

Кириченко А.В., Дубовик Е.В.

# Справочник HTML

КРАТКО # БЫСТРО # ПОД РУКОЙ



HTML5-теги \* Атрибуты  
Примеры использования \* Семантическая верстка  
Наглядные иллюстрации

Rescuer

**СЕРИЯ — СПРАВОЧНИК — СЕРИЯ**



---

**"Наука и Техника"**  
**г. Санкт-Петербург**



**HTML**

**КИРИЧЕНКО А.В., ДУБОВИК Е.В.**

# **Справочник** **HTML**

**КРАТКО # БЫСТРО # ПОД РУКОЙ**



---

**"Наука и Техника"**

**г. Санкт-Петербург**



УДК 004.42 ББК 32.973

ISBN 978-5-94387-275-4

Кириченко А.В., Дубовик Е.В.

**Справочник HTML. Кратко, быстро, под рукой** — СПб.: Наука и Техника, 2021. — 288 с., ил.

### Серия "Справочник"

---

Данный справочник содержит полную информацию об HTML5 в удобной и наглядной форме. Структура справочника позволяет быстро и удобно находить нужную информацию, получать примеры использования тех или иных тегов HTML.

Отдельное внимание уделено отраслевым рекомендациям по хорошему стилю использования HTML для верстки веб-сайтов, приведено описание приемов семантической верстки.

Справочник будет полезен всем, кто использует или изучает HTML: от начинающих до профессионалов.

Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Издательство не несет ответственности за возможный ущерб, причиненный в ходе использования материалов данной книги, а также за доступность материалов, ссылки на которые вы можете найти в этой книге. На момент подготовки книги к изданию все ссылки на интернет-ресурсы были действующими.

ISBN 978-5-94387-275-4



9 78- 5- 94387- 275- 4

Контактные телефоны издательства:  
(812) 412 70 26

Официальный сайт: [www.nit.com.ru](http://www.nit.com.ru)

© Кириченко А.В., Дубовик Е.В.

© Наука и Техника (оригинал-макет)

# Содержание

<b>ЧАСТЬ 1. ОБЩАЯ СТРУКТУРА HTML-ДОКУМЕНТА.....</b>	<b>15</b>
<b>1.1. ЧТО ТАКОЕ HTML? .....</b>	<b>16</b>
<b>1.2. HTML-ДОКУМЕНТ, ТЕГИ .....</b>	<b>18</b>
Введение в теги .....	18
<b>1.3. СТРУКТУРА HTML-ДОКУМЕНТА.....</b>	<b>23</b>
Атрибуты id и class .....	25
<b>1.4. ГЛОБАЛЬНЫЕ АТТРИБУТЫ HTML .....</b>	<b>25</b>
Другие глобальные атрибуты .....	28
<b>1.5. СОБЫТИЯ.....</b>	<b>32</b>
<b>ЧАСТЬ 2. СПРАВОЧНИК ТЕГОВ HTML5 .....</b>	<b>34</b>
<b>ТЕГ A .....</b>	<b>35</b>
<b>ТЕГ ABBR.....</b>	<b>38</b>
<b>ТЕГ ADDRESS .....</b>	<b>39</b>
<b>ТЕГ AREA .....</b>	<b>39</b>
<b>ТЕГ ARTICLE.....</b>	<b>41</b>
<b>ТЕГ ASIDE .....</b>	<b>42</b>
<b>ТЕГ AUDIO.....</b>	<b>44</b>
<b>ТЕГ B .....</b>	<b>46</b>

<b>TEG BASE .....</b>	<b>47</b>
<b>TEG BDI .....</b>	<b>49</b>
<b>TEG BDO .....</b>	<b>50</b>
<b>TEG BLOCKQUOTE.....</b>	<b>51</b>
<b>TEG BODY.....</b>	<b>52</b>
<b>TEG BR .....</b>	<b>52</b>
<b>TEG BUTTON .....</b>	<b>54</b>
<b>TEG CANVAS .....</b>	<b>59</b>
<b>TEG CAPTION .....</b>	<b>61</b>
<b>TEG CENTER .....</b>	<b>62</b>
<b>TEG CITE .....</b>	<b>63</b>
<b>TEG CODE .....</b>	<b>64</b>
<b>TEG COL .....</b>	<b>65</b>
<b>TEG COLGROUP .....</b>	<b>66</b>
<b>TEG COMMAND .....</b>	<b>68</b>
<b>TEG DATA .....</b>	<b>69</b>
<b>TEG DATALIST .....</b>	<b>70</b>
<b>TEG DD .....</b>	<b>71</b>
<b>TEG DEL .....</b>	<b>72</b>
<b>TEG DETAILS .....</b>	<b>73</b>
<b>TEG DFN .....</b>	<b>74</b>

<b>ТЕГ DIALOG .....</b>	<b>75</b>
<b>ТЕГ DIV .....</b>	<b>75</b>
<b>ТЕГ DL .....</b>	<b>79</b>
<b>ТЕГ DT .....</b>	<b>81</b>
<b>ТЕГ EM .....</b>	<b>81</b>
<b>ТЕГ EMBED .....</b>	<b>82</b>
<b>ТЕГ FIELDSET .....</b>	<b>83</b>
<b>ТЕГ FIGCAPTION .....</b>	<b>85</b>
<b>ТЕГ FIGURE .....</b>	<b>86</b>
<b>ТЕГ FOOTER .....</b>	<b>86</b>
<b>ТЕГ FORM .....</b>	<b>88</b>
<b>ТЕГИ H1,H2,H3,H4, H5, H6 .....</b>	<b>90</b>
<b>ТЕГ HEAD .....</b>	<b>92</b>
<b>ТЕГ HEADER .....</b>	<b>94</b>
<b>ТЕГ HGROUP .....</b>	<b>95</b>
<b>ТЕГ HR .....</b>	<b>95</b>
<b>ТЕГ HTML .....</b>	<b>97</b>
<b>ТЕГ I .....</b>	<b>98</b>
<b>ТЕГ IFRAME .....</b>	<b>98</b>
<b>ТЕГ IMG .....</b>	<b>101</b>
<b>ТЕГ INPUT .....</b>	<b>105</b>

<b>ТЕГ INS .....</b>	<b>127</b>
<b>ТЕГ KBD .....</b>	<b>128</b>
<b>ТЕГ KEYGEN .....</b>	<b>129</b>
<b>ТЕГ LABEL.....</b>	<b>130</b>
<b>ТЕГ LEGEND .....</b>	<b>131</b>
<b>ТЕГ LI .....</b>	<b>132</b>
<b>ТЕГ LINK .....</b>	<b>134</b>
<b>ТЕГ MAIN .....</b>	<b>135</b>
<b>ТЕГ MAP .....</b>	<b>136</b>
<b>ТЕГ MARK .....</b>	<b>138</b>
<b>ТЕГ MENU .....</b>	<b>138</b>
<b>ТЕГ MENUITEM .....</b>	<b>140</b>
<b>ТЕГ META.....</b>	<b>141</b>
<b>ТЕГ METER.....</b>	<b>145</b>
<b>ТЕГ &lt;NAV&gt;.....</b>	<b>146</b>
<b>ТЕГ NOSCRIPT .....</b>	<b>147</b>
<b>ТЕГ OBJECT .....</b>	<b>148</b>
<b>ТЕГ OL .....</b>	<b>150</b>
<b>ТЕГ OPTGROUP .....</b>	<b>153</b>
<b>ТЕГ OPTION.....</b>	<b>154</b>
<b>ТЕГ OUTPUT .....</b>	<b>156</b>

TEГ P .....	157
TEГ PARAM.....	158
TEГ PRE .....	159
TEГ PROGRESS .....	160
TEГ Q .....	162
TEГ RP .....	163
TEГ RT .....	164
TEГ RUBY .....	166
TEГ S .....	166
TEГ SAMP .....	167
TEГ SCRIPT .....	168
TEГ SECTION.....	169
TEГ SELECT .....	170
TEГ SMALL .....	173
TEГ SOURCE.....	174
TEГ SPAN .....	175
TEГ STRONG.....	177
TEГ STYLE.....	178
TEГ SUB .....	180
TEГ SUMMARY .....	180
TEГ SUP .....	181

<b>TEГ TABLE.....</b>	<b>182</b>
<b>TEГ TBODY .....</b>	<b>185</b>
<b>TEГ TD .....</b>	<b>186</b>
<b>TEГ TEXTAREA .....</b>	<b>192</b>
<b>TEГ TFOOT .....</b>	<b>195</b>
<b>TEГ TH .....</b>	<b>196</b>
<b>TEГ THEAD .....</b>	<b>198</b>
<b>TEГ TIME .....</b>	<b>198</b>
<b>TEГ TITLE.....</b>	<b>199</b>
<b>TEГ TR .....</b>	<b>201</b>
<b>TEГ TRACK .....</b>	<b>201</b>
<b>TEГ U .....</b>	<b>203</b>
<b>TEГ UL .....</b>	<b>204</b>
<b>TEГ VAR .....</b>	<b>206</b>
<b>TEГ VIDEO.....</b>	<b>207</b>
<b>TEГ WBR.....</b>	<b>209</b>
<b>TEГ XMP .....</b>	<b>210</b>

<b>ЧАСТЬ 3. СЕМАНТИЧЕСКАЯ ВЕРСТКА НА HTML .....</b>	<b>213</b>
<b>3.1. БАЗОВЫЕ АБСТРАКЦИИ РАЗМЕТКИ. ВЕРСТКА НА ОСНОВЕ БЛОКОВ DIV.....</b>	<b>214</b>
<b>3.2. СЕМАНТИЧЕСКИЕ ТЕГИ HTML5. ВЕРСТКА НА ОСНОВЕ СЕМАНТИЧЕСКОЙ РАЗМЕТКИ.....</b>	<b>216</b>
<b>3.3. ПРИМЕР СЕМАТИЧЕСКОЙ ВЕРСТКИ .....</b>	<b>222</b>
3.3.1. Блок HEAD .....	224
3.3.2. Блок HEADER .....	227
3.3.3. Блок SECTION – слайдер.....	231
3.3.4. Блок SECTION – Статьи и новости .....	233
3.3.5. Блок SECTION – видео .....	235
3.3.6. Блок SECTION – Архив журнала.....	237
3.3.7. Блок SECTION – подписка .....	238
3.3.8. Блок FOOTER .....	239
<b>ПРИЛОЖЕНИЕ 1. СТАНДАРТНЫЕ ЦВЕТА .....</b>	<b>244</b>
<b>ПРИЛОЖЕНИЕ 2. КОДЫ ЯЗЫКОВ.....</b>	<b>267</b>
<b>ПРИЛОЖЕНИЕ 3. МІМЕ-ТИПЫ .....</b>	<b>273</b>

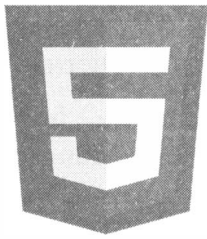




**HTML**

Данная книга для удобства поделена на 4 части:

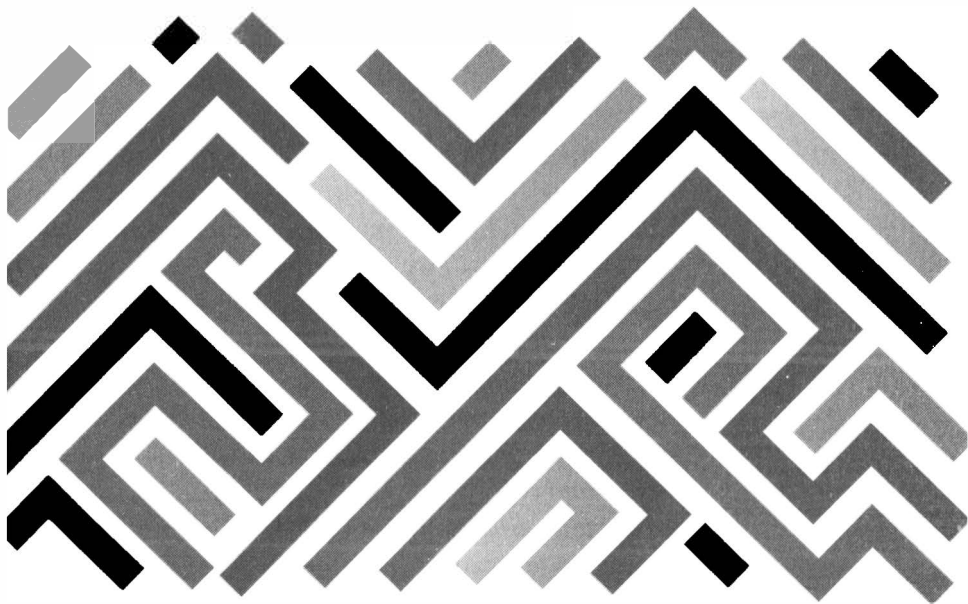
- В **первой** части рассмотрена общая структура HTML-документа и даны основные понятия: HTML-элементы, теги, атрибуты, события;
- Во **второй** части в алфавитном порядке идет описание-справочник всех тегов HTML5;
- **Третья** часть расскажет о приемах семантической верстки - базовые абстракции разметки, верстка на основе блоков div, верстка на основе семантической разметки, приведен пример семантической верстки главной страницы интернет-портала;
- **Четвертая** часть состоит из нескольких полезных приложений: RGB и HTML-значения для стандартного цвета; коды языков, которые можно использовать в теге lang; таблица соответствия расширений файлов MIME-типов.



**HTML**

# Часть 1.

## Общая структура HTML-документа



## 1.1. Что такое HTML?

Сеть Интернет (World Wide Web) – сеть глобального пользования, и поэтому вся информация в сети должна быть представлена на универсальном языке, который понимали бы все пользователи. Языком публикации, используемым в World Wide Web, является HTML (Hyper Text Markup Language – язык разметки гипертекста). Техническую информацию об этом языке (его спецификацию) на английском языке можно найти на сайте World Wide Web Consortium – [w3c.com](http://w3c.com).

Все интернет-сайты и интернет-страницы по своей сути представляют собой HTML-документы или наборы html-документов. В простейшем случае, html-документ хранится на сервере, и при обращении к нему по адресу из браузера он загружается и отображается в окне браузера. В то же время существуют языки программирования и скрипты, написанные на них, которые позволяют не хранить html-страницы на сервере, а динамически формировать эти страницы исходя из различных данных и действий пользователя. Такой возможностью обладает язык PHP, рассмотрение которого выходит за рамки этой книги. Кроме того, существуют языки программирования, на которых пишутся скрипты, выполняющие различные функции, встраиваемые

в HTML-документ. Таким языком является язык JavaScript.

В итоге html-документ представляет собой некий каркас, который определяет, что есть в документе и в виде каких структурных элементов в него включено. При этом содержимое этих структурных элементов может содержаться непосредственно в самом html-документе (например, текст), а может подгружаться из какого либо источника, например, из файла изображения, если в html-документе указано, что здесь должна быть картинка. Язык HTML позволяет задавать только структуру и не позволяет программировать алгоритмы, поэтому HTML является языком разметки.

Итак, HTML – язык разметки, предоставляющий разработчикам следующие возможности:

- Представлять информацию в сети в виде электронных документов, с информационным содержимым в виде форматированного текста, таблиц, списков, фотографий;
- Включать в документы звуковые фрагменты, видеоклипы, электронные таблицы и другие приложения и элементы мультимедиа;
- Осуществлять загрузку документов посредством активизации гипертекстовой ссылки;

- Разрабатывать формы для осуществления взаимодействия с удаленными службами (поисковыми роботами, on-line магазинами и т.п.)

До 1995 года не было каких-либо стандартов html-документа. Отсутствие стандартизации порождало проблему совместимости, позволяя разработчикам браузеров использовать собственные теги, о которых ничего не "знали" браузеры конкурентов. Нумерация стандартов начинается с 2.0 – видимо потому, что до этого стандарт отсутствовал, а сам HTML существовал (начиная с 1991 года).

Современным считается стандарт HTML 5, принятый 28 октября 2014 года. После вышли стандарты 5.1, 5.2 и 5.3 – с 2016 по 2018 гг.

## 1.2. HTML-документ, теги

### Введение в теги

Разметка документов заключается в том, что документ представляется в виде последовательности элементов. Например, чтобы отобразить в окне браузера простейшую текстовую информацию



**Рис. 1. Простейший текстовый HTML-документ**

HTML-документ должен иметь следующий вид:

```
<!DOCTYPE html>
<HTML>
  <HEAD>
    <TITLE>Мой первый html-документ</TITLE>
  </HEAD>
  <BODY>
    <H2 align="center">Я знаю HTML</H2>
    <P>
      Привет мир!
    </P>
  </BODY>
</HTML>
```

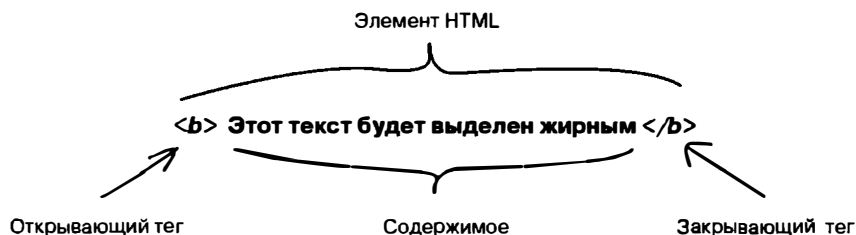


Для создания html-документа достаточно набрать его текст в каком-либо простом текстовом редакторе и сохранить его в файле с расширением .html (например, example1.html). Чтобы открыть его – щелкните по нему дважды мышкой, и содержимое html-файла откроется в браузере.

Основная структурная единица, с которой происходит работа в документе HTML – это HTML-элемент, который на профессиональном жаргоне принято называть **тегом**. Далее в книге мы повсеместно будем использовать слово **тег**, сначала еще указывая, что еще речь идет именно об html-элементе. Если посмотреть исходные файлы верстки сайтов в сети Интернет, электронных писем, многих печатных форм документов (например, используемые интернет-магазинами, в которых они выставляют покупателю счет), то все они представляют собой набор HTML тегов.

Как и языки программирования, язык разметки HTML постоянно развивается, публикуются новые стандарты, появляются новые имеющие свои семантические особенности теги.

Каждое объявление тега (HTML-элемента) обычно включает три части: **начальный** (открывающий) **тег**, **содержимое** и **конечный** (закрывающий) **тег**. Имя тега (HTML-элемента) отображается в начальном теге ( <имя элемента> ) и конечном теге ( </имя элемента> ). Пример:



Начальный тег может содержать определенные атрибуты, дополняющие его функционал. Для тега `a` основным атрибутом является `href`, указывающий адрес документа, на который указывает ссылка. Именно этот документ будет загружен в браузер (или скачан, если браузер не может отобразить его), если пользователь щелкнет по ссылке. Далее мы подробно поговорим о них.

Теги служат для разметки и на экране при просмотре интернет-страницы в браузере не отображаются.

В HTML имеют место теги (элементы), которые позволяют авторам опускать открывающие (например, элементы `HEAD` и `BODY`) и закрывающие (например, элементы `P` и `Li`) теги. В некоторых случаях допустимо отсутствие обоих тегов.

Существуют HTML-элементы, не имеющие содержимого. Например, элемент перехода на следующую строку `BR` его не имеет, т.к. его функция заключается в прерывании строки текста и переходе на следующую. Элементы такого типа никогда не имеют конечных тегов.

Все теги языка HTML выделяются символами-ограничителями ( < и > ), между которыми прописывается имя тега и, возможно, его атрибуты.

Единственным исключением являются теги комментария. Для них предусмотрено следующее написание:

```
<!--Текст комментария -->
```

Имена тегов и их атрибутов не чувствительны к регистру, т.е. не имеет значения заглавными или прописными буквами они написаны. Например, записи <bOdY> и <BoDu> абсолютно идентичны с точки зрения HTML.

Атрибуты – это свойства тега (элемента), которые имеют либо стандартное значение (используемое по умолчанию), либо значение, задаваемое разработчиком или сценарием (скриптом).

Набор атрибутов индивидуален для каждого тега (элемента), хотя некоторые из них могут повторяться у разных тегов (элементов).

Пары "атрибут=значение" помещаются перед закрывающей скобкой начального тега элемента, и если их несколько разделяются пробелами. Порядок их перечисления произволен. Например:

```
<BODY id = "pap">  
.....содержимое .....  
.....содержимое .....  
.....содержимое .....  
</BODY>
```

Некоторые атрибуты не имеют значения, тогда достаточно просто указания имени атрибута.

По умолчанию в HTML необходимо, чтобы все значения атрибутов были заключены в двойные (код ASCII 34) или одинарные (код ASCII 39) кавычки. В некоторых случаях допустимо этого не делать, но рекомендуется кавычки применять всегда.

Значение атрибута может включать только буквы (a – z, A – Z), цифры, знаки переноса (код ASCII 45) и точки (код ASCII 46). Включение одинарных кавычек в значение атрибута допустимо, если оно заключено в двойные кавычки, и наоборот.

При интерпретации тегов все браузеры придерживаются того правила, что при обработке неправильно заданного HTML-элемента или при использовании в документе элемента разметки не поддерживаемого данным браузером, он игнорируется.

## 1.3. Структура HTML-документа

Можно выделить общую структуру для файлов в формате HTML. Документ в формате HTML состоит из трех основных частей:

- строки, объявляющей файл как документ на языке HTML5;

- заголовка, заключенного в тег HEAD;
- тела документа, представляющего собой тег BODY, если документ имеет классическое, однооконное представление. В теле документа, собственно, и содержится вся, предлагаемая пользователю, информация.

Заголовок и тело документа заключены в тег-контейнер HTML. Все остальные теги разметки располагаются либо в заголовке, либо теле документа.

Простейший пример HTML-документа представлен ниже:

```
<!DOCTYPE html>
<html>
  <head>
    <title> Название документа </title>
  </head>
  <body>
    Здесь располагается текст страницы
  </body>
</html>
```

Тег `<!DOCTYPE html>` позволяет браузеру, в котором открывается данная страница, определить, что для верстки был использован стандарт языка HTML версии 5.

На данный момент у вас есть понимание, что такое HTML-документ. Далее, в разделе Справочник HTML-документов, мы подробно рассмотрим каждый тег HTML-документа.

## 1.4. Глобальные атрибуты HTML

При рассмотрении различных HTML-тегов мы изучаем атрибуты, свойственные тому или иному тегу. Однако есть набор атрибутов, называемых глобальными атрибутами, которые применимы абсолютно ко всем тегам HTML. То есть для любого тега могут быть заданы любые глобальные атрибуты.

### Атрибуты `id` и `class`

Важным правилом верстки сайтов является разделение структурной разметки и стилей оформления. То есть, в html есть возможность для текста заголовка H1 (тег H1) задать его выравнивание непосредственно в теге H1, но правильно будет создать CSS-правило для тега H1, которое будет располагаться отдельно от тега H1, и применяться ко всем тегам H1 в документе. Благодаря такой организации вам, во-первых, не надо будет искать, где именно в тексте html-документа надо подправить выравнивание в случае необходимости (все стилевые настройки располагаются обычно в отдельном файле или в заголовке html-файла), а во-вторых, вам

не надо будет каждый раз задавать для каждого тега H1 значение выравнивания текста – одно CSS-правило для тега H1 будет применено ко всем тегам H1 в html-документе. Соответственно, и в случае изменения выравнивания текста в CSS-правиле для тега H1 изменения будут применены ко всем тегам H1. Такой стиль верстки настолько считается правильным, что в версии HTML5 у тега H1 (и других) были исключены атрибуты задания оформления: выравнивания и т.п. Подразумевается, что все это будет делаться с помощью таблиц стилей CSS. Однако такие атрибуты все еще поддерживаются браузерами, так как они активно применялись в предыдущей версии HTML 4.01, и есть сайты, созданные в рамках предыдущего стандарта. Но мы с вами учимся создавать сайты согласно последним стандартам, а значит, будем отделять определение структурных элементов – тегов HTML, и их оформление – CSS-стили.

Когда вы для каждого тега задаете его внешний вид (то же выравнивание для заголовка H1), то вы можете для разных случаев применения одного и того же тега в HTML-документе задавать свои параметры (для одного заголовка H1 задать одно выравнивание, а для другого заголовка H1 на этой же html-странице – другое выравнивание).

Когда же вы создали одно стилевое CSS-правило для тега, то оно будет применяться ко всем включениям тега на html-странице. Как в таком слу-

чае для разных заголовков H1 задать разный вид в разных частях html-страницы?

Чтобы у верстальщика была возможность обращаться (и писать CSS-правила) не ко всем экземплярам тега на html-странице, а какому либо конкретному экземпляру, в HTML предусмотрено для каждого тега задавать его уникальное имя – атрибут `id`.

Также имеется возможность объединять несколько тегов под одним именем – задать для нескольких экземпляров одного тега одинаковый атрибут `class` (то есть объединить их в класс). И тогда, при обращении указывается тег и `id` конкретного тега (или `class` – при классификации экземпляров одного тега).

Итак, атрибуты `id` и `class` служат для идентификации содержащих их тегов. Через атрибут `id` тегу можно присвоить уникальное имя. Атрибут `class` причисляет тег к классу тегов, созданному разработчиком. Причем сам класс тегов образуется с первым, включенным в него тегом. Если тег причисляется к нескольким классам сразу, то тогда имена классов должны быть разделены пробелами.

Атрибут `id` служит в HTML для выполнения следующих функций:

- для осуществления выборочного обращения таблицы стилей к определенным элементам (задания их стиля);



- для указания цели (якоря) гипертекстовых ссылок, что позволяет им ссылаться не только на документ в целом, но и на его отдельные части;
- для реализации ссылки на определенный элемент сценария;
- для задания имени объекта, вставляемого в документ тегом *object*.

Атрибут `class` служит в HTML для выполнения следующих функций:

- для осуществления выборочного обращения таблицы стилей к определенной группе элементов (задания их стиля);
- для реализации особой обработки браузером определенной (заданной разработчиком) группы элементов.

Атрибуты `id` и `class` могут быть установлены почти для всех тегов языка HTML. Именованное HTML-тегов и объединение их в определенные группы позволяет осуществлять обращение к ним, что особенно полезно при применении к ним таблиц стилей.

## Другие глобальные атрибуты

С двумя основными глобальными атрибутами мы уже с вами знакомы – это атрибуты *id* и *class*. Осталось познакомиться с остальными.

Глобальные атрибуты HTML применяются практически ко всем тегам, поэтому о них мы поговорим отдельно, чтобы не повторять их для всех тегов:

- **accesskey** — позволяет получить доступ к элементу с помощью заданного сочетания клавиш. Допускается возможность указания нескольких символов. Содержимое данного атрибута — список разделенных пробелами символов. Браузер будет использовать первый, имеющийся в раскладке клавиатуры, символ из списка.
- **contenteditable** — сообщает, что элемент доступен для редактирования пользователем. Поддерживаются следующие значения:
  - > **true** — редактирование разрешено;
  - > **false** — редактирование не разрешено.
- **contextmenu** — устанавливает контекстное меню для элемента. Использование тега MENU на данный момент не стандартизировано разработчиками браузеров, а потому не рекомендуется
- **dir** — задает направление и отображение текста — слева направо или справа налево. Поле dir может принимать одно из следующих значений:
  - > **ltr**, что расшифровывается как left to right — слева направо;

- > ***rtl***, что расшифровывается как right to left – справа налево;
  - > ***auto*** — направление будет выбрано автоматически.
- **draggable** — позволяет определить, можно ли будет перетаскивать содержимое соответствующего тега или нет. Принимает следующие значения:
  - > ***true*** — можно перетаскивать;
  - > ***false*** — нельзя перетаскивать.
- **dropzone** — данный атрибут позволяет задать действия, которые выполняются после перетаскивания данных на содержимое соответствующего тега, а также задается допустимый тип перетаскиваемых данных. Поддерживаемые значения:
  - > ***copy*** — перетаскивание создаст копию перетаскиваемого элемента;
  - > ***move*** — перетаскиваемый элемент будет перемещен в новое расположение;
  - > ***link*** — перетаскивание создаст ссылку на перетаскиваемые данные.
  - > ***file: тип файла*** — задается допустимый тип файла (например, file: image/png).
- **hidden** — скрывает содержимое элемента от просмотра.
- **lang** — браузер использует значение параметра для правильного отображения некоторых национальных символов.

- **spellcheck** — указывает браузеру проверять или нет правописание и грамматику в тексте.
- **style** — применяется для определения стиля элемента с помощью правил CSS.
- **tabindex** — устанавливает порядок получения фокуса при переходе между элементами с помощью клавиши **Tab**. Атрибут принимает значения от 0 до 32767, определяющее порядковое место соответствующего тега в последовательности перехода между тегами html-страницы при последовательном нажатии клавиши "Tab"
- **title** — описывает содержимое элемента в виде всплывающей подсказки.
- **translate** — задает, можно ли переводить содержимое тега при автоматическом переводе всей страницы или нет. Может принимать значения:
  - > пустая строка или **"yes"** указывает, что содержимое тега должно быть переведено;
  - > **"no"** указывает, что содержимое тега не должно быть переведено.
- **xml:lang** — этот атрибут по своему действию похож на **lang**, но применяется только в XHTML-документах и указывает язык всего текста или его фрагмента.

## 1.5. События

Для большинства элементов HTML можно задать обработчики события. Например:

```
<button onclick="Check()">Проверка</button>
```

При нажатии кнопки произойдет событие *onclick*. В качестве обработчика этого события была назначена функция `Check()`, которая и будет вызвана.

Список событий и когда они происходят:

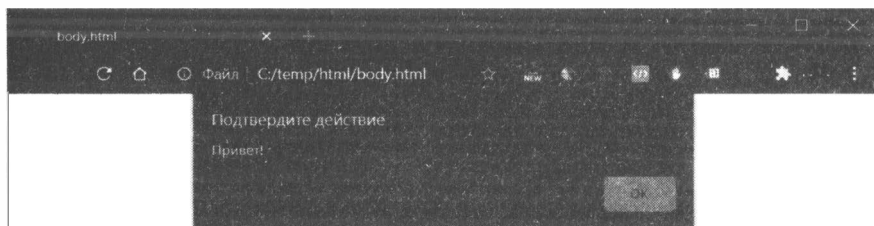
- ***onblur*** — потеря фокуса.
- ***onchange*** — изменение значения элемента формы.
- ***onclick*** — щелчок левой кнопкой мыши на элементе.
- ***ondblclick*** — двойной щелчок левой кнопкой мыши на элементе.
- ***onfocus*** — получение фокуса.
- ***onkeydown*** — клавиша нажата, но не отпущена.
- ***onkeypress*** — клавиша нажата и отпущена.
- ***onkeyup*** — клавиша отпущена.
- ***onload*** — документ загружен.
- ***onmousedown*** — нажата левая кнопка мыши.

- ***onmousemove*** — перемещение курсора мыши.
- ***onmouseout*** — курсор покидает элемент.
- ***onmouseover*** — курсор наводится на элемент.
- ***onmouseup*** — левая кнопка мыши отпущена.
- ***onreset*** — форма очищена.
- ***onselect*** — выделен текст в поле формы.
- ***onsubmit*** — форма отправлена.
- ***onunload*** — закрытие окна.

Еще пример:

```
<body onload="window.alert('Привет!')">
```

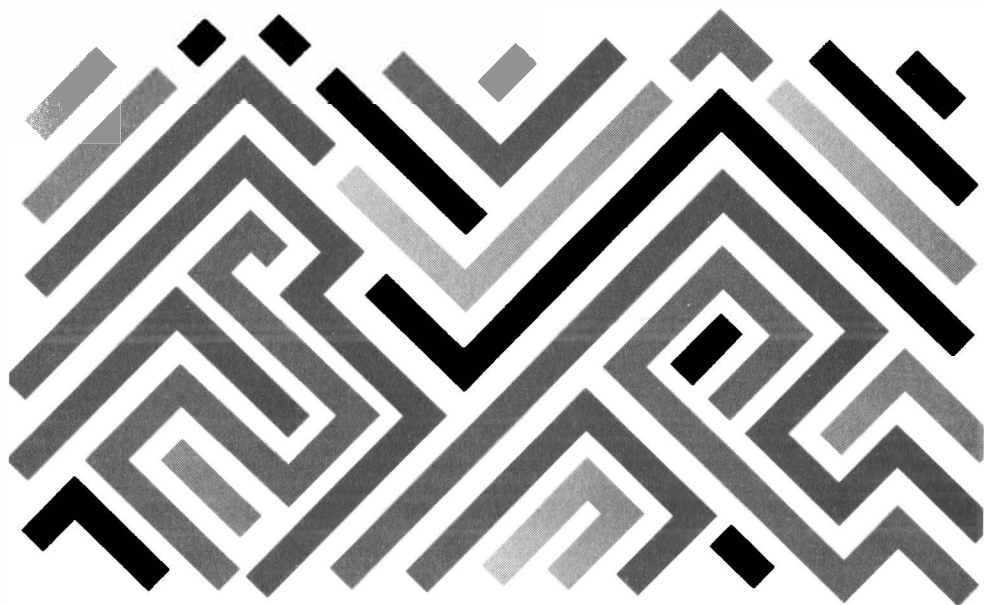
При загрузке страницы пользователь увидит уведомление "Привет!".



**Рис. 2. Пример установки обработчика события *onload***

## Часть 2.

# Справочник тегов HTML5



Данный раздел содержит информацию обо всех HTML-элементах с примерами и комментариями. Для каждого элемента приводится список доступных атрибутов и различные комментарии вроде его совместимости с тем или иным стандартом HTML.

## Тег A

Синтаксис:

```
<a>...</a>
```

Данный тег чаще всего используется для создания ссылок. Ссылки могут быть, как на другой документ, так и на определенную часть текущего документа, в этом случае говорят о якорях (anchors). Изначально данный тег и был предусмотрен для формирования якорей – ссылок на части текущего документа – отсюда и название этого тега <a> (anchor). Если атрибут href данного тега содержит адрес другого документа, то в качестве якоря будет использоваться этот документ.

Контентом тега (часть между открывающим и закрывающим тегами) обычно выступает текст или картинка. Но в HTML 5 допускается использование в теге а потокового контента (блочных элементов). Однако, не допускается включение в состав элемента а интерактивного контента, то есть внутри этого тега нельзя использовать сле-

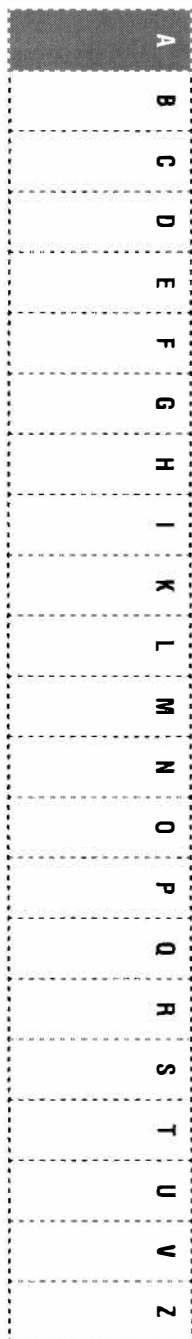
A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z



дующие теги: a, audio с элементами управления, button, details, embed, iframe, img с usemap, input, keygen, label, object с usemap, select, textarea и video).

Атрибуты:

- **download** = "имя\_файла" [только в HTML5.1] – указывает на использование ссылки для загружаемого ресурса. Позволяет указать имя файла, которое должно быть присвоено в локальной файловой системе сразу после загрузки файла. Позволяет переименовать файл на локальном компьютере пользователя, когда он загрузит его.
- **href** = "URI" – позволяет указать местоположение целевого документа или веб-ресурса. Можно создать ссылку, например, другой HTML-документ, на картинку, звуковой файл, PDF-документ, архив и т.д.
- **hreflang** = "код языка" – указывает основной язык целевого документа.
- **media** = "all | aural | braille | handheld | print | projection | screen | tty | tv" [HTML5] – указывает носитель целевого документа
- **rel** = "ключевое слово типа ссылки" – описывает связи между исходным и связанным документами. Для элемента a применяются следующие типы ссылочных связей – alternate, author, bookmark, help,



license, next, nofollow, noreferrer, prefetch, prev, search и tag.

- **target** = "текст" – содержит имя окна или области iframe, где будет отображаться целевой документ. Использование фреймов в современных документах не рекомендуется.
- **type** = "тип MIME" – указывает тип контента (тип MIME), например, text/html.
- Глобальные атрибуты HTML.

