

# **Язык разметки HTML**

Front end Core Morning– 05.09.2017

# План урока

1. Теги
2. Вложенность тегов
3. Атрибуты тегов
4. Комментарии
5. Теги верхнего уровня
6. Семантика
7. Блочные и строчные элементы
8. Теги заголовков
9. Теги форматирования текста
10. Ссылки
11. Изображения
12. Спецсимволы
13. Списки
14. Инструменты отладки
15. Валидация кода
16. Структура проекта

У нас есть текст вида:

## **Деревня**

Приветствую тебя, пустынный уголок,  
Приют спокойствия, трудов и вдохновенья,  
Где льется дней моих невидимый поток  
На лоне счастья и забвенья.

Выделим из него логические блоки

-> Начало заголовка **Деревня** -> Конец заголовка

-> Начало текста

Приветствую тебя, пустынный уголок, -> перенос строки

Приют спокойствия, трудов и вдохновенья, -> перенос строки

Где льется дней моих невидимый поток -> перенос строки

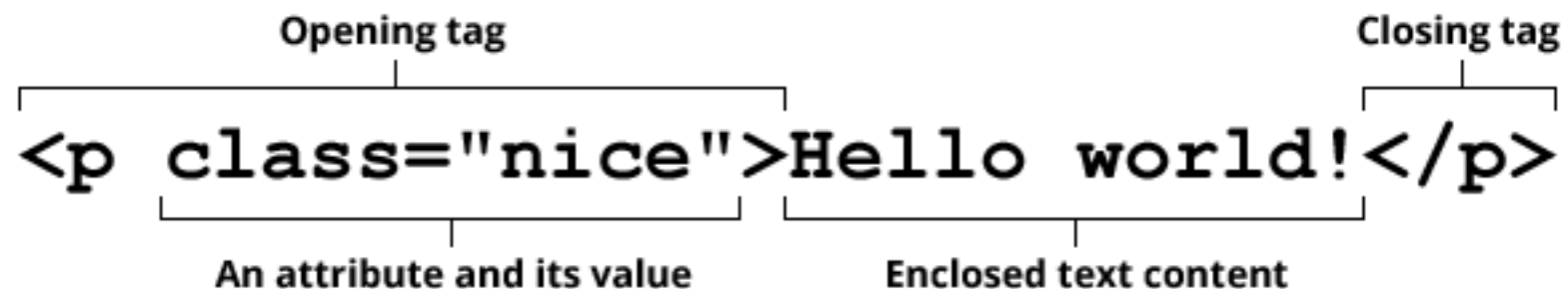
На лоне счастья и забвенья.

-> Конец текста

# Теги

Тег — элемент языка разметки,  
заключенный между символами < и >

*Anatomy of an HTML element*



## Теги бывают:

Двойные - > `<div></div>`

Одиночные -> `<img>`

## **Теги регистронезависимы**

`<html>` и `<HTML>`, или даже `<hTmL>` — одинаковые теги.

## В HTML теги можно не закрывать

<p> Текст в абзаце </p>

<p> Другой текст в абзаце </p>

Браузер сделает это за нас

Так делать **не стоит**

Это вызывает большое количество  
ошибок, т.к мы не контролируем  
Разметку



## Вложенность тегов

*<p>Теги могут быть <b>вложены</b></p>*

Тег должен быть закрыт,  
на том же уровне вложенности

*<p>Теги могут быть <b>вложены</p></b>*

# Атрибуты

  
`<div class="head"> содержимое </div>`

Атрибуты бывают:

- в двойных кавычках
- в одинарных кавычках
- без кавычек
- без значений

# Комментарии в HTML

<div> Это браузер видит </div>

<!--Комментарии будут проигнорированы-->

# Теги верхнего уровня

Весь HTML документ заключен внутри тега `<html></html>`

В теге `<head></head>` находится служебная информация о документе. (Имя документа, иконка, информация для поисковиков и т/д)

