



Лекция 1

Знакомимся с HTML

Цели этого занятия

- Научиться создавать элементарные веб-страницы;
- Определиться с глоссарием;
- Начать разбираться с технологиями.

Основные требования:

- HTML5/CSS3, SASS/LESS, Twitter Bootstrap, создание адаптивных UI
- Работа с макетами (Photoshop, Illustrator, Sketch)
- Хорошее знание Javascript (ES5, ES6), TypeScript будет плюсом
- Опыт работы со следующими фреймворками: Angular, React, Ember, Backbone
- Владение одним или несколькими инструментами сборки, такими как WebPack, Gulp, Grunt
- Опыт разработки мобильных приложений с помощью React Native, Ionic или Xamarin будет плюсом
- AJAX, Websockets, понимание RESTful
- Git/SVN
- Понимание основ SEO
- Базовые знания node.js или других backend-технологий будут плюсом, так же как и опыт работы с СУБД
- Хорошее знание английского

Глоссарий

HTML - документ, в котором мы указываем **контент** -- текст, формы, изображения, адреса загрузки скриптов и стилей.

CSS - документ, в котором указаны **параметры отображения** элементов (цвет, размер, положение).

JavaScript - **язык программирования сценариев**, сейчас широко используется везде, в том числе и в браузере.

Git - система контроля версий. Это **система, регистрирующая изменения в одном или нескольких файлах** с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.

Самостоятельная работа

В локальном репозитории находим файл **README.md** и на девятой строке пишем такой код:

```
* [Профиль на htmlacademy:] (ссылка-на-профиль) .
```

Сохраняем, пушим в форк и создаем pull request.

Что такое HTML тег?



```
~/Courses/basic-course/index.html - Sublime Text (UNREGISTERED)
index.html x
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Document</title>
6 </head>
7 <body>
8
9 </body>
10 </html>
```

Line 8, Column 5

Создать базовый скелет
страницы можно написав знак
“!” и нажав кнопку **Tab**

```
1 <!-- Говорим браузеру, что используем HTML5 -->
2 <!DOCTYPE html>
3 <!-- Сообщаем язык страницы -->
4 <html lang="en">
5 <head>
6     <!-- Указываем кодировку страницы -->
7     <meta charset="UTF-8">
8     <!-- Задаем название вкладки -->
9     <title>Document</title>
10 </head>
11 <body>
12     <!-- А вот здесь пишем весь основной код -->
13 </body>
14 </html>
```


Комментарии <!-- -->

Тег добавляет комментарий в код документа. Текст комментария не отображается на странице. Разрешается внутри комментария добавлять другие теги.

Создать комментарий можно
сокращением клавиш **Ctrl+⧵**

<!DOCTYPE html>

Тег необходим, чтобы браузер понимал, как следует интерпретировать текущую веб-страницу, поскольку есть несколько версий HTML. Кроме того, существует еще более строгий стандарт -- XHTML.

Раньше доктайп выглядел бы так:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

Самый главный тег <html>

В этом теге находится все содержимое сайта.

Никакие теги не должны выходить за границы <html></html>, так как это приведет к ухудшению SEO-позиции.

Заголовок сайта -- тег <head>

Какие теги содержит <head>?

<title>	Отображает название страницы на вкладке
<meta>	Содержит техническую информацию для браузера и поисковика
<link>	Подключает CSS, RSS и favicon
<script>	Подключает JavaScript-файлы

Тело документа <body>

Этот тег является контейнером для всех остальных тегов страницы.

Он всегда открывается после <head> и содержит весь контент страницы, который увидит пользователь.

Тип элементов

Строчные

- Занимают ширины не больше, чем необходимо;
- Являются непосредственной частью другого элемента;
- Привязаны к блоку.

(a, span, input)

Блочные

- Занимают всю доступную ширину окна;
- Начинаются с новой строки;
- Самодостаточны.

(div, h1, p, form, ul)

`<h1>Заголовки <h1> - <h6></h1>`

Заголовки разного уровня соответствуют важности контента после них.

Так, тег `<h1>` представляет собой наиболее важный заголовок (первого уровня), а тег `<h6>` служит для обозначения заголовка шестого уровня и является наименее значительным.

Абзац <p>

<p>Используется для хранения абзаца текста. Каждый новый абзац начинается с новой строки.</p>

<p>Между абзацами сохраняется расстояние.</p>

Строка

Используется для хранения текста, но, в отличие от абзаца, не начинается с новой строки, и занимает ровно столько места, сколько содержит в себе текста.

Позволяет выделять фрагменты в абзаце и задавать им стили.

Гиперссылка <a>

Позволяет ссылаться с одного документа на другой без использования адресной строки. В зависимости от присутствия атрибутов ***name*** или ***href*** тег <a> устанавливает ссылку или якорь.

Атрибут ***href*** указывает путь, по которому пользователь переходит при нажатии на ссылку. Например:

```
<a href="https://google.com.ua">Google it!</a>
```

Изображение

Предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG.

Адрес файла с картинкой задаётся через атрибут **src**.

Атрибут **alt** важен для поисковиков и отображает заданный текст тогда, когда картинка не может загрузиться.

Всегда указывай **alt** для изображений

Горизонтальный разделитель <hr>

Создает горизонтальную линию. Не нуждается в закрывающем теге.

