

Основы HTML

Предыдущий Обзор: начало работы в Интернете Следующий HTML (язык гипертекстовой разметки) — это код, который используется для структурирования веб-страницы и ее содержимого . Например, контент может быть структурирован в виде набора абзацев, списка маркированных пунктов или с использованием изображений и таблиц данных. Как следует из названия, эта статья даст вам общее представление о HTML и его функциях.

Так что же такое HTML? HTML — это язык разметки , определяющий структуру вашего контента. HTML состоит из ряда элементов , которые вы используете, чтобы заключить или обернуть различные части содержимого, чтобы оно выглядело определенным образом или действовало определенным образом. Окружающие теги могут сделать слово или изображение гиперссылкой на что-то еще, могут выделять слова курсивом, могут делать шрифт больше или меньше и так далее. Например, возьмем следующую строку контента:

My cat is very grumpy Если бы мы хотели, чтобы строка стояла отдельно, мы могли бы указать, что это абзац, заключив его в теги абзаца:

HTML Скопировать в буфер обмена

`<p>My cat is very grumpy</p>` Анатомия элемента HTML Давайте рассмотрим этот элемент абзаца немного подробнее.

элемент абзаца, включающий открывающий тег, содержимое с надписью «мой кот очень сварливый» и закрывающий тег Основные части нашего элемента следующие:

Открывающий тег: состоит из имени элемента (в данном случае p), заключенного в открывающие и закрывающие угловые скобки . Это указывает, где элемент начинается или начинает действовать — в данном случае, где начинается абзац. Закрывающий тег: это то же самое, что и открывающий тег, за исключением того, что он включает косую черту перед именем элемента. Здесь указывается, где заканчивается элемент — в данном случае там, где заканчивается абзац. Отсутствие закрывающего тега является одной из стандартных ошибок новичков и может привести к странным результатам. Содержимое: это содержимое элемента, в данном случае это просто текст. Элемент: открывающий тег, закрывающий тег и содержимое вместе составляют элемент. Элементы также могут иметь атрибуты, которые выглядят следующим образом:

Тег открытия абзаца с выделенным атрибутом класса: `class=editor-note` Атрибуты содержат дополнительную информацию об элементе, которую вы не хотите отображать в фактическом содержимом. Здесь — имя атрибута , а — значение атрибута . Атрибут позволяет вам присвоить элементу неunikальный идентификатор, который можно использовать для нацеливания на него (и любые другие элементы с таким же значением) с информацией о стиле и другими вещами. Некоторые атрибуты не имеют значения, например . `editor-note` `class` `class` `required`

Атрибуты, которые устанавливают значение, всегда имеют:

Пробел между ним и именем элемента (или предыдущим атрибутом, если элемент уже имеет один или несколько атрибутов). Имя атрибута, за которым следует знак равенства. Значение атрибута заключено в открывающие и закрывающие кавычки. Примечание. Значения простых атрибутов, которые не содержат пробелов ASCII (или каких-либо символов « ' ` = < > »), могут

оставаться без кавычек, но рекомендуется заключать в кавычки все значения атрибутов, так как это делает код более последовательным и понятным.

Вложенные элементы Вы также можете помещать элементы внутри других элементов — это называется вложением . Если бы мы хотели указать, что наш кот очень сварливый, мы могли бы обернуть слово «очень» в `<strong>` элемент, что означает, что слово должно быть сильно подчеркнуто:

HTML Скопировать в буфер обмена

`<p>My cat is <strong>very</strong> grumpy.</p>` Однако вам необходимо убедиться, что ваши элементы правильно вложены. В приведенном выше примере мы `<p>` сначала открыли элемент, затем сам `<strong>` элемент; поэтому мы должны `<strong>` сначала закрыть элемент, а затем `<p>` элемент. Неверно следующее:

HTML Скопировать в буфер обмена

`<p>My cat is <strong>very grumpy.</p></strong>` Элементы должны открываться и закрываться правильно, чтобы они были четко внутри или снаружи друг друга. Если они перекрываются, как показано выше, ваш веб-браузер попытается сделать наилучшее предположение о том, что вы пытались сказать, что может привести к неожиданным результатам. Так что не делай этого!

Пустые элементы Некоторые элементы не имеют содержимого и называются пустыми элементами . Возьмем `<img>` элемент, который у нас уже есть на нашей HTML-странице:

HTML Скопировать в буфер обмена

`<img src=«images/firefox-icon.png» alt=«My test image» />` Он содержит два атрибута, но не имеет закрывающего `</img>` тега и внутреннего содержимого. Это связано с тем, что элемент изображения не оборачивает содержимое, чтобы воздействовать на него. Его цель — встроить изображение в HTML-страницу в том месте, где оно появляется.

