

Занятие 1. Основы HTML

Введение

Доброго времени суток, читатель! Это текстовый вариант 1 урока из курса начинающего фронтенд разработчика.

Возможно, Вы незнакомы с понятие **фронтенд**? Вкратце фронтенд (Eng. *front-end*) - это та часть программного обеспечения, которая работает на стороне клиента (т.е. на вашем устройстве) и обменивается данными с сервером, который обычно называют **бэкэнд** (Eng. *backend*). 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. Ведь сейчас мы с вами находимся только в самом начале пути, ведущем в мир фронтенда.

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

HTML или Язык гипертекстовой разметки

HTML (Eng. *Hypertext markup language*) - это язык гипертекстовой разметки, который позволяет описать структуру веб-страниц с их содержимым (**контентом**).

Если говорить проще, то задача языка HTML - разбиение тестовых данных в понятные для браузера 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](#).

HTML элементы

Язык HTML представлен множеством элементов, каждый из которых имеет **семантику** (проще говоря, смысл) и некоторый базовый вид в браузере.

К примеру элемент параграф (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 поясняет браузеру, что это абзац и его нужно отделить небольшими отступами сверху и снизу.

Большинство тегов, доступных в языке являются парными, как `<p>` . Такие теги требуют закрывающей пары, которая отличается лишь наличием косой черты, вроде `</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 и запускать приведенные примеры кода. Это ускорит процесс обучения и гораздо интересней, чем простое чтение

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 пушистый.</p>
    <p>Он любит есть.</p>
  </body>
</html>
```

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

```
<!DOCTYPE html>
```

Он указывает браузеру на то, что документ состоит из HTML тегов и их не нужно выводить на страницу.

Сразу за ним следуют парные теги `<html>`

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

которые определяют элемент `html` - область, в которой необходимо писать HTML-код. Часто этот элемент называют **корневым** элементом веб-страницы.

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

```
<head>
  <title>>0 моем коте</title>
</head>
```

Этот элемент является контейнером для всего, что будет использоваться на странице, но не будет отображено на прямую, как контент. Сюда относятся файлы стилей и скрипты, о которых мы поговорим позднее, разного рода информация, вроде кодировки, ключевые слова по странице и прочее.

В нашей ситуации здесь содержится только элемент title, который устанавливает название вкладки в браузере, в панели открытых вкладок.

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

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

Именно эта зона будет отображена в браузере, когда вы откроете страницу. Всю информацию, которую вы хотите отобразить на странице необходимо помещать между тегами `<body>...</body>`.

Желая поделиться с миром информацией о своем коте, вы вероятно, напишите о нем парочку-другую строк.

Вот и сейчас мы поместили в `body` немного про пушистого:

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

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

Текстовое форматирование

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

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

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

Подразумевается, что все что мы пишем далее находится между тегами `<body></body>`. Современные браузеры поправят вас, если вы поместите информацию вне этих тегов, однако такая разметка будет считаться **невалидной**, т.е. некорректной с точки зрения правил языка.

Заголовки

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

```
<h1>0 мой кот!</h1>
```

Заголовки разделяют по степени важности.

```
<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 определяющий смысл дальнейшего текста, 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, если угодно

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>Белый кот</li>
  <li>Черная кошка</li>
  <li>Желто-рыжий котопёс</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>Белый кот</li>
  <li>Черная кошка</li>
  <li>Желто-рыжий котопёс</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>
  <dt>Кот Рыжий</dt>
  <dd>...</dd>
</dl>
```

Как видите, список определений несколько отличается от двух предыдущих списков. Однако он ничуть не сложнее: список состоит из пар "Термин"

- "Определение".

Термину соответствует элемент `dt` (Eng. *definition term*). В свою очередь определение записывается внутри элемента `dd` (Eng. *definition description*).