Примеры ссылок:

*Ссылка на локальный файл:*

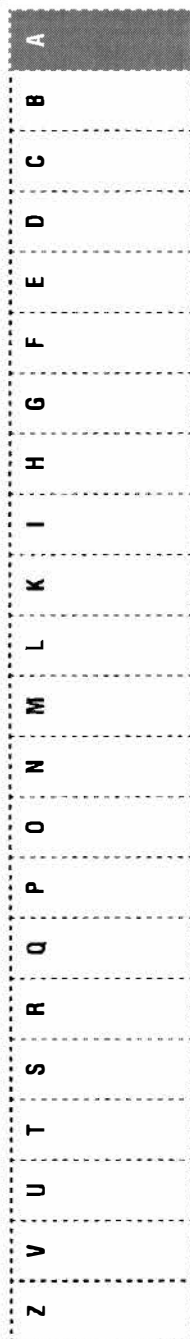
```
<a href="имя_файла.html">Документ будет
загружен в окно браузера</a>
<a href="архив.zip">Браузер предложит
загрузить архив</a>
```

*Ссылка на удаленный ресурс:*

```
<a href="https://nit.center">Будет
открыта главная страница сайта</a>
<a href="https://nit.center/1.
zip">Ссылка на удаленный архив</a>
```

*По этой ссылке будет загружен почтовый клиент для отправки сообщения на указанный адрес:*

```
<a href="mailto:den@example.com">den@
example.com</a>
```



*Ссылка на номер телефона – на мобильных устройствах браузер предложит набрать данный номер, на стационарном – будет загружена программа, позволяющая произвести набор, если такая установлена:*

```
<a href="tel:+71112223344">Номер телефона</a>
```

*Создание якоря и ссылка на него:*

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

```
...
```

```
<a href="#top">Наверх!</a>
```

Сначала мы создаем именованный якорь, а потом делаем ссылку на него.

## Тег abbr

Синтаксис:

```
<abbr>...</abbr>
```

Используется для обозначения аббревиатуры. Если нужно полное описание, используется атрибут *title*.

Атрибуты:

- **title** = "текст" – при использовании с элементом **abbr** данный атрибут обеспечивает полную расшифровку акронима.
- Глобальные атрибуты HTML.

А  
В  
С  
D  
Е  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
У  
V  
Z

Примеры:

```
<abbr title="Московский государственный  
университет">МГУ</abbr>
```

## Тег address

Синтаксис:

```
<address>...</address>
```

Используется для указания контактной информации в HTML-документе. Собственных атрибутов у этого тега нет, есть только глобальные атрибуты HTML.

- Глобальные атрибуты HTML.

Пример:

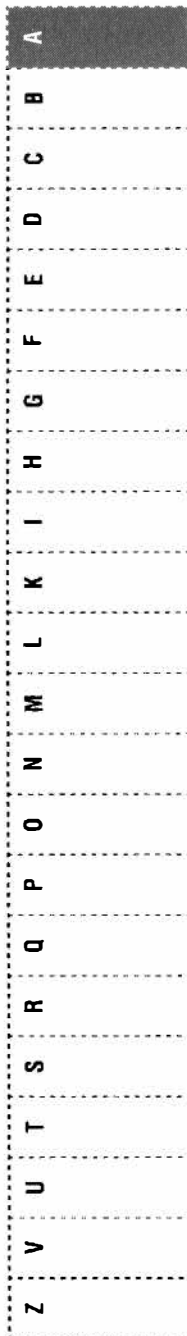
```
<address>  
ул. Колмогорова, 1  
Москва, Россия  
119991  
</address>
```

## Тег area

Синтаксис:

```
<area>
```

Включается в состав элемента *map*, который, в свою очередь, используется для создания клиентской карты изображений. Тег area позволяет определить специфическую область для клика.



Атрибуты:

- **alt** = "Текст" – задает описание изображения — на случай, если файл изображение недоступен или отображение картинок выключено для экономии трафика.
- **coords** = "значения" – содержит список значений координат, разделенных запятыми, которые описывают "горячую" область карты изображений.
- **download** = "имя файла" [только HTML5] – позволяет задать имя файла на компьютере пользователя после его загрузки.
- **href** = "URI" – адрес документа или ресурса, который станет доступным после того, как пользователь щелкнет по области, описанной в теге coords.
- **hreflang** = "код языка" – указывает основной язык целевого документа.
- **media** = "all | aural | braille | handheld | print | projection | screen | tty | tv" [HTML5] – указывает носитель целевого документа
- **rel** = "ключевое слово типа ссылки" – описывает связи между исходным и связанным документами. Для элемента a применяются следующие типы ссылочных связей — alternate, author, bookmark, help, license, next, nofollow, norereferrer, prefetch, prev, search и tag.

- **shape** = "rect | circle | poly | default" – определяет форму области.
- **target** = "текст" – содержит имя окна или области iframe, где будет отображаться целевой документ. Использование фреймов в современных документах не рекомендуется.
- **type** = "тип MIME" – указывает тип контента (тип MIME), например, text/html.
- Глобальные атрибуты HTML.

Пример:

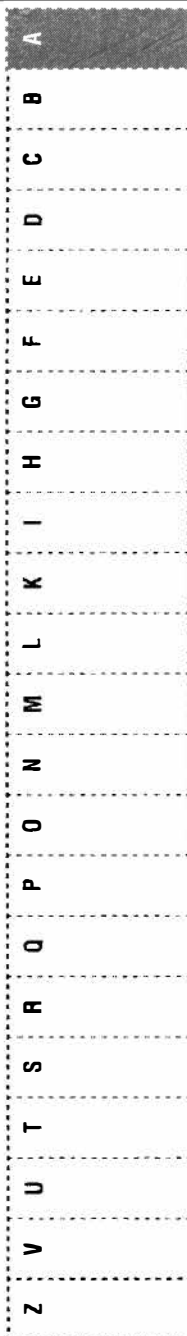
```
<map name= "space">
  <area                shape="rect"
    coords="200,25,280,100"
    href="https://nit.center" alt="">
  <area                shape="rect"
    coords="200,25,280,100"
    href="https://nit.center" alt="">
</map>
```

## Тег article

Синтаксис:

```
<article>...</article>
```

Тег article своими открывающим и закрывающим тегами определяет самостоятельную часть документа, страницы или сайта, предназначенную для независимого использования и обособленного восприятия. Это может быть статья в



блоге, статья в журнале или газете, какой-либо другой самостоятельный фрагмент содержимого. Теги `article` могут быть вложенными.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

**<article>**

```
<header>
```

```
<h1>Изучаем ZFS</h1>
```

```
<p>Каждые несколько лет, кто-то делает прогноз о том, сколько компьютерных ресурсов, скорее всего, понадобится в будущем. Позже, все смеются над наивностью этих прогнозов. В проектировании ZFS, Sun попыталась избежать этой ошибки.
```

```
</header>
```

```
<p>Далее следует контент статьи
```

```
<footer>
```

```
&copy; Виктор Черкасов
```

```
</footer>
```

**</article>**

## Тег `aside`

Синтаксис:

```
<aside>...</aside>
```

Тег `aside` представляет собой часть документа, чье содержимое только косвенно связано с основным содержимым документа. Чаще всего

представлен в виде боковой панели, сносок или меток. Внешний вид содержимого тега `aside` (а между его открывающим и закрывающим тегами может быть все что угодно в плане HTML) никак не определяется самим тегом `aside`, данным тегом просто придается определенный смысл фрагменту html-документа, а выделяется этот фрагмент. А вот как именно он будет отображен – в виде боковой панели, сноски или еще как-то, это вы уже сами зададите с помощью таблиц стилей.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<article>
  <h1>Изучаем ZFS</h1>
  <p>Каждые несколько лет, кто-
то делает прогноз о том, сколько
компьютерных ресурсов, скорее
всего, понадобится в будущем.
Позже, все смеются над наивностью
этих прогнозов. В проектировании
ZFS, Sun попыталась избежать этой
ошибки.
  ...
  <aside>
    <h1>Другие статьи по этой теме:</
    h1>
    <ul>
      <li><a href="zfs.html">Подробно о
      ZFS</a></li>
      <li><a href="fs.
      html">Файловая система Linux</a></
```





```

    li>
  </ul>
</aside>
</article>

```

## Тег audio

Синтаксис:

```
<audio>...</audio>
```

Позволяет вставить звуковой файл в HTML-документ. Адрес звукового файла задается атрибутом *src*. Поскольку разные браузеры поддерживают разные форматы звуковых файлов, элемент *audio* включает наборы параметров для форматов файлов. Также этот тег поддерживает альтернативный контент, отображаемый браузером, который не поддерживает данный тег.

Браузеры постоянно совершенствуются. В таблице 1 приводится информация о поддержке аудио-форматов различными браузерами на момент написания этих строк. Но со временем может все поменяться.

**Таблица 1. Аудио-форматы и браузеры**

Формат	IE	Chrome	Opera	Safari	Firefox
ogg/vorbis	-	+	+	-	+
wav	-	+	+	+	+

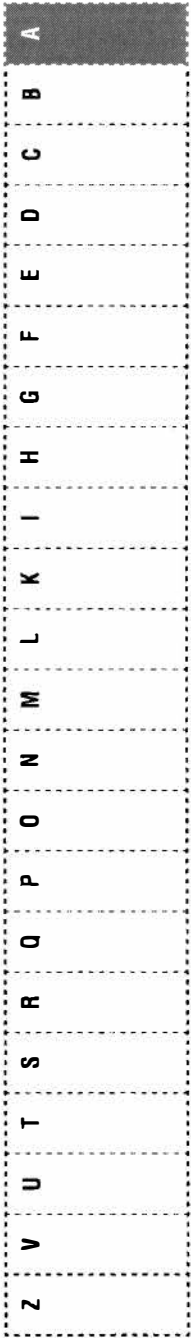
mp3	+	+	-	+	-
AAC	+	+	-	+	-

Атрибуты:

- **autoplay** (или **autoplay="autoplay"**) — звук начинает играть сразу после загрузки страницы.
- **controls** (или **controls="controls"**) — добавляет панель управления к аудиофайлу.
- **loop** (или **loop="loop"**) — повторяет воспроизведение звука с начала после его завершения.
- **muted** (или **muted="muted"**) — отключает (приглушает) звук.
- **preload = "none | metadata | auto"** — используется для загрузки файла вместе с загрузкой веб-страницы. Значение *none* предотвращает предварительную загрузку, *metadata* осуществляет выборку метаданных ресурса, но не загружает сам звуковой файл, *auto* — обеспечивает автоматическую загрузку файла.
- **src** = "путь к файлу" — указывает путь к воспроизводимому файлу.
- Глобальные атрибуты HTML.

Пример:

```
<audio autoplay>
```



A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

```
<source src="audio/music.ogg"
type='audio/ogg; codecs=vorbis'>
```

```
<source src="audio/music.mp3"
type="audio/mpeg">
```

Тег `audio` не поддерживается вашим браузером.

```
<a href="audio/music.mp3">Скачайте
музыку</a>.
</audio>
```

Тег `source` позволяет указать несколько источников звукового файла – в разных форматах (для обеспечения совместимости с разными браузерами), что и продемонстрировано в этом примере. Также мы задаем текст, который будет показан, если браузер не поддерживает тег `<audio>`.

## Тег b

Синтаксис:

```
<b>...</b>
```

Используется для выделения полужирным текст, заслуживающий внимания. Собственных атрибутов не имеет, но можно использовать глобальные атрибуты HTML.

Пример:

```
<b>Этот текст будет выделен жирным</b>
```

Атрибуты:

- Глобальные атрибуты HTML.

# Тег base

Синтаксис:

```
<base>
```

Тег base используется для явного задания полного URL-адреса документа. Это бывает полезно ввиду того, что общепринятым стилем задания гипертекстовых ссылок является относительная их адресация. То есть при задании ссылки на документ указывается не полный его URL-адрес, а задается его месторасположение относительно текущего адреса. Так вот тег BASE как раз и задает адрес, относительно которого и будут браться относительные ссылки в HTML-документе.

Пример относительной ссылки:

```
<A href="/BOOK1/chapter1.html">
```

Такой стиль полезен тем, что при переносе всего дерева HTML-документов в другое место (сайт), не требуется изменять ссылки в самих HTML-документах.

Использование тега base позволяет реализовывать относительные ссылки в том случае, когда HTML-документ перемещен (или скопирован), а все остальное дерево документов, на которые он ссылается, нет. Адрес его поменялся (например, документ лежал на [www.anekdot.ru](http://www.anekdot.ru), а теперь у Вас дома, на диске С), однако при активизации относительной ссылки, она будет взята браузером

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

относительно исходного адреса, прописанного в теге `base`.

Атрибуты:

- **href** = "URL" – задает URL документа, используемый в качестве базового при решении относительных идентификаторов URL.
- **target** = "имя" – определяет заданное по умолчанию целевое окно или фрейм для всех ссылок, используемых в документе. Использование фреймов не рекомендуется!
- Глобальные атрибуты HTML.

Начальный тег `base` обязателен, конечный тег запрещен.

Тег `base` можно использовать и в заголовке и в теле документа, причем несколько раз. Область действия тега `base` определяется от места его задания и до конца документа, или до следующего объявления тега `base`, если таковой имеется.

Пример:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.01 //EN">
<HTML>
  <HEAD>
    <TITLE>Документ с
    использованием элемента BASE</
    TITLE>
    <BASE href="https://www.nit.
```

```

center">
</HEAD>
<BODY>
    ...текст документа...
    ...текст документа...<a
href= "/images/ret.gif">Ссылка на
изображение ret.gif</a>
    ...текст документа...

<a href= "chapter2.html">Переход к
следующей главе</a>
    ...текст документа...
    ...текст документа...
    ...текст документа...
</BODY>
</HTML>

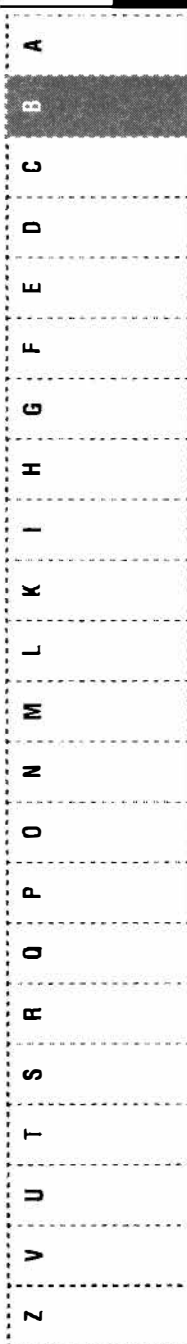
```

В этом примере переход по относительным ссылкам задается относительно URL-адреса `https://www.nit.center`. Таким образом, заданные в этом документе ссылки в абсолютном варианте всегда (независимо от месторасположения документа) будут иметь следующий вид: `https://www.nit.center/images/ret.gif` и `https://www.nit.center/chapter2.html`. Если базовый адрес задан бы не был, то эти ссылки интерпретировались бы относительно каталога, в котором находится данный документ. Соответственно при перемещении документа изменялись бы цели относительных ссылок.

## Тег bdi

Синтаксис:

```
<bdi>...</bdi>
```



Применяется для выделения текста, который может читаться в направлении, противоположном направлению окружающего текста.

Атрибуты:

- **dir** = "**ltr** | **rtl** | **auto**" – указывает направление чтения текста: *ltr* (слева направо), *rtl* (справа налево), *auto* (определяется браузером). Если атрибут не указан, используется значение *auto*.
- Глобальные атрибуты HTML.

Пример:

```
<p dir="rtl"><bdi>шн</bdi> – суббота</p>
```

## Тег bdo

Синтаксис:

```
<bdo>...</bdo>
```

Элемент применяется для выделения текста, который читается в обратном направлении.

Атрибуты:

- **dir** = "**ltr** | **rtl** | **auto**" – указывает направление чтения текста: *ltr* (слева направо), *rtl* (справа налево), *auto* (определяется браузером). Если атрибут не указан, используется значение *auto*.
- Глобальные атрибуты HTML.

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

Пример:

```
<p><bdo dir="rtl">А роза упала на лапу
Азора</bdo></p>
```

## Тег blockquote

Синтаксис:

```
<blockquote>...</blockquote>
```

Используется для отображения длинной цитаты, которая была позаимствована из другого источника. Текст, обозначенный этим тегом, традиционно отображается как выровненный блок с отступами слева и справа (примерно по 40 пикселей), а также с отступами сверху и снизу.

Атрибуты:

- **cite** = "URL" — адрес, который указывает на источник цитаты.

Пример:

```
<blockquote>
  <p>Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed diam nonummy nibh euismod tincidunt
ut laoreet dolore magna aliquam
erat volutpat. Ut wisis enim ad minim
veniam, quis nostrud exerci tution
ullamcorper suscipit lobortis nisl ut
aliquip ex ea commodo consequat.</p>
</blockquote>
```

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z



## Тег body

Синтаксис:

```
<body>...</body>
```

В теле документа располагается содержательная его часть. В качестве тела документа определен элемент-контейнер *body*. Начальный и конечный теги являются необязательными (необходимость их использования определяется контекстом).

Атрибуты:

- Глобальные атрибуты HTML
- Обработчики событий HTML (onload, onunload, ononline и т.д.)

Пример:

```
<html>
  <head>
    <title>Пример</title>
  </head>
  <body>
    <h1>Контент документа</h1>
  </body>
</html>
```

## Тег br

Синтаксис

```
<br>
```

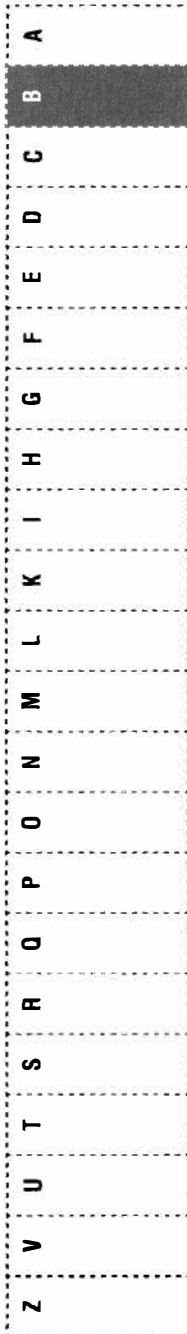
Формирует разрыв строки в контенте, например, в почтовом адресе. Собственных атрибутов нет, только глобальные атрибуты HTML.

Пример:

```
<address>
ул. Колмогорова, 1<br>
Москва, Россия<br>
119991<br>
</address>
```



Рис. 3. Демонстрация использования тега <br>



# Тег button

Синтаксис:

```
<button>...</button>
```

Тег создает кнопку и по своему действию напоминает тег `<input type="button">`. Кнопки используются, как правило, в формах данных. Действие кнопки задается, как правило, с помощью обработчика события *onclick* (будет далее показано в примере).

Атрибуты:

- **autofocus** – отображает кнопку с выделенным фокусом (кнопка будет сразу выделена и доступна для нажатия).
- **disabled** – кнопка будет выключена и недоступна для нажатия. Позже кнопку можно будет активировать с помощью JavaScript, но рассмотрение данного приема выходит за рамки этой книги.
- **form** = "идентификатор формы" – явно связывает кнопку с формой, идентификатор которой указывается с помощью данного атрибута.
- **formaction** = "URL-ссылка" – указывает приложение, обрабатывающее форму. Выполняет те же функции, что и атрибут *action* для тега *form*.

- **formenctype** = "тип контента" – задает способ кодирования данных формы. При значении `application/x-www-form-urlencoded` вместо пробелов ставится `+`, символы вроде русских букв кодируются их шестнадцатеричными значениями (например, `%D0%9F%D0%B5%D1%82%D1%8F` вместо Петя). Значение `multipart/form-data` означает, что данные не кодируются. Это значение используется при отправке файла. Значение `text/plain` означает, что пробелы заменяются знаком `+`, а остальные символы не кодируются.
- **formmethod** = "метод отправки формы" – указывает метод пересылки данных формы. Допустимые значения `GET` и `POST`, аналогично, как и для атрибута *method* тега *form*.
- **formnovalidate** – отменяет проверку данных, введенных пользователем в форме на корректность.
- **formtarget** = "имя" – указывает целевое окно, в котором отображаются результаты передачи данных формы. Выполняет ту же функцию, что и атрибут `target` для тега *form*.
- **menu** = "текст" – задает меню, которое будет отображено при нажатии кнопки, если для ее атрибута **type** используется значение *menu*.

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

- **name** = "текст" – присваивает имя элементу управления – кнопке.
- **type** = "submit|reset|button|menu" – определяет поведение кнопки: *submit* (отправка формы, по умолчанию), *button* (клиентская кнопка, управляемая с помощью кода JavaScript), *reset* (сброс формы к значениям по умолчанию), *menu* (отображает меню, заданное атрибутом menu).
- **value** = "текст" – присваивает значение кнопки. Поведение кнопки задается атрибутом *type*.
- Глобальные атрибуты HTML.

Примеры:

Кнопка, при нажатии на которую происходит переход по ссылке. Это достигается благодаря вложению тега `button` в тег `a`:

```
<a href="https://nit.
center"><button>Перейти на сайт</
button></a>
```

Кнопка отправки формы:

```
<button type="submit"
name="go">Отправить</button>
```

Кнопка, при нажатии на которую вызывается JS-функция `ShowAlert()`:

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

```
<button onclick="ShowAlert()">Показать  
окно</button>
```

В популярном фреймворке Bootstrap для красивого отображения кнопок предназначены следующие стили:

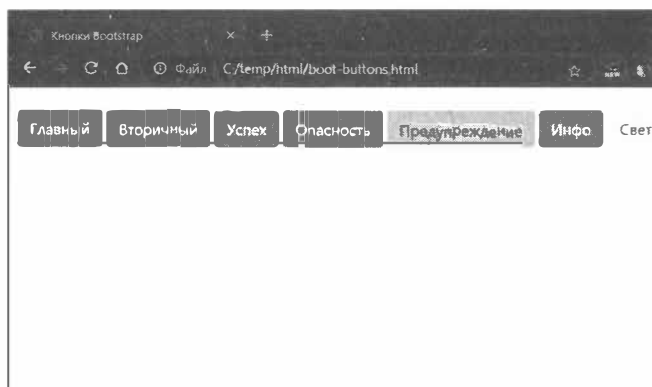
- **btn** – элемент страницы будет выглядеть именно как кнопка.
- **btn-primary** – главная кнопка (фон кнопки – синий).
- **btn-secondary** – вторичная кнопка (фон кнопки – серый).
- **btn-success** – кнопка успешного действия (фон кнопки – зеленый).
- **btn-danger** – кнопка опасности. Используется для обозначения опасных действий, например, удаления. Цвет кнопки – красный.
- **btn-warning** – кнопка-предупреждение (фон – желтый).
- **btn-info** – информационная кнопка (например, кнопка в диалоге). Цвет фона – синий, но оттенок другой, в отличие от btn-primary.
- **btn-light** – светлая кнопка (цвет кнопки – светло-серый).
- **btn-dark** – темная кнопка (цвет кнопки – темно-серый, "мокрый асфальт").

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

- **btn-link** – кнопка в виде ссылки. Цвет фона – белый, цвет текста – синий.

Рассмотрим пример применения этих стилей:

```
<html>
<head>
<title>Кнопки Bootstrap</title>
<!-- Подключаем стили Bootstrap -->
<link rel="stylesheet" href="https://
stackpath.bootstrapcdn.com/bootstrap/
4.5.2/css/bootstrap.min.css">
</head>
<body>
<br>&nbsp;
<button type="button" class="btn btn-
primary">Главный</button>
<button type="button" class="btn btn-
secondary">Вторичный</button>
<button type="button" class="btn btn-
success">Успех</button>
<button type="button" class="btn btn-
danger">Опасность</button>
<button type="button" class="btn btn-
warning">Предупреждение</button>
<button type="button" class="btn btn-
info">Инфо</button>
<button type="button" class="btn btn-
light">Светлый</button>
<button type="button" class="btn btn-
dark">Темный</button>
<button type="button" class="btn btn-
link">Линк</button>
</body>
</html>
```



*Рис. 4. Кнопки Bootstrap*

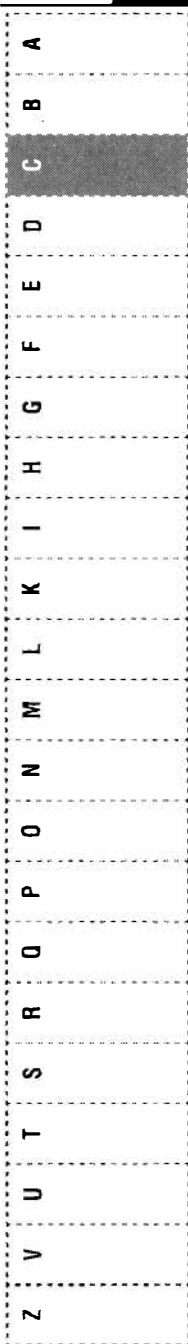
Наверняка вы видели такие кнопки в современных веб-приложениях. Теперь вы знаете, как создать такие же красивые кнопки в своих страницах.

## Тег canvas

Синтаксис:

```
<canvas>...</canvas>
```

Создает область, в которой при помощи JavaScript можно рисовать разные объекты, выводить изображения, трансформировать их и менять свойства.





## Атрибуты:

- **height** = "число" – высота области холста, выраженная в пикселях.
- **width** = "число" – ширина области холста, выраженная в пикселях.
- Глобальные атрибуты HTML.

## Пример:

```

<html>
<head>
<script type="application/x-javascript">
function draw() {
    var canvas = document.
getElementById("box");
    var ctx = canvas.getContext("2d");
    ctx.fillStyle = "rgb(170, 110, 250)";
    ctx.fillRect(60, 70, 85, 100);
}
</script>
</head>
<body onload="draw()">
<canvas id="box" width="350"
height="350"></canvas>
</body>
</html>

```

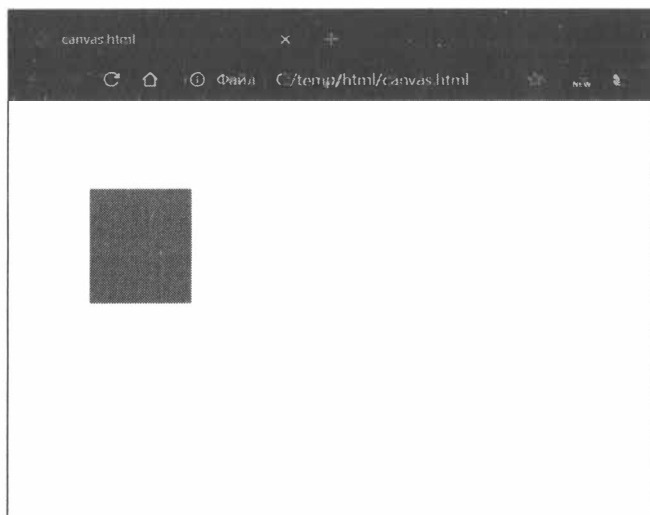


Рис. 5. Использование тега `canvas`

## Тег `caption`

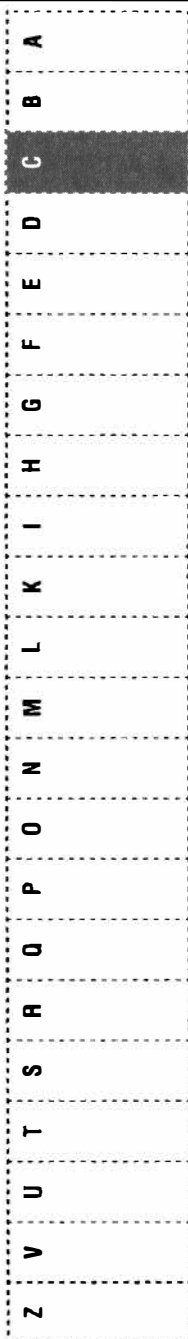
Синтаксис:

```
<caption>...</caption>
```

Предназначен для создания заголовка к таблице и может использоваться только внутри контейнера `<table>`, причем только сразу после открывающегося текста.

Атрибуты:

- **align** — определяет выравнивание заголовка по горизонтали.



- **valign** — устанавливает расположение заголовка до или после таблицы.
- Глобальные атрибуты HTML.

Пример:

```
<table>
  <caption>Текст</caption>
  <tr>
    <td>...</td>
  </tr>
</table>
```

## Тег center

Синтаксис:

```
<center>...</center>
```

Выравнивает содержимое тега по центру экрана. В современных HTML-спецификациях данный тег использовать запрещено, а выравнивание текста осуществляется посредством CSS.

Атрибуты:

- Только глобальные атрибуты HTML

Пример:

```
<center>По центру</center>
```

# Тег cite

Синтаксис:

```
<cite>...</cite>
```

Используется для формирования ссылки на другую статью, книгу, журнал. Помечает заголовок печатной работы.

Атрибуты:

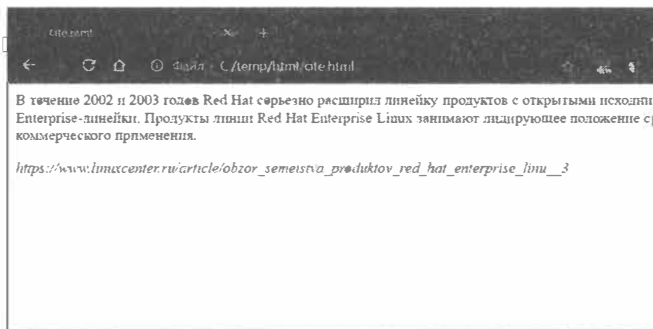
- Глобальные атрибуты HTML

Пример:

```
<p>В течение 2002 и 2003 годов Red Hat серьезно расширил линейку продуктов с открытыми исходниками выпуском семейства дистрибутивов Enterprise-линейки. Продукты линии Red Hat Enterprise Linux занимают лидирующее положение среди дистрибутивов Linux для коммерческого применения.
```

```
<p><cite>https://www.linuxcenter.ru/article/obzor\_semeistva\_produktov\_red\_hat\_enterprise\_linu\_\_3</cite>
```

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z



**Рис. 6. Демонстрация тега cite**

## Тег code

Синтаксис:

```
<code>...</code>
```

Позволяет выделить фрагмент компьютерного кода, который будет отображаться как встроенный элемент. Браузеры отображают содержимое этого тега с помощью моноширинного шрифта.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<code>
    &lt;?php
        echo "Hello!";
    ?&gt;
</code>
```

Рис. 7. Пример отображения тега *code*

## Тег col

Синтаксис:

```
<col>...</col>
```

Создает столбец в таблице (или столбцы с помощью атрибута `span`). Свойства атрибута могут применяться ко всем ячейкам в столбце. Данный элемент не выполняет структурное группирование столбцов, а является пустым элементом, обеспечивающим совместное использование атрибутов. Его нужно включать после элемента заголовка и перед произвольными элементами строки или группы строк вместе с элементом *table*.

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

### Атрибуты:

- **span** = "число" – указывает количество столбцов, которые попадают в область действия элемента col.
- Глобальные атрибуты HTML.

### Пример:

```
<table width="400" border="0"
cellpadding="5" cellspacing="0">
  <col width="150" valign="top">
  <col width="250" valign="top">
  <tr>
    <td style="background: #D0D28E">Столбец 1
    </td>
    <td style="background: #DDDDDD">Столбец 2
    </td>
  </tr>
</table>
```

## Тег colgroup

### Синтаксис:

```
<colgroup>...</colgroup>
```

Позволяет группировать столбцы таблицы. Таблица может отображаться вся сразу, а может последовательно. В первом случае браузер ждет загрузки содержимого всех ячеек таблицы и только после этого отображает таблицу. Для убыстрения загрузки рекомендуется организовывать последовательное отображение таблицы.

В этом случае сначала загружаются и отображаются границы таблицы (если они есть), затем по очереди загружается и отображается содержимое ячеек. При этом загрузка всей таблицы по времени не меняется, но зато значительно уменьшается время ожидания первого представления таблицы (первой отображенной информации), что делает ее загрузку гораздо привлекательней. Для обеспечения последовательного отображения таблицы браузеру необходимо указать ее предварительные размеры. Для этого и используются теги **colgroup**. У данного тега есть атрибут `span = "число"`, который и задает количество столбцов, попадающих в область действия тега **colgroup**. Тег **colgroup** используется после тега **caption** (при наличии) и перед тегами строки (**tr**) или группы строк.

Атрибуты:

- **span** = "число" — количество колонок, к которым следует применять атрибуты.
- Глобальные атрибуты HTML.

Пример:

```
<table>
  <colgroup id="info">
    <col span="2" width="200">
    <col span="1" width="100">
  </colgroup>
  <thead>
    ...
  продолжение разметки таблицы
```

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z



# Тег command

Синтаксис:

`<command>`

Данный элемент используется вместе с элементом *menu* и является интерактивным. Он представляет действие меню. В HTML5.1 данный элемент не рекомендуется к использованию, а вместо него нужно использовать элемент *menuitem*.

Атрибуты:

- **checked** – указывает на то, что данная команда должна быть выбранной.
- **command** = "идентификатор мастер-команды" – если используется данный атрибут, то состояние тега *command* устанавливается на основании мастер-команды.
- **disabled** – указывает, что данная команда меню временно недоступна.
- **icon** = "URL картинки" – задает местонахождение изображения, используемого в качестве кнопки для команды.
- **label** = "текст" – задает текст команды меню.
- **radiogroup** = "группа" – задает название группы команд, если выбран тип команды *radio*.

- **title** = "текст" – создает подсказку для команды.
- **type** = "command/checkbox/radio" – задает тип команды. Ключевое слово *command* создает обычную команду вместе со связанным действием. Ключевое слово *checkbox* создает команду в виде чек-бокса, *radio* – в виде зависимого переключателя.
- Глобальные атрибуты HTML.

Пример:

```
<menu>
  <command onclick="alert('Команда 1')">
    label="Команда 1">
  <command onclick="alert('Команда 2')">
    label="Команда 2">
  <command onclick="alert('Команда 3')">
    label="Команда 3">
</menu>
```

## Ter data

Синтаксис:

```
<data>...</data>
```

Данный элемент поддерживается только в HTML5.1. Представляет распознаваемый компьютером эквивалент для контента тега, используемый при обработке данных. Может использоваться для всех видов данных, в том числе для времени, даты, веса и т.д.

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

Атрибуты:

- **value** = "значение" – поддерживает распознаваемый компьютером эквивалент для контента элемента.
- Глобальные атрибуты HTML.

Пример:

```
<data value="2021-02-12">12 февраля 2021
года</data>
```

## Ter datalist

Синтаксис:

```
<datalist>...</datalist>
```

Данный тег используется вместе с элементом управления *input* с целью создания нового управления *list*. Данный тег создает раскрывающееся меню, включающее набор предварительно заданных вариантов. Используется для реализации функции "автодополнения" ввода, когда пользователь вводит данные в поле со списком и получает список доступных вариантов. Разница между элементом *datalist* и *select* в том, что в первом случае пользователю не нужно выбирать один из заранее заданных вариантов, а можно вводить в поле все что угодно.

Атрибуты:

- Только глобальные атрибуты

Пример:

```
<input list="fruit">
<datalist id="fruit">
  <option value="Яблоко">
  <option value="Апельсин">
</datalist>
```

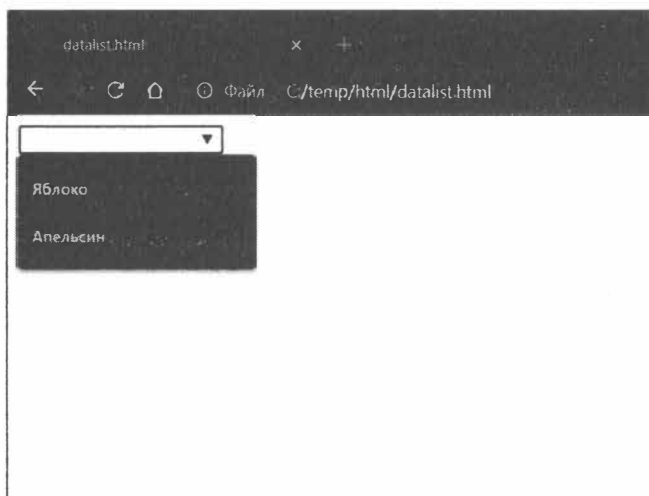


Рис. 8. Тег `datalist` в действии

## Тег `dd`

Синтаксис:

```
<dd>...</dd>
```

Входит в тройку элементов `<dl>`, `<dt>`, `<dd>`, предназначенных для создания списка определений. Каждый такой список начинается с кон-

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

тейнера `<dl>`, куда входит тег `<dt>` создающий термин и тег `<dd>` задающий определение этого термина. Закрывающий тег `</dd>` не обязателен.

Атрибуты:

- Глобальные атрибуты HTML.

Пример:

```
<dl>
  <dt>Термин 1</dt>
  <dd>Определение термина 1</dd>
  <dt>Термин 2</dt>
  <dd>Определение термина 2</dd>
</dl>
```

## Тег del

Синтаксис:

```
<del>...</del>
```

Посредством данного тега отмечается удаленный фрагмент документа. Используется для выделения правок в различных документах.

Атрибуты:

- **cite** = "URL-ссылки" – используется для создания ссылки на исходный документ, в котором указана причина изменения.
- **datetime** = "ГГГГ-ММ-ДДТчч:мм:ссTZD" – позволяет указать дату и время изменения.

- Глобальные атрибуты HTML.

## Ter details

Синтаксис:

```
<details>...</details>
```

Представляет виджет замыкания. Тег **details** может включать тег **summary**, с помощью которого формируется резюме или заголовок для деталей, за которым следует дополнительный потоковый контент.

Атрибуты:

- **open** – указывает на то, что область, описываемая тегом **details**, должна быть в открытом или явном состоянии, когда документ загружается.
- Глобальные атрибуты HTML.

Пример:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>details</title>
  </head>
  <body>
    <details>А это подробности!</details>
  </body>
</html>
```

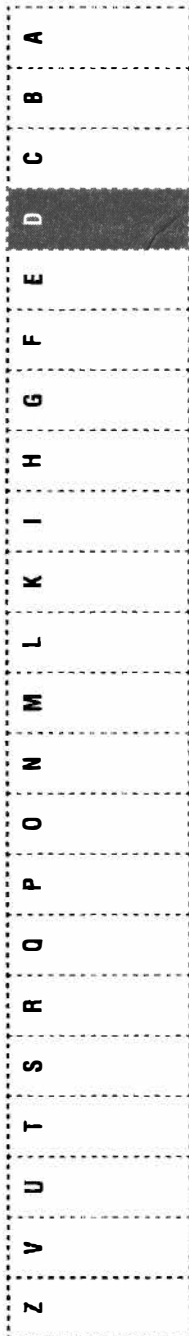




Рис. 9. Тег *details*

## Ter dfn

Синтаксис:

```
<dfn>...</dfn>
```

Когда в документе указывается новый термин, он выделяется курсивом и дается его определение. Тег **dfn** используется для выделения таких терминов при их первом появлении в тексте. Браузеры отображают содержимое тега *dfn* с помощью курсива.

