

Занятие 1. Основы HTML

Введение

Доброго времени суток, читатель! Это первая глава из курса начинающего фронтенд разработчика.

Возможно, Вы незнакомы с понятие **фронтенд**? Фронтенд (eng. *Front-end*) - это та часть программного обеспечения, которая работает на стороне клиента (т.е. на вашем устройстве) и обменивается данными с серверной частью, которую обычно называют **бэкенд** (eng. *Back-end*). Все достаточно просто.

Современный фронтенд-разработчик сталкивается с довольно разнообразным спектром задач, так что скучать не приходится. Начиная от разработки пользовательских **интерфейсов**, до написания и внедрения **скриптов**, отвечающих за логику в веб-приложениях.

Но пока это все вдалеке от нас. Сейчас мы с вами только в самом начале пути, ведущем в мир Фронтенда.

И как очевидно бы это не звучало, начнем мы познавать этот мир с самых основ - с **HTML**.

HTML или Язык гипертекстовой разметки

HTML (eng. *Hypertext markup language*) - это язык гипертекстовой разметки, который позволяет описать структуру веб-страниц и её **контента** (содержимое страницы).

Задача языка HTML - разбиение тестовых данных в понятные браузеру блоки, которые можно связать с уже существующей информацией при помощи гиперссылок.

Для того, чтобы начать использовать HTML достаточно просто открыть блокнот и сохранить файл с расширением `.html`. После сохранения, иконка файла сменится на иконку вашего браузера, установленного по умолчанию.

В жизни для редактирования HTML документов вместо блокнота обычно используют один из существующих редакторов кода, вроде [Atom](#), [Sublime Text 3](#) или [Visual Studio Code](#).

Я рекомендую Atom, однако выбор редактора кода - дело вкуса.

HTML элементы

Язык HTML представлен множеством элементов, каждый из которых имеет

семантику - смысл, который важен для браузеров и поисковых систем - и некоторый базовый вид в браузере.

К примеру элемент `p` (eng. *Paragraph*):

```
<p>Мой кот очень пушистый</p>
```

Он состоит из открывающего и закрывающего **тегов** (`<p>` и `</p>` , соответственно), а также контента `Мой кот очень пушистый` .

Вместе они образуют **HTML-элемент**, в данном случае элемент-параграф, который браузер распознает как текстовый абзац и после этого определит, что его нужно отделить небольшими отступами сверху и снизу.

Большинство тегов, доступных в языке являются парными, как например `<p>` . Такие теги требуют закрывающей пары, которая отличается лишь наличием косой черты перед названием элемента - `</p>` .

Базовая структура HTML документа

Для лучшего понимания рекомендую создать на вашем локальном диске папку MyProject, а в ней создать файл page.html и самостоятельно писать код и просматривать результат в браузере. Это гораздо интересней, чем простое чтение, а также ускоряет процесс обучения.

Попробуем разобраться со структурой рядового HTML документа, который обычно и называют **веб-страницей**.

```
<!DOCTYPE html>
<html>
  <head>
    <title>О моем коте</title>
  </head>
  <body>
    <p>Мой кот очень толстый.</p>
    <p>Он любит есть.</p>
  </body>
</html>
```

Первая строчка любой веб-страницы это DOCTYPE - формат документа:

```
<!DOCTYPE html>
```

Формат документа указывает браузеру на то, что веб-страница состоит из содержащего HTML-код контента, который нужно отформатировать перед отображением на странице.

Сразу за ним следует парный тег `<html>`

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

Он определяет элемент `html` - область, в которой необходимо писать весь HTML-код. Также его часто называют **корневым** элементом веб-страницы (или просто **корень**).

Весь код, который мы пишем внутри `html`, можно разбить на две основные области. Первая - это элемент `head`.

```
<head>
  <title>>О моем коте</title>
</head>
```

Этот элемент является контейнером для всех ресурсов, которые будут использоваться на странице, но не будут отображены напрямую, как контент. Сюда относятся файлы стилей и скрипты, о которых мы поговорим позднее, разного рода дополнительная информация, вроде кодировки, ключевых слов и прочего.

В нашей ситуации здесь содержится только элемент `title`, который устанавливает название веб-страницы в браузере, отображаемое в панели открытых вкладок.

Второй областью является элемент `body`, который называется **телом страницы**.

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

Именно эта зона будет отображена в браузере, когда вы откроете страницу. Это значит, что всю информацию, которую вы хотите показать на странице, необходимо помещать внутрь элемента `body`.

Вот и сейчас, Желая поделиться с миром подробностями о своем коте, вы вероятно, напишите о нем парочку-другую строк.

```
<body>
  <p>Мой кот очень толстый.</p>
  <p>Он любит есть.</p>
</body>
```

А вообще у меня не толстый кот, а пушистый...

Текстовое форматирование

Думаю, уже пора перейти к одной из базовых возможностей HTML - текстовому форматированию.

В HTML достаточно много элементов, которые позволяют объяснить браузеру, что же содержится на странице.

Один из самых популярных элементов в этой группе мы уже встречали - это элемент `p`.

