**Конспект по теме "Что такое HTML. Теги для разметки текста и атрибуты"**

**Что такое браузер**

**Браузер** — прикладное программное обеспечение для просмотра веб-страниц. В глобальной сети браузеры используют для запроса, обработки, манипулирования и отображения содержания веб-сайтов.

Браузер может быть:

* клиентским приложением на вашем компьютере;
* приложением на смартфоне;
* другим приложением, в которое встроен браузер;
* браузер может быть доступен и на других устройствах (Smart TV, часы, компьютер автомобиля, холодильник).

Вы сообщаете браузеру что отобразить. Для этого вы указываете ему адрес документа или URL (Uniform Resource Locator) — единый указатель ресурса.

протокол://<хост>/<путь>#<якорь>

**Протокол** — схема обращения к ресурсу (для веб-страниц это http или https).

**Хост** — полностью прописанное доменное имя сервера.

**Путь** — уточняющая информация о месте нахождения ресурса, очень похожа на путь в файловой системе.

**Якорь** — идентификатор «якоря», ссылающегося на некоторую часть (раздел) открываемого документа.

Например, адрес страницы программы, на которой можно зарегистрироваться на курс: <http://netology.ru/programs/html-css-base>.

К URL мы еще вернемся, когда будем разбираться с ссылками и изображениями.

Что делает браузер:

1. Формирует и отправляет запрос на сервер (HTTP).
2. Получает ответ сервера (HTTP).
3. Разбирает ответ сервера, документ (HTML).
4. **Рисует и отображает полученную страницу в окне (HTML и CSS).**
5. Реагирует на действия пользователя (CSS и JavaScript).

HTML документ, полученный с сервера, содержит:

* текстовую информацию;
* структуру документа (заголовки, абзацы и т.д.);
* ссылки на другие ресурсы (изображения, видео, аудио, стили).

**Семантика**

Веб-страницы в первую очередь должны представлять информацию в удобном виде. Давайте подумаем, из чего она состоит. Вот, что мы видим в книгах, журналах, газетах:

* текст, разбитый на абзацы;
* заголовки разных уровней;
* иллюстрации;
* цитаты;
* акценты по тексту.

Их внешний вид может отличаться в различных источниках. Но суть от этого не меняется. Вы с первого взгляда определите заголовок в новой книге. Именно такую информацию браузер должен уметь правильно представлять на экране. Значит, нам необходимо сообщить ему, что является заголовком, абзацем, где сделать акцент, и когда должны следовать иллюстрации.

**Разметка**

Представим, что нам нужно изобрести универсальный способ описания документов, по которому браузер сможет их правильно отобразить на экране. Перед нами стоят задачи:

1. Для создания не должно требоваться сложное программное обеспечение.
2. Дать возможность указать браузеру, что является заголовком, абзацем, иллюстрацией.
3. Процесс создания документа должен быть простым.
4. Сам подход должен быть легок в изучении.

**Для создания не должно требоваться сложное программное обеспечение** — пусть документ будет обычным текстом, тогда для создания потребуется обычный текстовый редактор. Текстовый редактор — блокнот. Текстовый процессор — MS Word.

**Дать возможность указать браузеру, что является заголовком, абзацем, иллюстрацией** — добавим в текст документа простые текстовые метки, по которым браузер поймет, где начинается и заканчивается заголовок, абзац, где должны быть иллюстрации, на какой текст сделать акцент.

**Процесс создания документа должен быть простым** — просто ограничим количество возможных меток, оставив только самое необходимое.

**Сам подход должен быть легок в изучении** — а это мы проверим по итогам курса.

К меткам есть дополнительные требования:

* Метка должна быть уникальной и не использоваться в обычной письменной речи. Поэтому Заголовок — плохой вариант метки, как и символ \*.
* Она должна состоять из небольшого числа символов, иначе их набор будет утомителен. Поэтому *ТутНачинаетсяЗаголовокОтнынеИВоВекиВеков* — плохой вариант метки, хоть и вряд ли встретится в самом тексте.
* По метке должна быть понятна ее суть. Поэтому *ТНЗ29* — тоже плохой вариант метки, хоть короткий и уникальный.