Атрибуты:

- Глобальные атрибуты HTML.

Пример:

```
<p><dfn>Информатика</dfn> — наука о  
методах и процессах сбора, хранения,  
обработки, передачи, анализа и оценки  
информации с применением компьютерных  
технологий</p>
```

## Тег dialog

Синтаксис:

```
<dialog>...</dialog>
```

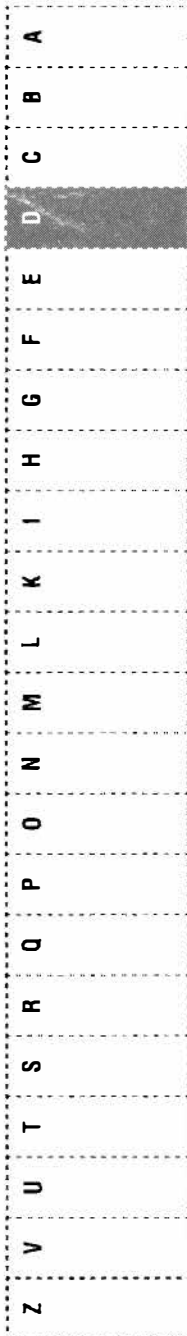
Позволяет создать плавающее диалоговое окно, например, окно входа на сайт или окно регистрации. Данный тег поддерживается только в спецификации HTML5.1. На данный момент (февраль-март 2021 года) данный тег не поддерживается ни одним браузером.

## Тег div

Синтаксис:

```
<div>...</div>
```

Тег **div** является блочным элементом и используется для выделения фрагмента документа с целью изменения внешнего вида его содержимого. Внешний вид содержимого изменяется с помощью CSS-стилей, а для самого тега `<div>`





используется атрибут **class** для указания CSS-класса или атрибут **id** с именем селектора, чтобы обращаться к тегу через JavaScript.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <title>Ter DIV</title>
  <style type="text/css">
    .block1 {
      width: 300px;
      background: #ddd;
      padding: 5px;
      padding-right: 30px;
      border: solid 1px black;
      float: left;
    }
    .block2 {
      width: 300px;
      background: #dc0;
      padding: 5px;
      border: solid 1px black;
      float: left;
      position: relative;
      top: 50px;
      left: -30px;
    }
  </style>
</head>
<body>
```

```
<div class="block1">Съешь ещё этих  
мягких французских булок,  
да выпей чаю</div>  
<div class="block2">Съешь ещё этих  
мягких французских булок,  
да выпей чаю</div>  
  
</body>  
</html>
```

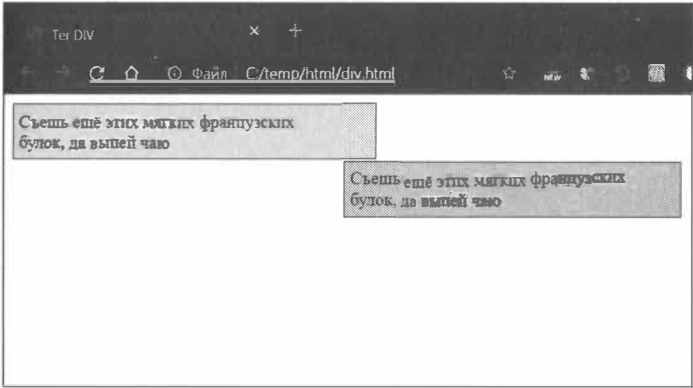


Рис. 10. Использование тега div

В современном мире тег **div** используется для разметки (layout) страницы. Используя CSS, можно с легкостью создать страницу, состоящую из нескольких колонок, используя тег **div**. Для облегчения этого процесса (чтобы самому не определять CSS-правила), вы можете использовать фреймворк Bootstrap. Рассмотрим пример:

```
<html>  
<head>
```

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

```

<title>Колонки Bootstrap</title>
<link rel="stylesheet" href="https://
stackpath.bootstrapcdn.com/bootstrap/
4.5.2/css/bootstrap.min.css">
</head>
<body>
<div class="container">
  <div class="row">
    <div class="col-sm">
      Съешь ещё этих мягких французских
булок, да выпей чаю
    </div>
    <div class="col-sm">
      Съешь ещё этих мягких французских
булок, да выпей чаю
    </div>
    <div class="col-sm">
      Съешь ещё этих мягких французских
булок, да выпей чаю
    </div>
  </div>
</div>
</body>
</html>

```

Подробное описание Bootstrap, к сожалению, выходит за рамки этой книги, но только что вы создали "резиновую" разметку (которая будет адаптироваться под размер экрана устройства) на три колонки, не написав ни одного CSS-правила! Как обойтись без Bootstrap и создать аналогичную разметку без его стилей, будет показано далее в этой книге. Читайте внимательно!

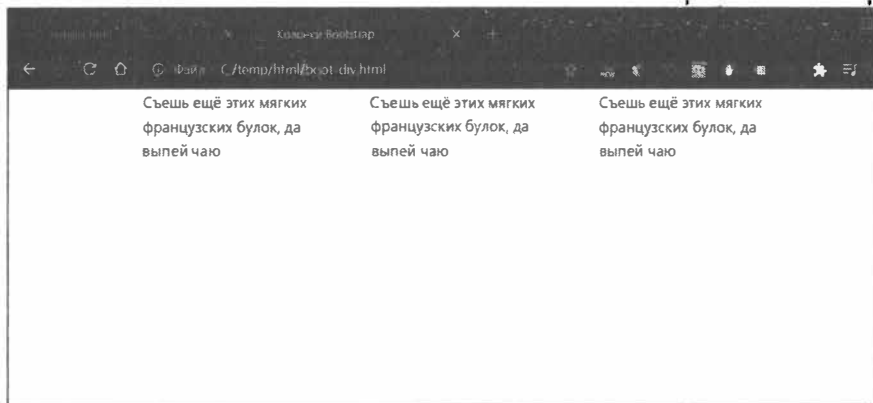


Рис. 11. Разметка с помощью Bootstrap

## Тег dl

Синтаксис:

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

Данный тег демонстрирует список описаний. Каждый элемент в этом списке является парой "имя-значение", состоящей из имени (тег dt) и его значения (dd). Списки описаний используются для терминов и определений, FAQ и другой подобной информации.

Атрибуты:

- Глобальные атрибуты HTML

H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

## Пример:

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Список определений</title>
    <style>
      dd {
        font-style: italic; /* Курсивное
начертание текста */
      }
      dt {
        margin-top: 1em; /* Отступ сверху */
      }
    </style>
  </head>
  <body>

    <dl>
      <dt>GIF</dt>
      <dd>Graphics Interchange Format</dd>
      <dt>JPEG</dt>
      <dd>Joint Photographic Experts Group</
dd>
    </dl>

    </body>
  </html>
```



Рис. 12. Использование тега `dl`

## Тег `dt`

Синтаксис:

```
<dt>...</dt>
```

Выделяет имя тега внутри списка описаний. См. описание тега `dl`.

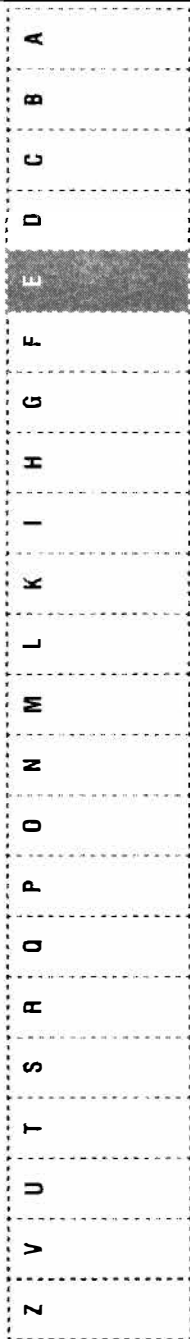
- Глобальные атрибуты HTML

## Тег `em`

Синтаксис:

```
<em>...</em>
```

Тег `<em>` предназначен для акцентирования текста. Браузеры отображают такой текст курсивным начертанием.



Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<em>Текст</em>
```

## Тег embed

Синтаксис:

```
<embed>...</embed>
```

Используется для загрузки и отображения объектов (например, видеофайлов, флэш-роликов, некоторых звуковых файлов и т.д.), которые исходно браузер не понимает. Как правило, такие объекты требуют подключения к браузеру специального плагина или запуска вспомогательной программы.

Атрибуты:

- **height** = "число", **weight** = "число" – задает высоту и ширину объекта, выраженного в пикселях.
- **src** = "URL-ссылка" – задает адрес объекта, который будет отображен на странице.
- **type** = "MIME-тип" – задает MIME-тип отображаемого объекта.
- Глобальные атрибуты HTML.

Пример:

```
<embed src= "movies/film.mov" width="240"
height="196"
pluginspage= "http://www.apple.com/
quicktime/download/">
<noembed><img src= "film.gif"> У вас не
установлен
подключаемый модуль.</noembed>
</embed>
```

## Ter fieldset

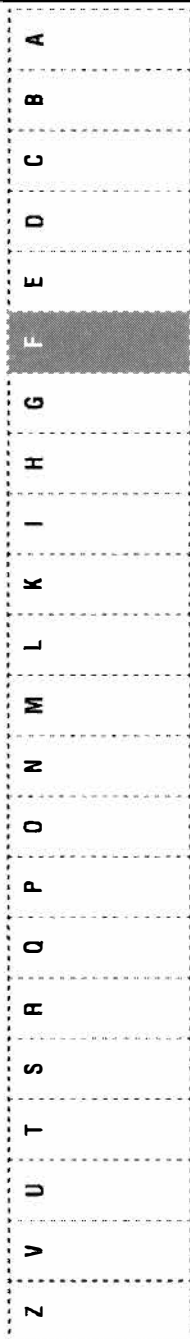
Синтаксис:

```
<fieldset>...</fieldset>
```

Данный тег позволяет сформировать группу связанных элементов управления формой и меток. Обычно теги `<fieldset>` включаются в состав тега `form`.

Атрибуты:

- **disabled** – отключает элементы управления, включенные в действие области тега `fieldset`. Браузеры по-разному отображают отключенные элементы управления, но чаще всего их выделяют серым цветом.
- **form** = "идентификатор формы" – позволяет связать набор полей с определенной формой. При этом тег `fieldset` не должен быть потомком элемента `form`, который применяется к `fieldset`.
- **name** = "название" – задает имя набору полей.





- Глобальные атрибуты HTML

Пример:

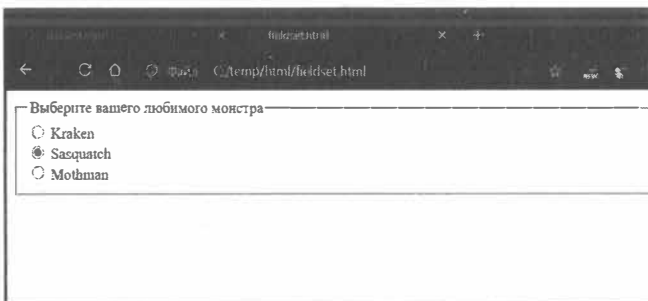
```
<form>
  <fieldset>
    <legend>Выберите вашего любимого
монстра</legend>

    <input type="radio" id="kraken"
name="monster">
    <label for="kraken">Kraken</label><br/>

    <input type="radio" id="sasquatch"
name="monster">
    <label for="sasquatch">Sasquatch</
label><br/>

    <input type="radio" id="mothman"
name="monster">
    <label for="mothman">Mothman</label>
  </fieldset>
</form>
```

Пример показывает, как элемент `<fieldset>` группирует части HTML-формы, а вложенный элемент `<legend>` задает заголовок для `<fieldset>`.



**Рис. 13. Использование тега `fieldset`**

# Тег figcaption

Синтаксис:

```
<figcaption>...</figcaption>
```

Позволяет задать заголовок или легенду, относящуюся к тегу **figure**, который является родительским по отношению к тегу **figcaption**.

Атрибуты:

- Глобальные атрибуты HTML

```
<figure>  
    
  <figcaption>Моя кошка</figcaption>  
</figure>
```

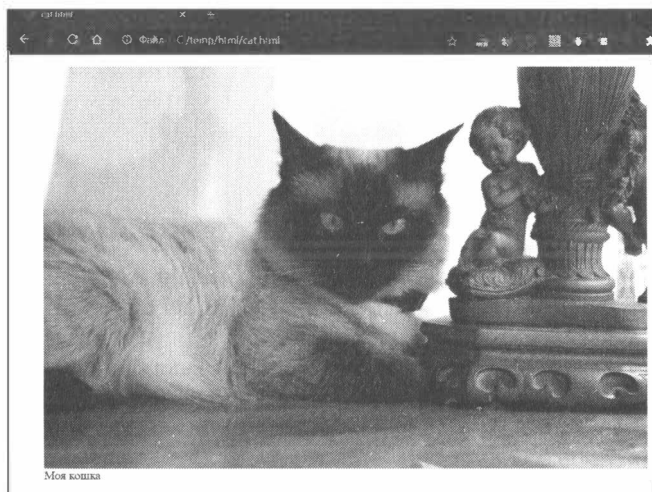


Рис. 14. Рисунок и подпись

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

A  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

## Ter figure

Синтаксис:

```
<figure>...</figure>
```

Используется для группирования любых элементов, например, изображений и подписей к ним.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

*См. figcaption*

## Ter footer

Синтаксис:

```
<footer>...</footer>
```

Ter FOOTER представляет собой нижний колонтитул (футер, подвал), содержащий информацию о документе, его авторе, контакты и т.п. Такая область (набор тегов) огораживается открывающим и закрывающим тегами FOOTER и называется "подвалом" или "футером". Подвал обычно располагается внизу интернет-страницы (html-документа).

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<html>
  <head>
    <title>   Название   документа
  </title>
  </head>
  <body>

    <header>
      Шапка сайта
    </header>
    <nav>
      Навигация
      <ul>
        <li><a href="#">Ссылка 1</a></li>
        <li><a href="#">Ссылка 2</a></li>
      </ul>
    </nav>
    <section>
      Секция 1
      <article>
        <h1>Заголовок статьи</h1>
        <p>Контент</p>
        <h2>Подзаголовок статьи</h2>
      </article>
    </section>
    <section>
      Секция 2
      <article>
        <h1>Заголовок статьи 2</h1>
        <p>Контент</p>
        <h2>Подзаголовок статьи</h2>
      </article>
    </section>

    <footer>
      <p>Дата, адрес, авторство</p>
    </footer>
  </body>
</html>
```

A  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

# Ter form

Синтаксис:

```
<form>...</form>
```

Позволяет создать интерактивную форму, содержащей элементы управления, предназначенные для сбора вводимых пользователем данных, а также другого контента страницы. В документе может быть несколько форм, но они не могут вкладываться одна в другую и перекрываться.

Атрибуты:

**accept-charset** = "список кодировок" – позволяет определить список кодировок символов, которые должны приниматься серверов для обработки текущей формы. Чаще всего используется кодировка utf-8. Но так как в последнее время и серверы, и браузеры перешли на эту кодировку, то данный атрибут можно вообще не указывать.

**action** = "адрес сценария обработки формы" – указывает адрес приложения (сценария), которое будет обрабатывать формы. Данный атрибут является обязательным.

**autocomplete** = "on/off" – позволяет браузеру заполнить поля формы автоматически (on) или требует, чтобы пользователь каждый раз сам вводил информацию (off).

**enctype** = "тип контента" – задает способ кодирования данных формы. При значе-

нии application/x-www-form-urlencoded вместо пробелов ставится +, символы вроде русских букв кодируются их шестнадцатеричными значениями (например, %D0%9F%D0%B5%D1%82%D1%8F вместо Петя). Значение multipart/form-data означает, что данные не кодируются. Это значение используется при отправке файла. Значение text/plain означает, что пробелы заменяются знаком +, а остальные символы не кодируются.

**method** = "GET/POST" – указывает HTTP-метод, используемый для передачи данных формы. При использовании метода GET информация, введенная пользователем, будет добавлена к URL и отправлена вместе с URL-ссылкой.

**name**="имя" – присваивает имя форме.

**novalidate** – указывает на то, что форма не была верифицирована при передаче данных.

**target** = "имя" – указывает цель для результатов передачи данных формы, которые должны быть загружены. Атрибут позволяет указать окно, в которое будет загружен результат обработки формы. Можно использовать специальные целевые значения `_top`, `_bottom`, `_parent`, `_self`.

- Глобальные атрибуты HTML.

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

Пример:

```
<form method="post" action="login.php">
  <p><input type="text" name="login"
    class="form-control"
    placeholder="Логин">
  <p><input type="password"
    name="password"
    class="form-control"
    placeholder="Пароль">
  <p><button type="submit" class="btn
    btn-primary">Войти</button>
</form>
```

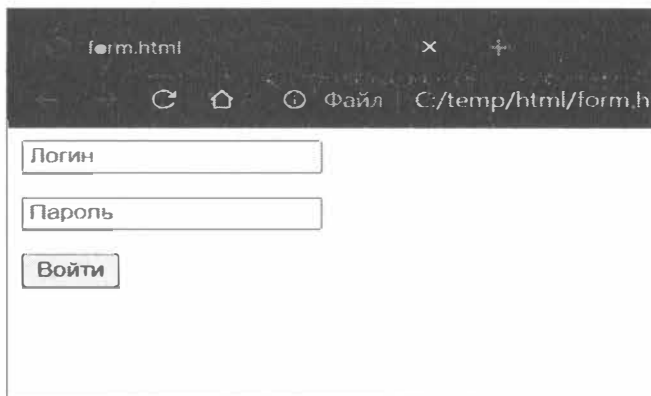


Рис. 15. Созданная форма

## Теги H1, H2, H3, H4, H5, H6

Синтаксис:

```
<h1>...</h1>
```

```
<h2>...</h2>
<h3>...</h3>
<h4>...</h4>
<h5>...</h5>
<h6>...</h6>
```

Язык HTML определяет шесть заголовков разного уровня, демонстрирующие важность секции, расположенной после заголовка. Тег **h1** представляет самый важный заголовок первого уровня, а тег **h6** используется для обозначения заголовка шестого уровня и является наименее значимым.

Теги h1,...,h6 относятся к блочным элементам, они всегда начинаются с новой строки, а после них другие элементы отображаются на следующей строке. Кроме того, перед заголовком и после него добавляется пустое пространство.

При использовании тегов H1, H2, H3, H4, H5, H6 задание начального тега является обязательным. Конечный тег может не указываться, тогда заголовком будет считаться все, что расположено после начального тега.

Атрибуты:

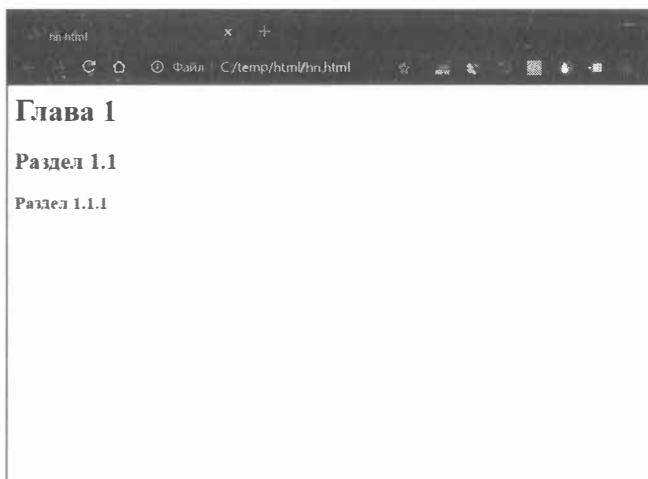
- Глобальные атрибуты HTML

Пример:

```
<h1>Глава 1</h1>
<h2>Раздел 1.1</h2>
<h3>Раздел 1.1.1</h3>
```

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z





**Рис. 16. Заголовки h1, h2, h3**

## Тег head

Синтаксис:

```
<head>...</head>
```

Тег `<head>` (элемент заголовка) предназначен для содержания в нем информации о документе, а именно:

- название документа (тег TITLE)
- полный URL документа (тег BASE)
- управляющую информацию (тег META)

- список ссылок (тег LINK)
- описание стилей (тег STYLE)
- задание скриптов (тег SCRIPT)

Тег HEAD имеет один необязательный атрибут *profile*, указывающий на внешний файл мета-элементов. В качестве значения атрибута задается адрес URL. Задание тега HEAD в общем виде выглядит так:

```
<HEAD profile="URL">
    ...список элементов заголовка...
    ...список элементов заголовка...
    ...список элементов заголовка...

</HEAD>
```

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<head>
  <meta charset="utf-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1">
  <title>Книги из магазина</title>
  <link rel="stylesheet" href="plugins/
fontawesome-free/css/all.min.css">
  <link rel="stylesheet" href="dist/css/
adminlte.min.css">
  <style>h1 {color: red; }</style>
</head>
```

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

# Ter header

Синтаксис:

```
<header>...</header>
```

Своими открывающим и закрывающим тегами определяет заголовочную часть HTML-документа. Обычно в этой части (в качестве содержимого тега HEADER) располагаются теги навигации, изображение логотипа, заголовок с названием сайта и т.п. Внутри содержимого могут быть любые теги за исключением тегов HEADER и FOOTER.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<html>
  <head>
    <title> Название документа </
title>
  </head>
  <body>
    <header>
      Шапка сайта
    </header>
    <nav>
      Навигация
      <ul>
        <li><a href="#">Ссылка 1</a></li>
        <li><a href="#">Ссылка 2</a></li>
      </ul>
    </nav>
  ...
```

# Тег hgroup

Синтаксис:

```
<hgroup>...</hgroup>
```

Используется для группирования заголовков веб-страницы или раздела. Внутри располагаются теги заголовков от <h1> до <h6>. Данный тег помечен, как подлежащий удалению из HTML5, поэтому его использование не рекомендуется.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<hgroup>
    <h1>Заголовок 1</h1>
    <h2>Заголовок 2</h2>
</hgroup>
```

# Тег hr

Синтаксис:

```
<hr>
```

Используется для отображения горизонтальной линии, как правило, такая линия используется для разделения различных блоков текста. В последнее время данный тег используется все

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

реже, но, тем не менее, он поддерживается современными браузерами.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque et arcu metus. Quisque semper dignissim felis. Nullam sed commodo lorem, at bibendum arcu.
```

```
<hr>
```

```
<p>Nullam sed commodo lorem, at bibendum arcu. Maecenas tincidunt vehicula velit et euismod. Donec iaculis viverra consectetur.
```

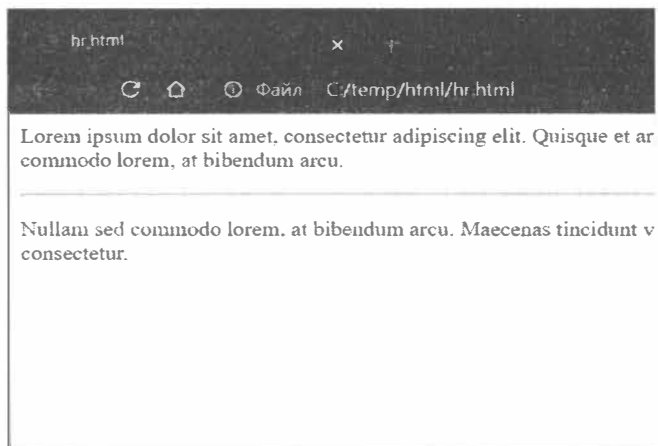


Рис. 17. Тег `hr`

# Тег html

Синтаксис:

```
<html>...</html>
```

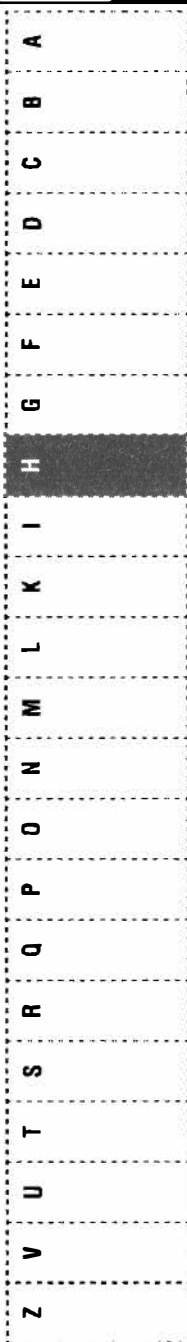
Заголовок и тело документа заключены в тег-контейнер html. Все остальные теги разметки располагаются либо в заголовке, либо теле документа. Данный тег является корневым, то есть все остальные теги документа вкладываются в этот тег.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<html>
    <head>
        <title> Название документа </title>
    </head>
    <body>
        Здесь располагается текст страницы
    </body>
</html>
```



## Тег i

Синтаксис:

```
<i>...</i>
```

Устанавливает курсивное начертание шрифта. Данный тег можно использовать совместно с другими тегами, которые определяют начертание текста.

Атрибуты:

- Глобальные атрибуты HTML.

Пример:

```
<i>Этот текст будет курсивным</i>
```

## Тег iframe

Синтаксис:

```
<iframe>...</iframe>
```

Создает плавающий фрейм, позволяющий загружать в область заданных размеров другие независимые документы. Данный тег является контейнером, содержание которого игнорируется браузерами, которые не поддерживают данный тег. Для таких браузеров можно указать

текст, который увидят пользователи. Текст должен располагаться между тегами `<iframe>` и `</iframe>`.

Атрибуты:

- **allowfullscreen** – объект, находящийся во фрейме, может использовать запрос на переход в полноэкранный режим.
- **height** = "число" – указывает высоту встроенного фрейма.
- **name** = "имя" – задает имя фрейма, которое можно использовать для ссылки на него с помощью целевых ссылок.
- **sandbox** = "allow-forms / allow-pointer-lock / allow-popups / allow-same-origin / allow-scripts / allow-top-navigation" – используется для активизации скриптов, всплывающих окон, подключаемых модулей во внедренных документах.
- **seamless** – при использовании этого атрибута браузер будет трактовать внедренный документ так, как будто бы он является частью родительского документа для целевых ссылок, структуры документа и CSS. Наиболее частое применение этого атрибута – применения стилей CSS к внедренному документу.
- **src** = "URL-ссылка" – задает адрес документа, который будет внедрен во фрейм `iframe`.
- **width** = "число" – задает ширину фрейма.

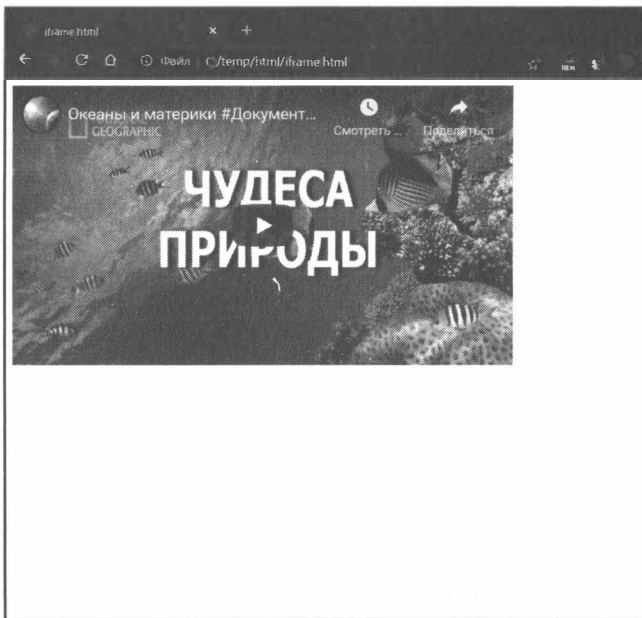
A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z



А  
В  
С  
Д  
Е  
Ф  
Г  
Н  
И  
К  
Л  
М  
Н  
О  
Р  
Q  
R  
S  
Т  
У  
V  
Z

Пример:

```
<iframe width="560" height="315"
src="https://www.youtube.com/
embed/PnaXw_qUg3A" frameborder="0"
allow="accelerometer; autoplay;
clipboard-write; encrypted-media;
gyroscope; picture-in-picture"
allowfullscreen></iframe>
```



**Рис. 18. Видео встроено в страницу**

- Глобальные атрибуты HTML

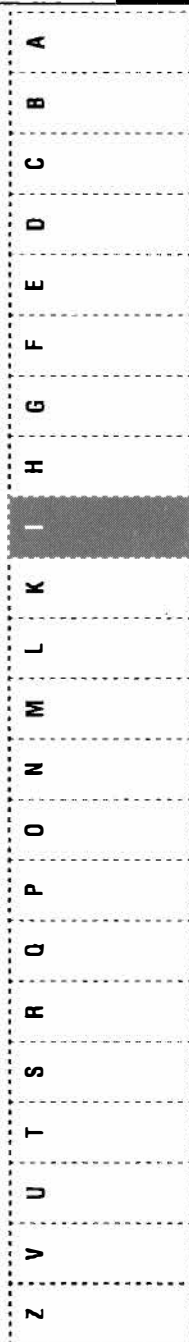
# Тег `img`

Синтаксис:

```
<img>
```

Внедрение иллюстративных изображений в HTML осуществляется посредством тега `IMG`, имеющего следующие атрибуты:

- **src** = "адрес картинки" — обязательный атрибут, задающий URL-адрес (полный или относительный) расположения изображения
- **width** = "число", **height** = "число" — геометрические размеры изображения, задаваемые либо в пикселах, либо в процентах видимого пространства окна браузера
- **alt** = "альтернативный текст" — атрибут, имеющий в качестве значения текстовую строку, которая является альтернативной текстовой информацией текущего изображения.
- **usemap** = "карта" — использовать изображение совместно с клиентской навигационной картой, имя которой указывается в качестве значения этого атрибута.
- **ismap** — использовать изображение совместно с серверной навигационной картой, имя которой указывается в качестве значения этого атрибута.
- Глобальные атрибуты HTML.



А  
В  
С  
Д  
Е  
Ф  
Г  
И  
К  
Л  
М  
Н  
О  
Р  
Д  
В  
С  
Т  
У  
А  
З

Тег IMG используется только с открывающим тегом.

Пример:

```

```

Одна из частых задач – не просто вставка изображения, а вставка слайдера (карусели). Используя Bootstrap, вы можете создать слайдер вовсе без знания JavaScript и CSS. Все, что вам нужно – это использовать стандартные стили Bootstrap. Рассмотрим пример:

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <title>Карусель изображений</title>
  <meta charset="utf-8">
    <meta name="viewport"
content="width=device-width, initial-
scale=1">
  <link rel="stylesheet" href="https://
maxcdn.bootstrapcdn.com/bootstrap/3.4.1/
css/bootstrap.min.css">
  <script src="https://ajax.googleapis.
com/ajax/libs/jquery/3.5.1/jquery.min.
js"></script>
    <script src="https://maxcdn.
bootstrapcdn.com/bootstrap/3.4.1/js/
bootstrap.min.js"></script>
</head>
<body>
```

```

<div class="container">
  <h2>Карусель изображений</h2>
  <div id="myCarousel" class="carousel
slide" data-ride="carousel">
    <!-- Индикаторы (кнопки внизу) -->
    <ol class="carousel-indicators">
      <li data-target="#myCarousel" data-
slide-to="0" class="active"></li>
      <li data-target="#myCarousel" data-
slide-to="1"></li>
      <li data-target="#myCarousel" data-
slide-to="2"></li>
    </ol>

    <!-- Обертки для слайдов -->
    <div class="carousel-inner">
      <div class="item active">
        
      </div>

      <div class="item">
        
      </div>

      <div class="item">
        
      </div>
    </div>

    <!-- Кнопки Назад и Вперед -->
    <a class="left carousel-control"
href="#myCarousel" data-slide="prev">
      <span class="glyphicon glyphicon-

```

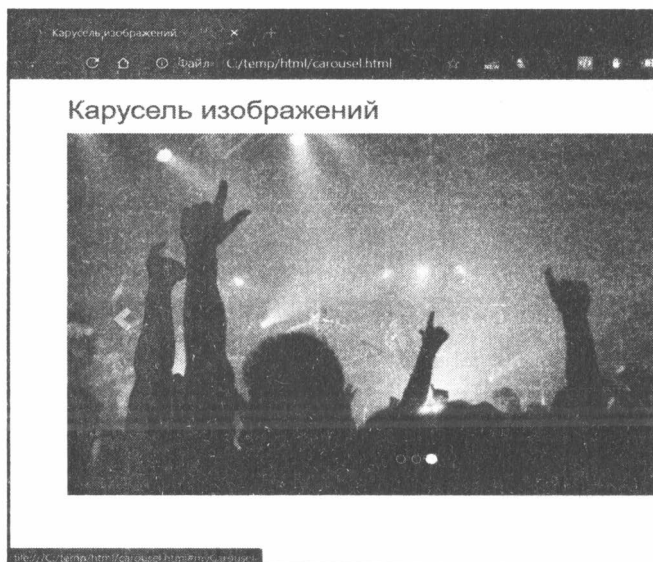
A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

```

chevron-left"></span>
    <span class="sr-only">Назад</span>
  </a>
  <a class="right carousel-control"
href="#myCarousel" data-slide="next">
    <span class="glyphicon glyphicon-
chevron-right"></span>
    <span class="sr-only">Вперед</span>
  </a>
</div>
</div>

</body>
</html>

```



**Рис. 19. Слайдер с использованием Bootstrap**

# Тег input

Синтаксис:

```
<input>...</input>
```

Данный тег является одним из самых универсальных элементов формы и позволяет создавать различные элементы интерфейса и обеспечивать взаимодействие с пользователем. С помощью этого тега можно создать текстовые поля, кнопки, переключатели и флажки. Данный тег не требует обязательного помещения в контейнер **form**, но, как правило, он редко используется по отдельности, поскольку введенные данные в большинстве случаев отправляются на сервер, а для этого нужен тег **form**.

У этого тега очень много атрибутов. Рассмотрим их:

- **type** = "тип" — основной атрибут тега **input**, который задает тип элемента формы:

*text* — текстовое поле.

*password* — поле с паролем.

*radio* — переключатель.

*checkbox* — флажок.

*button* — кнопка.

*submit* — кнопка для отправки формы.

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

	A
	B
	C
	D
	E
	F
	G
	H
	I
	J
	K
	L
	M
	N
	O
	P
	Q
	R
	S
	T
	U
	V
	W
	X
	Y
	Z

*reset* – кнопка очистки формы.

*file* – поле для отправки файла.

*image* – кнопка с изображением.

*color* – средство выбора цвета.

*date* – средство выбора даты.

*time* – средство выбора даты.

*date-time* – средство выбора даты и времени.

*email* – поле для ввода e-mail.

*tel* – поле для ввода телефона.

*url* – поле для ввода url.

*search* – поисковое поле.

*month* – поле выбора месяца.

*week* – поле выбора недели.

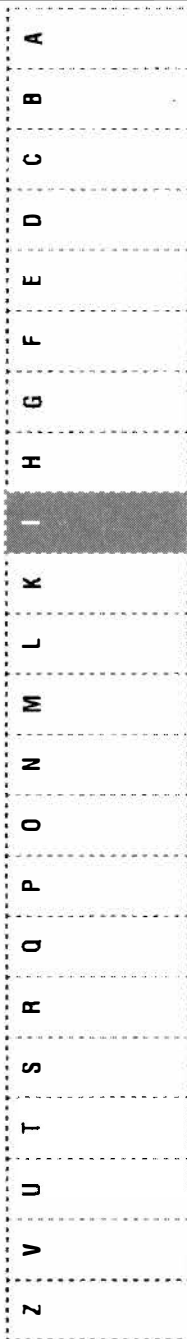
*number* – поле ввода целого числа.

*hidden* – скрытое поле.

- **accept** = "тип **MIME**" – указывает список (разделенных запятыми) типов контента, которые будут корректно восприниматься сервером, который выполняет обработку данных формы. Используется для фильтрации несовместимых файлов при отображении запроса пользователю, в котором предлагается выбрать файлы для

отправки на сервер. Работает только с типом *file*.

