | \21D2 | Двойная стрелка направо |
| **⇓** | &dArr; | \21D3 | Двойная стрелка вниз |
| **⇔** | &hArr; | \21D4 | Двойная стрелка влево-вправо |
| **➠** | &#10144; | \27A0 | Летящая стрела |
| **➤** | &#10148; | \27A4 | Наконечник стрелы |
| **➥** | &#10149; | \27A5 | Изогнутая стрела, указывающая вниз и вправо |
| **➦** | &#10150; | \27A6 | Изогнутая стрела, указывающая вверх и вправо |
| **➳** | &#10163; | \27B3 | Стрела направо |
| **↺** | &#8634; | \21BA | Круглая стрелка с наконечником против часовой стрелки |
| **↻** | &#8635; | \21BB | Круглая стрелка с наконечником по часовой стрелке |
| **⇧** | &#8679; | \21E7 | Толстая полая стрелка вверх |
| **↩** | &#8617; | \21A9 | Стрелка налево с крючком |
| **➫** | &#10155; | \27AB | Наклонённая вниз объёмная стрелка |
| **⬇** | &#11015; | \2B07 | Закрашенная стрелка вниз |
| **⬆** | &#11014; | \2B06 | Закрашенная стрелка вверх |

### 4. Карточные масти

|  |  |  |  |
| --- | --- | --- | --- |
| **Вид** | **HTML-код** | **CSS-код** | **Описание** |
| **♠** | &spades; | \2660 | "Пики" |
| **♣** | &clubs; | \2663 | "Трефы" |
| **♥** | &hearts; | \2665 | "Червы" |
| **♦** | &diams; | \2666 | "Бубны" |
| **♡** | &#9825; | \2661 | Контур "Червы" |
| **♢** | &#9826; | \2662 | Контур "Бубны" |
| **♤** | &#9828; | \2664 | Контур "Пики" |
| **♧** | &#9831; | \2667 | Контур "Трефы" |

### 5. Деньги

|  |  |  |  |
| --- | --- | --- | --- |
| **Вид** | **HTML-код** | **CSS-код** | **Описание** |
| **¢** | &cent; | \FFE0 | Цент |
| **£** | &pound; | \FFE1 | Фунт стерлингов |
| **₽** | &#8381; | \20BD | Российский рубль |
| **¥** | &yen; | \00A5 | Йена или юань |
| **€** | &euro; | \20AC | Евро |
| **$** | &#36; | \0024 | Доллар |
| **₴** | &#8372; | \20B4 | Знак гривны |
| **₹** | &#8377; | \20B9 | Индийская рупия |
| **圓** | &#22291; | \5713 | Китайский юань |
| **₸** | &#8376; | \20B8 | Казахстанский тенге |

### 6. Знаки зодиака

|  |  |  |  |
| --- | --- | --- | --- |
| **Вид** | **HTML-код** | **CSS-код** | **Описание** |
| **♈** | &#9800; | \2648 | Овен |
| **♉** | &#9801; | \2649 | Телец |
| **♊** | &#9802; | \264A | Близнецы |
| **♋** | &#9803; | \264B | Рак |
| **♌** | &#9804; | \264C | Лев |
| **♍** | &#9805; | \264D | Дева |
| **♎** | &#9806; | \264E | Весы |
| **♏** | &#9807; | \264F | Скорпион |
| **♐** | &#9808; | \2650 | Стрелец |
| **♑** | &#9809; | \2651 | Козерог |
| **♒** | &#9810; | \2652 | Водолей |
| **♓** | &#9811; | \2653 | Рыбы |

# 1.10. HTML-генераторы

**HTML-генераторы** и полезные сервисы облегчат работу с HTML-кодом, протестируют отображение вашего сайта в разных браузерах, а также дадут информацию о поддержке браузерами тех или иных тегов и технологий.

## Подборка HTML-генераторов

**Image-Maps** — сервис для создания карт-изображений.

**Mobirise** — оффлайн приложение для Windows и Mac, с помощью которого можно создавать небольшие веб-сайты, лендинги, онлайн-резюме и портфолио, промо-сайты для приложений, событий, продуктов и услуг.

**FREE Banner Maker** — сервис для HTML5-баннеров.

**HTML minifier** — сервис для минимизации HTML-кода.

**Mobile HTML5** таблица совместимости HTML5 API на мобильных и планшетных браузерах с тестированием на реальных устройствах.

**Initializr** — генератор HTML5-шаблонов, которые помогут начать работу над вашим новым проектом. Он построен на базе HTML5 Boilerplate, мощнейшем HTML5-шаблоне, созданном Paul Irish и Divya Manian. С его помощью вы элементарно создадите HTML5-проект с необходимыми настройками по вашему выбору. По сути, это облегченная версия Boilerplate. Выбираемыми параметрами являются: структура, серверные опции (.htaccess, nginx.conf, web.config), подключение jquery, аналитики google, выбор между css и less и тд. Как только вы настроите конфигурацию шаблона, нажмите Download и вы получите набор файлов, согласно сделанному выбору.

**Browsershots** делает скриншоты дизайна вашего сайта в разных операционных системах и браузерах. Проверка может длиться несколько часов, в зависимости от количества файлов на сайте.

Сервис **Can I use…** тестирует поддержку основных возможностей HTML5 для всех версий всех браузеров. Он позволяет выделить именно те возможности, которые требуются вам. Для работы с сервисом нужно выбрать вкладку **Tables**, затем вкладку **Compatibility tables**, после чего выбрать требуемую возможность (возможности), установив флажок. Также можно производить поиск через поле **Search**.

Сервис **HTML5Test** проверяет, насколько тот или иной браузер готов работать со стандартом HTML5.

Сервис **HTML5 Please** содержит информацию о поддержке браузерами всех новых API стандарта HTML5.

**FormLinter** найдёт на сайте формы и проверит, есть ли у них проблемы с доступностью.

Полезные ссылки

* 1. **Веб-разработка. Frontend** https://stepik.org/course/100971/syllabus

# **Основы современной верстки** [**https://ru.hexlet.io/courses/layout-designer-basics**](https://ru.hexlet.io/courses/layout-designer-basics)

* 1. HTML <https://html5book.ru/html-html5/>
  2. [http://htmlbook.ru](https://vk.com/away.php?to=http%3A%2F%2Fhtmlbook.ru&post=11193554_493&cc_key=)
  3. <https://htmlacademy.ru>
  4. https://code-basics.com/ru/languages/html