Анатомия HTML-документа Это обобщает основы отдельных элементов HTML, но сами по себе они неудобны. Теперь мы рассмотрим, как отдельные элементы объединяются в целую HTML-страницу. Давайте вернемся к коду, который мы поместили в наш `index.html` пример (с которым мы впервые познакомились в статье «Работа с файлами »):

HTML Скопировать в буфер обмена

`<!doctype html> <html lang=«en-US»>`

```
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width" />
  <title>My test page</title>
</head>
<body>
  
</body>
```

`</html>` Здесь мы имеем следующее:

<!DOCTYPE html>— тип документа . Это необходимая преамбула. В глубине веков, когда HTML был молод (около 1991/92), типы документов должны были служить ссылками на набор правил, которым HTML-страница должна была следовать, чтобы считаться хорошим HTML, что могло означать автоматическую проверку ошибок и другие действия. полезные вещи. Однако в наши дни они мало что делают и в основном просто необходимы, чтобы убедиться, что ваш документ ведет себя правильно. Это все, что вам нужно знать на данный момент. — <html>элемент. Этот элемент охватывает весь контент на всей странице и иногда называется корневым элементом. Также включает в себя langатрибут, задающий основной язык документа. <head></head>— <head>элемент. Этот элемент действует как контейнер для всего материала, который вы хотите включить в HTML-страницу, кроме содержимого , которое вы показываете посетителям вашей страницы. Это включает в себя такие вещи, как ключевые слова и описание страницы, которые вы хотите отображать в результатах поиска, CSS для стилизации нашего контента, объявления наборов символов и многое другое. <meta charset=«utf-8»>— Этот элемент задает набор символов, который должен использовать ваш документ, равный UTF-8, который включает в себя большинство символов подавляющего большинства письменных языков. По сути, теперь он может обрабатывать любой текстовый контент, который вы можете на него поместить. Нет причин не устанавливать это, и это может помочь избежать некоторых проблем в дальнейшем. <meta name=«viewport» content=«width=device-width»>— Этот элемент области просмотра обеспечивает отображение страницы с шириной области просмотра, не позволяя мобильным браузерам отображать страницы шире области просмотра, а затем уменьшать их. <title></title>— <title>элемент. Это устанавливает заголовок вашей страницы, который отображается на вкладке браузера, в которую загружена страница. Он также используется для описания страницы, когда вы добавляете ее в закладки/избранное. <body></body>— <body>элемент. Он содержит весь контент, который вы хотите показывать веб-пользователям, когда они посещают вашу страницу, будь то текст, изображения, видео, игры, воспроизводимые звуковые дорожки или что-то еще. Изображений Снова обратим внимание на элемент:

HTML Скопировать в буфер обмена

 Как мы уже говорили, он вставляет изображение на нашу страницу в том месте, где оно появляется. Он делает это с помощью srcатрибута (source), который содержит путь к нашему файлу изображения.

Мы также включили alt(альтернативный) атрибут. В altатрибуте вы указываете описательный текст для пользователей, которые не могут видеть изображение, возможно, по следующим причинам:

Они слабовидящие. Пользователи со значительными нарушениями зрения часто используют инструменты, называемые программами чтения с экрана, чтобы зачитывать им замещающий текст. Что-то пошло не так, из-за чего изображение не отображается. Например, попробуйте намеренно изменить путь внутри вашего srcатрибута, чтобы сделать его неверным. Если вы сохраните и перезагрузите страницу, вы должны увидеть что-то вроде этого вместо изображения: Слова: мой тестовый образ Ключевые слова для замещающего текста — «описательный текст». Замещающий текст, который вы пишете, должен предоставить читателю достаточно информации, чтобы получить представление о том, что передает изображение. В этом примере наш текущий текст «Мое тестовое изображение» совсем не годится. Гораздо лучшей альтернативой нашему логотипу Firefox был бы «Логотип Firefox: пылающая лиса, окружающая Землю».

Попробуйте придумать лучший альтернативный текст для вашего изображения прямо сейчас.

Примечание. Узнайте больше о специальных возможностях в нашем обучающем модуле по специальным возможностям .

Разметка текста В этом разделе будут рассмотрены некоторые важные элементы HTML, которые вы будете использовать для разметки текста.