- **alt** = "текст" — альтернативный текст для кнопки с изображением. Только для типа ввода *image*.
- **autocomplete** = "on/off" — позволяет браузеру заполнить поля формы автоматически (on) или требует, чтобы пользователь каждый раз сам вводил информацию (off).
- **autofocus** — устанавливает фокус в поле формы.
- **border** — толщина рамки вокруг изображения.
- **checked** — предварительно активированный переключатель или флажок.
- **dirname** = "ltr/rtl" — задает направление текста элемента.
- **disabled** — блокирует доступ и изменение элемента.
- **form** — связывает поле с формой по ее идентификатору.
- **formaction** = "URL-ссылка" — указывает приложение, обрабатывающее форму. Выполняет те же функции, что и атрибут *action* для тега **form**.

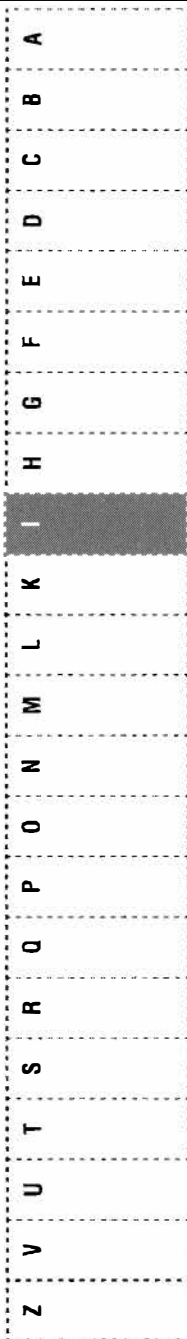




	A
	B
	C
	D
	E
	F
	G
	H
	I
	J
	K
	L
	M
	N
	O
	P
	Q
	R
	S
	T
	U
	V
	W
	X
	Y
	Z

- **formenctype** = "тип контента" — задает способ кодирования данных формы. При значении `application/x-www-form-urlencoded` вместо пробелов ставится `+`, символы вроде русских букв кодируются их шестнадцатеричными значениями (например, `%D0%9F%D0%B5%D1%82%D1%8F` вместо Петя). Значение `multipart/form-data` означает, что данные не кодируются. Это значение используется при отправке файла. Значение `text/plain` означает, что пробелы заменяются знаком `+`, а остальные символы не кодируются.
- **formmethod** = "метод отправки формы" — указывает метод пересылки данных формы. Допустимые значения `GET` и `POST`, аналогично как и для атрибута `method` тега `form`.
- **formnovalidate** — отменяет проверку данных, введенных пользователем в форме на корректность.
- **formtarget** = "имя" — указывает целевое окно, в котором отображаются результаты передачи данных формы. Выполняет ту же функцию, что и атрибут `target` для тега `form`.
- **height** = "высота" — указывает высоту изображения кнопки. Только для типа `image`.
- **list** = "идентификатор типа данных" — указывает на список вариантов, которые можно выбрать при вводе текста.

- **max** = "число или строка" — верхнее значение для ввода числа или даты. Значение *max* не должно быть меньше значения *min*.
- **maxlength** = "число" — максимальное количество символов разрешенных в тексте. Применяется для типов `text`, `password`, `email`, `search`, `tel`, `url`.
- **min** = "число или строка" — нижнее значение для ввода числа или даты. Не должно быть больше, чем *max*.
- **multiple** — позволяет загрузить несколько файлов одновременно.
- **name** = "текст" — имя поля, предназначено для того, чтобы обработчик формы мог его идентифицировать.
- **pattern** = "регулярное выражение JS" — устанавливает шаблон ввода. Применяется к типам ввода данных `text`, `password`, `email`, `tel`, `search`, `url`.
- **placeholder** = "текст" — выводит подсказывающий текст.
- **readonly** — устанавливает, что поле не может изменяться пользователем. Используется для типов `text`, `password`, `email`, `tel`, `search`, `url`, `number`, `date`, `time`, `datetime`, `month`, `week`.
- **required** — обязательное для заполнения поле. Пользователю не получится от-



править форму, если не заполнены все *required*-поля.

- **size** = "число" — ширина текстового поля.
- **src** = "URL-ссылка" — адрес графического файла для поля с изображением. Если в качестве типа ввода данных используется *image*, то данный атрибут указывает на местоположения изображения, используемого в качестве кнопки.
- **step** = "any/число" — шаг приращения для числовых полей.
- **value** = "значение" — значение элемента. Применяется ко всем типам ввода, кроме *image*.
- **width** = "ширина" — ширина поля ввода в пикселях.
- Глобальные атрибуты HTML

Рассмотрим примеры применения этого тега.

## Кнопки: **button**, **reset**, **submit**

Для создания кнопки используется тип **button**. Создается кнопка, которую можно нажать. При этом поведение кнопки не определяется тегом. При использовании кнопки разработчик сам должен запрограммировать ее действие с помощью обработчика события *onclick*.

```
<input type="button" name="simpleBtn"
onclick="Show()" value="Просто кнопка">
```

Тип **submit** создает кнопку отправки форму, а тип **reset** – кнопку сброса формы к значениям по умолчанию. Внешне все эти кнопки выглядят одинаково:

```
<input type="button" name="simpleBtn"
value="Просто кнопка">
<p><input type="reset" name="resetBtn"
value="Сброс">
<p><input type="submit" name="submitBtn"
value="Отправить">
```



Рис. 20. Кнопки

Атрибут *value* задает текст кнопки. При этом ширина кнопки подстраивается под введенный текст. Высота кнопки выбирается браузером автоматически. Для изменения внешнего вида кнопок принято использовать CSS.

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

Рассмотрим пример установки обработчика кнопки:

```
<h1 id="text">Текст, который будем
скрывать и показывать</h1>
<p><input type="button"
onclick="document.
getElementById('text').hidden = true;"
value="Скрыть">
<p><input type="button"
onclick="document.
getElementById('text').hidden = false;"
value="Показать">
```

В данном примере у нас есть элемент с `id = "text"`. Обработчик кнопки Скрыть устанавливает свойство *hidden* этого элемента в *true*, чем скрывает его. Аналогично, обработчик кнопки Показать устанавливает это свойство в *false* и элемент снова отображается.

## Независимые переключатели: checkbox

Тип ввода *checkbox* создает флажок в области действия тега **form**. Флажок еще называется независимым переключателем, поскольку его состояние (вкл. или выкл.) не зависит от других аналогичных элементов.

Для независимых переключателей часто используют атрибуты:

*autofocus* – фокус ввода будет по умолчанию на этом элементе.

*checked* – флажок будет по умолчанию в состоянии "Включено".

*disabled* – элемент ввода отключен и не будет использоваться при отправке формы.

*name* – обязательный атрибут при передаче данных на сервер, иначе не будет понятно, какой переключатель был включен.

*required* – обязательный переключатель (без включения отправка формы невозможна).

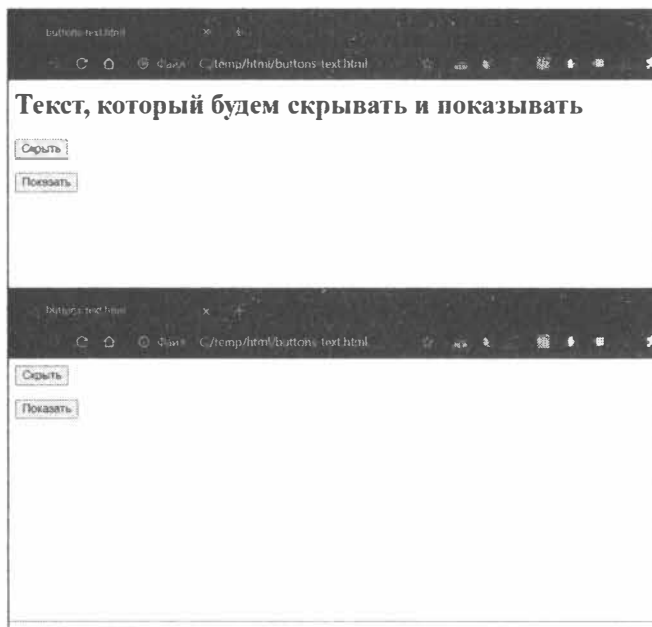


Рис. 21. Включение/выключение отображения элемента

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

Пример:

```
<p>Выберите количество комнат:
<input type="checkbox" name="rooms[]"
value="1" checked>1
<input type="checkbox" name="rooms[]"
value="2">2
<input type="checkbox" name="rooms[]"
value="3">3
<input type="checkbox" name="rooms[]"
value="4">4
```



**Рис. 22. Независимые переключатели**

Первый переключатель включен по умолчанию, а все остальные выключены. Здесь мы использовали одну хитрость. Хотя выключатели и независимые и могут включаться/выключаться независимо друг от друга, мы для каждого из них указали одно и то же имя – `rooms[]`. Сделано это было умышлено, чтобы каждый переключатель был передан на сервер под именем `rooms`. Квадратные скобки нужны, чтобы сценарий на сер-

вере работал с переменной `rooms` как с массивом, что более удобно, чем разбирать вручную четыре разных переменных – `room1`, `room2`, `room3`, `room4`. Атрибут *value* задает значение, которое будет передано на сервер, если данный флажок выбран. Если пользователь выбрал 2 и 3 комнатные квартиры, то массив на сервере будет выглядеть так:

```
$rooms = array("2", "3");
```

### Выбор цвета: color

Тип ввода *color* отображает средство выбора цвета (рис. 23). Атрибут *value* задает начальное значение цвета.

```
<input type="color" name= "background" value = "#DDDDDD">
```



Рис. 23. Средство выбора цвета

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z



## Выбор даты и времени: **date**, **datetime**, **datetime-local**, **time**, **month**, **week**

Для ввода даты и времени предусмотрено целых 6 типов ввода:

**date** – выбор даты (используется чаще всего).

**datetime** – выбор даты и времени.

**datetime-local** – выбор даты и времени в локальном часовом поясе.

**time** – выбор времени.

**month** – выбор месяца.

**week** – выбор недели.

Рассмотрим примеры таких полей ввода:

```
<p><input type="date" name="date" value =  
"2021-02-13">  
<p><input type="datetime" name="datetime"  
value = "2021-02-13T07:34:00+2:00">  
<p><input type="datetime-local"  
name="dtl" value = "2021-02-13T07:34:00">  
<p><input type="month" name="month" value  
= "2021-07">  
<p><input type="time" name="time" value  
= "07:34:00">  
<p><input type="week" name="week" value  
= "2021-W10">
```

Основной атрибут – *value*. Он задает начальное значение даты/времени. Для **datetime** значение даты указывается по стандарту ISO. Дата и время в этом случае передаются как время UTC. Для типа ввода **datetime-local** время указывается в местном часовом поясе. Внешний вид самих элементов ввода и диалогов, которые будут открываться для выбора значений, зависит от браузера. Далее все иллюстрации сделаны в браузере Chrome.

Как видите, почему-то Chrome не отобразил кнопку открытия диалога для типа **datetime**. Даже если удалить строку *value* (что исключает неправильность установки значения даты), то он отображает данный элемент все равно, как обычное текстовое поле.

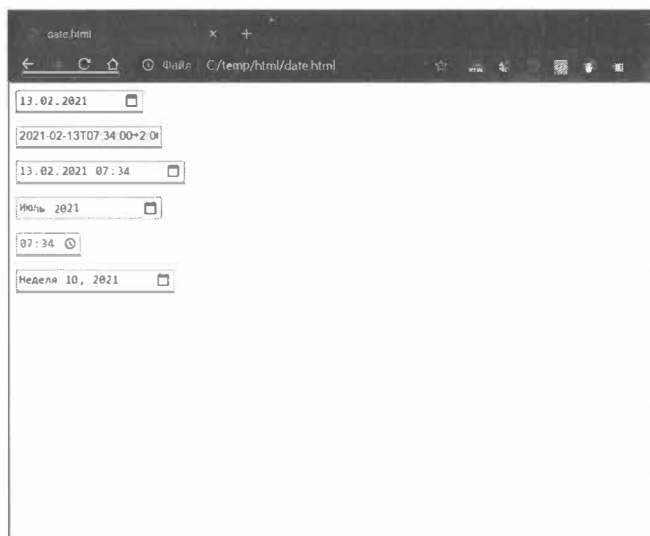
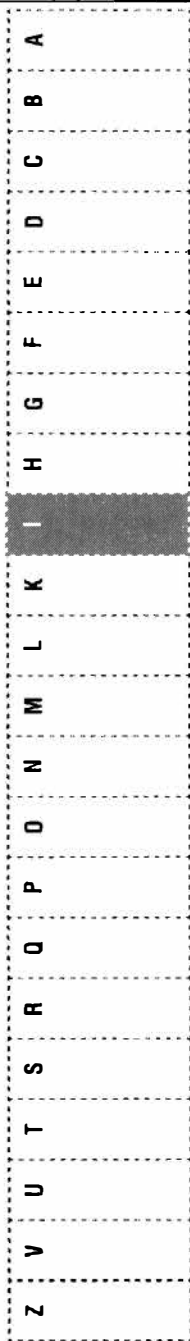
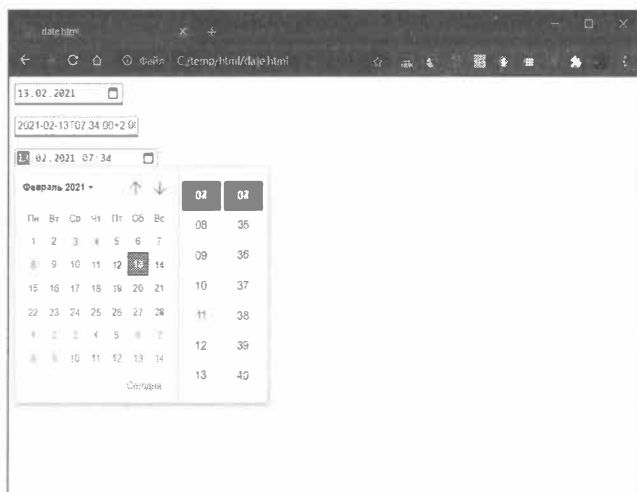


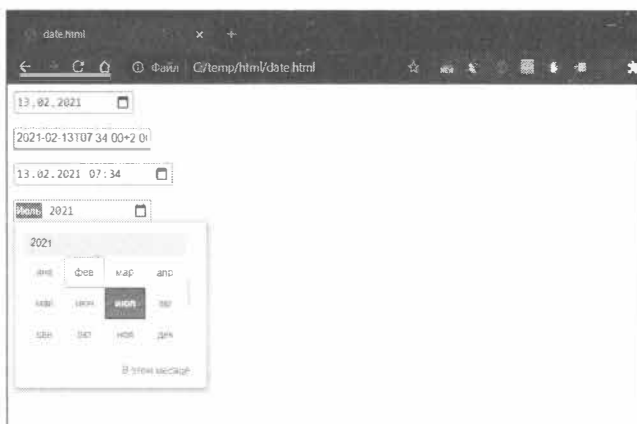
Рис. 24. Начальная форма данных





**Рис. 25. Диалог выбора локальной даты и времени**

Напомним, что для полей даты и времени вы можете задавать атрибуты *min* и *max* – так вы можете ограничивать минимальные и максимальные временные значения.



**Рис. 26. Выбор месяца**

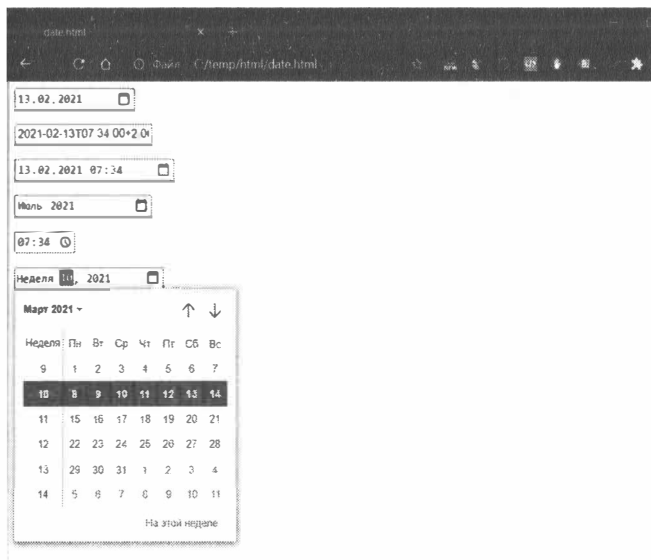


Рис. 27. Выбор недели

## Текстовые поля: email, text, password, search

Все текстовые поля выглядят одинаково. Исключение составляет лишь поле *password*, где вводимые символы заменяются на точки или звездочки (зависит от браузера):

```
<p><input type="text" name="text"
placeholder="Введите текст">
<p><input type="email" name="email"
placeholder="Введите e-mail">
<p><input type="password" name="pswd"
placeholder="Введите пароль">
<p><input type="search" name="s"
placeholder="Поиск...">
```

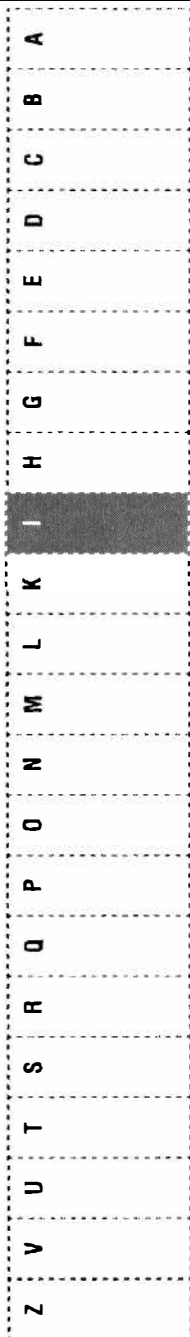




Рис. 28. Текстовые поля

Поговорим об атрибутах этих полей:

**placeholder** = "текст" – позволяет оставить подсказку для пользователя. В отличие от атрибута *value*, данное значение не передается на сервер при отправке формы.

**value** = "значение" – начальное значение поля, будет передано на сервер при передаче формы.

**minlength** = "число", **maxlength** = "число" – позволяет указать минимальную и максимальную длину значения (в знаках). Эти атрибуты удобно использовать для ограничения длины паролей. Как правило, в большинстве случаев пароль должен быть не менее 6 символов, поэтому можно указать `minlength="6"` для поля пароля.

**size** = "число" – позволяет регулировать размер поля, устанавливает ширину в знаках-местах.

**required** – пока не заполнены все *required*-поля, форма не будет передана на сервер.

**name** = "имя" – имя элемента, которое будет передано на сервер.

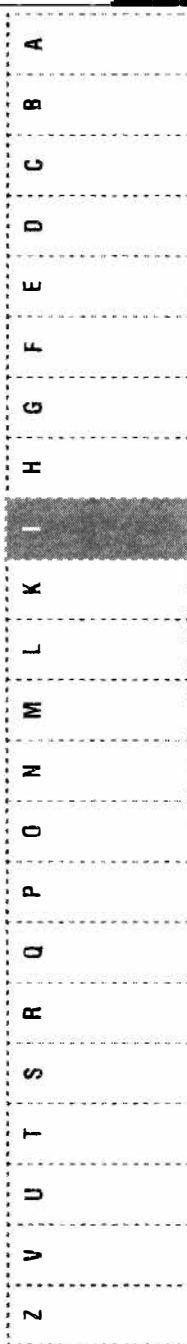
## Ввод телефона: tel

По сути, поле для ввода телефона – это обычное текстовое поле для ввода номера телефона. По сути, оно мало чем отличается от текстового поля, просто позволяет указать браузеру, что данное поле предназначено для ввода номера телефона. Важно, что передать значение на сервер у вас не получится до тех пор, пока значение поля не будет похоже на номер телефона.

```
<input type="tel" name="phone" value="000-000-0000">
```

## Ввод URL: url

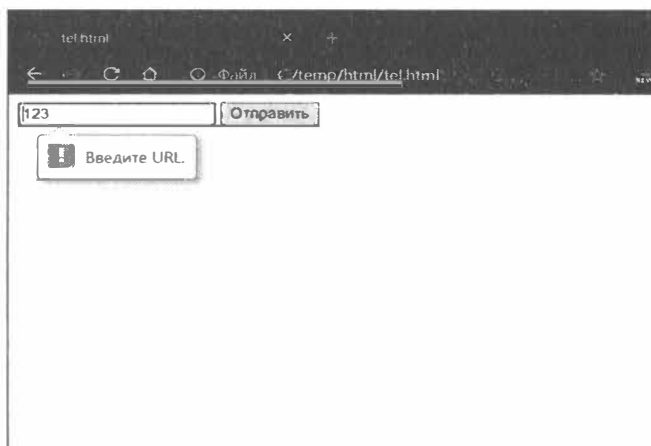
Такая ситуация и для поля типа **url**. Поле отображается как обычное текстовое поле и его функционал никак не препятствует вводу значений, отличных от URL. Зато отправить форму на сервер у вас не получится, пока данное поле не будет содержать URL-адрес:



```

<form      action="http://nit.center/test.
php">
    <input      type="url"      name="url"
value="123">
    <input      type="submit"   name="url"
value="Отправить">
</form>

```



**Рис. 29.** Форма не отправится, пока вы не введете URL

## Зависимые переключатели: radio

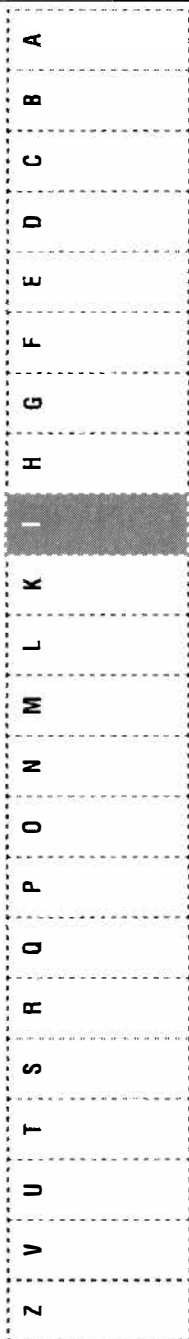
Зависимые переключатели отличаются от независимых тем, что в одной группе зависимых переключателей (их еще называют радио-кнопками) может быть включен только один из переключателей.

Рассмотрим пример, аналогичный независимым переключателям:

```
<p>Выберите количество комнат:
<input type="radio" name="rooms"
value="1" checked>1
<input type="radio" name="rooms"
value="2">2
<input type="radio" name="rooms"
value="3">3
<input type="radio" name="rooms"
value="4">4
```



Рис. 30. Зависимые переключатели





## Ввод числовых значений: number, range

Данные типы ввода предназначены для ввода числовых значений:

```
<p><input type="number" min="0" max="100"  
name="num" value="30">
```

```
<p><input type="range" min="0" max="100"  
name="ran" value="30">
```

Атрибуты *min* и *max* задают минимальные и максимальные значения. Разница между этими полями в том, что типа **number** создает поле со счетчиком (стрелками вверх/вниз), а поле **range** – формирует ползунок, позволяющий выбирать значение.

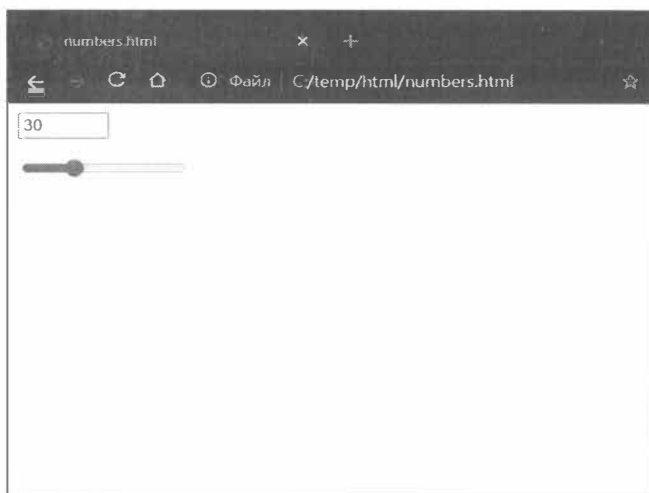


Рис. 31. Поля для ввода числовых значений

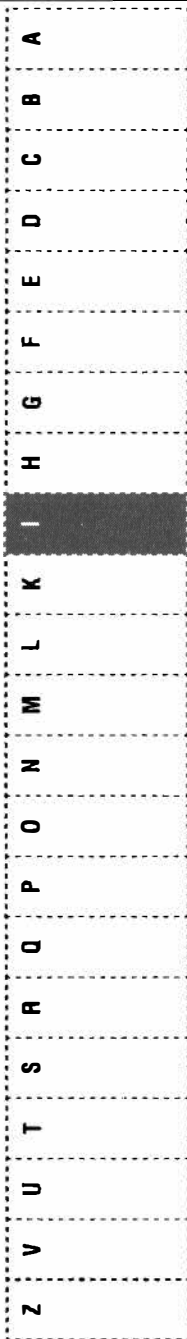
## Выбор файла: file

Поле для выбора файла, который должен быть загружен на сервер. Важно при этом отключить MIME-кодирование формы данных, установив значение `multipart/form-data` для атрибута `enctype`:

```
<form method="post" action="upload.php"
enctype="multipart/form-data">
  <p>Выберите файл для отправки на
сервер:<br>
      <input type="file"
name="picture" size="28">
  </p>
</form>
```

Если вам интересно, что будет дальше, приводим код PHP-сценария `upload.php`, который и выполняет загрузку на сервер картинки:

```
<?php
// Имя файла на сервере
$uploadfile = "images/test.jpg";
// Обратите внимание: название элемента
массива $_FILES должно совпадать
// с именем поля file в форме. В нашем
случае — picture
if (copy($_FILES['picture']['tmp_name'],
$uploadfile)) {
    // В случае успешной загрузки картинки
перенаправляем пользователя
    // на страницу test.html, где он может
просмотреть загруженную картинку
    header('Location: test.html');
}
```

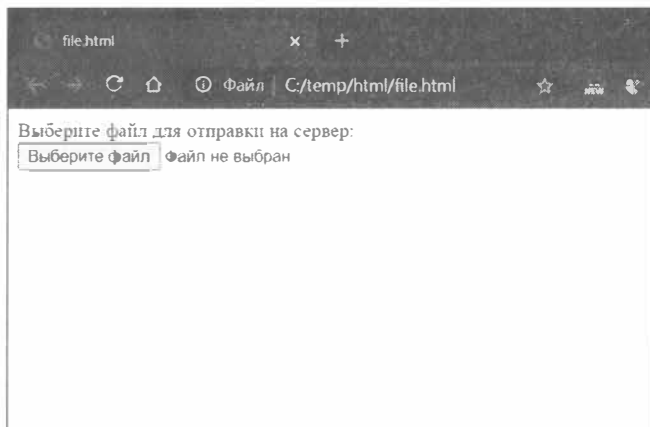


```

else
    die('Невозможно загрузить картинку.
    Обратитесь к разработчику');

```

```
?>
```



*Рис. 32. Поле для выбора файла*

## Скрытый элемент: hidden

Используется для передачи на сервер текстовых значений, скрытых от пользователя. Пользователь их не видит и не может редактировать. Применение такого типа следующее: сценарий формирует страницу редактирования статьи, а скрытое поле при этом содержит ID статьи (чтобы сценарий сохранения статьи знал, в какую статью нужно внести изменения):

```



```

## Тип image

Позволяет создать применить изображение в качестве заменителя кнопки *submit*. Данное поле позволяет нарисовать красивую произвольную картинку и использовать ее в качестве кнопки *submit*, то есть кнопки отправки формы:

```
<input type="image" src="img/button.gif"
alt="Отправить">
```

## Тег ins

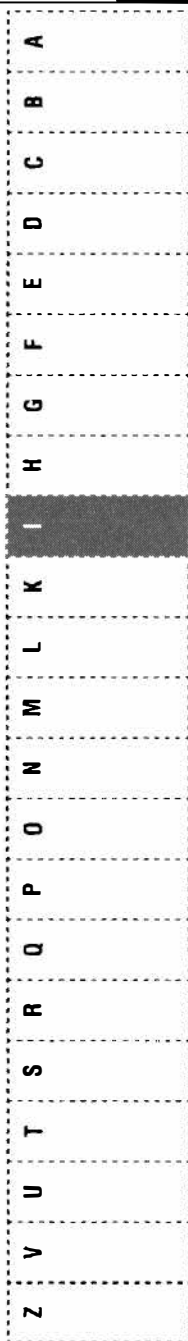
Синтаксис:

```
<ins>...</ins>
```

Предназначен для выделения текста, который был добавлен в новую версию документа (вставка нового текста). Подобное форматирование позволяет отследить, какие изменения в тексте документа были сделаны. Данный элемент можно использовать в паре с тегом **del**, который помечает текст, помеченный для удаления. Тег **ins** используется для выделения встроенных или блочных элементов.

Атрибуты:

- **cite** = "URL-ссылки" – используется для создания ссылки на исходный документ, в котором указана причина изменения.



- **datetime** = "ГГГГ-ММ-ДДТчч:мм:ссTZD"  
– позволяет указать дату и время изменения.
- Глобальные атрибуты HTML

Пример:

```
<del>Этот текст удален в старой версии
документа</del><ins> А этот вставлен!</
ins>
```

## Ter kbd

Синтаксис:

```
<kbd>...</kbd>
```

Название данного тега представляет собой сокращение от слова "keyboard" (клавиатура). Указывает текст, задаваемый пользователем.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<p>Введите код скидки. Пример: <kbd>111-
AX-GG</kbd>
```

# Тег keygen

Синтаксис:

<keygen>

Данный тег используется для генерации пары ключей — закрытого и открытого. Когда форма отправляется на сервер, закрытый ключ сохраняется на локальном компьютере, а открытый ключ передается вместе с формой. Сами ключи необходимы для шифрования и расшифровки данных, создания и проверки цифровой подписи.

Атрибуты:

- **challenge** = "строка вызова" — поддерживает строку вызова, которая будет передаваться вместе с ключом.
- **disabled** — блокирует элемент.
- **keytype** = "ключевое слово" — позволяет указать тип генерируемого ключа, например, `rsa`
- **name** = "имя" — задает имя, которое будет передано на сервер.
- Глобальные атрибуты HTML

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

Пример:

```
<form method="post" action="chiper.php">
  <keygen name="public"
challenge="112233445566778899"
keytype="RSA">
  <input type="submit"
value="Отправить">
</form>
```

## Тег label

Синтаксис:

```
<label>...</label>
```

Используется для связывания информации с различными тегами управления. Один элемент **label** ("Надпись") связывается только с одним элементом формы. Атрибут *for* указывает на элемент, с которым связана надпись.

Атрибуты:

- **for = "id"** – задает идентификатор элемента ввода, с которым связан данная надпись.
- Глобальные атрибуты HTML

Примеры:

```
<label>Имя пользователя: <input
type="text" name="login"></label>
```

Здесь надпись связывается с текстовым полем, предназначенным для ввода логина.

А здесь мы явно задаем **id** элемента, с которым связывается надпись:

```
<label for="username">Имя пользователя:
</label>
<input type="text" id="username"
name="login">
```

## Тег legend

Синтаксис:

```
<fieldset>
  <legend>Текст</legend>
</fieldset>
```

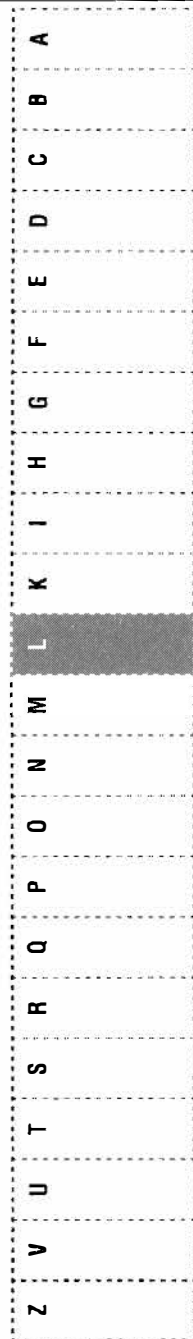
Данный тег применяется для создания заголовка группы элементов формы, которая определяется с помощью тега `<fieldset>`. Группа элементов обозначается в браузере с помощью рамки, а текст, который располагается внутри контейнера `<legend>`, встраивается в эту рамку.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<fieldset>
```

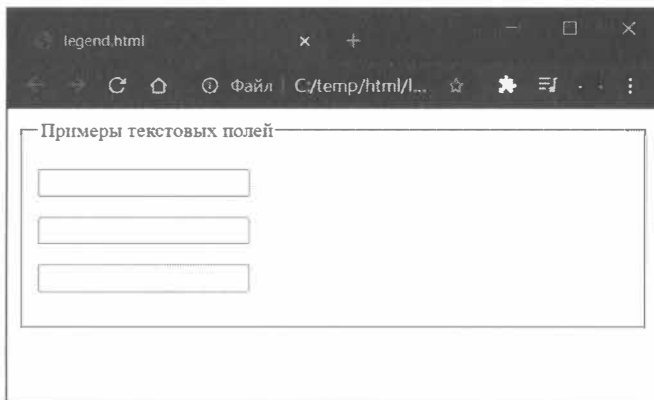




```

<legend>Примеры текстовых полей</
legend>
  <p><input type="text" name="login">
  <p><input          type="password"
name="pswd">
  <p><input          type="email"
name="email">
</fieldset>

```



**Рис. 33. Пример использования тега legend**

## Тег li

Синтаксис:

```
<li>...</li>
```

Данный тег используется внутри списков – нумерованного (ol) и маркированного (ul).

Атрибуты:

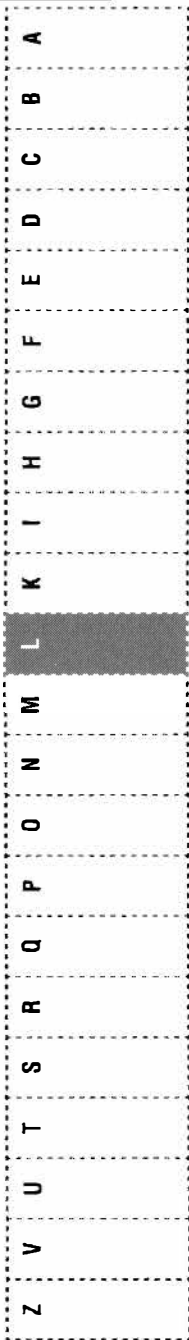
- **value** = "число" — указывает номер элемента. Только для нумерованных списков (ol). При этом нумерация последующих элементов списка возрастает, начиная с указанного числа.
- Глобальные атрибуты HTML.

Пример:

```
<ol>  
  <li>Первый элемент списка</li>  
  <li>Второй элемент списка</li>  
  <li>Третий элемент списка</li>  
</ol>
```



*Рис. 34. Нумерованный список*



## Тег link

Синтаксис:

```
<link>
```

Данный тег указывает на связь, отношение между содержащим его документом и другим ресурсом сети. На данный момент единственное его практическое применение заключается в подключении внешних каскадных таблиц стилей (CSS), ответственных за визуальное представление документа.

Информация о документе, предоставляемая элементом LINK может быть использована некоторыми поисковыми машинами для оптимизации поиска.

Атрибуты:

- **href** = "URL-ссылка" – указывает адрес связанного документа.
- **hreflang** = "код языка" – задает основной язык для связанного документа.
- **media** = "all / screen / print / handheld / projection / tty / tv / projection / braille / aural" [HTML5] – указывает мультимедийный объект, по отношению к которому применяется связанный ресурс. Как правило, тег используется для сопоставления таблиц стилей с соответствующим мультимедийным объектом.

- **rel** = "ключевое слово типа ссылки" – описывает связи между исходным и связанным документами. Для элемента **link** применяются следующие типы ссылочных связей – *alternate*, *author*, *help*, *icon*, *license*, *next*, *prefetch*, *prev*, *search*, *stylesheet* и *tag*.
- **size** = "any / два пиксельных размера" – указывает размеры пиктограмм, если атрибуту *rel* элемента **link** присвоено значение *icon*.
- **type** = "ресурс" – указывает мультимедийный тип или тип контента для связанного ресурса. Значение *text/css* указывает, что связанный документ является внешней таблицей стилей.
- Глобальные атрибуты HTML.

Пример:

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

В качестве значения атрибута *href* указывается путь к файлу стилей CSS.