**Язык разметки** — набор символов или последовательностей, вставляемых в текст для передачи информации о его выводе или строении.

**HTML**

**HTML** — *HyperText Markup Language* — «язык гипертекстовой разметки». Cтандартизированный язык разметки документов во Всемирной паутине. HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области вёрстки. HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения.

Просто текст в HTML-документе браузер никак не форматирует. Переносы строк заменяет на пробел, а несколько подряд идущих пробелов показывает как один. Для разметки используются специальные метки, которые обозначают начало и конец абзаца, заголовка и все остальные элементы. Чаще всего их называют тегами. Набор тегов ограничен, и у каждого из них своя роль.

Стандарт поддерживается Консорциумом Всемирной паутины W3C. В стандарт как раз входит допустимый набор тегов, их роль и прочие особенности их использования. Текущая версия стандарта HTML 5.1

**Вёрстка веб-страниц** — это создание структуры html-кода и описание стилей, размещающих элементы веб-страницы (изображения, текст и т. д.) в окне браузера, согласно разработанному макету, таким образом, чтобы элементы дизайна выглядели аналогично макету.

Тэг

Теги в HTML заключены в угловые скобки < >. Нам привычнее их называть символами меньше и больше. Так как нам нужно обозначать начало и конец элемента, например, абзаца, то существует понятие открывающего и закрывающего тега. Тег p позволяет выделить абзац текста (Paragraph по-английски).

123<p>Абзац</p>

<p>Абзац</p>

<p>Абзац</p>

Для разметки заголовков предусмотрено 6 тегов, что позволяет нам различать заголовки разного уровня h1-h6 (Heading).

12345<h1>Заголовок</h1>

<p>Абзац</p>

<h2>Подзаголовок</h2>

<p>Абзац</p>

<p>Абзац</p>

Для разметки блоков цитат, врезок и просто выделенных блоков текста предусмотрен тег blockquote.

1234<h1>Заголовок</h1>

<p>Абзац</p>

<blockquote>Цитата</blockquote>

<p>Абзац</p>

Если не закрыть один из тегов, это может привести к ошибке отображения.

Вложенность тегов

Цитата может состоять из нескольких абзацев текста. Нам потребуется обозначить их внутри тега цитаты. В HTML можно помещать одни теги внутрь других, поэтому мы можем текст внутри цитаты также разметить тегами абзацев:

1234567<h1>Заголовок</h1>

<p>Абзац</p>

<blockquote>

<p>Абзац</p>

<p>Абзац</p>

</blockquote>

<p>Абзац</p>

При этом теги p внутри цитаты будут дочерними для blockquote. А тег blockquote будет родительским для вложенных тегов p. Вспомним ошибку, которую мы разбирали ранее:

1234<h1>Заголовок

<p>Абзац</p>

<blockquote>Цитата

<p>Абзац</p>

Теперь понятно: браузер посчитал, что первый тег p и blockquote вложены в h1. А второй тег p в свою очередь вложен в blockquote.

Не все теги можно вкладывать в другие теги. Например, логика подсказывает, что нет смысла вкладывать абзацы или цитаты в заголовок. Но, например, то, что тег blockquote нельзя вкладывать в абзац, уже не так очевидно. Какие теги в какие можно вкладывать, тоже регулируется стандартом. И там все подчинено очень простой логике. Например, блок цитаты может состоять из нескольких абзацев. А абзац не может состоять из нескольких блоков цитат. Для вставки цитаты по тексту есть тег q.

1<p> Текст <q>Цитата</q> Текст </p>

Теги должны быть вложены только целиком. Т.е. нельзя вложить один тег в два других:

12<p> Текст <q> Начало цитаты </p>

<p> Конец цитаты </q> </p>

Ошибки с нарушением вложенности тегов тоже могут привести к неправильном отображению. Браузеры сейчас достаточно умны и исправят код. Но полагаться на это не стоит.