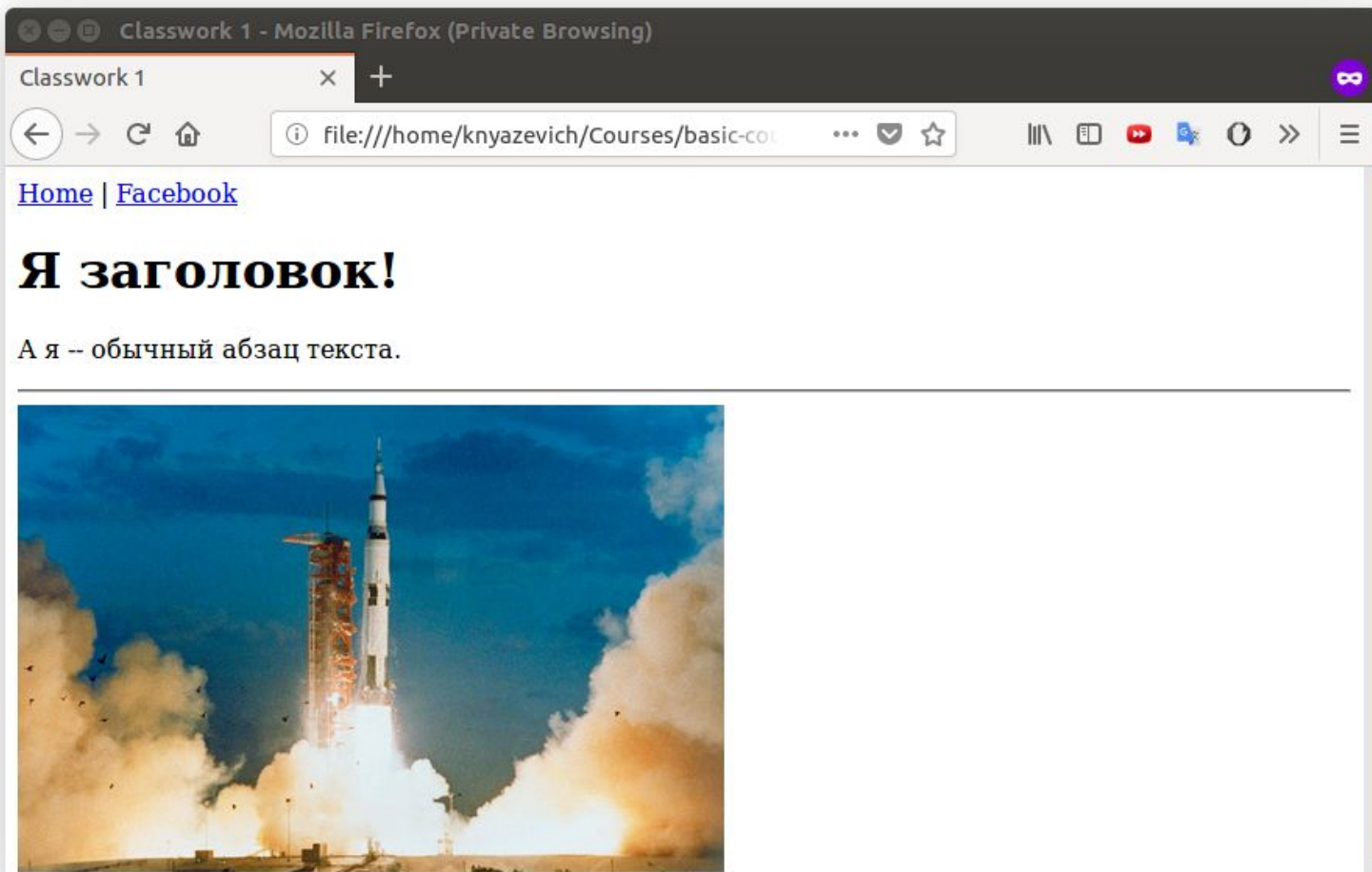
Перевод строки

Устанавливает перевод строки в том месте, где этот тег находится. **Используется в исключительных случаях:** для создания переносов в адресах, стихах, там где нужно добиться точного переноса. Закрывающий тег не нужен.

Самостоятельная работа

Создаем страницу **index.html**, которая содержит всю необходимую базовую разметку и соответствует изображению на следующем слайде.

- Ссылка **Home** никуда не ведет, она заглушечная.
Facebook ведет на facebook.com.
- Картинка кликабельна. При нажатии открывается полная версия.



Проверку проходят исключительно проекты, которые соответствуют всем базовым критериям и сданы в срок. Каждый студент имеет 4 (четыре) попытки сдачи проекта и не более одной в день, поэтому рекомендуем грамотно расходовать свое время и не оставлять все на последний день

Разобранные критерии

- На верстку выпускного макета выделяется **14 дней**, точные даты будут озвучены на восьмой лекции
- На каждое крупное изменение (добавление header, создание новой секции) создается commit. В конце работы локальный репозиторий push'ится на удаленный, где наставники в случайное время могут посмотреть историю и делать замечания на вкладке "Issues". Если замечания не исправлены -- код точно не пройдет проверку.

Разобранные критерии

- Коммиты должны иметь осмысленные названия, которые отражают общее содержание изменений. Чистоте истории коммитов уделяется особое внимание, чистые коммиты важны не меньше, чем чистый код.
- Вёрстка идентично отображается в последних версиях браузеров Chrome, Opera, Edge и Firefox.
- JavaScript\jQuery не используется.

Разобранные критерии

- В файле должен присутствовать DOCTYPE, favicon, стили и шрифты подключаются через `<link>`, присутствуют все базовые элементы.
- Выполнена HTML-разметка всех страниц и всех элементов на страницах.
- Страница проходит проверку валидатором (<http://validator.w3.org> или плагин W3C Validator в Sublime Text 3).
- Код должен выглядеть аккуратно и понятно для стороннего человека. Предпочтителен Google HTML Styleguide (<https://google.github.io/styleguide/htmlcssguide.html#HTML>).