## Тег main

Синтаксис:

```
<main>...</main>
```

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

А  
В  
С  
Д  
Е  
Ё  
З  
И  
К  
Л  
М  
Н  
О  
П  
Р  
С  
Т  
У  
Ф  
Х  
Ц  
Ч  
Ш  
Щ  
Ъ  
Ы  
Ь  
Э  
Ю  
Я

Используется для основного содержимого документа. Содержимое должно быть уникальным и не включать типовые блоки вроде шапки сайта, подвала, навигации, боковой панели, формы поиска и т. п.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<body>
<header>
    Это заголовок сайта
</header>
<main>
    <h1>Контент страницы</h1>
</main>
<footer>
    Подвал сайта
</footer>
</body>
```

## Тег map

Синтаксис:

```
<map>...</map>
```

Данный тег используется для создания карты изображений. Он служит контейнером для тегов **area**, которые определяют активные области для карты изображений. Теги **area** формируют невидимую область, которую можно использовать для навигации по сайту.

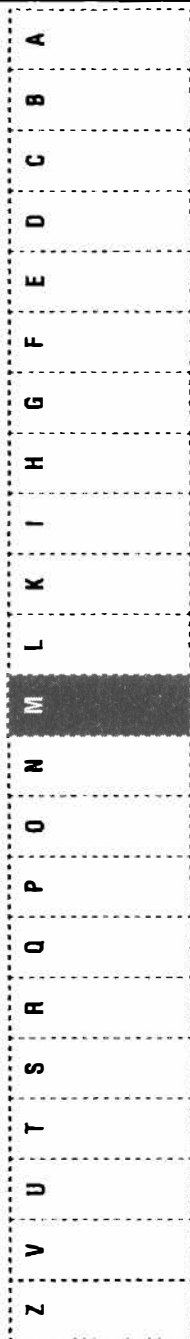
димые зоны на изображении, которые являются ссылками на HTML-документы. Цель использования тега **map** – в связывании тега **img** с картой-изображением.

Атрибуты:

- **id** = "ID" – присваивает идентификатор карте изображений, в результате на нее можно сослаться из скрипта JavaScript.
- **name** = "имя" – присваивает карте изображений имя, используемое для ссылки внутри элемента **img**.
- Глобальные атрибуты HTML.

Пример:

```
<map id="map" name="map">
  <area shape="circle"
coords="664,34,9" target="_parent"
title="Ростов-на-Дону" href="http://
example.com/map_reg.php?id=1" />
  <area shape="circle"
coords="154,334,9" target="_parent"
title="Симферополь" href="http://
example.com/map_reg.php?id=2" />
  <area shape="circle"
coords="599,322,9" target="_parent"
title="Краснодар" href="http://example.
com/map_reg.php?id=3" />
  <area shape="circle"
coords="667,506,9" target="_parent"
title="Сочи" href="http://example.com/
map_reg.php?id=4" />
...
</map>
```



## Тег mark

Синтаксис:

```
<mark>...</mark>
```

Данный тег помечает текст как выделенный. Такой текст ничем не отличается от обычного, но его вид может быть изменен с помощью стилей. Отмеченный текст рассматривается как наиболее важный для пользователя.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<article>
```

Съешь ещё этих мягких <mark>французских  
булок</mark>, да выпей чаю.

```
</article>
```

## Тег menu

Синтаксис:

```
<menu>...</menu>
```

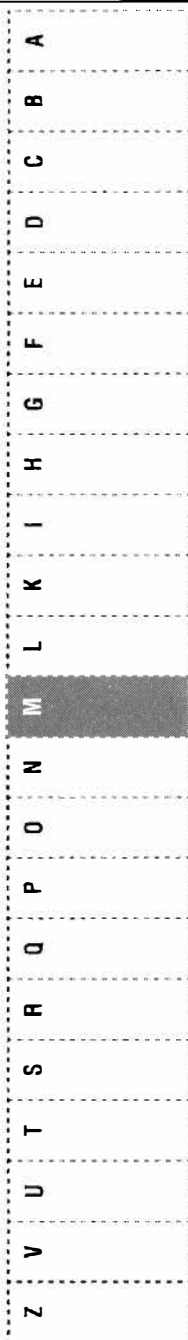
С данным тегом не все просто. До HTML4 он использовался для отображения списка пунктов

меню. Внутри контейнера **main** использовались теги **li**, подобно списку **ol**. В HTML4 данный тег вышел из употребления и вместо него рекомендовалось использовать маркированный список **ol**. В HTML5 тег **menu** снова стал актуальным, но уже в другом качестве. Теперь он выступает контейнером для тега `<command>` и создания меню.

Атрибуты:

- **label** = "текст" [HTML5] – указывает метку для меню, которая может отображаться во вложенных меню.
- **type** = "popup / toolbar" [HTML5] – идентифицирует тип меню, которое будет объявлено. При использовании **popup** мы получаем всплывающее меню или меню для элемента *button*, в качестве которого в форме выбрано значение **menu**. Значение *toolbar* подразумевает использование меню в качестве панели инструментов, с которой можно взаимодействовать.
- Глобальные атрибуты HTML.

```
<menu type="popup">
  <menuitem onclick="cut()"
label="Вырезать">
  <menuitem onclick="copy()"
label="Копировать">
  <menuitem onclick="paste()"
label="Вставить">
</menu>
```





# Ter menuitem

Синтаксис:

```
<menuitem>...</menuitem>
```

Данный тег поддерживается HTML5.1 и более новыми стандартами. Представляет команду, которая может быть выбрана из всплывающего меню.

Атрибуты:

- **checked** – указывает на то, что данная команда будет выбрана.
- **command** = "идентификатор мастер-команды" – ссылается на команду *menuitem*, которая была определена и установлена где-то в другом месте в документе или приложении.
- **default** – указывает на то, что данная команда будет командой по умолчанию.
- **disabled** – отключает команду.
- **icon** = "URL-ссылка" – указывает местоположение изображения, которое представляет команду.
- **label** = "подпись" – задает имя команды, показанной в меню.
- **type** = "command / checkbox / radio" – задает тип команды либо обычная ко-

манда вместе со связанным действием (command), переключатель (checkbox), или единственный параметр, выбираемый в списке (radio).

- Глобальные атрибуты HTML.

Пример:

```
<menu type="popup">
  <menuitem onclick="cut()"
  label="Вырезать">
  <menuitem onclick="copy()"
  label="Копировать">
  <menuitem onclick="paste()"
  label="Вставить">
</menu>
```

## Тег meta

Синтаксис:

```
<meta>
```

Тег META используется для задания некоторых свойств интернет-страницы (а именно: автора, списка ключевых слов, кодировки и т.п.), благодаря чему позволяет управлять обработкой HTML-документа. Этот тег, вместе с тегом TITLE является наиболее используемым при задании заголовка.

При задании META начальный тег обязателен, конечный тег запрещен.

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

## Атрибуты:

- **name** = "имя" – указывает имя свойства;
- **content** = "значение" – задает значение свойства;
- **scheme** = "схема" – указывает имя схемы, используемой для обработки значения свойства;
- **http-equiv** = "http-эквивалент" – используется вместо атрибута name для указания имени http-сообщения;
- **lang** = "язык" – информация о языке. Обязательный атрибут.
- **dir** = "направление" – указывает направление текста. Необязательный атрибут.
- Глобальные атрибуты HTML.

Каждый элемент META содержит в себе пару свойство-значение. Атрибут *name* (*http-equiv*) указывает свойство, атрибут *content* – значение.

## Например:

```
<META http-equiv=content-type content=
"text/html; charset=utf-8" >
<META name="description"
content="Обучение, управление проектами,
сертификация РМР">
```

```
<META name="keywords" content="PMP,
проект, управление проектами">
```

В первом случае указана кодировка HTML-документа, во втором случае – описание документа, в третьем – ключевые слова. Оба последних тега META обычно имеют одинаковое или близкое значение в целях повышения эффективности обнаружения их поисковыми роботами, так как именно по этим параметрам в значительной степени осуществляется отбор нужных документов.

Атрибут *http-equiv* может использоваться вместо атрибута *name*, для задания свойств HTML-документа на уровне http-заголовка. Через атрибут *http-equiv* осуществляется доступ к полям HTTP-заголовка. Первое полезное применение этого атрибута заключалось в осуществлении принудительной перезагрузки документа браузером. Для этого с помощью атрибута *http-equiv* используется http-оператор **refresh**. Время, через которое надо произвести перезагрузку указывается через атрибут *content*, а адрес загружаемого документа – атрибутом *url* оператора **refresh**.

Пример написания:

```
<META http-equiv="Refresh" content="5;
new_document.html">
```

При таком задании, через 5 секунд после загрузки текущего документа браузер автоматически перейдет к загрузке документа `new_document.html`.

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

Практически во всех HTML-документах тег META используется для их описания посредством задания списка ключевых слов и краткой информации о содержимом документа. Ключевые слова, вместе с названием документа, помогают поисковым машинам найти документ. В своих отчетах они выдают название документа, прописанное в элементе TITLE, и краткое его описание, заданное через элемент META.

Для указания списка ключевых слов и краткой информации о документе в заголовке используются два META элемента:

```
<META name="description"
content="Обучение, управление проектами,
сертификация PMP">
<META name="keywords" content="PMP,
проект, управление проектами">
```

Через тег META можно указать кодировку содержимого документа. Элемент META тогда принимает вид:

```
<META http-equiv="Content-
type" content="text/
html; charset=windows-1251">
```

С помощью тега META можно запретить кэширование документа, что бывает полезно при частом обновлении документа. Для осуществления этой операции в элемент META включается оператор cache-control (в HTTP 1.0 применялся оператор Pragma). Оператор устанавливается в

положение no-cache. META-элемент тогда примет вид:

```
<META http-equiv="Cache-control"
content="no-cache">
```

При кэшировании документа можно указать время, до которого имеет место соответствие кэшированного документа с его оригиналом на сервере. В данном случае используется HTTP-оператор Expires и элемент META принимает вид:

```
<META http-equiv="Expires"
content="Monday, 18-May-2023 00:00:01">
```

**Примечание:** в данном случае дата задается для корректировки HTTP-заголовка и поэтому должна иметь следующий формат, указанный в примере.

## Ter meter

Синтаксис:

```
<meter>...</meter>
```

Используется для вывода значения в некотором известном диапазоне. Используется преимущественно для отображения числовых значений, например, количества результатов поиска, объема жидкости, давления и др. .

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

## Атрибуты:

- **high** = "число" – позволяет задать высший предел для меры.
- **low** = "число" – задает низший предел для меры.
- **max** = "число" – позволяет задать наибольшее значение диапазона.
- **min** = "число" – задает наименьшее значение диапазона.
- **optium** = "число" – показывает диапазон, который рассматривается как оптимальный для меры.
- **value** = "число" – задает действительное значение для меры.
- Глобальные атрибуты HTML.

## Примеры:

```
<meter>50%</meter>
```

```
<meter min="0" max="200">150</meter>
```

## Тег &lt;nav&gt;

## Синтаксис:

```
<nav>...</nav>
```

Тег NAV своим открывающим и закрывающим тегами задает область html-документа, в которой собраны теги (ссылки) навигации (как внутри текущего документа, так и ведущих на другую страницу). Тег не поддерживается HTML4.

Артикул:

- Глобальные атрибуты HTML.

Пример:

```
<header>
  Шапка сайта
</header>
<nav>
  Навигация
  <ul>
    <li><a href="#">Ссылка 1</a></li>
    <li><a href="#">Ссылка 2</a></li>
  </ul>
</nav>
```

## Тег noscript

Синтаксис:

```
<noscript>...</noscript>
```

Позволяет определить альтернативный контент, который будет использоваться в случае невозможности выполнения скрипта. Данный контент может отображаться, если браузер не поддерживает сценарии, если поддержка скриптов выключена.

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z



чена или браузер не может понять язык написания сценария. Если `noscript` находится в области *head*, а поддержка скриптов выключена, то могут использоваться лишь теги `style`, `link`, `meta`.

Артикул:

- Глобальные атрибуты HTML

Пример:

```
<noscript>
```

```
    Для нормального функционирования
    страницы включите JavaScript
</noscript>
```

## Тег object

Синтаксис:

```
<object>...</object>
```

Тег сообщает браузеру, как загружать и отображать объекты, которые исходно браузер не понимает. Как правило, такие объекты требуют подключения к браузеру специального модуля, который называется плагином, или запуска вспомогательной программы.

Спецификация HTML 4 разрешает вкладывать несколько тегов **object** с разным содержанием друг в друга. Это позволяет отображать тот контент, который понимает браузер, при отсутствии

нужного плагина. Например, внешний тег **object** загружает видеофайл, а для случая, когда соответствующий кодек (программа для сжатия и восстановления видеоданных) не установлен, внутренний тег **object** показывает графическое изображение. Дополнительно внутрь контейнера **object** можно поместить тег **param**, который передает дополнительные параметры для отображения объекта. Тег **object** может содержать несколько тегов **param**.

Атрибуты:

- **data** = "URL-адрес" – указывает адрес ресурса, синтаксис зависит от объекта.
- **form** = "идентификатор формы" – связывает объект с элементом *form* на странице.
- **height** = "число" – задает высоту объекта в пикселях.
- **name** = "тест" – название объекта, на который будут ссылаться сценарии.
- **type** = "тип MIME" – задает тип MIME для объекта внутри тега.
- **usemap** = "URL" – задает карту изображений, которая будет использоваться вместе с объектом.
- **width** = "число" – ширина объекта в пикселях.
- Глобальные атрибуты HTML.

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

Пример:

```
<object      type="application/x-shockwave-
flash"
            data="images/car.swf" width="400"
height="300">
    <param name="quality" value="high">
    <param name="wmode" value="opaque">
</object>
```

Данный тег часто использовался для встраивания Flash-объектов, но в 2021 году плагин Flash является запрещенным и вскоре следует ожидать отказ от использования тега **object**.

## Тег ol

Синтаксис:

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

Язык HTML обладает возможностями предоставления информации в виде списков. Список служит для добавления структуры в документ. Причем эта структура отображается визуально, например: список покупок, меню сайта, пошаговое описание каких-либо действий, толковый словарь и т.п.

В HTML различают:

- Неупорядоченные списки (список покупок)

- Упорядоченные списки (пошаговое описание, в котором каждый шаг пронумерован).
- Список определений (толковый словарь)

В любом списке должен присутствовать хотя бы один элемент списка, иначе он будет проигнорирован.

Неупорядоченные списки создаются тегом UL, упорядоченные списки – тегом OL. Списки обоих типов состоят из последовательности элементов списка, которые задаются тегом LI. Тег LI, а точнее его содержание (например, название покупки), является обособленной частью списка. Неупорядоченные списки отображаются браузерами как маркированные, а упорядоченные – как пронумерованные. Пример структуры неупорядоченного списка (нумерованного):

```
<ol>
  <li>

  </li>
  <li>

  </li>
  <li>

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

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

## Атрибуты:

- **reversed** – изменяет направление числовой последовательности на обратное, от наибольшего значения к наименьшему.
- **start** = "число" – начинает нумерацию с заданного числа.
- **type** = "1 / a / A / i / I" – определяет систему нумерации:
  - 1 – арабские цифры.
  - a – буквы нижнего регистра.
  - A – буквы верхнего регистра.
  - i – римские цифры нижнего регистра.
  - I – римские цифры верхнего регистра.
- Глобальные атрибуты HTML.

## Пример:

```
<ol>
  <li>Яблоко</li>
  <li>Апельсин</li>
  <li>Киви</li>
</ol>
```

Для дополнительного примера см. тег **li**.

# Тег optgroup

Синтаксис:

```
<optgroup>...</optgroup>
```

Определяет логическую группу элементов **option**, которая находится внутри управления меню *select*. Тег **optgroup** не может содержать в себе другие теги **optgroup** (нельзя вкладывать **optgroup** друг в друга).

Атрибуты:

- **disabled** – отключает группу параметров.
- **label** = "текст" – задает подпись для группы параметров.
- Глобальные атрибуты HTML.

Пример:

```
Выберите смартфон<br>
<select name="phone" size="7"
multiple="multiple">
  <optgroup label="Apple">
    <option>iPhone 12</option>
    <option>iPhone 11</option>
  </optgroup>
  <optgroup label="Samsung">
    <option>Galaxy S</option>
    <option>A50</option>
  </optgroup>
</select>
```

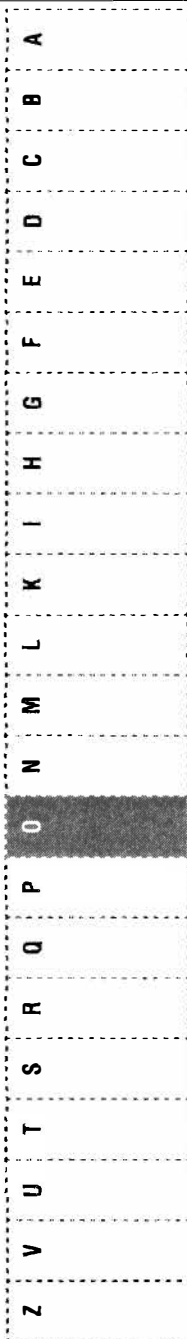




Рис. 35. Демонстрация использования тега `optgroup`

## Ter option

Синтаксис:

```
<option>...</option>
```

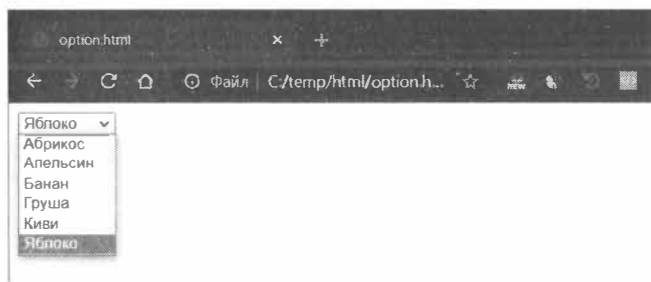
Тег **option** используется внутри тега **select** и определяет отдельные элементы списка. Ширина списка определяется самым длинным текстом, указанным в теге **option**, а также может быть изменена с помощью стилей CSS. Если планируете отправлять выбранный элемент на сервер, то список *select* нужно поместить внутри формы и присвоить его атрибуту *name* имя. В качестве контента элемента **option** используется значение, которое передается приложению обработки формы

Атрибуты:

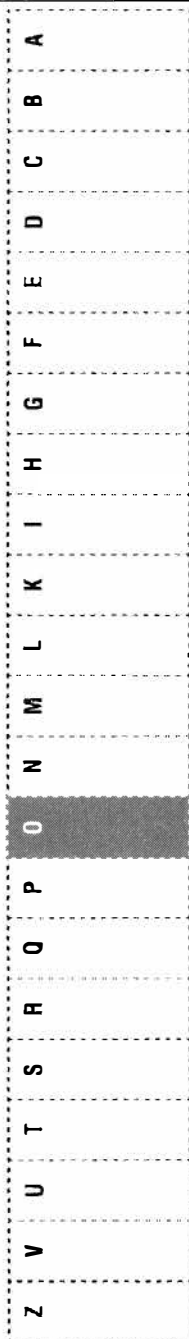
- **disabled** — заблокировать для доступа элемент списка.
- **label** — указание метки пункта списка.
- **selected** — заранее устанавливает определенный пункт списка выделенным.
- **value** — значение пункта списка, которое будет отправлено на сервер или прочитано с помощью скриптов.
- Глобальные атрибуты HTML.

Пример:

```
<select name="fruit">
  <option>Абрикос</option>
  <option>Апельсин</option>
  <option>Банан</option>
  <option>Груша</option>
  <option>Киви</option>
  <option selected>Яблоко</option>
</select>
```



**Рис. 36.** Последний вариант является выбранным по умолчанию





# Ter output

Синтаксис:

```
<output>...</output>
```

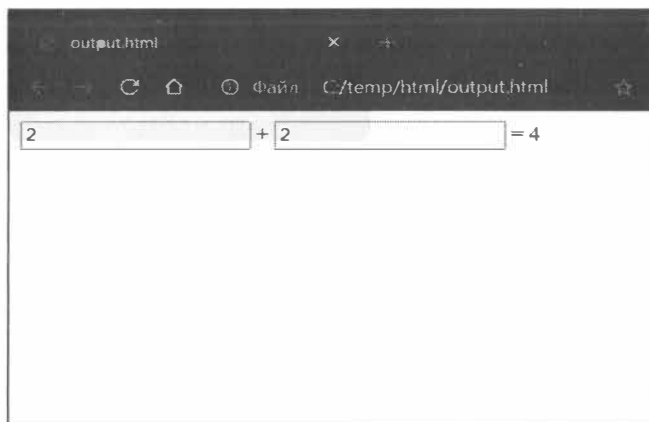
Представляет область, используемую в качестве результата вычисления, например, результат вычисления сценария. Не поддерживается в HTML4.

Атрибуты:

- **for** = "текст" — создает связь между результатом вычисления и именованным элементом, который подвержен влиянию вычисления.
- **form** = "идентификатор формы" — явно связывает output с определенной формой.
- **name** = "имя" — задает имя элемента output.
- Глобальные атрибуты HTML.

Пример:

```
<form oninput="myCalc.value = parseInt(a.
value) + parseInt(b.value)">
<input type="number" id= "a"> + <input
type="number" id="b">
= <output      name="myCalc"      for="a
b">[результат a+b]</output>
</form>
```



*Рис. 37. Пример использования тега output*

## Тег p

Синтаксис:

`<p>...</p>`

Определяет текстовый абзац. Данный тег является блочным элементом, всегда начинается с новой строки, абзацы текста, идущие друг за другом, разделяются между собой отступами. Величиной отступов можно управлять с помощью стилей. Если закрывающего тега нет, считается, что конец абзаца совпадает с началом следующего блочного элемента.

Атрибуты:

- **align** = "выравнивание" – управляет выравниваем текста внутри тега p. Доступ-

ные значения: *left* (по левому краю), *right* (по правому краю), *center* (по центру), *justify* (по ширине).

- Глобальные атрибуты HTML.

Пример:

```
<p>Это новый абзац текста</p>
<p>А это еще один абзац</p>
```

## Тег param

Синтаксис:

```
<param>
```

Данный тег используется внутри контейнера **object**. Его назначение – передача параметру объекту, который внедряется в веб-страницу посредством тега **object**. Один тег **param** используется для передачи объекту одного параметра. Имя параметра задается атрибутом *name*, а значение – атрибутом *value*.

Атрибуты:

- **name** – определяет имя параметра.
- **value** – определяет значение параметра.
- Глобальные атрибуты HTML.

Пример:

```
<param name="autoStart" value="false">
```

## Тег pre

Синтаксис:

```
<pre>...</pre>
```

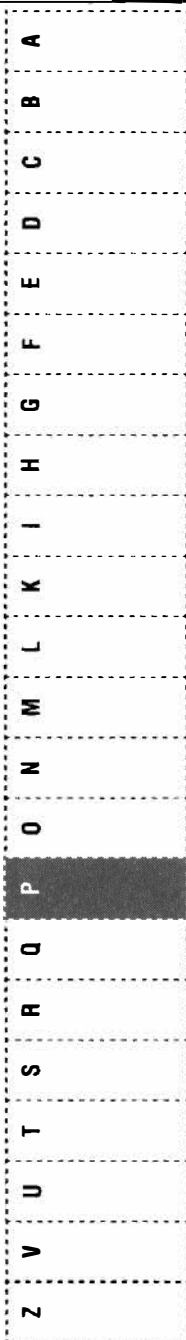
Выводит предварительно отформатированный текст, строки которого отображаются в том виде, в котором они записаны в теге **pre**. При этом внутри данного тега можно использовать различные пробельные символы вроде табуляции, множественные пробелы и переводы строки – они не будут удалены, как без этого тега. По умолчанию текст внутри блока **pre** отображается с помощью моноширинного шрифта (Courier), что позволяет сохранить интервалы и выравнивание. При желании шрифт можно изменить с помощью стилей.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<pre>
    Это пример
        текста    с    произвольным
выравниванием внутри
                        тега pre
</pre>
```



# Тег progress

Синтаксис:

```
<progress>...</progress>
```

Визуально представляет собой ход выполнения какой-то задачи, например, загрузки файла. Изменение значения индикатора прогресса происходит через JavaScript. Данный тег не поддерживается HTML4.

Атрибуты:

- **max** = "число" – максимальное значение, например, 100.
- **value** = "число" – показывает, какая часть задачи выполнена, например, 50.
- Глобальные атрибуты HTML.

Пример:

```
<!DOCTYPE html>
<html>
<style>
#myProgress {
    width: 100%;
    background-color: #ddd;
}
#myBar {
    width: 1%;
}
```

```

    height: 30px;
    background-color: #4CAF50;
}
</style>
<body>

<h1>Индикатор прогресса</h1>

<div id="myProgress">
  <div id="myBar"></div>
</div>

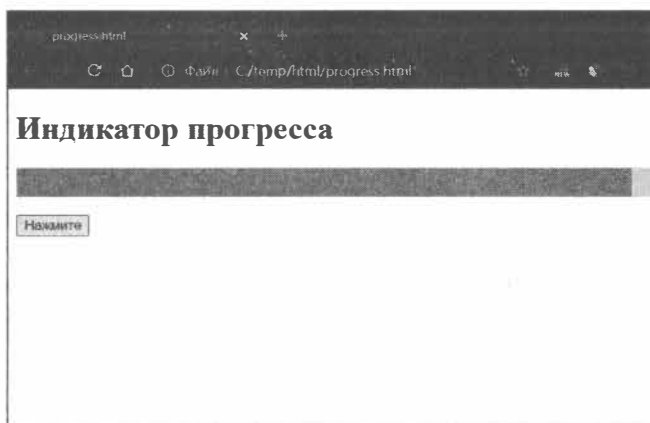
<br>
<button      onclick="move()">Нажмите</
button>

<script>
var i = 0;
function move() {
  if (i == 0) {
    i = 1;
    var elem = document.
getElementById("myBar");
    var width = 1;
    var id = setInterval(frame, 10);
    function frame() {
      if (width >= 100) {
        clearInterval(id);
        i = 0;
      } else {
        width++;
        elem.style.width = width + "%";
      }
    }
  }
}
</script>

</body>
</html>

```

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z



*Рис. 38. Индикатор прогресса в действии*

## Тег q

Синтаксис:

`<q>...</q>`

Используется для выделения короткой цитаты, которая может включаться в строку. Тег **blockquote** используется для выделения длинных цитат. Но когда нужно включить цитату в строку, то лучше использовать тег **q**, поскольку он не создает лишних отступов вокруг цитаты.

Атрибуты:

- **cite** = "URL" – источник цитаты.
- Глобальные атрибуты HTML.

Пример:

```
<p>Рассмотрим пример <q>использования  
тега q</q> внутри абзаца</p>
```

## Тег rp

Синтаксис:

```
<rp>...</rp>
```

Используется для вывода текста в браузерах, которые не поддерживают тег `<ruby>`. В остальных браузерах текст, заключенный в контейнер `<rp>`, не отображается. Тег `<rp>` должен идти до или после тега `<rt>`. Тег не поддерживается HTML4.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<ruby>  
  текст  
  <rp>это текст</rp><rt>здесь аннотация</  
rt><rp>это текст</rp>  
</ruby>
```



Рис. 39. Тег `rp`

## Тег `rt`

Синтаксис:

```
<rt>...</rt>
```

Тег `<rt>` добавляет аннотацию сверху или снизу от текста, заключенного в контейнер `<ruby>`. Сама аннотация выводится уменьшенным шрифтом. Подобная форма записи используется для азиатских языков (японский, китайский), но может использоваться и в "мирных целях", когда нужно написать один текст над другим (см. рис. 25). Тег не поддерживается в HTML4.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>rt</title>
    <style>
      .date { font-size: 2.5em; }
      .date ruby { background: #0078c9;
color: #fff; }
      .date rt { background: #a0dae8; color:
#000; padding: 2px;
margin-top: 20px; }
    </style>
  </head>
  <body>
    <p class="date">
      <ruby>
        15<rt>февраля</rt></ruby>
      </p>
    </body>
  </html>
```

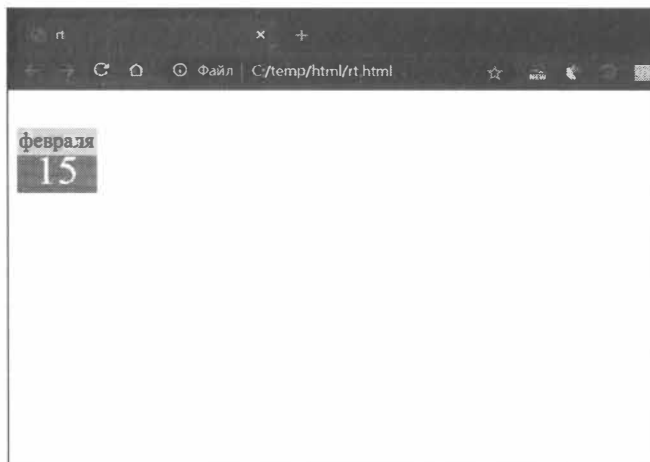
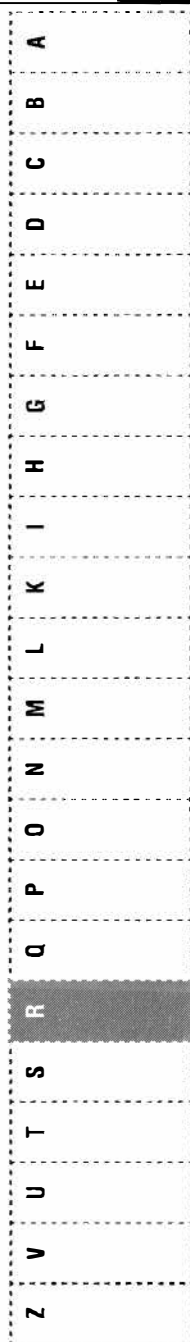


Рис. 40. Использование тега `rt`



## Тег ruby

Синтаксис:

```
<ruby>...</ruby>
```

Предназначен для добавления небольшой аннотации сверху или снизу от заданного текста. Внутри этого тега могут использоваться теги **rt** и **rp**. Тег **rt** формирует аннотацию к тексту, а **rp** используется в браузерах, которые не поддерживают *ruby*. Название тега происходит от слова "рубин", которым в типографии обозначается небольшой шрифт. Он не имеет никакого отношения к языку программирования Ruby. Тег не поддерживается HTML4.

Атрибуты:

- Глобальные атрибуты HTML.

Пример был приведен при описании тега **rt**.

## Тег s

Синтаксис:

```
<s>...</s>
```

Отображает текст как перечеркнутый. Этот тег аналогичен тегу **strike**, но в отличие от него имеет сокращенную форму записи подобно тегам **b**,

А  
В  
С  
Д  
Е  
Ф  
Г  
Н  
И  
К  
Л  
М  
Н  
О  
Р  
Д  
Р  
С  
Т  
У  
В  
З

**i** и **u**. Вместо этого тега настоятельно рекомендуется использовать стили.

Атрибуты:

- Глобальные атрибуты HTML.

Пример:

```
<s>Этот текст будет перечеркнутым</s>
```

## Тег samp

Синтаксис:

```
<samp>...</samp>
```

Предназначен для отображения кода компьютерной программы или скрипта. Как правило, браузеры отображают с помощью моноширинного текста.

Атрибуты:

- Глобальные атрибуты HTML.

Пример:

```
<samp>
  echo "Hello!";
</samp>
```

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

# Ter script

Синтаксис:

```
<script>...</script>
```

Предназначен для описания скриптов, может содержать ссылку на программу или ее текст на определенном языке. Скрипты могут располагаться во внешнем файле, а могут находиться внутри тега **script**. Небольшие скрипты, как правило, хранятся внутри тега script, а длинные – в отдельных файлах.

Атрибуты:

- **async** – указывает на то, что сценарий внутри script должен выполняться асинхронно (не поддерживается HTML4)
- **charset** = "кодировка" – задает кодировку документа, в котором находится внешний скрипт. Данный атрибут не применяется к скриптам, которые описаны внутри тега script.
- **defer** – указывает браузеру на то, что внешний сценарий должен быть выполнен один раз после завершения анализа страницы.
- **src** = "URL" – задает URL документа, в котором находится скрипт.

- **type** = "тип контента" – задает язык, на котором написан скрипт. В HTML5 по умолчанию подразумевается, что используется JavaScript, поэтому значение `text/javascript` указывать не нужно. В HTML4 нужно обязательно указывать этот атрибут.
- Глобальные атрибуты HTML.

Пример:

```
<script>
window.alert('Hello!');
</script>
```

## Ter section

Синтаксис:

```
<section>...</section>
```

Тег `section` – своим открывающим и закрывающим тегами задает раздел – тематическую группу содержимого интернет-страницы – который должен восприниматься обособленно. Секция может иметь свою собственную обособленную структуру, в частности в секции может быть свой заголовок (тег `HEADER`) и подвал (тег `FOOTER`). Также допускается вложение одной секции в другую. Тег не поддерживается HTML4.

Атрибуты:

- Глобальные атрибуты HTML

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

Пример:

```
<section>
  Секция 1
<article>
  <h1>Заголовок статьи</h1>
  <p>Контент</p>
  <h2>Подзаголовок статьи</h2>
</article>
</section>
<section>
  Секция 2
<article>
  <h1>Заголовок статьи 2</h1>
  <p>Контент</p>
  <h2>Подзаголовок статьи</h2>
</article>
</section>
```

## Тег select

Синтаксис:

```
<select>...</select>
```

Создает выпадающий список или список с прокруткой. Список содержит несколько вариантов, задаваемых тегами **option**. В зависимости от атрибутов **select**, пользователь может выбрать только один или несколько вариантов списка. Внутри **select**, кроме **option**, можно также использовать тег **optgroup**.

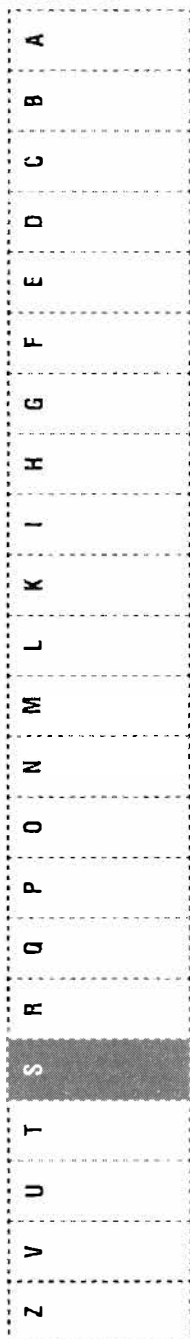
Атрибуты:

- **autofocus** — автоматическое получение фокуса ввода.
- **disabled** — блокирует доступ к элементу и его изменение, пока какой-то скрипт на странице не разблокирует его.
- **form** = "идентификатор формы" — позволяет связать поле с формой по ее идентификатору.
- **multiple** — позволяет выбрать несколько элементов списка сразу.
- **name** = "имя" — имя поля, на которое можно сослаться в JavaScript-сценариях.
- **required** — обязательное для заполнения поле.
- **size** = "число" — количество отображаемых строк списка.
- Глобальные атрибуты HTML.

Примеры:

## Классический раск список:

```
<select name="year" size="1">
  <option value="2021">2021</option>
  <option value="2020">2020</option>
  <option value="2019">2019</option>
  <option value="2018">2018</option>
```

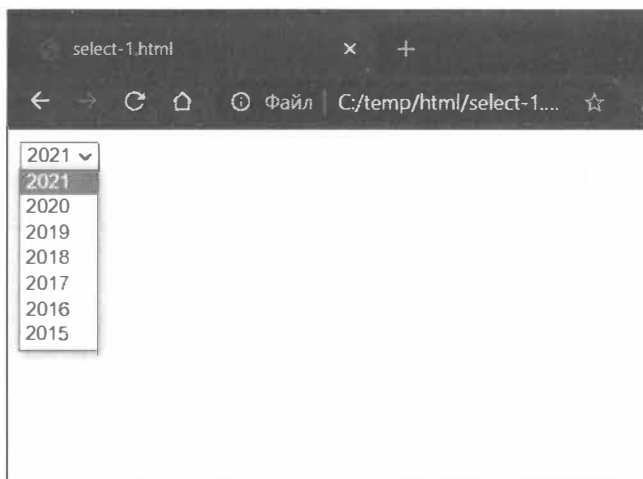




```

<option value="2017">2017</option>
<option value="2016">2016</option>
<option value="2015">2015</option>
</select>

```



*Рис. 41. Обычный выпадающий список*

## Список с прокруткой и возможностью выбора нескольких вариантов:

```

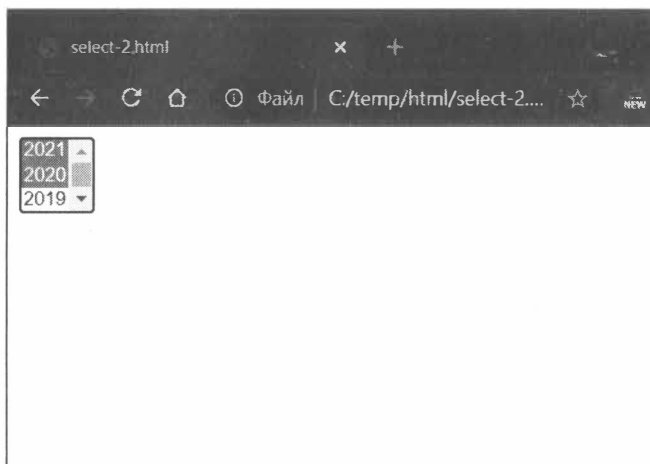
<select name="year" size="3" multiple>
  <option value="2021">2021</option>
  <option value="2020">2020</option>
  <option value="2019">2019</option>
  <option value="2018">2018</option>

```

```

    <option value="2017">2017</option>
    <option value="2016">2016</option>
    <option value="2015">2015</option>
</select>

```



**Рис. 42. Список с возможностью выбора нескольких вариантов**

## Ter small

Синтаксис:

```
<small>...</small>
```

Отображает контекст небольшим текстом. Применяется для небольших заметок, например, для

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

замечания о юридической ответственности за неправомерное использование контента.