1234567<p>

Текст

<q>Начало цитаты</q>

</p>

<p>

<q>Конец цитаты</q>

</p>

Взаимосвязь тегов

Мы уже знаем, что вложенные теги считаются дочерними для того тега, в который они вложены. А этот тег будет для них непосредственным родителем.

123<родитель>

<дочерний></дочерний>

</родитель>

Если теги вложены в один и тот же тег и идут друг за другом, то такие теги называются соседями.

12345<родитель>

<сосед></сосед>

<сосед></сосед>

<сосед></сосед>

</родитель>

При этом соседи, которые следуют раньше, — предшественники. А позже — последователи. Ближайший предшественник — предыдущий тег. Ближайший последователь — следующий.

1234567<родитель>

<предшественник></предшественник>

<предыдущий></предыдущий>

<тег></тег>

<следующий></следующий>

<последователь></последователь>

</родитель>

Все теги, вложенные в тег и вложенные во вложенные теги, — потомки.

12345678<родитель>

<потомок></потомок>

<потомок></потомок>

<потомок>

<потомок></потомок>

<потомок></потомок>

</потомок>

</родитель>

Теги для расстановки акцентов по тексту:

* **q** — цитата по тексту;
* **strong** — сделать акцент на слове или фразе, обозначить их важность;
* **em** — выделить слово.

В эти теги нельзя вкладывать никакие другие.

Форматирование кода

Стандарт не регламентирует, как форматировать код. Переносы строк преобразуются в пробелы, несколько пробелов подряд преобразуются в один. Но опыт разработчиков привел к тому, что форматирование кода стало нерегламентированным стандартом индустрии. И все это для увеличения скорости работы верстальщика:

* Каждый уровень вложенности обозначается отступом.
* Отступ составляет два пробела.
* Если у тега нет вложенных, его весь можно записать в одну строку.
* Но если строка слишком длинная, то теги нужно разместить отдельно от содержимого.

Вот так:

12345678<h1>Заголовок</h1>

<p>Короткий абзац</p>

<blockquote>

<p>

Длинный абзац

</p>

<p>Абзац с <em>вложенным</em> тегом </p>

</blockquote>

Атрибуты тегов

Чтобы сообщить браузеру дополнительную информацию о содержимом тегов или задать специальные свойства, используются атрибуты. Атрибуты задаются в открывающем теге:

1234<blockquote cite="А.С. Пушкин"> Мой дядя самых честных правил,

Когда не в шутку занемог,

Он уважать себя заставил.

И лучше выдумать не мог. </blockquote>

Значение атрибута задается в двойных кавычках " " после символа =.

Атрибутов может быть несколько, тогда они перечисляются через пробел.

Есть универсальные атрибуты, которые можно указать в любом теге. Например:

* **id** — позволяет задать тегу универсальный идентификатор, чтобы в дальнейшем ссылаться на него;
* **class** — позволяет задать один или несколько через пробел определенных пользователем классов для дальнейшего управления внешним видом элемента через стили.

Также у некоторых тегов есть специальные атрибуты, доступные только им. Например, у тега blockquote и q есть атрибут cite, который позволяет указать автора или источник цитаты:

1234<blockquote cite="А.С. Пушкин"> Мой дядя самых честных правил,

Когда не в шутку занемог,

Он уважать себя заставил.

И лучше выдумать не мог. </blockquote>

С остальными мы познакомимся при изучении тегов.

Гиперссылка

**Гипертекст** — текст, «который разветвляется как бы сам по себе или выполняет действия по запросу».

Именно благодаря гипертексту интернет стал таким популярным и таким, каким мы его сейчас видим. Так как же нам сделать текст, который разветвляется сам по себе? В HTML для этого предусмотрены ссылки в тексте, которые позволяют по клику перейти на другую страницу или к другой части текущей страницы. Для того чтобы слово или фразу сделать гиперссылкой, её нужно разметить тегом a:

1<p>Текст <a>ссылка</a> текст</p>

Чтобы сообщить браузеру, куда перенаправить пользователя, укажем информацию об этом в специальном атрибуте href (hyper reference – гиперссылка):