Заголовки Элементы заголовков позволяют указать, что определенные части вашего контента являются заголовками или подзаголовками. Точно так же, как книга имеет основное название, заголовки глав и подзаголовки, HTML-документ тоже может. HTML содержит 6 уровней заголовков, `<h1>` - `<h6>` , хотя обычно вы будете использовать максимум от 3 до 4:

HTML Скопировать в буфер обмена

`<!-- 4 heading levels: --> <h1>My main title</h1> <h2>My top level heading</h2> <h3>My subheading</h3> <h4>My sub-subheading</h4>` Примечание. Все, что находится в HTML между `<!--` и `-->` является комментарием HTML . Браузер игнорирует комментарии при отображении кода. Другими словами, на странице их не видно — только в коде. HTML-комментарии позволяют вам писать полезные заметки о вашем коде или логике.

Теперь попробуйте добавить подходящий заголовок на HTML-страницу прямо над `<img>`элементом.

Примечание. Вы увидите, что ваш уровень заголовка 1 имеет неявный стиль. Не используйте элементы заголовков, чтобы сделать текст больше или жирнее, потому что они используются для доступности и по другим причинам, таким как SEO . Постарайтесь создать осмысленную последовательность заголовков на своих страницах, не пропуская уровни.

Пункты Как объяснялось выше, `<p>`элементы предназначены для содержания абзацев текста; вы будете часто использовать их при разметке обычного текстового содержимого:

HTML Скопировать в буфер обмена

`<p>This is a single paragraph</p>` Добавьте свой образец текста (вы должны взять его из Как будет выглядеть ваш сайт?) в один или несколько абзацев, размещенных непосредственно под вашим `<img>`элементом.

Списки Большая часть веб-контента представляет собой списки, и в HTML есть специальные элементы для них. Списки разметки всегда состоят как минимум из 2-х элементов. Наиболее распространенными типами списков являются упорядоченные и неупорядоченные списки:

Неупорядоченные списки предназначены для списков, в которых порядок элементов не имеет значения, например для списка покупок. Они завернуты в `<ul>`элемент. Упорядоченные списки предназначены для списков, где порядок элементов имеет значение, например, рецепт. Они завернуты в `<ol>`элемент. Каждый элемент внутри списков помещается внутрь элемента `<li>`(элемент списка).

Например, если мы хотим превратить часть следующего фрагмента абзаца в список

HTML Скопировать в буфер обмена

`<p>`

```
At Mozilla, we're a global community of technologists, thinkers, and
builders
working together...
```

</p> Мы могли бы изменить разметку на это

HTML Скопировать в буфер обмена

```
<p>At Mozilla, we're a global community of</p>
```

```
<ul>
```

```
<li>technologists</li>
<li>thinkers</li>
<li>builders</li>
```

```
</ul>
```

<p>working together...</p> Попробуйте добавить упорядоченный или неупорядоченный список на страницу примера.

Ссылки Ссылки очень важны — именно они делают сеть сетью! Чтобы добавить ссылку, нам нужно использовать простой элемент — <a>«а» является краткой формой для «якоря». Чтобы превратить текст в абзаце в ссылку, выполните следующие действия:

Выберите текст. Мы выбрали текст «Манифест Mozilla». Оберните текст в <a>элемент, как показано ниже: HTML Скопировать в буфер обмена

```
<a>Mozilla Manifesto</a> Присвойте <a>элементу hrefатрибут, как показано ниже: HTML
Скопировать в буфер обмена
```

```
<a href=»«>Mozilla Manifesto</a> Заполните значение этого атрибута веб-адресом, на который
вы хотите установить ссылку: HTML Скопировать в буфер обмена
```

```
<a href=»https://www.mozilla.org/en-US/about/manifesto/»>
```

Mozilla Manifesto

 Вы можете получить неожиданные результаты, если опустите часть [https:](https://)или <http://>, называемую протоколом , в начале веб-адреса. После создания ссылки щелкните по ней, чтобы убедиться, что она отправляет вас туда, куда вы хотели. Примечание: href поначалу может показаться довольно неясным выбор имени атрибута. Если вам трудно его запомнить, помните, что это гипертекстовая ссылка . Добавьте ссылку на свою страницу сейчас, если вы еще этого не сделали. Заключение Если вы выполнили все инструкции в этой статье, у вас должна получиться страница, похожая на приведенную ниже (вы также можете просмотреть ее здесь): Скриншот веб-страницы с логотипом Firefox, заголовком о том, что Mozilla — это круто, и двумя абзацами текста-заполнителя. Если вы застряли, вы всегда можете сравнить свою работу с нашим готовым примером кода на GitHub. Здесь мы только поцарапали поверхность HTML. Чтобы узнать больше, перейдите к нашей теме Learning HTML .

From:
<https://wwoss.direct.quickconnect.to/> - **worldwide open-source software**

Permanent link:
https://wwoss.direct.quickconnect.to/doku.php?id=software:development:learn:getting_started_with_the_web:html_basics&rev=1692672098

Last update: **2023/08/22 05:41**