Атрибуты:

- Глобальные атрибуты HTML.

Пример:

```
<p><small>&copy; 2021 Моя компания/small>
```

## Ter source

Синтаксис:

```
<source>
```

Используется внутри тегов **video** и **audio**, указывая источник видео или аудио соответственно. Не поддерживается HTML4. Позволяет указать несколько версий мультимедиа-файла. Напомним, что разные браузеры поддерживают разные типы мультимедиа-контента. Верстальщик может указать несколько источников видео/аудио – в разных форматах, чтобы быть уверенным, что пользователи всех браузеров увидят/услышат мультимедиа-контент. Браузер будет просматривать список элементов source до тех пор, пока не найдет формат, который может воспроизводить.

Атрибуты:

- **media** = "all | aural | braille | handheld | print | projection | screen | tty | tv" – задает среду

для воспроизведения мультимедиа-файлов.

- **src** = "URL-ссылка" – указывает источник мультимедиа-файла.
- **type** = "Тип MIME" – указывает тип мультимедиа-контента. Также может включать MIME-параметр `codecs=`, задающий необходимый для воспроизведения контента кодек.
- Глобальные атрибуты HTML.

Пример:

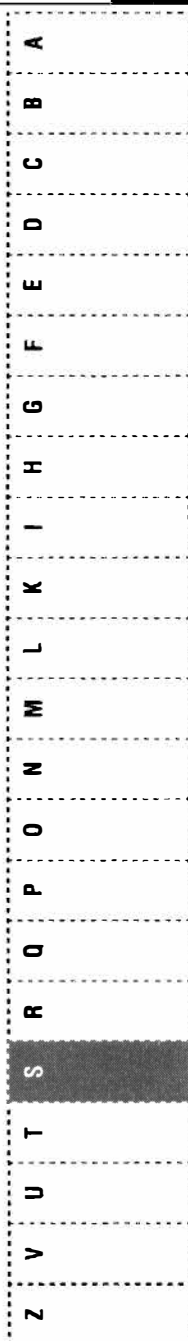
```
<video>
  <source src="film.ogv" type="video/ogg; codecs='theora, vorbis'">
  <source src="filv.mp4" type="video/mp4">
Ваш браузер не поддерживает тег video
</video>
```

## Тег span

Синтаксис:

```
<span>...</span>
```

Предназначен для определения строчных элементов документа. Отличие от блочных элементов, таких как **p** или **div**: посредством тега **span** можно выделить часть информации внутри других тегов и установить для нее свой стиль. На-



пример, внутри абзаца (тега `p`) можно изменить цвет и размер первой буквы, если добавить начальный и конечный тег `span` и определить для него стиль текста. Описывать каждый раз стиль внутри тега не очень хорошо, поэтому можно выделить стиль в CSS-таблицу, а для тега добавить атрибут *class* или *id* с именем селектора.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ter SPAN</title>
    <style>
      body {
        font-family: Arial, sans-serif; /*
Рубленый шрифт */
      }
      .letter {
        color: red; /* Красный цвет символов
*/
        font-size: 200%; /* Размер шрифта в
процентах */
        font-family: serif; /* Шрифт с
засечками */
        position: relative; /* Относительное
позиционирование */
        top: 5px; /* Сдвиг сверху */
      }
    </style>
  </head>
```

```

<body>

  <p><span class="letter">С</span>ъешь
ещё этих мягких французских булок, да
выпей чаю</p>

</body>
</html>

```

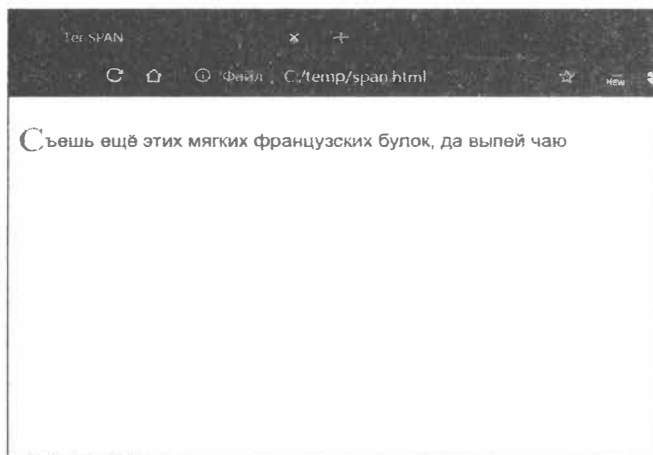


Рис. 43. Пример использования тега `span`

## Ter strong

Синтаксис:

```
<strong>...</strong>
```

Тег **strong** предназначен для акцентирования текста. Браузеры отображают такой текст жирным начертанием.



Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<p>Обычный текст, а это <strong>текст,
на который нужно обратить внимание</
strong></p>
```

## Тег style

Синтаксис:

```
<style>...</style>
```

Используется для включения стилевых правил в документ. Тег **style** не нужно путать с глобальным атрибутом *style* – это разные вещи. Атрибут *style* позволяет применить стиль к конкретному элементу страницы, а тег **style** – применить стили сразу ко всем элементам страницы. В HTML4 данный тег обязательно должен находиться внутри тега **head**. В HTML5 его можно использовать в контексте документа.

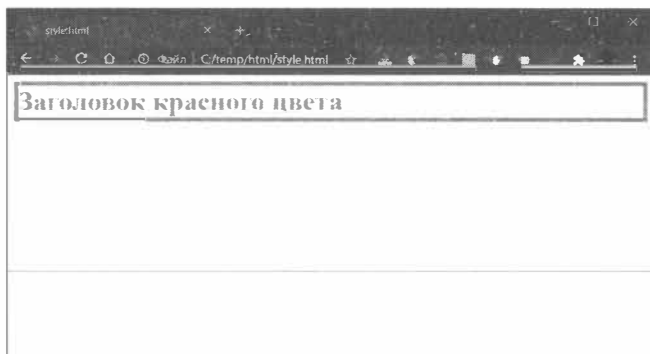
Атрибуты:

- **media** = "all | aural | braille | handheld | print | projection | screen | tty | tv" – задает среду для воспроизведения мультимедиа-файлов.

- **scoped** – применяет стили только к локальному дереву документов, а не к целому документу. Не поддерживается HTML4.
- **type = "text/css"** – задает тип контента. в HTML5 указывать тип контента необязательно, а вот в HTML4 данный параметр обязательный.
- Глобальные атрибуты HTML.

Пример:

```
<html>
  <head>
    <style type="text/css">
h1 {color: red; border: 5px solid red; }
    </style>
  </head>
  <body>
    <h1>Заголовок красного цвета</h1>
  </body>
</html>
```



*Рис. 44. Заголовок красного цвета с рамкой*

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z



## Тег sub

Синтаксис:

```
<sub>...</sub>
```

Позволяет отформатировать текст в виде нижнего индекса. Правила хорошего тона подразумевают использовать этот тег только, когда нижний индекс имеет специфическое значение. Для типографских эффектов можно использовать стили. Как правило, тег **sub** используется в различных формулах.

Атрибуты:

- Глобальные атрибуты HTML.

Пример:

```
<p>H<sub>2</sub>O</p>
```

## Тег summary

Синтаксис:

```
<summary>...</summary>
```

Создает резюме, заголовок или легенда для соответствующего предка тега **details**. Тег **summary** должен идти первым внутри **details**.

Атрибуты:

- Глобальные атрибуты HTML

А  
В  
С  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

Пример:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>summary</title>
  </head>
  <body>
    <details>
      <summary>Информация об авторе</summary>
      <p>Иван Иванов</p>
    </details>
  </body>
</html>
```

## Тег sup

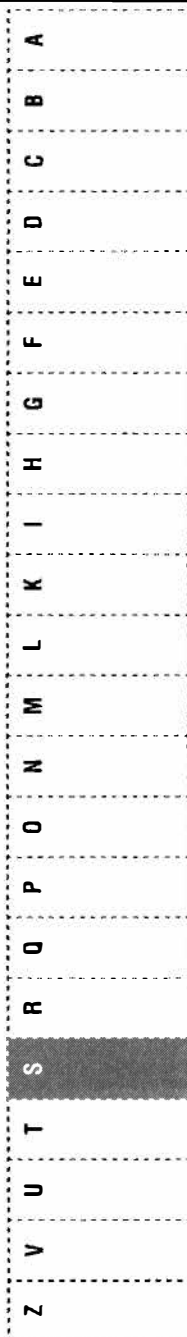
Синтаксис:

```
<sup>...</sup>
```

Позволяет отформатировать текст в виде верхнего индекса. Правила хорошего тона подразумевают использовать этот тег только, когда верхний индекс имеет специфическое значение. Для типографских эффектов можно использовать стили. Как правило, тег sup используется в различных формулах.

Атрибуты:

- Глобальные атрибуты HTML.



Пример:

```
<p>x=y<sup>2</sup></p>
```

## Тег table

Синтаксис:

```
<table>...</table>
```

Создает таблицу, используемую для отображения информации в табличной форме. Для создания таблицы понадобится сам тег **table**, а также теги **tr** и **td**. Тег **tr** описывает строки таблицы, а **td** – колонки внутри каждой строки.

В HTML-документе может содержаться любое количество таблиц, причем допускается вложение таблиц друг в друга. Каждая таблица является содержимым тега-контейнера **table** и состоит, по крайней мере, из одной строки.

Атрибуты:

- **border** = "ширина рамки" – позволяет указать ширину рамки (в пикселях), которая обрамляет таблицу и ее ячейки. Если присвоить этому атрибуту значение 0, рамка не будет отображаться. По умолчанию используется значение 1.
- **sortable** – в HTML5.1 данный атрибут позволяет сортировать таблицу по столбцу. Используется вместе с атрибутом *sorted* элемента **th**, значение которого может

присваиваться вручную или браузером, когда пользователь щелкает на столбце.

- Глобальные атрибуты HTML.

Рассмотрим примеры.

## Простая таблица:

```
<table border="1">
  <tr>
    <th>1-ая ячейка 1-ой строки</th>
    <th>2-ая ячейка 1-ой строки</th>
    <th>3-ая ячейка 1-ой строки</th>
  </tr>
  <tr>
    <td>1-ая ячейка 2-ой строки</td>
    <td>2-ая ячейка 2-ой строки</td>
  </tr>
</table>
```

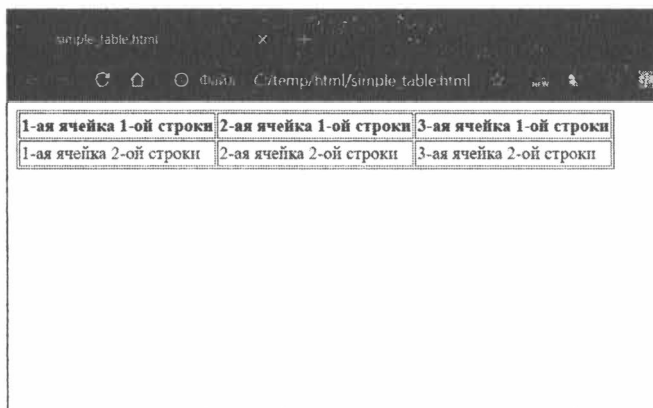
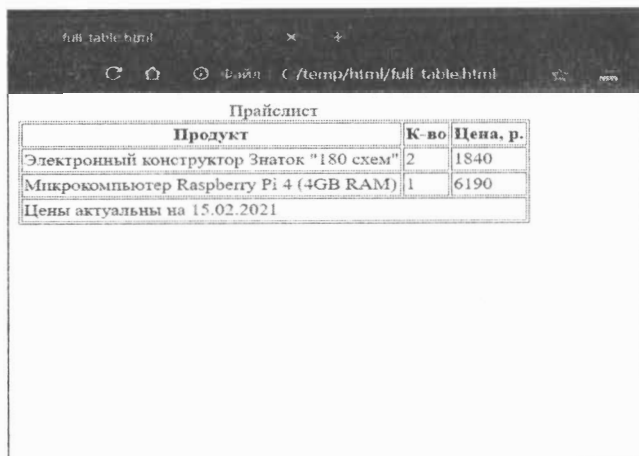


Рис. 45. Простая таблица

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z

## Полная модель таблицы – с тегами caption, tbody, thead, tfoot, colgroup, col:

```
<table border>
  <caption>Прайслист</caption>
  <colgroup id="prices">
    <col span="2">
    <col span="1">
  </colgroup>
  <thead>
    <tr>
      <th>Продукт</th><th>К-во</th><th>Цена,
р.</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td colspan="3">Цены актуальны на
15.05.2021</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>Электронный конструктор Знаток
"180 схем"</td>
      <td>2</td>
      <td>1840</td>
    </tr>
    <tr>
      <td>Микрокомпьютер Raspberry Pi 4
(4GB RAM)</td>
      <td>1</td>
      <td>6190</td>
    </tr>
  </tbody>
</table>
```



The screenshot shows a web browser window with the address bar displaying 'C:/temp/html/full table.html'. The page content is a table titled 'Прайс-лист' (Price List). The table has three columns: 'Продукт' (Product), 'К-во' (Quantity), and 'Цена, р.' (Price, rubles). The data rows are: 'Электронный конструктор Знаток "180 схем"' (Electronic constructor Znatok '180 schemes') with quantity 2 and price 1840; and 'Микрокомпьютер Raspberry Pi 4 (4GB RAM)' with quantity 1 and price 6190. A footer note states 'Цены актуальны на 15.02.2021' (Prices are valid as of 15.02.2021).

Прайс-лист		
Продукт	К-во	Цена, р.
Электронный конструктор Знаток "180 схем"	2	1840
Микрокомпьютер Raspberry Pi 4 (4GB RAM)	1	6190

Цены актуальны на 15.02.2021

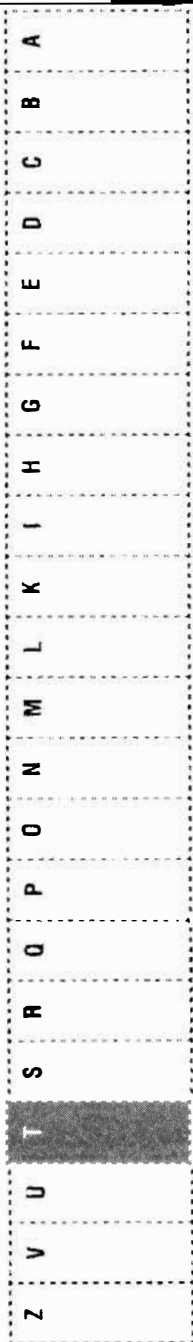
Рис. 46. Полная таблица

## Ter tbody

Синтаксис:

```
<tbody>...</tbody>
```

Предназначен для хранения одной или нескольких строк таблицы. Это позволяет создавать структурные блоки, к которым можно применять единое оформление через стили, а также управлять их видом через скрипты. Элементы `tbody`, `thead`, `tfoot` могут ускорить отображение больших таблиц, обеспечить механизм для прокрутки тела таблицы независимо от ее заголовка и футера или повторять заголовок таблицы и футер при печати длинной таблицы на нескольких страницах.



Атрибуты:

- Глобальные атрибуты HTML

Пример:

*см. table*

## Ter td

Синтаксис:

`<td>...</td>`

Определяет ячейку таблицы данных. Ячейка может содержать произвольный контент, в том числе другую таблицу.

Атрибуты:

- **headers** = "идентификатор ссылки" – атрибут, значением которого является список имен (id) ячеек, выступающих в роли заголовка (описания) для содержащей его ячейки.
- **rowspan** = "число" – атрибут, отвечающий за объединение соседних строк в рамках одного столбца (т.е. соседних ячеек в столбце) В качестве своего значения принимает натуральные числа. По умолчанию установлен в значении, равном единице.

- **colspan** = "число" — атрибут, отвечающий за объединение соседних столбцов в рамках одной строки (т.е. соседних ячеек в строке). В качестве своего значения принимает натуральные числа. По умолчанию установлен в значении, равном единице.
- Глобальные атрибуты HTML.

Пример:

```
<table border="1">
  <tr>
    <td colspan="2">Ячейка
занимает сразу два места</td>
  </tr>
  <tr>
    <td>Ячейка 1 Строка 2</td>
    <td>Ячейка 2 Строка 2</td>
  </tr>
</table>
```

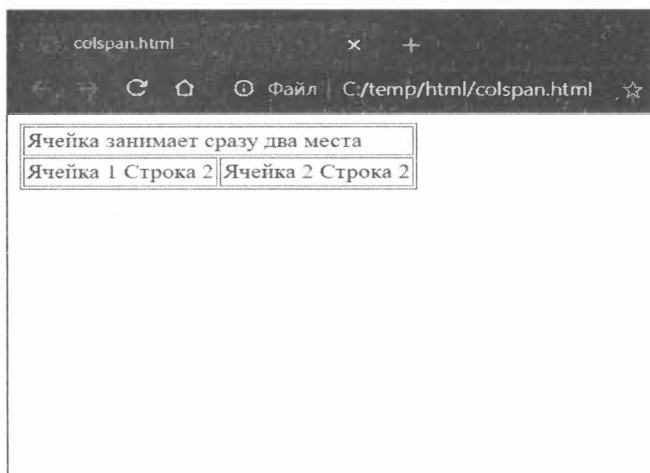
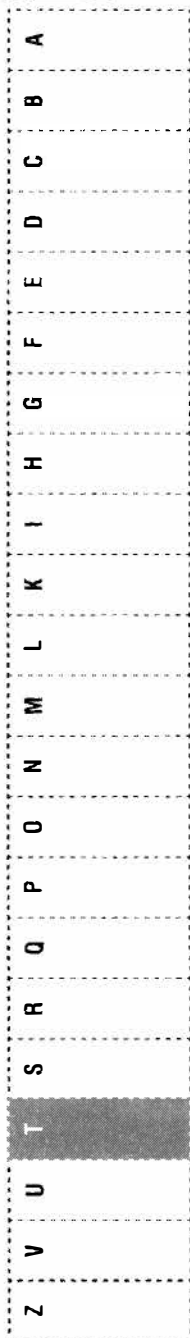


Рис. 47. Демонстрация атрибута **colspan**





Раньше таблицы очень часто использовались для расположения элементов на странице, например, рассмотрим пример верстки страницы с тремя колонками:

```
<table border="0" width="100%">
  <tr>
    <td valign="top" width="33%">
      Lorem ipsum dolor sit amet,
      consectetur adipiscing elit.
      Aenean sit amet porttitor enim.
      Curabitur eleifend metus ac tortor
      luctus efficitur. Pellentesque sem
      elit, dictum tincidunt efficitur at,
      dapibus quis odio.
    </td>
    <td valign="top" width="33%">
      Lorem ipsum dolor sit amet,
      consectetur adipiscing elit.
      Aenean sit amet porttitor enim.
      Curabitur eleifend metus ac tortor
      luctus efficitur. Pellentesque sem
      elit, dictum tincidunt efficitur at,
      dapibus quis odio. Aliquam ultrices
      augue sed eros varius iaculis.
      Orci varius natoque penatibus et
      magnis dis parturient montes,
      nascetur ridiculus mus.
    </td>
    <td valign="top" width="33%">
      Lorem ipsum dolor sit amet,
      consectetur adipiscing elit.
      Aenean sit amet porttitor enim.
```

```
Curabitur eleifend metus ac tortor
luctus efficitur. Pellentesque sem
elit, dictum tincidunt efficitur at,
dapibus quis odio.
</td>
</tr>
</table>
```

A
B
C
D
E
F

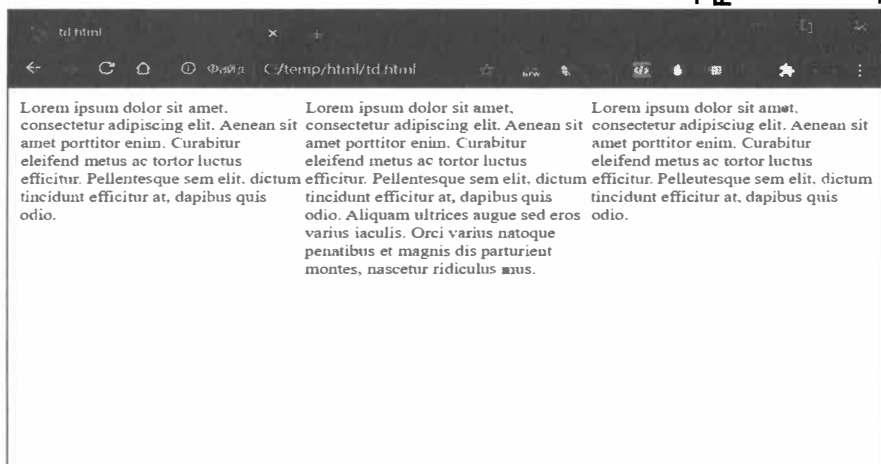


Рис. 48. Верстка страницы с помощью таблиц

Здесь мы растягиваем ширину таблицы на всю ширину экрана (`width="100%"`), затем мы создаем три колонки одинаковой ширины (`width="33%"`) и задаем тип вертикального выравнивания контента каждой колонки (`valign="top"` – по верхнему краю). Сейчас такой тип верстки не используется, а вместо таблиц используется тег **div** с заданными в CSS правилами отображения.

O
P
Q
R
S
T
U
V
Z

```

<html>
  <head>
    <title>Макет на 3 колонки</
title>
  <style>
    .container { position: relative;
/* Относительное позиционирование
*/ }
    .container DIV { position: absolute;
/* Абсолютное позиционирование */ }
    .col1 {width: 33%;}
    .col2 {width: 34%; left: 33%;}
    .col3 {width: 33%; right: 0;}
  </style>
  </head>
  <body>
    <div class="container">
      <div class="col1">
        Lorem ipsum dolor sit amet,
        consectetur adipiscing elit.
        Aenean sit amet porttitor enim.
        Curabitur eleifend metus ac tortor
        luctus efficitur. Pellentesque sem
        elit, dictum tincidunt efficitur at,
        dapibus quis odio.
      </div>
      <div class="col2">
        Lorem ipsum dolor sit amet,
        consectetur adipiscing elit.
        Aenean sit amet porttitor enim.
        Curabitur eleifend metus ac tortor
        luctus efficitur. Pellentesque sem
        elit, dictum tincidunt efficitur at,
        dapibus quis odio. Aliquam ultrices

```

```

augue sed eros varius iaculis.
Orci varius natoque penatibus et
magnis dis parturient montes,
nascetur ridiculus mus.
</div>
<div class="col3">
Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Aenean sit amet porttitor enim.
Curabitur eleifend metus ac tortor
luctus efficitur. Pellentesque sem
elit, dictum tincidunt efficitur at,
dapibus quis odio.
</div>
</div>
</body>
</html>

```

A

B

C

D

E

F

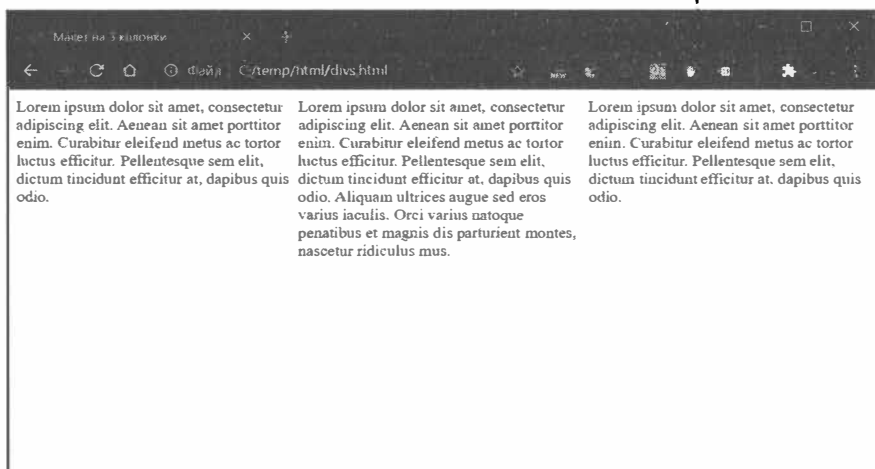
G

H

I

K

L



U

V

Z

Рис. 49. Верстка страницы с помощью div

## Ter textarea

Синтаксис:

```
<textarea>...</textarea>
```

Отображает многострочное поле для ввода текста. Контент, выводимый в этом поле, задается между двумя тегами – открывающимся и закрывающимся. Если нужно создать подсказку о том, что нужно ввести в этот текст, используйте атрибут *placeholder* или *title*, но не оставляйте ее между тегами `<textarea>` и `</textarea>`, иначе при отправке формы она будет отправлена на сервер.

Атрибуты:

- **autocomplete** = "on | off" – позволяет браузеру заполнять поле автоматически (значение *on*) или отключает эту возможность (значение *off*). Атрибут не поддерживается HTML4.
- **autofocus** – показывает, что элемент получает фокус при загрузке формы. Не поддерживается HTML4.
- **cols** = "число" – задает максимальное количество символов в строке. Большинство браузеров автоматически обертывают вводимые пользователем данные так, что каждая строка была меньше значения, заданного с помощью атрибута *cols*. Если

атрибут *wrap* содержит значение *hard*, то атрибут *cols* является обязательным.

- **dirname** = "имя поля ввода" – указывает имя поля *input*, которое включает инструкции по направлению текста (*ltr* или *rtl*) для области текста. Не поддерживается HTML4.
- **disabled** – деактивирует элемент управления. Дальнейшая активация возможна с помощью JavaScript.
- **form** = "идентификатор формы" – жестко связывает элемент управления с определенной формой. Не поддерживается HTML4.
- **maxlength** = "число" – задает максимальное количество символов, которые можно ввести в текстовую область. Не поддерживается HTML4.
- **name** = "имя" – задает имя элементов. Это имя передается приложению, которое обрабатывает форму.
- **placeholder** = "подсказка" – позволяет задать подсказку или пример, который поможет пользователю ввести корректные данные. Не поддерживается HTML4.
- **readonly** – элемент управления будет доступен только для чтения, его содержимое невозможно будет изменить.

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
Z



- **required** – элемент становится обязательным для заполнения. Если элемент не заполнен, отправка формы невозможна. Не поддерживается HTML4.
- **rows** = "число" – указывает высоту поля в знаках. Если пользователь ввел больше строк, то появляется полоска прокрутки, а само поле прокручивается вниз, чтобы принять новые строки.
- **wrap** = "hard | soft" – если этот атрибут имеет значение *hard*, то в конце строк по мере их появления в текстовом поле будут добавлены жесткие разрывы. При использовании этого значения обязательно нужно задать атрибут *cols*. Если значение *soft*, то символы разрыва не будут переданы на сервер.
- Глобальные атрибуты HTML.

Пример:

```
<textarea cols="50" rows="10">
Текстовая область
</textarea>
```



Рис. 50. Текстовая область

## Тег tfoot

Синтаксис:

```
<tfoot>...</tfoot>
```

Определяет нижний футер таблицы. Данный тег может включаться после элементов `caption`, `colgroup`, `thead`. Может появляться до тега **tbody** или после него. В таблице может быть только один элемент **tfoot**.

Атрибуты:

- Глобальные атрибуты HTML.

Пример:

см. *table*



# Ter th

Синтаксис:

```
<th>...</th>
```

Определяет ячейку заголовка таблицы. Текст в такой ячейке отображается браузером обычно жирным шрифтом и выравнивается по центру. Тег **th** должен размещаться внутри контейнера **tr**, который в свою очередь располагается внутри тега **table**. С точки зрения разметки ячейки **th** работают так же, как и ячейки **td**.

Атрибуты:

- **rowspan** = "число" — атрибут, отвечающий за объединение соседних строк в рамках одного столбца (т.е. соседних ячеек в столбце). В качестве своего значения принимает натуральные числа. По умолчанию установлен в значении, равном единице.
- **colspan** = "число" — атрибут, отвечающий за объединение соседних столбцов в рамках одной строки (т.е. соседних ячеек в строке). В качестве своего значения принимает натуральные числа. По умолчанию установлен в значении, равном единице.
- **headers** = "идентификатор ссылки" — атрибут, значением которого является список имен (id) ячеек, выступающих в роли заголовка (описания) для содержащей его ячейки.

- **scope** = "row | col | rowgroup | colgroup" – позволяет указать, к какой группе ячеек применяется заголовок. Например, при значении *row* ячейка заголовка применяет последовательные ячейки в одной и той же строке.
- **sorted** = "reversed|число больше 0" – обеспечивает возможность сортировки столбцов таблицы с помощью атрибута *sortable*. Только HTML5.1.
- Глобальные атрибуты HTML.

Пример:

```
<table>
  <tr>
    <th>Наименование</th>
    <th>Остаток, шт.</th>
    <th>Цена, р.</th></td>
  </tr>
  <tr>
    <td>Электронный конструктор Знаток
      "180 схем"</td>
    <td>2</td>
    <td>1840</td>
  </tr>
</table>
```

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
Z

А  
В  
С  
Д  
Е  
Ф  
Г  
Н  
И  
К  
Л  
М  
Н  
О  
Р  
Р  
Р  
С  
Т  
У  
У  
Х

## Ter thead

Синтаксис:

```
<thead>...</thead>
```

Определяет заголовок таблицы. Должен содержать хотя бы одну строку (tr) с элементами **th** или **td**. Может использоваться для дублирования заголовков, когда полная таблицы прерывается на страницах или для создания статического заголовка, который будет отображаться над основной частью прокручиваемой таблицы.

Атрибуты:

- Глобальные атрибуты HTML.

Пример:

*см. table*

## Ter time

Синтаксис:

```
<time>...</time>
```

Помечает текст внутри тега `<time>` как дата, время или оба значения. Может указываться непосредственно внутри контейнера `<time>`, либо задаваться через атрибут *datetime*. Может

использоваться для передачи распознаваемой компьютером информации о времени или преобразования информации о времени в другие форматы, например, можно преобразовать 5 р.п. в 17:00.

Атрибуты:

- `datetime` = "ГГГГ-ММ-ДДТчч:мм:ссTZD" – задает дату и время. Если задать этот атрибут, то контент тега **time** может быть пустым.
- Глобальные атрибуты HTML.

Пример:

Время работы: с `<time>9:00</time>` по `<time>18:00</time>`

## Тег title

Синтаксис:

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

Тег **title** задает название HTML-документа (интернет-страницы). Этот тег разметки, строго говоря, не является обязательным, однако его использование настоятельно рекомендуется. Он присваивает документу название, независимое от имени файла, которое отображается в строке заголовка браузера. Также это имя используется

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
X

по умолчанию при добавлении документа в папку "Избранное". Для документа, тег **title** которого не задан, браузер в качестве его имени будет использовать надпись "Без заголовка" либо полный адрес документа.

Открывающий и закрывающий теги TITLE являются обязательными. Его содержимое представляет текстовую строку неограниченной длины.

Название документа должно кратко характеризовать его содержание. Учитывая это, а также возможную минимизацию окна браузера, рекомендуется в названии документа ограничиться 50-60 символами. Содержание тега TITLE не должно включать в себя других тегов разметки. Например, нельзя с помощью тега **I** вывести название документа курсивом, т.е. запись:

```
<TITLE><I> название документа </I>
</TITLE>
```

недопустима.

Запись: `<I><TITLE> название документа </TITLE></I>`

также неприемлема, так как тег **I** не является тегом заголовка.

Атрибуты:

- Глобальные атрибуты HTML.

Пример:

```
<head>
  <title>Заголовок документа</title>
</head>
```

## Тег tr

Синтаксис:

```
<tr>...</tr>
```

Определяет строку ячеек внутри таблицы. Элемент **tr** может использоваться внутри элемента **table**, **thead**, **tfoot**, **tbody**. Внутри себя он может хранить элементы **td** и **th**.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

*см. table*

## Тег track

Синтаксис:

```
<track>
```

Позволяет авторам указать текстовую дорожку для медийных элементов **<audio>** и **<video>**.

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
X

Такая дорожка обычно содержит субтитры на разных языках, комментарии, заголовки и др. Не поддерживается HTML4.

Атрибуты:

- **default** – помечает данную дорожку как дорожку по умолчанию, если это не противоречит предпочтениям пользователя.
- **kind = "subtitles | captions | descriptions | chapters | metadata"** – позволяет уточнить назначение внешнего ресурса, а именно:

*subtitles* – субтитры.

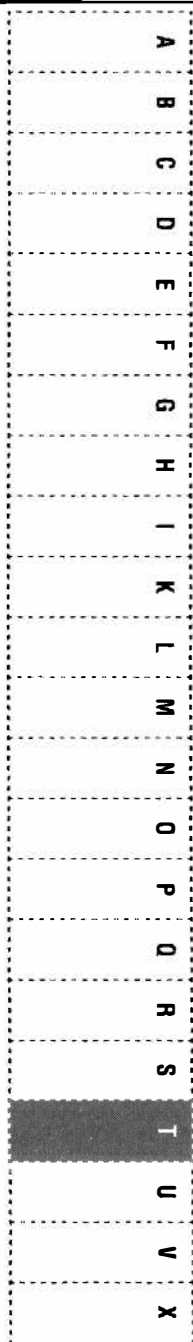
*captions* – полное описание аудио-файла, включая звуковые эффекты, для тех случаев, когда звук недоступен или для людей со слабым слухом.

*descriptions* – ресурс является аудио-дорожкой, описывающей видеосюжеты, которые в силу каких-то причин не могут быть просмотрены.

*chapters* – заголовки глав, используемых для навигации по мультимедиа-ресурсу.

*metadata* – дорожка, предназначенную для использования сценариями. По умолчанию subtitles.

- **label** = "текстовая строка" – указывает заголовок для дорожки, который может быть отображен браузером.
- **src** = "URL" – указывает адрес ресурса.



- **srclang** = "двухбуквенный код языка" – указывает язык данных текстовой дорожки (см. прил. 2).
- Глобальные атрибуты HTML.

Пример:

```
<video src="film.mp4">  
  <track kind="subtitles" src="film.  
en.vtt" srclang="en" label="English">  
  <track kind="subtitles" src="film.  
ru.vtt" srclang="ru" label="Русский">  
</video>
```

Тег u

Синтаксис:

```
<u>...</u>
```

Добавляет подчеркивание к тексту. Этот тег осуждается спецификацией HTML, взамен рекомендуется использовать стили. Помните, что подчеркнутый текст легко спутать с гиперссылкой, поэтому использовать этот элемент не рекомендуется.

Атрибуты:

- Глобальные атрибуты HTML

A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
X



Пример:

```
<u>Подчеркнутый</u> текст
```

## Тег ul

Синтаксис:

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

Создает маркированный список элементов, в котором элементы (li) не упорядочиваются. По умолчанию браузеры отображают элементы в таких списках с маркерами в виде кружочков.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<ul>
  <li>Абрикос</li>
  <li>Апельсин</li>
  <li>Банан</li>
  <li>Киви</li>
  <li>Яблоко</li>
</ul>
```



*Рис. 51. Пример маркированного списка*

Изменить тип маркированного списка можно посредством CSS-свойства `list-style-type`:

**circle** — маркер в виде кружка.

**disc** — маркер в виде точки.

**square** — маркер в виде квадрата.

Пример:

```
<html>
<head>
  <meta charset="utf-8">
  <title>list-style-type</title>
  <style>
    ul {
      list-style-type: square; /* Квадрат
*/
    }
  </style>
</head>
<body>
```



```

<ul>
    <li>Абрикос</li>
    <li>Апельсин</li>
    <li>Банан</li>
    <li>Киви</li>
    <li>Яблоко</li>
</ul>

```

```

</body>

```

```

</html>

```



**Рис. 52. Изменение типа списка**

## Тег var

Синтаксис:

```

<var>...</var>

```

Тег VAR — отмечает заключенный в себе текст, как экземпляр переменной или аргумента какой-либо программы. Браузерами по умолчанию отображается курсивом.

Атрибуты:

- Глобальные атрибуты HTML.

## Тег video

Синтаксис:

```
<video>...</video>
```

Посредством элемента *video* осуществляется внедрение видеофайла в страницу. Не поддерживается HTML4. Адрес видеоресурса указывается посредством атрибута *src*. Используя тег **source**, есть возможность задать несколько источников внутри тега **video**. Также внутри **video** можно задать альтернативный контент – на тот случай, если браузер не поддерживает его.

**Таблица 2. Кодеки и браузеры**

Браузер	IE	Chrome	Opera	Safari	Firefox
ogg/vorbis	-	+	+	-	+
wav	-	-	+	+	+
mp3	+	+	-	+	-
AAC	+	+	-	+	-
ogg/theora	-	+	+	-	+
H.264	+	+	-	+	-
WebM	-	+	+	-	+

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
X

## Атрибуты:

- **autoplay** – автовоспроизведение видео.
- **controls** – указывает, что браузер должен показать набор элементов управления (кнопки Play/Pause, ползунок, кнопки управления позицией и некоторые другие).
- **height** = "число" – задает высоту окна видеопроигрывателя в пикселях.
- **loop** – указывает на то, что в мультимедийный файл должен воспроизводиться с начала после завершения.
- **mediagroup** = "текст" – устанавливает ссылку на несколько мультимедийных элементов путем присваивания им одного и того же значения *mediagroup*.
- **muted** – отключает звук.
- **poster** = "URL" – задает адрес постера – графического файла, который выводится как заполнитель перед началом воспроизведения видео.
- **preload** = "none | metadata | auto" – говорит браузеру, должен ли мультимедиа-файл загружаться автоматически. Значение *none* говорит о том, что автоматически загружать файл не нужно. Значение *metadata* не приводит к загрузке медиа-файла, загружаются только мета-данные. Значение *auto* говорит о том, что файл должен быть

загружен автоматически, но окончательное решение – за браузером.