1234567<p>Наша <a href="#about">Компания</a> всем компаниям компания</p>

<p>текст</p>

<p>текст</p>

<h2 id="about">О компании</h2>

<p>текст</p>

<p>текст</p>

<p>текст</p>

Самый простой способ задать ссылку — использовать так называемый якорь внутри страницы. Мы можем задать уникальный идентификатор любому тегу с помощью атрибута id и потом сослаться на него из этого же документа, добавив к идентификатору символ решетки # в начале. Браузер прокрутит страницу так, чтобы содержимое этого тега оказалась видно в окне. Это удобно для больших документов или для создания оглавлений. Для того чтобы ссылаться на другие документы, нам нужно изучить, что такое URL.

URL в ссылках

Адрес, который содержит протокол и хост и может содержать дополнительно путь и якорь, называется абсолютным: <http://netology.ru/programs/html-css-base>

Адрес без указания протокола и хоста является абсолютным относительно сервера. При переходе по такой ссылке будет использован протокол и хост открытой страницы: [/programs/html-css-base](https://github.com/netology-code/wm-materials/blob/master/programs/html-css-base).Если эта ссылка на странице <http://netology.ru/development/programs>, то при переходе мы попадем на страницу <http://netology.ru/programs/html-css-base>. А если на странице <http://www.ruscircus.ru/study/pc.shtml>, то при переходе мы попадем на страницу <http://www.ruscircus.ru/programs/html-css-base>.

Адрес без протокола, хоста и ведущего слеша / называется относительным. И при его расчете учитывается не только протокол и хост страницы, но и ее адрес. Принцип тот же, что и при перемещении по файлам и папка на компьютере: programs/html-css-base. Если открыть на странице <http://netology.ru/>, то перейдем на страницу <http://netology.ru/programs/html-css-base>. Если на странице <http://netology.ru/development/programs>, то перейдем на страницу <http://netology.ru/development/programs/html-css-base>. А если на странице <http://netology.ru/development/programs/>, то перейдем на страницу <http://netology.ru/development/programs/programs/html-css-base>.

Особый случай пути . и .. . — текущая папка .. — родительская папка.

Использование ссылок

1<p> Я учусь на курсе Основы HTML и CSS</p>

Добавим ссылку на курс в тексте:

1<p> Я учусь на курсе <a href="http://netology.ru/programs/html-css-base"> Основы HTML и CSS </a> </p>

Если вы хотите сделать заголовок ссылкой, вложите в него тег а.

1<h1> <a href="/news/">Новости компании «Рога и копыта»</a> </h1>

Для ссылки можно задать подсказку в атрибуте title, которая покажется при наведении указателя мыши на ссылку:

12345<h1>

<a href="/news/" title="Переход к разделу Новости компании">

Новости компании «Рога и копыта»

</a>

</h1>

Картинки

Для вставки картинки в документ используется тег img. Этот тег не имеет содержимого и закрывающего тега. Это так называемый одиночный тег. URL файла с изображением указывается в специальном атрибуте src:

123<h1>Заголовок</h1>

<img src="путь">

<p>Текст</p>

После загрузки картинки браузер определит её размер в пикселях и отобразит в полный размер в указанном месте.

Хотя размером изображения можно управлять с помощью стилей, рекомендуется готовить изображения нужного размера, так как размер напрямую влияет на вес файла, а значит и на скорость передачи по сети.

Если картинка не загрузится по каким-либо причинам, можно указать альтернативный текст, который браузер отобразит вместо картинки в этом случае. Там можно написать о том, что изображено на картинке. Для этого есть специальный атрибут alt:

123<h1>Заголовок</h1>

<img src="путь" alt="альтернативный текст">

<p>Текст</p>

Если картинку вставить в тексте, то она будет вести себя как символ, но в полный размер. И в 99% случаев вы получите не то, что ожидали. Такой вариант удобен для вставки символьных изображений, таких как emoji. В остальных случаях помещайте картинку за пределами параграфа.

12<h1>Заголовок</h1>

<p><img src="путь">Текст</p>