Подразумевается, что все, что мы пишем далее в этой секции, находится внутри элемента `body`. Современные браузеры поправят вас, если вы поместите информацию вне элемента, однако такая разметка будет считаться **невалидной** - некорректной с точки зрения правил языка HTML.

Заголовки

Один из самых простых способов сделать текст удобным для чтения, помимо выделения абзацев, является разделение его на смысловые части - секции, которым предшествуют заголовки.

```
<h1>О мой кот!</h1>
```

Заголовки разделяют по степени важности (перечислены по убыванию):

```
<h1>О мой кот!</h1>
...
<h2>Тоже мой, только поменьше</h2>
...
<h3>Средних размеров кот</h3>
...
<h4>Кот "Четвертинка"</h4>
...
<h5>Небольшой Кот</h5>
...
<h6>Кот, которого почти никто не замечает</h6>
```

Здесь `h1` - самый главный заголовок, обычно являющийся названием статьи или документа, а `h6` заголовок, выделяющий небольшой подраздел в основном тексте. В общем случае, заголовки одного типа могут повторяться на странице.

Правилом хорошего тона является отсутствие пропусков в номерах заголовков. То есть вместо набора из `h1`, `h4` и `h6`, стоит делать выбор в пользу комбинации `h1`, `h2` и `h3`.

Это создаёт грамотную структуру представляемого контента, которую принято

называть **outline** (eng. *Outline* - схема, набросок).

Списки

При написании докладов, статей и прочей текстовой информации достаточно часто возникает потребность собрать какие-то мысли в список, перечислить их по порядку. В HTML для этих целей используются 3 типа списков:

1. Несортированный список (eng. *Unsorted list*).

```
<ul>
  <li>Кот Белый</li>
  <li>Кот Черный</li>
  <li>Кот "Рыжик"</li>
</ul>
```

Для того, чтобы добавить пункт в список необходимо использовать элемент `li` (eng. *List item*). Каждый элемент данного списка имеет маркер - черный кружок слева от пункта.

2. Упорядоченный список (eng. *Ordered list*)

```
<ol>
  <li>Кот Белый</li>
  <li>Кот Черный</li>
  <li>Кот "Рыжик"</li>
</ol>
```

Упорядоченный отличается от несортированного наличием нумерации каждого элемента списка (от 1 и по порядку).

3. Список определений (eng. *Description list*)

```
<dl>
  <dt>Кот Белый</dt>
  <dd>
    Традиционный дизайн почти всех классов котов.
    Имеется существенный недостаток - на белом цвете
    сильно заметны загрязнения и, как следствие,
    необходимо часто проводить процедуру отчистки.
  </dd>
  <dt>Кот Черный</dt>
  <dd>
    Готический дизайн, характерный для Пражских котов.
    Когда-то имел скверную репутацию, поскольку считалось,
    что если черный кот перейдет дорогу, то будут
    неприятности. Недостатком является неуловимость
    данной разновидности во тьме ночной.
  </dd>
</dl>
```

```
<dt>Кот "Рыжик"</dt>
<dd>...</dd>
</dl>
```

Как видите, список определений несколько отличается от двух предыдущих списков, но он тоже достаточно прост.

Каждый элемент в списке представлен парой "Термин" - "Определение". Термину соответствует элемент `dt` (eng. *Definition term*). В свою очередь определение записывается внутри элемента `dd` (eng. *Definition description*).

Цитаты

Не грех процитировать умную мысль, даже если она принадлежит тебе.

Для того, чтобы добавить цитату на страницу достаточно воспользоваться элементом `blockquote` :

```
<p>Марк Твен однажды сказал:</p>
<blockquote cite="ru.wikiquote.org/wiki/Марк_Твен">
    "Ничто так не нуждается в исправлении, как чужие привычки."
</blockquote>
```

Такой вид цитаты является **блочным**, то есть он, как и абзац, отделяется от остальной части контента. Если же вы хотите добавить в абзац, то следует использовать элемент `q` .

```
<p>Марк Твен однажды сказал: <q cite="ru.wikiquote.org/wiki/Марк_Твен">
    Ничто так не нуждается в исправлении, как чужие привычки.</q></p>
```

При использовании элемента `q` (eng. *Quote*) нет необходимости писать кавычки, поскольку браузер добавит их автоматически.

“Важные” теги

Кажется каждый хоть раз сталкивался с рекламой содержащей броскую фразу, вроде **“Купи кота, а то потом не будет”**, написанной большим и жирным шрифтом. Задача такой рекламы, очевидно, обратить на себя внимание. В тексте эта задача - достаточно частое явления.

HTML предоставляет ряд элементов, которые могут использоваться с целью усиления значения определенных фраз или привлечения внимания читателя к ним. К категории “читатель” можно также отнести и поисковые системы, поскольку они уделяют некоторое внимание выделенным фразам, особенно если это **ключевые слова**.

Раз уж мы затронули вопрос привлечения внимания, то давайте рассмотрим

элемент `em` (eng. *Emphasize*), в чьи обязанности входит создание контраста или акцента на указанном тексте. По умолчанию браузеры отображают содержание этого элемента курсивом.

```
<p><em>Насколько</em> ваш кот пушистый?</p>
```

— *Насколько* ваш кот пушистый?