- **src** = "URL-ссылка" – адрес видео-файла.
- **width** = "число" – ширина окна видеопроигрывателя в пикселях.
- Глобальные атрибуты HTML.

Пример:

```
<video      src="film.avi"      width="640"
height="480"
      poster="poster.jpg"      type="video/
ogg; codecs='theora, vorbis'">
      Ваш браузер не поддерживает тег
video
</video>
```

## Тег wbr

Синтаксис:

```
<wbr>
```

Задаёт точку разрыва в длинном слове. При необходимости строка будет разорвана именно в том месте, где стоит данный тег. Не поддерживается HTML4.

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
W  
X

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<p>Самое длинное слово из химии</p>
  <p class="word">метоксихлоп<br>диэтилам
ино<br>метил
<br>бутил<br>амино<br>акридин</p>
```

## Тег xmp

Синтаксис:

```
<xmp>...</xmp>
```

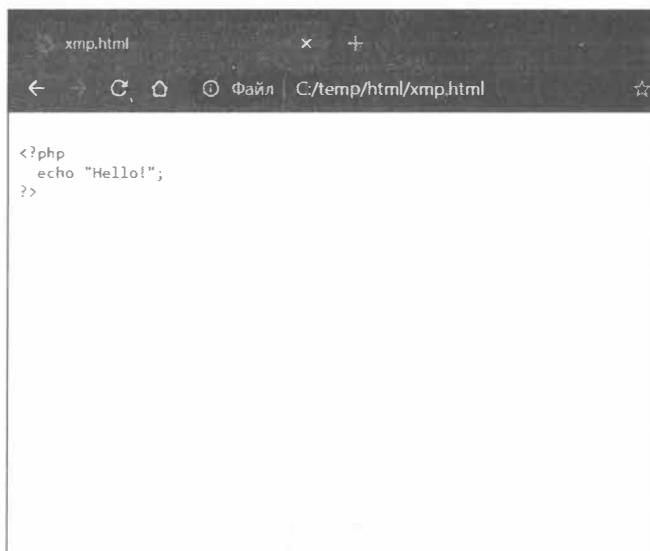
Отображает содержимое контейнера "как есть" и шрифтом фиксированной ширины. Пока тег `<xmp>` не закрыт, все теги внутри него отображаются как обычный текст. Идеально подходит для вывода программных кодов, поскольку все символы никак не преобразуются браузером.

Атрибуты:

- Глобальные атрибуты HTML

Пример:

```
<xmp>
<?php
    echo "Hello!";
?>
</xmp>
```



**Рис. 53. Использование тега *xmp***

Также рекомендуем ознакомиться с прилагаемыми к книге приложениями, из которых вы узнаете дополнительную полезную информацию, а именно HTML-коды цветов, двухбуквенные коды языков и коды MIME-типов.

A  
B  
C  
D  
E  
F  
G  
H  
I  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
X

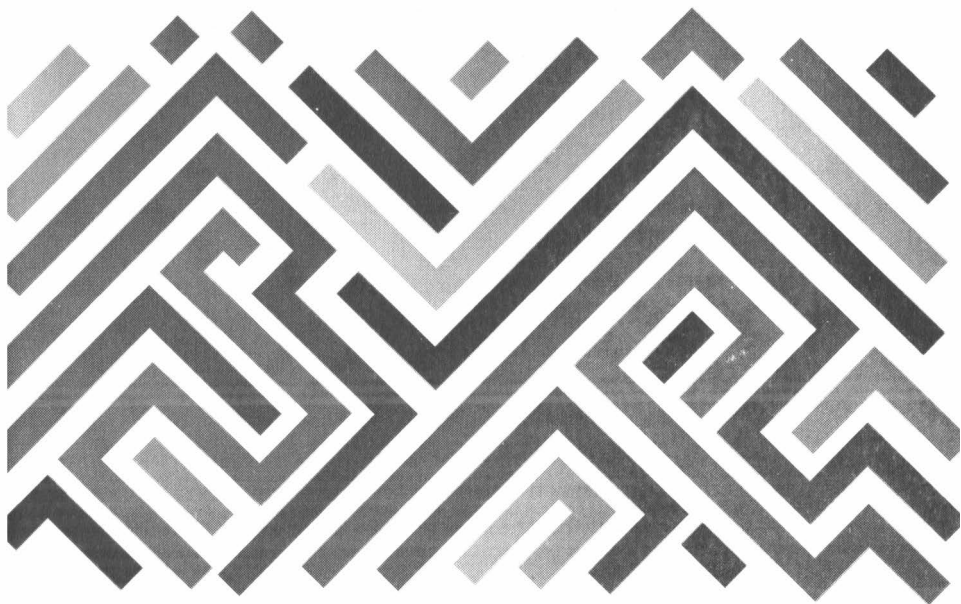




**HTML**

## **Часть 3.**

# **Семантическая верстка на HTML**



## 3.1. Базовые абстракции разметки. Верстка на основе блоков **DIV**

В общем случае все HTML-элементы подразделяются на элементы уровня блока (или блокообразующие элементы) и на строковые элементы. Различие между ними заключается в следующем:

- Элементы уровня блока представляют собой ярко выраженные элементы структуры. В своем содержимом такие элементы могут иметь как просто какую-либо информацию, так и другие блокообразующие или строковые элементы. Элементом уровня блока, например, является элемент абзаца `P`, в содержимом которого могут быть заданы другие абзацы, списки, таблицы и т.д., которые будут по отношению к нему дочерними.
- Строковые элементы представляют собой не столько элемент структуры, сколько саму информацию, обладающую определенным свойством (их еще называют элементами встраиваемой информации). Например, элемент `I` не создает структурное подразделение, а просто утверждает, что текст в его содержимом должен отобра-

жаться курсивом. В содержимом строковых элементов не могут быть блокообразующие элементы, только строковые.

Разность между строковыми элементами и элементами уровня блока хорошо заметна при использовании каскадных таблиц стилей (CSS): под блокообразующие элементы выделяются стилевые блоки (прямоугольные области экрана, обладающие определенными визуальными свойствами).

Строка браузера, в которой присутствует элемент уровня блока, не может содержать ничего, кроме него. Например, если на строке расположен заголовок, то на ней уже не может быть никакой другой информации.

Универсальным элементом уровня блока является элемент DIV, универсальным строковым элементом - SPAN. Смысл их существования заключается в том, что они не имеют никаких своих специфических атрибутов и свойств, а основным своим назначением имеют привнесение в HTML-документ структуры. Выделять некую информацию в блок осуществляется с той целью, чтобы потом к ней можно было бы целенаправленно обратиться: задать определенное свойство с помощью каскадных таблиц стилей, использовать при работе сценария и т.п. Использование обычных, не универсальных элементов зачастую бывает не совсем удобно ввиду того, что они, в соответствии со своим назначением, имеют свои

особенности. Например, блокообразующий элемент абзаца P выделяет свое содержимое сверху и снизу дополнительными пустыми строками.

Элементы DIV и SPAN обладают одинаковым набором необязательных глобальных атрибутов (см. конец данной главы)

## **3.2. Семантические теги HTML5.**

### **Верстка на основе семантической разметки**

В последних релизах языка верстки HTML, в особенности версии 5, была добавлена поддержка большого количества семантических тегов. Данное изменение имеет наибольшее значение для верстальщиков, веб-разработчиков и других пользователей данного языка. Использование семантических тегов, позволяет сделать разметку более читаемой, более понятной человеку, что упрощает редактирование и поддержку подобных файлов. С другой стороны, для браузеров, в большинстве случаев, использование данных тегов аналогично использованию тега <div>.

Использование семантических тегов позволяет оценить качество выполненной верстки. Срав-

ните два одинаковых фрагменты разметки. В первом случае, не используются семантические теги:

```
<div>
  <div>
    ...
  </div>
  <p>...</p>
</div>
<div>
  <h3>...</h3>
  <a>...</a>
</div>

<div>
...
</div>
```

Попытка проанализировать данный фрагмент разметки с точки зрения семантики практически невозможно, т.к. подобная структура может встречаться практически в любой области HTML документа, например, в шапке, в футере и в основном теле странице – теге `<body>`.

Во втором случае, используются семантические теги:

```
<section>
  <nav>
    ...
  </nav>
  <p>...</p>
</section>

<aside>
  <h3>...</h3>
  <a>...</a>
```

```
</aside>

<article>
...
</article>
```

Анализируя данный фрагмент разметки, можно сделать несколько выводов:

1. В разметке имеется три самостоятельных блока `<section>`, `<aside>`, `<article>`
2. Блок `<section>` представляет собой автономный раздел страницы. Если проводить аналогии со страницей-лэндингом, то это может быть раздел "Наши преимущества" и раздел "Наши услуги"
3. В блоке `<aside>` описано содержимое, которое может быть связано только косвенно с основным содержанием страницы. Наиболее частое применение – разметка боковых панелей.
4. Блок `<article>` содержит часть документа, которая может быть использована повторно в любой другой части документа, т.к. представляет собой самостоятельный фрагмент. В качестве примера можно привести запись в блоге или фрагмент статьи.

Далее приведем описание тегов семантической разметки HTML5, а в главе, посвященной рассмотрению фронтенда практического примера, будет показано их практическое применение. Обратите внимание, что сами семантические теги практически никак не определяют внешний вид своего содержимого. Например, классический тег BODY является по своей сути семантическим тегом – он говорит, что все, что находится между его открывающим и закрывающим тегами является телом html-документа. И все. Внешний вид можно настраивать с помощью стилевых CSS-таблиц, которые применяются к этому тегу, но сам тег этим "не занимается".

- Тег ARTICLE – своими открывающим и закрывающим тегами определяет самостоятельную часть документа, страницы или сайта, предназначенную для независимого использования и обособленного восприятия – это может быть статья в блоге, статья в журнале или газете, какой-либо другой самостоятельный фрагмент содержимого. Теги ARTICLE могут быть вложенными.
- Тег ASIDE представляет собой часть документа, чье содержимое только косвенно связано с основным содержимым документа. Чаще всего представлен в виде боковой панели, сносок или меток. Внешний вид содержимого тега ASIDE (а между его открывающим и закрывающим тегами может быть все что угодно в плане HTML) никак не определяется самим тегом



ASIDE, данным тегом просто придается определенный смысл фрагменту html-документа, а выделяется этот фрагмент. А вот как именно он будет отображен – в виде боковой панели, сноски или еще как-то – это вы уже сами зададите с помощью таблиц стилей.

Пример:

```
<article>
```

```
<p>
```

Стандарт управления проектами РМР  
был разработан американским институтом  
PMI

```
</p>
```

```
<aside>
```

```
<p>
```

Что примечательно, создатели  
первой версии стандарта потом отказались  
от последующих версий РМР, так как,  
по их мнению, те не соответствуют  
первоначальной идее стандарта.

```
</p>
```

```
</aside>
```

```
<p>
```

Стандарт РМР содержит в себе следующие  
области знаний ...

```
</p>
```

```
</article>
```

- Тег **HEADER** – своими открывающим и закрывающим тегами определяет заголовочную часть HTML-документа. Обычно в этой части (в качестве содержимого тега **HEADER**) располагаются теги навигации, изображение логотипа, заголовки с названием сайта и т.п. Внутри содержимого мо-

гут быть любые теги за исключением тегов HEADER и FOOTER.

- Тег FOOTER представляет собой нижний колонтитул (футер, подвал), содержащий информацию о документе, его авторе, контакты и т.п. Такая область (набор тегов) ограждается открывающим и закрывающим тегами FOOTER и называется "подвалом" или "футером". Подвал обычно располагается внизу интернет-страницы (html-документа).
- Тег SECTION – своим открывающим и закрывающим тегами задает раздел – тематическую группу содержимого интернет-страницы – который должен восприниматься обособленно. Секция может иметь свою собственную обособленную структуру, в частности в секции может быть свой заголовок (тег HEADER) и подвал (тег FOOTER).
- Тег NAV своим открывающим и закрывающим тегами задает область html-документа, в которой собраны теги (ссылки) навигации (как внутри текущего документа, так и ведущих на другую страницу).

Пример семантической верстки:

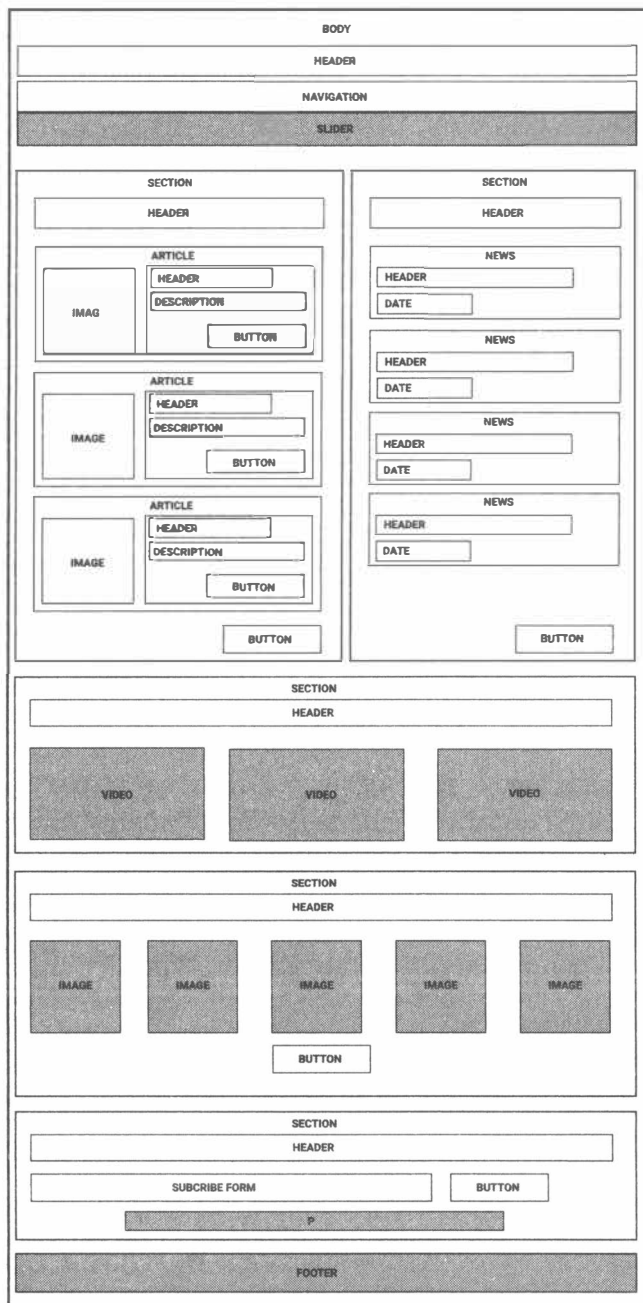
```
<html>
  <head>
    <title> Название документа </title>
  </head>
  <body>
```

```
<header>
  Шапка сайта
</header>
<nav>
  Навигация
  <ul>
    <li><a href="#">Ссылка 1</a></li>
    <li><a href="#">Ссылка 2</a></li>
  </ul>
</nav>
<section>
  Секция 1
  <article>
    <h1>Заголовок статьи</h1>
    <p>Контент</p>
    <h2>Подзаголовок статьи</h2>
  </article>
</section>
<section>
  Секция 2
  <article>
    <h1>Заголовок статьи 2</h1>
    <p>Контент</p>
    <h2>Подзаголовок статьи</h2>
  </article>
</section>

<footer>
  <p>Дата, адрес, авторство</p>
</footer>
</body>
</html>
```

### 3.3. Пример сематической верстки

В качестве примера рассмотрим семантическую верстку главной страницы интернет-портала, структурная схема которой представлена на рисунке:



Далее разберем верстку главной страницы, при этом подробно рассмотрим наиболее крупные структурные блоки:

- блок HEAD
- блок HEADER
- блоки SECTION
- блок FOOTER

В общем виде, структура главной страницы имеет вид, представленный на рисунке выше.

Основное содержимое страницы, отображающееся в браузере, расположено внутри тега BODY.

### 3.3.1. Блок HEAD

В данном блоке описывается основная информация о странице, указывается ее заголовок, перечисляются подключаемые стилевые файлы, шрифты, скрипты, прописываются дополнительные теги META, используемые для SEO-оптимизации страницы в поисковых системах.

Код данного блок приведен в листинге 3.1.

#### Листинг 3.1. Код блока HEAD

```
<head>
```

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

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

```
<title>НИТ -учебный центр по управлению проектами</title>

<link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@100;200;300;400;500;600;700;800;900&display=swap" rel="stylesheet">

<link href="https://fonts.googleapis.com/css2?family=PT+Serif:ital,wght@0,400;0,700;1,400;1,700&display=swap" rel="stylesheet">

<link rel="stylesheet" href="assets/css/style.css">

<link rel="stylesheet" href="assets/css/normalize.css">

<link rel="stylesheet" href="assets/css/plugins/font-awesome.min.css">

<link rel="stylesheet" href="assets/css/plugins/swiper-bundle.css">

</head>
```

Разберемся, какие внутренние и сторонние ресурсы подключаются к странице нашего сайта, а также, какие настройки мы указываем для работы. Разбор блока приведен в таблице 3.3.

**Таблица 3**  
**Разбор блока HEAD**

Тег	Описание, Назначение
<code>&lt;meta charset="UTF-8"&gt;</code>	Указывает на используемую на странице кодировку текста UTF-8
<code>&lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;</code>	<p>Определяет, как страница будет отображаться на дисплеях разного разрешения – адаптировать свое содержимое, либо сохранять пропорции и размеры.</p> <p>В данном случае фрагмент <code>content="width=device-width, initial-scale=1.0"</code> указывает на то, что ширина страницы будет адаптироваться под ширину устройства</p>
<code>&lt;title&gt;НИТ -учебный центр по управлению проектами&lt;/title&gt;</code>	В данном теге определяется название страницы, которое будет указываться в адресной строке браузера
<code>&lt;link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@100;200;300;400;500;600;700;800;900&amp;display=swap" rel="stylesheet"&gt;</code>	В данном теге подключается шрифт Montserrat. Ссылку для подключения можно сгенерировать на странице <a href="https://fonts.google.com">fonts.google.com</a> . Данный шрифт относится к категории Sans Serif, т.е. шрифт без засечек. На нашем сайте, данный шрифт используется в основном для заголовков разного уровня, кнопок, пунктов меню и т.п.
<code>&lt;link href="https://fonts.googleapis.com/css2?family=PT+Serif:ital,wght@0,400;0,700;1,400;1,700&amp;display=swap" rel="stylesheet"&gt;</code>	В данном теге подключается шрифт PT Serif. Ссылку для подключения можно сгенерировать на странице <a href="https://fonts.google.com">fonts.google.com</a> . Данный шрифт относится к категории Serif, т.е. шрифт с засечками. На нашем сайте, данный шрифт используется в основном для больших текстовых блоков, например, содержании статьи, описание презентации и т.п.

<pre>&lt;link rel="stylesheet" href="assets/css/style.css"&gt;</pre>	<p>В данном теге подключается основной стилевой файл <code>style.css</code>. Если упустить данный тег, то написанные нами стили попросту не применятся к странице</p>
<pre>&lt;link rel="stylesheet" href="assets/css/normalize. css"&gt;</pre>	<p>В данном теге подключается дополнительный стилевой файл, который называется <code>normalize.css</code>. В данном файле содержатся "нормализующие" стили для многих базовых тегов, улучшающих их отображение в разных браузерах. Скачать файл можно по ссылке: <a href="https://necolas.github.io/normalize.css/">https://necolas.github.io/normalize.css/</a></p>
<pre>&lt;link rel="stylesheet" href="assets/css/plugins/font- awesome.min.css"&gt;</pre>	<p>В данном теге подключается дополнительный стилевой файл иконочного шрифта Font Awesome. Подключение данных таблиц стилия позволяет нам упростить использование на нашем сайте иконок из популярной библиотеки. Узнать подробнее о Font Awesome и способах его подключения к сайту можно по ссылке: <a href="https://fontawesome.ru/">https://fontawesome.ru/</a></p>
<pre>&lt;link rel="stylesheet" href="assets/css/plugins/ swiper-bundle.css"&gt;</pre>	<p>Данный тег указывается только для страниц, где используется подключаемый сторонний слайдер Swiper. В данном теге мы подключаем таблицу стилей для этого слайдера. Подробнее о подключении слайдера Swiper к нашему сайту будет рассказано в главе про JavaScript.</p>

### 3.3.2. Блок HEADER

В данном блоке содержится структура главного меню страницы и всего сайта в целом. Как и на многих сайтах, для разных страниц меню остается единообразным, меняются только стили активных



пунктов меню. Т.к. меню – это видимая часть сайта, которая отображается пользователю, то данные и последующие блоки располагаются внутри тега BODY. Код данного блока представлен в листинге 3.2:

### Листинг 3.2. Код блока HEADER:

```
<header class="main-header">
  <div class="container">
    <div class="header-top">
      <a href="http://nit.center/" class="main-logo-link">
        
      </a>
      <div class="header-top-content col-lg-9">
        <ul class="contacts-list">
          <li class="contacts-item">
            <a href="mailto:info@nit.center"
class="contacts-link"> info@nit.center
            </a>
          </li>
          <li class="contacts-item">
            <a href="tel:+78123090686"
class="contacts-link"> +7(812)000-00-00
            </a>
          </li>
          <li class="contacts-item">
            <a href="tel:+74992838606"
class="contacts-link"> +7(499)000-00-00
            </a>
          </li>
        </ul>
        <ul class="tags-list">
          <li class="tags-item">
```

```
<a href="http://nit.center/presentation/6hats"
class="tag-link"> Шесть думающих шляп
    </a>
</li>
<li class="tags-item">
    <a href="http://nit.center/landing.php" class="tag-
link"> Сдай РМР успешно
        </a>
    </li>
<li class="tags-item">
    <a href="http://nit.center/articles/kernel"
class="tag-link"> Инновации
        </a>
    </li>
<li class="tags-item">
    <a href="http://nit.center/education.php"
class="tag-link"> Обучение
        </a>
    </li>
<li class="tags-item">
    <a href="http://nit.center/news.php"
class="tag-link"> Новости
        </a>
    </li>
</ul>
</div>
<button class="navbar-toggler">
    <span class="toggler-icon"></span>
    <span class="toggler-icon"></span>
    <span class="toggler-icon"></span>
</button>
</div>
</div>
```

```
<div class="navigation-wrapper">
  <div class="container">
    <nav class="main-navigation">
      <ul class="main-navigation-list">
        <li class="main-navigation-item">
          <a href="http://nit.center" class="main-navigation-
link" active>Главная</a></li><li class="main-navigation-
item">
          <a href="http://nit.center/education.php"
class="main-navigation-link" >Направления обучения</a></
li><li class="main-navigation-item">
          <a href="http://nit.center/landing.php"
class="main-navigation-link" >Сертификация РМР</a></li><li
class="main-navigation-item">
          <a href="http://nit.center/archive.php"
class="main-navigation-link" >Архив журнала Управление</
a></li><li class="main-navigation-item">
          <a href="http://nit.center/articles.php"
class="main-navigation-link" >База знаний</a>
          <ul class="submenu-list"><li class="main-
navigation-item">
            <a href="http://nit.center/articles.
php" class="main-navigation-link">Статьи</a></li><li
class="main-navigation-item">
            <a href="http://nit.center/presentations.
php" class="main-navigation-link">Презентации</a></li></
ul>
          </li><li class="main-navigation-item">
            <a href="http://nit.center/about.php" class="main-
navigation-link" >О компании</a></li>
          </ul>
        </nav>
      </div>
    </div>
  </header>
```

### 3.3.3. Блок SECTION – слайдер

В данном блоке содержится код слайдера для главной страницы. Особенность слайдера заключается в наличии миниатюр слайдов в боковой части, с помощью которых также доступна навигация между слайдами. Для "оживления" слайдера используется библиотека одного из популярных слайдеров Swiper. Если внимательно изучить код верстки данного блока, то можно увидеть, что у некоторых тегов присутствует класс, имеющий в названии составную "swiper". Это значит, что данные блоки структуры слайдера, соответствуют документации Swiper. Листинг данного блока:

#### Листинг 3.3. Блок SECTION – слайдер

```
<section class="slider-section">
  <h1 class="visually-hidden"></h1>
  <div class="container">
    <div class="slider">
      <div class="slider-col-1 main-slider-thumbs-container">
        <ul class="preview-list swiper-wrapper">
          <li class="preview-item swiper-slide"
active data-slide-to="1">
            
          </li><li class="preview-item swiper-slide"
data-slide-to="2">
            
          </li><li class="preview-item swiper-slide"
data-slide-to="3">
            
          </li>
        </ul>
      </div>
    <div class="slider-col-2">
```

```
<div class="slider-item-field main-slider-container">
  <div class="slider-item-wrapper swiper-wrapper">
    <div class="slider-item swiper-slide">
      <a target=_blank href="http://nit.center/">
        
      </a>
    </div><div class="slider-item swiper-slide">
      <a target=_blank href="http://nit.center/">
        
      </a>
    </div><div class="slider-item swiper-slide">
      <a target=_blank href="http://nit.center/landing.php">
        
      </a>
    </div>

  </div>
  <div class="slider-navigation">
    </div>
  </div>
</div>
</section>
```

### 3.3.4. Блок SECTION – Статьи и новости

Данный блок содержит в себе две колонки, в которых представлены списки: в левом – список со статьями, в правом – список новостей. В листинге данного блока указано по одному элементу для статей и новостей, чтобы не перегружать листинг однотипными элементами.

#### Листинг 3.4.

```
<div class="articles-news-books-area row-section-field">
  <div class="row-sections-wrapper container">
    <div class="row">
      <section class="useful-articles double-
block">
        <div class="articles-row row">
          <div class="section-title">
            <span class="sub-title">Статьи</
span>
            <h2>Полезные статьи</h2>
          </div>
        </div>
        <ul class="articles-list">
          <li class="articles-item">
            <div class="article-image">
              
```

```

        </div>
        <div class="article-description">
            <h3>ФОРМАТ ПРОЕКТНОЙ ДЕЯТЕЛЬНОСТИ
ПО РЕАЛИЗАЦИИ БИЗНЕС-ПРОЕКТОВ В РАМКАХ ОБРАЗОВАТЕЛЬНОГО
ПРОЦЕССА СОГЛАСНО МОДЕЛИ УНИВЕРСИТЕТА 3.0.</h3>
            <p>
                В данной статье описан опыт
реализации бизнес-проекта в рамках образовательного
процесса. На примере конкретного проекта разобрана
среднесрочная форма проектной деятельности,
рекомендованная авторами к внедрению в образовательный
процесс в ВУЗах РФ. Приве
            </p>
            <a href="http://nit.center/
article/format_proektnoi_deyatel_nosti_po_realizatsii_
biznes_proektov" class="button">Читать далее</a>
        </div>
    </li>
    ...

</ul>
<div class="show-more-wrapper">
    <a href="articles.php"
class="colored-btn">Читать все статьи</a>
</div>
</section>
<div class="double-block">
    <section class="news">
        <div class="news-row row">
            <div class="section-title">
                <span class="sub-
title">Новости мира СПО</span>
                <h2>Новости</h2>
            </div>

```

```
</div>
<ul class="news-list">
  <li class="news-item">
    <a href="http://nit.center/
news/53f57dcd-616b-11eb-a414-ac1f6bbd4544" class="news-
header">
        Выручка московских театров
выросла почти на 60% после смягчения ограничений
    </a>
    <span class="news-
date">2021-01-28</span>
  </li>
  ...

</ul>
<div class="show-more-wrapper">
  <a href="news.php"
class="colored-btn">Все новости</a>
</div>
</section>
</div>
</div>
</div>
```

### 3.3.5. Блок SECTION – видео

В данном блоке расположены видео в три колонки. Из нового, для главной страницы, можно отметить использование тегов IFRAME. Код блока представлен в листинге 3.5:



**Листинг 3.5.**

```
<section class="video-section">
  <div class="container">
    <div class="video-wrapper">
      <div class="video-row row">
        <div class="section-title col-lg-6">
          <span class="sub-title">Видео</span>
          <h2>Видео про управление от
университета Синергия</h2>
        </div>
      </div>
      <ul class="video-content-row row">
        <li class="tetra-block col-lg-4">
          <iframe height="228" src="https://
www.youtube.com/embed/AwfkCcdldKw" frameborder="0"
allow="accelerometer; encrypted-media; gyroscope;
picture-in-picture" allowfullscreen></iframe>
        </li><li class="tetra-block col-lg-4">
          <iframe height="228" src="https://
www.youtube.com/embed/6Aw5hRKZg4g" frameborder="0"
allow="accelerometer; encrypted-media; gyroscope;
picture-in-picture" allowfullscreen></iframe>
        </li><li class="tetra-block col-lg-4">
          <iframe height="228" src="https://
www.youtube.com/embed/leZJps7wwYw" frameborder="0"
allow="accelerometer; encrypted-media; gyroscope;
picture-in-picture" allowfullscreen></iframe>
        </li>
      </ul>
    </div>
  </div>
</section>
```

### 3.3.6. Блок SECTION – Архив журнала

Данный блок представляет собой пятиколоночную секцию, каждый элемент-журнал представляет собой картинку с текстовым блоком-подписью. В данном листинге также не указаны повторяющиеся элементы списка, обернутые в тег `<li>`. Листинг данной секции:

#### Листинг 3.6.

```
<section class="management-archive">
  <div class="management-archive-shape">
    
  </div>
  <div class="container">
    <div class="management-archive-row row">
      <div class="section-title">
        <span class="sub-title">Все выпуски журнала</span>
        <h2>Архив журнала "Управление"</h2>
      </div>
    </div>
    <ul class="penta-wrapper row">
      <li class="penta-block management-archive-item">
        <div class="management-archive-item-wrapper">
          
          <span class="archive-number">Выпуск 3/2018</span>
          <a target=_blank href="http://nit.center/lf/
pdf/2018-3.pdf" class="download-issue-btn">
            Скачать в PDF
          </a>
        </div>
      </li>
    </ul>
  </div>
</section>
```

```
        </li>
        ...
    </ul>
    <div class="show-more-wrapper">
        <a href="archive.php" class="colored-
btn">Перейти в архив</a>
    </div>
</div>
</section>
```

### 3.3.7. Блок SECTION – подписка

Наличие на сайте формы обратной связи для клиентов – это правило хорошо тона для любого разработчика. Т.к. большинство сайтов или интернет-порталов нацелены на привлечение клиента и побуждение его на совершение покупки, то такой элемент, как форма обратной связи просто необходим. Для создания форм используется тег FORM, в котором располагаются поля для ввода пользователем данных, создаваемые тегом INPUT. Листинг данного блока:

#### Листинг 3.7:

```
<section class="subscribe">
    <div class="container">
        <div class="subscribe-wrapper">
            <div class="subscribe-row row">
                <div class="section-title">
                    <h2>Подписаться на рассылку</h2>
                </div>
            </div>
            <div class="subscribe-row row">
                <div class="col-lg-8">
```

```
<form class="subscribe-form"
name="subscribe-form" method="POST">
    <label>
        <input type="email"
name="subscribe-email" placeholder="Введите ваш почтовый
ящик" maxlength="40" required>
    </label>
    <div class="subscribe-button-wrapper">
        <button class="subscribe-button
colored-btn" type="submit">Подписаться</button>
    </div>
</form>
<p class="subscribe-description">
    Один раз в месяц мы делимся с
подписчиками самой актуальной и полезной информацией
по управлению проектами: новости, лайфхаки и секреты
подготовки к РМР, успешные кейсы. Просто оставьте свой
адрес электронной почты и убедитесь сами!
</p>
</div>
</div>
</div>
</section>
```

### 3.3.8. Блок FOOTER

Последний блок на данной странице – это футер. В футерах сайтов обычно указывают контактную информацию, ссылки на профили в социальных сетях, иногда дублируют основное меню, либо делают отдельное меню для футера. В нашем проекте в футере содержится название сайта, а также набор ссылок на профи-

ли в социальной сети. Как и другие блоки, данный блок обернут в семантический тег `FOOTER`, который позволяет понять разработчикам и браузерам его назначение. Листинг данного блока:

### Листинг 3.7:

```
<footer class="main-footer">
  <div class="container">
    <div class="footer-wrapper">
      <p class="footer-credits">nit.center © 2020 -
2021</p>
      <ul class="social-list">
        <li class="social-item">
          <a href="" aria-label="Вконтакте">
            <svg xmlns="http://www.
w3.org/2000/svg" width="27" height="15" viewBox="0 0
26.14 14.91">

            </svg>

          </a>
        </li>
        <li class="social-item">
          <a href="" aria-label="Instagram">
            <svg xmlns="http://www.
w3.org/2000/svg" width="20" height="20" viewBox="0 0 20
20">

            </svg>

          </a>
        </li>
        <li class="social-item">
          <a href="" aria-label="YouTube">
```

```
        <svg xmlns="http://www.
w3.org/2000/svg" width="20" height="20" viewBox="0 0 512
352">

            </svg> </a>

        </li>
        <li class="social-item">
            <a href="" aria-label="Facebook">
                <svg xmlns="http://www.
w3.org/2000/svg" width="19" height="22" viewBox="0 0
10.15 21.74">

                    </svg>

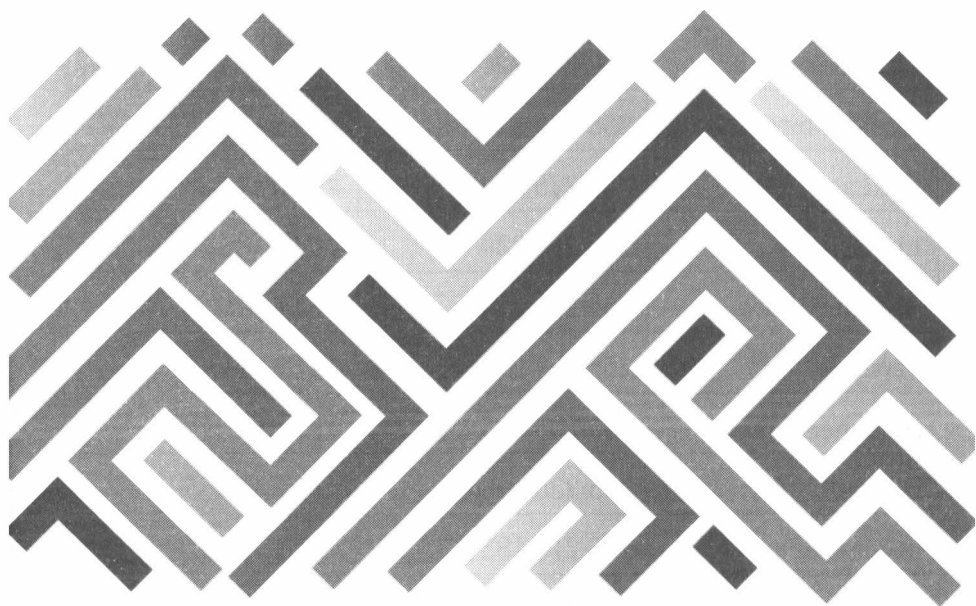
                </a>
            </li>

        </ul>
    </div>
</div>
</footer>
```

В данном блоке, иконки социальных сетей созданы с помощью интеграции кода SVG-изображений непосредственно в верстку страниц. Данная реализация является одним из способов добавления на страницу векторных изображений, которые можно легко редактировать с помощью стилевых файлов. Однако из-за объемистости кода SVG изображений в листинге 3.7 он опущен.



# Приложения





# Приложение 1. Стандартные цвета

Таблица 1. RGB и HTML-значения для стандартного цвета

Цвет	R G B	HTML-код
Snow	255 250 250	#FFFAFA
GhostWhite	248 248 255	#F8F8FF
WhiteSmoke	245 245 245	#F5F5F5
Gainsboro	220 220 220	#DCDCDC
FloralWhite	255 250 240	#FFFAF0
OldLace	253 245 230	#FDF5E6
Linen	250 240 230	#FAF0E6
AntiqueWhite	250 235 215	#FAEBD7
PapayaWhip	255 239 213	#FFEFD5
BlanchedAlmond	255 235 205	#FFEBCD
Bisque	255 228 196	#FFE4C4
PeachPuff	255 218 185	#FFDAB9
NavajoWhite	255 222 173	#FFDEAD
Moccasin	255 228 181	#FFE4B5

Cornsilk	255 248 220	#FFF8DC
Ivory	255 255 240	#FFFFF0
LemonChiffon	255 250 205	#FFFACD
Seashell	255 245 238	#FFF5EE
Honeydew	240 255 240	#F0FFF0
MintCream	245 255 250	#F5FFFA
Azure	240 255 255	#F0FFFF
AliceBlue	240 248 255	#F0F8FF
lavender	230 230 250	#E6E6FA
LavenderBlush	255 240 245	#FFF0F5
MistyRose	255 228 225	#FFE4E1
White	255 255 255	#FFFFFF
Black	0 0 0	#000000
DarkSlateGray	47 79 79	#2F4F4F
DimGrey	105 105 105	#696969
SlateGrey	112 128 144	#708090
LightSlateGray	119 136 153	#778899
Grey	190 190 190	#BEBEBE
LightGray	211 211 211	#D3D3D3
MidnightBlue	25 25 112	#191970

NavyBlue	0 0 128	#000080
CornflowerBlue	100 149 237	#6495ED
DarkSlateBlue	72 61 139	#483D8B
SlateBlue	106 90 205	#6A5ACD
MediumSlateBlue	123 104 238	#7B68EE
LightSlateBlue	132 112 255	#8470FF
MediumBlue	0 0 205	#0000CD
RoyalBlue	65 105 225	#4169E1
Blue	0 0 255	#0000FF
DodgerBlue	30 144 255	#1E90FF
DeepSkyBlue	0 191 255	#00BFFF
SkyBlue	135 206 235	#87CEEB
LightSkyBlue	135 206 250	#87CEFA
SteelBlue	70 130 180	#4682B4
LightSteelBlue	176 196 222	#B0C4DE
LightBlue	173 216 230	#ADD8E6
PowderBlue	176 224 230	#B0E0E6
PaleTurquoise	175 238 238	#AFEEEE
DarkTurquoise	0 206 209	#00CED1
MediumTurquoise	72 209 204	#48D1CC

Turquoise	64 224 208	#40E0D0
Cyan	0 255 255	#00FFFF
LightCyan	224 255 255	#E0FFFF
CadetBlue	95 158 160	#5F9EA0
MediumAquaMarine	102 205 170	#66CDAA
AquaMarine	127 255 212	#7FFFD4
DarkGreen	0 100 0	#006400
DarkOliveGreen	85 107 47	#556B2F
DarkSeaGreen	143 188 143	#8FBC8F
SeaGreen	46 139 87	#2E8B57
MediumSeaGreen	60 179 113	#3CB371
LightSeaGreen	32 178 170	#20B2AA
PaleGreen	152 251 152	#98FB98
SpringGreen	0 255 127	#00FF7F
LawnGreen	124 252 0	#7CFC00
Green	0 255 0	#00FF00
Chartreuse	127 255 0	#7FFF00
MedSpringGreen	0 250 154	#00FA9A
GreenYellow	173 255 47	#ADFF2F
LimeGreen	50 205 50	#32CD32

YellowGreen	154 205 50	#9ACD32
ForestGreen	34 139 34	#228B22
OliveDrab	107 142 35	#6B8E23
DarkKhaki	189 183 107	#BDB76B
PaleGoldenrod	238 232 170	#EEE8AA
LtGoldenrodYello	250 250 210	#FAFAD2
LightYellow	255 255 224	#FFFFE0
Yellow	255 255 0	#FFFF00
Gold	255 215 0	#FFD700
LightGoldenrod	238 221 130	#EEDD82
goldenrod	218 165 32	#DAA520
DarkGoldenrod	184 134 11	#B8860B
RosyBrown	188 143 143	#BC8F8F
IndianRed	205 92 92	#CD5C5C
SaddleBrown	139 69 19	#8B4513
Sienna	160 82 45	#A0522D
Peru	205 133 63	#CD853F
Burlywood	222 184 135	#DEB887
Beige	245 245 220	#F5F5DC
Wheat	245 222 179	#F5DEB3

SandyBrown	244 164 96	#F4A460
Tan	210 180 140	#D2B48C
Chocolate	210 105 30	#D2691E
Firebrick	178 34 34	#B22222
Brown	165 42 42	#A52A2A
DarkSalmon	233 150 122	#E9967A
Salmon	250 128 114	#FA8072
LightSalmon	255 160 122	#FFA07A
Orange	255 165 0	#FFA500
DarkOrange	255 140 0	#FF8C00
Coral	255 127 80	#FF7F50
LightCoral	240 128 128	#F08080
Tomato	255 99 71	#FF6347
OrangeRed	255 69 0	#FF4500
Red	255 0 0	#FF0000
HotPink	255 105 180	#FF69B4
DeepPink	255 20 147	#FF1493
Pink	255 192 203	#FFC0CB
LightPink	255 182 193	#FFB6C1
PaleVioletRed	219 112 147	#DB7093

Maroon	176 48 96	#B03060
MediumVioletRed	199 21 133	#C71585
VioletRed	208 32 144	#D02090
Magenta	255 0 255	#FF00FF
Violet	238 130 238	#EE82EE
Plum	221 160 221	#DDA0DD
Orchid	218 112 214	#DA70D6
MediumOrchid	186 85 211	#BA55D3
DarkOrchid	153 50 204	#9932CC
DarkViolet	148 0 211	#9400D3
BlueViolet	138 43 226	#8A2BE2
Purple	160 32 240	#A020F0
MediumPurple	147 112 219	#9370DB
Thistle	216 191 216	#D8BFD8
Snow1	255 250 250	#FFFAFA
Snow2	238 233 233	#EEE9E9
Snow3	205 201 201	#CDC9C9
Snow4	139 137 137	#8B8989
Seashell1	255 245 238	#FFF5EE
Seashell2	238 229 222	#EEE5DE

Seashell3	205 197 191	#CDC5BF
Seashell4	139 134 130	#8B8682
AntiqueWhite1	255 239 219	#FFEFDB
AntiqueWhite2	238 223 204	#EEDFCC
AntiqueWhite3	205 192 176	#CDC0B0
AntiqueWhite4	139 131 120	#8B8378
Bisque1	255 228 196	#FFE4C4
Bisque2	238 213 183	#EED5B7
Bisque3	205 183 158	#CDB79E
Bisque4	139 125 107	#8B7D6B
PeachPuff1	255 218 185	#FFDAB9
PeachPuff2	238 203 173	#EECBAD
PeachPuff3	205 175 149	#CDAF95
PeachPuff4	139 119 101	#8B7765
NavajoWhite1	255 222 173	#FFDEAD
NavajoWhite2	238 207 161	#EECFA1
NavajoWhite3	205 179 139	#CDB38B
NavajoWhite4	139 121 94	#8B795E
LemonChiffon1	255 250 205	#FFFACD
LemonChiffon2	238 233 191	#EEE9BF



LemonChiffon3	205 201 165	#CDC9A5
LemonChiffon4	139 137 112	#8B8970
Cornsilk1	255 248 220	#FFF8DC
Cornsilk2	238 232 205	#EEE8CD
Cornsilk3	205 200 177	#CDC8B1
Cornsilk4	139 136 120	#8B8878
Ivory1	255 255 240	#FFFFFF0
Ivory2	238 238 224	#EEEEEE0
Ivory3	205 205 193	#CDCDC1
Ivory4	139 139 131	#8B8B83
Honeydew1	240 255 240	#F0FFF0
Honeydew2	224 238 224	#E0EEE0
Honeydew3	193 205 193	#C1CDC1
Honeydew4	131 139 131	#838B83
LavenderBlush1	255 240 245	#FFF0F5
LavenderBlush2	238 224 229	#EEE0E5
LavenderBlush3	205 193 197	#CDC1C5
LavenderBlush4	139 131 134	#8B8386
MistyRose1	255 228 225	#FFE4E1
MistyRose2	238 213 210	#EED5D2

MistyRose3	205 183 181	#CDB7B5
MistyRose4	139 125 123	#8B7D7B
Azure1	240 255 255	#F0FFFF
Azure2	224 238 238	#E0EEEE
Azure3	193 205 205	#C1CDCD
Azure4	131 139 139	#838B8B
SlateBlue1	131 111 255	#836FFF
SlateBlue2	122 103 238	#7A67EE
SlateBlue3	105 89 205	#6959CD
SlateBlue4	71 60 139	#473C8B
RoyalBlue1	72 118 255	#4876FF
RoyalBlue2	67 110 238	#436EEE
RoyalBlue3	58 95 205	#3A5FCD
RoyalBlue4	39 64 139	#27408B
Blue1	0 0 255	#0000FF
Blue2	0 0 238	#0000EE
Blue3	0 0 205	#0000CD
Blue4	0 0 139	#00008B
DodgerBlue1	30 144 255	#1E90FF
DodgerBlue2	28 134 238	#1C86EE

DodgerBlue3	24 116 205	#1874CD
DodgerBlue4	16 78 139	#104E8B
SteelBlue1	99 184 255	#63B8FF
SteelBlue2	92 172 238	#5CACEE
SteelBlue3	79 148 205	#4F94CD
SteelBlue4	54 100 139	#36648B
DeepSkyBlue1	0 191 255	#00BFFF
DeepSkyBlue2	0 178 238	#00B2EE
DeepSkyBlue3	0 154 205	#009ACD
DeepSkyBlue4	0 104 139	#00688B
SkyBlue1	135 206 255	#87CEFF
SkyBlue2	126 192 238	#7EC0EE
SkyBlue3	108 166 205	#6CA6CD
SkyBlue4	74 112 139	#4A708B
LightSkyBlue1	176 226 255	#B0E2FF
LightSkyBlue2	164 211 238	#A4D3EE
LightSkyBlue3	141 182 205	#8DB6CD
LightSkyBlue4	96 123 139	#607B8B
SlateGray1	198 226 255	#C6E2FF
SlateGray2	185 211 238	#B9D3EE

SlateGray3	159 182 205	#9FB6CD
SlateGray4	108 123 139	#6C7B8B
LightSteelBlue1	202 225 255	#CAE1FF
LightSteelBlue2	188 210 238	#BCD2EE
LightSteelBlue3	162 181 205	#A2B5CD
LightSteelBlue4	110 123 139	#6E7B8B
LightBlue1	191 239 255	#BFEFFF
LightBlue2	178 223 238	#B2DFEE
LightBlue3	154 192 205	#9AC0CD
LightBlue4	104 131 139	#68838B
LightCyan1	224 255 255	#E0FFFF
LightCyan2	209 238 238	#D1EEEE
LightCyan3	180 205 205	#B4CDCD
LightCyan4	122 139 139	#7A8B8B
PaleTurquoise1	187 255 255	#BBFFFF
PaleTurquoise2	174 238 238	#AEEEEEE
PaleTurquoise3	150 205 205	#96CDCD
PaleTurquoise4	102 139 139	#668B8B
CadetBlue1	152 245 255	#98F5FF
CadetBlue2	142 229 238	#8EE5EE

CadetBlue3	122 197 205	#7AC5CD
CadetBlue4	83 134 139	#53868B
Turquoise1	0 245 255	#00F5FF
Turquoise2	0 229 238	#00E5EE
Turquoise3	0 197 205	#00C5CD
Turquoise4	0 134 139	#00868B
Cyan1	0 255 255	#00FFFF
Cyan2	0 238 238	#00EEEE
Cyan3	0 205 205	#00CDCD
Cyan4	0 139 139	#008B8B
DarkSlateGray1	151 255 255	#97FFFF
DarkSlateGray2	141 238 238	#8DEEEE
DarkSlateGray3	121 205 205	#79CDCD
DarkSlateGray4	82 139 139	#528B8B
Aquamarine1	127 255 212	#7FFFD4
Aquamarine2	118 238 198	#76EEC6
Aquamarine3	102 205 170	#66CDAA
Aquamarine4	69 139 116	#458B74
DarkSeaGreen1	193 255 193	#C1FFC1
DarkSeaGreen2	180 238 180	#B4EEB4

DarkSeaGreen3	155 205 155	#9BCD9B
DarkSeaGreen4	105 139 105	#698B69
SeaGreen1	84 255 159	#54FF9F
SeaGreen2	78 238 148	#4EEE94
SeaGreen3	67 205 128	#43CD80
SeaGreen4	46 139 87	#2E8B57
PaleGreen1	154 255 154	#9AFF9A
PaleGreen2	144 238 144	#90EE90
PaleGreen3	124 205 124	#7CCD7C
PaleGreen4	84 139 84	#548B54
SpringGreen1	0 255 127	#00FF7F
SpringGreen2	0 238 118	#00EE76
SpringGreen3	0 205 102	#00CD66
SpringGreen4	0 139 69	#008B45
Green1	0 255 0	#00FF00
Green2	0 238 0	#00EE00
Green3	0 205 0	#00CD00
Green4	0 139 0	#008B00
Chartreuse1	127 255 0	#7FFF00
Chartreuse2	118 238 0	#76EE00

Chartreuse3	102 205 0	#66CD00
Chartreuse4	69 139 0	#458B00
OliveDrab1	192 255 62	#C0FF3E
OliveDrab2	179 238 58	#B3EE3A
OliveDrab3	154 205 50	#9ACD32
OliveDrab4	105 139 34	#698B22
DarkOliveGreen1	202 255 112	#CAFF70
DarkOliveGreen2	188 238 104	#BCEE68
DarkOliveGreen3	162 205 90	#A2CD5A
DarkOliveGreen4	110 139 61	#6E8B3D
Khaki1	255 246 143	#FFF68F
Khaki2	238 230 133	#EEE685
Khaki3	205 198 115	#CDC673
Khaki4	139 134 78	#8B864E
LightGoldenrod1	255 236 139	#FFEC8B
LightGoldenrod2	238 220 130	#EEDC82
LightGoldenrod3	205 190 112	#CDBE70
LightGoldenrod4	139 129 76	#8B814C
LightYellow1	255 255 224	#FFFFE0
LightYellow2	238 238 209	#EEEEED1

LightYellow3	205 205 180	#CDCDB4
LightYellow4	139 139 122	#8B8B7A
Yellow1	255 255 0	#FFFF00
Yellow2	238 238 0	#EEEE00
Yellow3	205 205 0	#CDCD00
Yellow4	139 139 0	#8B8B00
Gold1	255 215 0	#FFD700
Gold2	238 201 0	#EEC900
Gold3	205 173 0	#CDAD00
Gold4	139 117 0	#8B7500
Goldenrod1	255 193 37	#FFC125
Goldenrod2	238 180 34	#EEB422
Goldenrod3	205 155 29	#CD9B1D
Goldenrod4	139 105 20	#8B6914
DarkGoldenrod1	255 185 15	#FFB90F
DarkGoldenrod2	238 173 14	#EEAD0E
DarkGoldenrod3	205 149 12	#CD950C
DarkGoldenrod4	139 101 8	#8B658B
RosyBrown1	255 193 193	#FFC1C1
RosyBrown2	238 180 180	#EEB4B4



RosyBrown3	205 155 155	#CD9B9B
RosyBrown4	139 105 105	#8B6969
IndianRed1	255 106 106	#FF6A6A
IndianRed2	238 99 99	#EE6363
IndianRed3	205 85 85	#CD5555
IndianRed4	139 58 58	#8B3A3A
Sienna1	255 130 71	#FF8247
Sienna2	238 121 66	#EE7942
Sienna3	205 104 57	#CD6839
Sienna4	139 71 38	#8B4726
Burlywood1	255 211 155	#FFD39B
Burlywood2	238 197 145	#EEC591
Burlywood3	205 170 125	#CDAA7D
Burlywood4	139 115 85	#8B7355
Wheat1	255 231 186	#FFE7BA
Wheat2	238 216 174	#EED8AE
Wheat3	205 186 150	#CDBA96
Wheat4	139 126 102	#8B7E66
Tan1	255 165 79	#FFA54F
Tan2	238 154 73	#EE9A49

Tan3	205 133 63	#CD853F
Tan4	139 90 43	#8B5A2B
Chocolate1	255 127 36	#FF7F24
Chocolate2	238 118 33	#EE7621
Chocolate3	205 102 29	#CD661D
Chocolate4	139 69 19	#8B4513
Firebrick1	255 48 48	#FF3030
Firebrick2	238 44 44	#EE2C2C
Firebrick3	205 38 38	#CD2626
Firebrick4	139 26 26	#8B1A1A
Brown1	255 64 64	#FF4040
Brown2	238 59 59	#EE3B3B
Brown3	205 51 51	#CD3333
Brown4	139 35 35	#8B2323
Salmon1	255 140 105	#FF8C69
Salmon2	238 130 98	#EE8262
Salmon3	205 112 84	#CD7054
Salmon4	139 76 57	#8B4C39
LightSalmon1	255 160 122	#FFA07A
LightSalmon2	238 149 114	#EE9572

LightSalmon3	205 129 98	#CD8162
LightSalmon4	139 87 66	#8B5742
Orange1	255 165 0	#FFA500
Orange2	238 154 0	#EE9A00
Orange3	205 133 0	#CD8500
Orange4	139 90 0	#8B5A00
DarkOrange1	255 127 0	#FF7F00
DarkOrange2	238 118 0	#EE7600
DarkOrange3	205 102 0	#CD6600
DarkOrange4	139 69 0	#8B4500
Coral1	255 114 86	#FF7256
Coral2	238 106 80	#EE6A50
Coral3	205 91 69	#CD5B45
Coral4	139 62 47	#8B3E2F
Tomato1	255 99 71	#FF6347
Tomato2	238 92 66	#EE5C42
Tomato3	205 79 57	#CD4F39
Tomato4	139 54 38	#8B3626
OrangeRed1	255 69 0	#FF4500
OrangeRed2	238 64 0	#EE4000

OrangeRed3	205 55 0	#CD3700
OrangeRed4	139 37 0	#8B2500
Red1	255 0 0	#FF0000
Red2	238 0 0	#EE0000
Red3	205 0 0	#CD0000
Red4	139 0 0	#8B0000
DeepPink1	255 20 147	#FF1493
DeepPink2	238 18 137	#EE1289
DeepPink3	205 16 118	#CD1076
DeepPink4	139 10 80	#8B0A50
HotPink1	255 110 180	#FF6EB4
HotPink2	238 106 167	#EE6AA7
HotPink3	205 96 144	#CD6090
HotPink4	139 58 98	#8B3A62
Pink1	255 181 197	#FFB5C5
Pink2	238 169 184	#EEA9B8
Pink3	205 145 158	#CD919E
Pink4	139 99 108	#8B636C
LightPink1	255 174 185	#FFAEB9
LightPink2	238 162 173	#EEA2AD

LightPink3	205 140 149	#CD8C95
LightPink4	139 95 101	#8B5F65
PaleVioletRed1	255 130 171	#FF82AB
PaleVioletRed2	238 121 159	#EE799F
PaleVioletRed3	205 104 137	#CD6889
PaleVioletRed4	139 71 93	#8B475D
Maroon1	255 52 179	#FF34B3
Maroon2	238 48 167	#EE30A7
Maroon3	205 41 144	#CD2990
Maroon4	139 28 98	#8B1C62
VioletRed1	255 62 150	#FF3E96
VioletRed2	238 58 140	#EE3A8C
VioletRed3	205 50 120	#CD3278
VioletRed4	139 34 82	#8B2252
Magenta1	255 0 255	#FF00FF
Magenta2	238 0 238	#EE00EE
Magenta3	205 0 205	#CD00CD
Magenta4	139 0 139	#8B008B
Orchid1	255 131 250	#FF83FA
Orchid2	238 122 233	#EE7AE9

Orchid3	205 105 201	#CD69C9
Orchid4	139 71 137	#8B4789
Plum1	255 187 255	#FFBBFF
Plum2	238 174 238	#EEAEEE
Plum3	205 150 205	#CD96CD
Plum4	139 102 139	#8B668B
MediumOrchid1	224 102 255	#E066FF
MediumOrchid2	209 95 238	#D15FEE
MediumOrchid3	180 82 205	#B452CD
MediumOrchid4	122 55 139	#7A378B
DarkOrchid1	191 62 255	#BF3EFF
DarkOrchid2	178 58 238	#B23AEE
DarkOrchid3	154 50 205	#9A32CD
DarkOrchid4	104 34 139	#68228B
Purple1	155 48 255	#9B30FF
Purple2	145 44 238	#912CEE
Purple3	125 38 205	#7D26CD
Purple4	85 26 139	#551A8B
MediumPurple1	171 130 255	#AB82FF
MediumPurple2	159 121 238	#9F79EE

MediumPurple3	137 104 205	#8968CD
MediumPurple4	93 71 139	#5D478B
Thistle1	255 225 255	#FFE1FF
Thistle2	238 210 238	#EED2EE
Thistle3	205 181 205	#CDB5CD
Thistle4	139 123 139	#8B7B8B
grey11	28 28 28	#1C1C1C
grey21	54 54 54	#363636
grey31	79 79 79	#4F4F4F
grey41	105 105 105	#696969
grey51	130 130 130	#828282
grey61	156 156 156	#9C9C9C
grey71	181 181 181	#B5B5B5
gray81	207 207 207	#CFCFCF
gray91	232 232 232	#E8E8E8
DarkGrey	169 169 169	#A9A9A9
DarkBlue	0 0 139	#00008B
DarkCyan	0 139 139	#008B8B
DarkMagenta	139 0 139	#8B008B
DarkRed	139 0 0	#8B0000
LightGreen	144 238 144	#90EE90

## Приложение 2. Коды языков

Коды языков, которые можно использовать в теге **lang**, приведены в таблице ниже.

**Таблица 2. Коды языков**

Язык	Код
Абхазский	ab
Азербайджанский	az
Аймарский	ay
Албанский	sq
Английский	en
Американский английский	en-us
Арабский	ar
Армянский	hy
Ассамский	as
Африкаанс	af
Башкирский	ba
Белорусский	be
Бенгальский	bn
Болгарский	bg
Бретонский	br
Валлийский	cy



Венгерский	hu
Вьетнамский	vi
Галисийский	gl
Голландский	nl
Греческий	el
Грузинский	ka
Гуарани	gn
Датский	da
Зулу	zu
Иврит	iw
Идиш	ji
Индонезийский	in
Интерлингва (искусственный язык)	ia
Ирландский	ga
Исландский	is
Испанский	es
Итальянский	it
Казахский	kk
Камбоджийский	km
Каталанский	ca

Кашмирский	ks
Кечуа	qu
Киргизский	ky
Китайский	zh
Корейский	ko
Корсиканский	co
Курдский	ku
Лаосский	lo
Латвийский, латышский	lv
Латынь	la
Литовский	lt
Малагасийский	mg
Малайский	ms
Мальтийский	mt
Маори	mi
Македонский	mk
Молдавский	mo
Монгольский	mn
Науру	na
Немецкий	de

Непальский	ne
Норвежский	no
Пенджаби	pa
Персидский	fa
Польский	pl
Португальский	pt
Пуштунский	ps
Ретороманский	rm
Румынский	ro
Русский	ru
Самоанский	sm
Санскрит	sa
Сербский	sr
Словацкий	sk
Словенский	sl
Сомали	so
Суахили	sw
Суданский	su
Тагальский	tl
Таджикский	tg

Тайский	th
Тамильский	ta
Татарский	tt
Тибетский	bo
Тонга	to
Турецкий	tr
Туркменский	tk
Узбекский	uz
Украинский	uk
Урду	ur
Фиджи	fj
Финский	fi
Французский	fr
Фризский	fy
Хауса	ha
Хинди	hi
Хорватский	hr
Чешский	cs
Шведский	sv
Эсперанто (искусственный язык)	eo

Эстонский	et
Яванский	jw
Японский	ja

## Приложение 3. MIME-типы

Приложение 3 содержит таблицу соответствия расширений файлов MIME-типов. Такая табличка пригодится любому дизайнеру, особенно, если он дополнительно желает освоить Web-программирование.

**Таблица 3. Таблица MIME-типов**

Тип MIME	Расширение
animation/narrative	nml
application/activemessage	
application/andrew-inset	
application/applefile	
application/atomicmail	
application/dca-rft	
application/dec-dx	
application/dsptype	tsp
application/fastman	lcc
application/font-tdpfr	pfr
application/futuresplash	spl
application/mac-binhex40	hqx
application/mac-compactpro	cpt

application/macwriteii	
application/mspowerpoint	pot pps ppt ppz
application/msword	
application/news-message-id	
application/news-transmission	
application/octet-stream	bin class dms exe sea
application/oda	oda
application/pdf	pdf
application/postscript	ai eps ps
application/remote-printing	
application/rtf	rtf
application/slate	
application/softvision	svi
application/wita	
application/wordperfect5.1	
application/x-authorware-bin	aab
application/x-authorware-map	aam
application/x-authorware-seg	aas
application/x-bcpio	bcpio
application/x-bzip2	bz2

application/x-calquick	cqk
application/x-cdlink	vcd
application/x-cocoa	cco
application/x-compress	Z
application/x-cpio	cpio
application/x-csh	csh
application/x-director	dcr dir dxr
application/x-dvi	dvi
application/x-excel	xls
application/x-expandedbook	ebk
application/x-gtar	gtar
application/x-gzip	gz
application/x-hdf	hdf
application/x-httpd-cgi	cgi
application/x-javascript	js ls mocha
application/x-koan	skd skm skp skt
application/x-latex	latex
application/x-lzh	lha lzh
application/x-mapserver	mps
application/x-mascot	mct



application/x-mif	mif
application/x-netcdf	cdf nc
application/x-ns-proxy-auto-config	pac
application/x-pixelscoter	mpp
application/x-pointplus	css
application/x-sh	sh
application/x-shar	shar
application/x-shockwave-flash	swf
application/x-sprite	spr sprite
application/x-spt	spt
application/x-stuffit	sit
application/x-sv4cpio	sv4cpio
application/x-sv4crc	sv4crc
application/x-tar	tar tgz
application/x-tcl	tcl
application/x-tex	tex
application/x-texinfo	texi texinfo
application/x-troff	roff t tr
application/x-troff-man	man
application/x-troff-me	me

application/x-troff-ms	ms
application/x-ustar	ustar
application/x-wais-source	src
application/x-xdma	xdm xdma
application/zip	zip
audio/basic	au snd
audio/echospeech	es
audio/midi	kar mid midi
audio/mpeg	mp2 mp3 mpga
audio/tsplayer	tsi
audio/voxware	vox
audio/x-aiff	aif aifc aiff
audio/x-bamba	aba
audio/x-chacha	cha
audio/x-mio	mio
audio/x-pn-realaudio	ra ram rm
audio/x-pn-realaudio-plugin	rpm
audio/x-twinvq	vqf vql
audio/x-twinvq-plugin	vqe
audio/x-wav	wav

chemical/x-csml	csm
chemical/x-embl-dl-nucleotide	emb
chemical/x-gaussian-input	gau
chemical/x-mdl-molfile	mol
chemical/x-mopac-input	mop
chemical/x-pdb	pdb
chemical/x-xyz	xyz
i-world/i-vrml	ivr
image/fif	fif
image/gif	gif
image/ief	ief
image/jpeg	jpe jpeg jpg
image/png	png
image/tiff	tif tiff
image/vasa	mcf
image/x-cmu-raster	ras
image/x-freehand	fh fh4 fh5 fh7 fhc
image/x-ico	ICO
image/x-jps	jps
image/x-portable-anymap	pnm

image/x-portable-bitmap	pbm
image/x-portable-graymap	pgm
image/x-portable-pixmap	ppm
image/x-rgb	rgb
image/x-xbitmap	xbm
image/x-xpixmap	xpm
image/x-xres	swx
image/x-xwindowdump	xwd
message/external-body	
message/news	
message/partial	
message/rfc822	
multipart/alternative	
multipart/appledouble	
multipart/digest	
multipart/mixed	
multipart/parallel	
plugin/listenup	ptlk
plugin/wanimate	waf wan
text/html	htm html

text/plain	txt
text/richtext	rtx
text/tab-separated-values	tsv
text/x-setext	etx
text/x-sgml	sgm sgml
text/x-speech	talk
text/x-vcard	vcf
video/mpeg	mpe mpeg mpg
video/quicktime	mov qt
video/vnd.vivo	viv vivo
video/x-bamba	vba
video/x-msvideo	avi
video/x-sgi-movie	movie
video/x-tango	tgo
video/x-vif	vif
workbook/formulaone	vtb
world/x-panoramix	pan
x-conference/x-cooltalk	ice
x-world/x-d96	d96 mus
x-world/x-svr	svr

x-world/x-vrml	vrml wrl
x-world/x-vrt	vrt

## Послесловие

Наша книга подошла к концу. Надеемся, что вам удалось найти в ней нужную информацию и то, что вас интересует. Мы постарались собрать в этом Справочнике все актуальную информацию по HTML5, которая потребуется вам для изучения и программирования на HTML5. Авторы также выражают благодарность сотрудникам ООО "Цифровые Бизнес-Платформы" за участие в разработке и написании этой книги. Надеемся, что вам будут интересны и другие Справочники из этой же серии. Все пожелания и отзывы вы можете оставлять на почтовом ящике издательства:

[nitmail@nit.com.ru](mailto:nitmail@nit.com.ru)



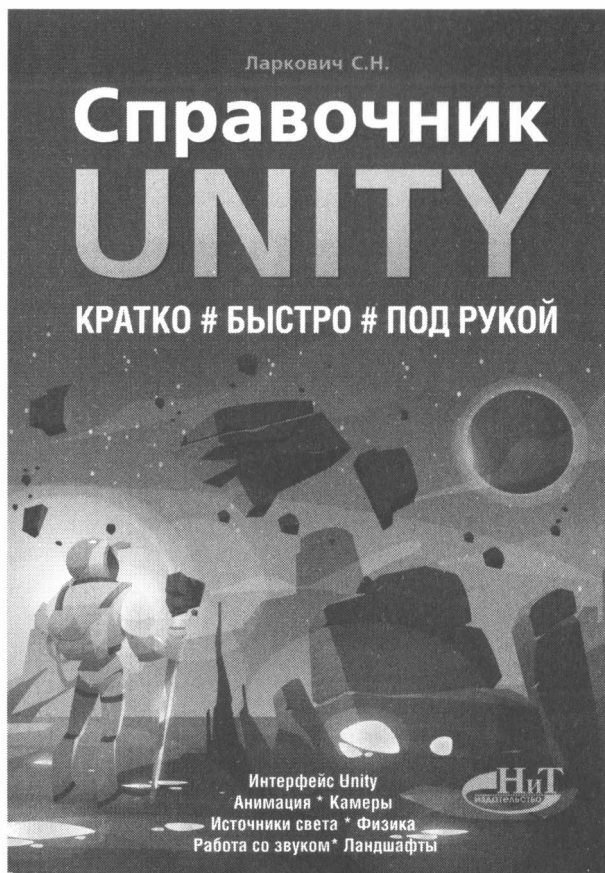
**HTML**



***Издательство "Наука и Техника" рекомендует:***



**Издательство "Наука и Техника" рекомендует:**



Эта книга будет полезна всем, кто хочет освоить или уже осваивает самый популярный из современных и доступных игровых движков - Unity. В справочнике в удобном формате собрано множество полезной информации для работы с Unity: компоненты для создания игровых объектов и сцен; трансформация объектов; камеры и их свойства; источники света и геометрия освещения; материалы, текстуры, шейдеры; создание ландшафта; 3D-физика и 2D-физика в Unity; коллайдеры, сочленения и компоненты; работа со звуком и звуковым оформлением; анимация; полезные сочетания клавиш и различные трюки Unity.

Эта книга поможет вам увереннее работать с Unity и в удобной форме оперативно получать информацию, которая будет у вас всегда под рукой. Кроме того, книга может использоваться в качестве дополнения к каким-либо учебным видеокурсам по Unity, а также для систематизации имеющихся знаний. Подходит как начинающим, так и опытным разработчикам на Unity.



# Книжный магазин

издательства «Наука и Техника»  
приглашает за покупками

... ➤ **Предлагаем широкий ассортимент  
технической литературы ведущих  
издательств (более 2000 наименований):**

- Компьютерная литература
- Радиозлектроника
- Телекоммуникации и связь
- Транспорт, строительство
- Научно-популярная медицина,  
педагогика, психология

... ➤ **Чем привлекателен наш магазин:**

- низкие цены;
- ежедневное пополнение ассортимента;
- поиск книг под заказ;
- обслуживание за наличный  
и безналичный расчет;
- гибкая система скидок;
- комплектование библиотек;
- обеспечение школ учебниками  
по информатике;
- возможна доставка.

**Наш адрес:** г. Санкт-Петербург  
пр. Обуховской Обороны д. 107  
ст. метро Елизаровская

**Справки о наличии книг по тел. 412-70-26**

**E-mail: [admin@nit.com.ru](mailto:admin@nit.com.ru)**  
(рассылка ассортиментного прайс-листа по запросу)

Мы работаем с 10 до 19 часов без обеда и выходных  
(в субботу и воскресенье до 18 час)



Издательство «Наука и Техника»

**КНИГИ ПО КОМПЬЮТЕРНЫМ ТЕХНОЛОГИЯМ,  
МЕДИЦИНЕ, РАДИОЭЛЕКТРОНИКЕ**

## **Уважаемые читатели!**

Книги издательства «Наука и Техника» вы можете:

➤ **заказать в нашем интернет-магазине БЕЗ ПРЕДОПЛАТЫ по ОПТОВЫМ ценам**

**www.nit.com.ru**

- более 3000 пунктов выдачи на территории РФ, доставка 3—5 дней
- более 300 пунктов выдачи в Санкт-Петербурге и Москве, доставка — на следующий день

### **Справки и заказ:**

- на сайте **www.nit.com.ru**
  - по тел. (812) 412-70-26
  - по эл. почте nitmail@nit.com.ru

➤ **приобрести в магазине издательства по адресу:**

Санкт-Петербург, пр. Обуховской обороны, д.107

М. Елизаровская, 200 м за ДК им. Крупской

Ежедневно с 10.00 до 18.30

Справки и заказ: тел. (812) 412-70-26

➤ **приобрести в Москве:**

«Новый книжный» Сеть магазинов  
ТД «БИБЛИО-ГЛОБУС»

тел. (495) 937-85-81, (499) 177-22-11  
ул. Мясницкая, д. 6/3, стр. 1, ст. М «Лубянка»  
тел. (495) 781-19-00, 624-46-80

Московский Дом Книги,  
«ДК на Новом Арбате»

ул. Новый Арбат, 8, ст. М «Арбатская»,  
тел. (495) 789-35-91

Московский Дом Книги,  
«Дом технической книги»

Ленинский пр., д.40, ст. М «Ленинский пр.»,  
тел. (499) 137-60-19

Московский Дом Книги,  
«Дом медицинской книги»

Комсомольский пр., д. 25, ст. М «Фрунзенская»,  
тел. (499) 245-39-27

Дом книги «Молодая гвардия»

ул. Б. Полянка, д. 28, стр. 1, ст. М «Полянка»  
тел. (499) 238-50-01

➤ **приобрести в Санкт-Петербурге:**

Санкт-Петербургский Дом Книги  
Буквоед. Сеть магазинов

Невский пр. 28, тел. (812) 448-23-57  
тел. (812) 601-0-601

➤ **приобрести в регионах России:**

г. Воронеж, «Амитель» Сеть магазинов  
г. Екатеринбург, «Дом книги» Сеть магазинов  
г. Нижний Новгород, «Дом книги» Сеть магазинов  
г. Владивосток, «Дом книги» Сеть магазинов  
г. Иркутск, «Продавить» Сеть магазинов  
г. Омск, «Техническая книга» ул. Пушкина, д.101

тел. (473) 224-24-90  
тел. (343) 289-40-45  
тел. (831) 246-22-92  
тел. (423) 263-10-54  
тел. (395) 298-88-82  
тел. (381) 230-13-64

## **Мы рады сотрудничеству с Вами!**

Кириченко А.В., Дубовик Е.В.

# Справочник HTML

**КРАТКО # БЫСТРО # ПОД РУКОЙ**

**Группа подготовки издания:**

Зав. редакцией компьютерной литературы: *М. В. Финков*

Редактор: *Е. В. Финков*

Корректор: *А. В. Громова*

12+

---

ООО «Наука и Техника»

Лицензия №000350 от 23 декабря 1999 года.

192029, г. Санкт-Петербург, пр. Обуховской обороны, д. 107.

Подписано в печать 16.02.2021. Формат 60х90 1/16.

Бумага офсетная. Печать офсетная. Объем 18 п. л.

Тираж 1400. Заказ 239.

Отпечатано ООО «Принт-М»

142300, Московская область,

г. Чехов, ул. Полиграфистов, дом 1

Кириченко А.В., Дубовик Е.В.

# Справочник HTML

## КРАТКО # БЫСТРО # ПОД РУКОЙ

Данный справочник содержит полную информацию об HTML5 в удобной и наглядной форме. Структура справочника позволяет быстро и удобно находить нужную информацию, получать примеры использования тех или иных тегов HTML. Отдельное внимание уделено отраслевым рекомендациям по хорошему стилю использования HTML для верстки веб-сайтов, приведено описание приемов семантической верстки. Справочник будет полезен всем, кто использует или изучает HTML: от начинающих до профессионалов.

Издательство «Наука и Техника» рекомендует:



ISBN 978-5-94387-275-4



9 78- 5- 94387- 275- 4

Издательство "Наука и Техника"  
г. Санкт-Петербург

Для заказа книг:

(812) 412-70-26

e-mail: [nitmail@nit.com.ru](mailto:nitmail@nit.com.ru)

[www.nit.com.ru](http://www.nit.com.ru)