Браузеры, помимо `em`, выделяют курсивом и элемент `i` (eng. *Italic*). Разница лишь в том, что `i` не имеет никакой семантики (смысла) и используется обычно там, где нужно визуально изменить текст, который чем-то отличается от остального. Например, при использовании названия книги в абзаце.

```
<p><i>Насколько</i> ваш кот пушистый?</p>
```

— *Насколько* ваш кот пушистый?

Далее идет `strong`. Он подчеркивает важность выделенной информации и по умолчанию отображается в браузере, как текст с жирным начертанием.

```
<p>Мой кот <strong>очень</strong> пушистый.</p>
```

— Мой кот **очень** пушистый.

Кроме элемента `strong`, имеется еще элемент `b` (eng. *Bold*), который также позволяет отобразить текст жирным, однако он, как и `i`, не имеет никакой семантики и обычно используется в презентационных целях.

```
<p>Мой кот <b>очень</b> пушистый.</p>
```

— Мой кот **очень** пушистый.

Все эти теги можно комбинировать друг с другом. Например, получим жирный и наклонный текст.

```
<p>Ваш кот <strong><i>достаточно</i></strong> пушистый?</p>
```

— Ваш кот ***достаточно*** пушистый?

Порядок для комбинирования в данном случае не важен, главное соблюдайте правильно уровни вложенности тегов, то есть тот тег, который открылся первым, должен закрыться последним.

Атрибуты HTML-элементов

Думаю, вы заметили, что внутри открывающего тега `<blockquote>` присутствовал **атрибут** `cite=...`. Атрибуты - это дополнительные параметры, которые настраивают элементы, регулируют их поведение различным способом или позволяют передать важную для браузера информацию о элементе.

В данной ситуации атрибут `cite` тега `<blockquote>` сообщает браузеру адрес ресурса, которая является источником цитаты - ru.wikiquote.org/wiki/Марк_Твен.

Все атрибуты располагаются внутри открывающего тега, сразу после его имени в формате:

```
<тег атрибут="значение" атрибут2="значение2" ...>
```

Тег и атрибуты (в том числе между собой) разделяются пробелом, перед закрывающим знаком `>` пробел необязателен.

Существует множество атрибутов, среди которых 16 являются **глобальными** (общими для всех элементов). Подробнее атрибуты мы разберем позднее.

Изображения

Веб - это огромная база знаний, в которой содержится невероятно количество текстовой информации (преимущественно о котах, конечно же). Однако обойтись лишь текстовой информацией трудно.

Человек по своей природе гораздо лучше усваивает визуальный контент. Так почему бы не добавить немного изображений?

```

```



Заметьте, что тег `` не является парным, а значит не имеет закрывающего тега.

Атрибут `src` является неотъемлемым для элемента `img`, поскольку он определяет расположение интересующего нас изображения.

В отличие от атрибута `cite`, который не интересен пользователю, поскольку он никак не влияет на контент, однако помогает поисковой системе, `img` влияет на содержимое страницы: без него браузер бы не знал, откуда загружать фотографию этого прелестного создания.

В свою очередь атрибут `alt` тоже оказывается достаточно значимым для `img`, поскольку заменяет текст-подсказкой изображение, которое по какой-то причине не может быть загружено на страницу.

Он напрямую влияет на **доступность** нашей страницы для людей с ограниченными возможностями, так как программы для чтения текста с экрана (eng. *Screen readers*) озвучивают именно значение этого атрибута.

Список тегов, рассматриваемых в главе

Структура

- `<!DOCTYPE html>` - формат документа, указывающей браузеру, что на странице содержится HTML
- `<html>` - корневой элемент страницы, содержит весь HTML код
- `<head>` - контейнер для вспомогательной информации о документе (ключевые слова, описание и т.д.) и дополнительных ресурсов (скрипты, стили и прочее)
- `<title>` - определяет название документа, отображаемое во вкладке браузера
- `<body>` - тело страницы, содержащее основной контент

Разметка текста

- `<p>` - текстовый абзац
- `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>` - заголовки разного уровня (перечислены по убыванию)
- `<blockquote>` - блочная цитата, отделяется от остального текста отступами
 - `cite="URI"` - атрибут, позволяющий указать адрес ресурса, с которого взята цитата
- `<q>` - цитата внутри основного текста, кавычки добавляются автоматически.
 - `cite="URI"` - тоже, что и у `blockquote`

Текстовое форматирование

- `` - важный текст, имеет жирное начертание

- `` - акцент на тексте, отображается наклонным шрифтом
- `` - текст с жирным начертанием
- `<i>` - текст курсивом

Списки

- `` - несортированный список, элементы маркируются символом (обычно черный круг)
- `` - сортированный список, элементы маркируются цифрами по порядку
- `` - элемент списка
- `<dl>` - список определений
- `<dt>` - термин, содержащийся в списке определений
- `<dd>` - определение термина

Изображения

- `` - элемент изображения на странице
 - `src="Path"` - атрибут, указывающий путь (или адрес) к файлу с изображением