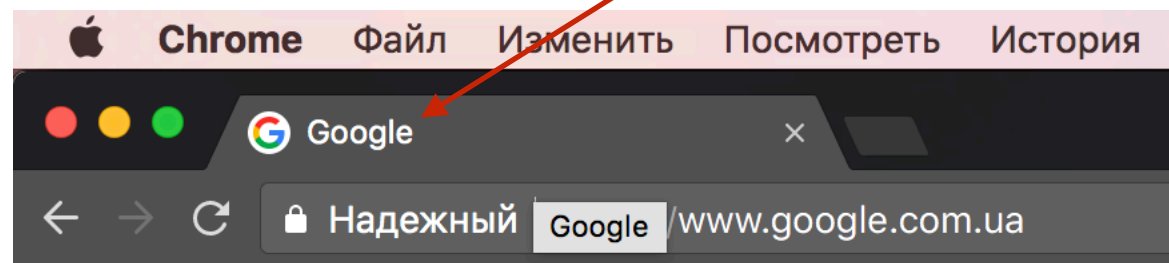
В теге `<body></body>` находится весь контент нашего документа (Текст, изображения и т/д)

# Служебные теги

Внутри тега <meta> мы помещаем служебную информацию, например определение кодировки нашего документа

```
<meta charset="UTF-8">
```

Внутри тега <title> мы помещаем заголовок нашей страницы, который виден вот тут:



**<!DOCTYPE> предназначен для указания типа текущего документа**

```
<!DOCTYPE [Элемент верхнего уровня] [Публичность]  
"[Регистрация]//[Организация]//[Тип] [Имя]//[Язык]" "[URL]">
```

*Например*

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

*Doctype в HTML5*

```
<!DOCTYPE html> <!--html5-->
```

## **Создадим html страницу**

1. Откройте любой текстовый редактор
2. В верхнем меню выберите Файл -> Сохранить как
3. Сохраните файл с именем и расширением index.html
4. Нажмите два раза левой кнопкой мыши
5. Файл должен открыться в браузере по умолчанию

# Добавим в наш файл теги верхнего уровня

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Первая страничка</title>  
  </head>  
  <body>  
  
  </body>  
</html>
```



# Добавим в наш файл немного текста

```
<!DOCTYPE html>
<html>
  <head>
    <title>Первая страничка</title>
  </head>
  <body>
    Деревня
```

Приветствую тебя, пустынный уголок,  
Приют спокойствия, трудов и вдохновенья,  
Где льется дней моих невидимый поток  
На лоне счастья и забвенья.

```
</body>
</html>
```

# Выполним разметку этого текста

```
<!DOCTYPE html>
<html>
  <head>
    <title>Первая страничка</title>
  </head>
  <body>
    <h1>Деревня</h1>
    <p>
      Приветствую тебя, пустынный уголок, <br>
      Приют спокойствия, трудов и вдохновенья,<br>
      Где льется дней моих невидимый поток <br>
      На лоне счастья и забвенья.
    </p>
  </body>
</html>
```

# **Семантика в HTML**

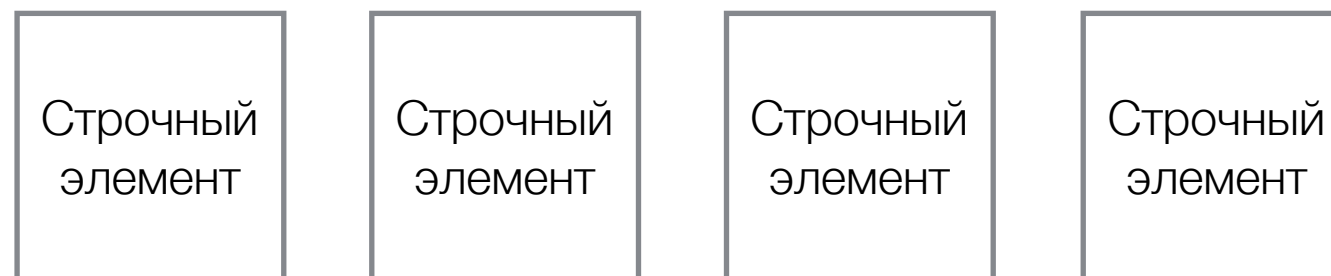
Каждый тег имеет значение

# Блочные и строчные элементы

Блочный элемент занимает всю ширину экрана



Строчные элементы находятся  
в одной строке друг с другом



# Посмотрим как выглядят строчные и блочные элементы в потоке

Блочный элемент

Строчный  
элемент

Строчный  
элемент

Блочный элемент

Строчный  
элемент

## Пример блочных и строчных тегов.

`<div>`Блочный элемент`</div>`

`<span>`Строчный элемент`</span>`

`<span>`Строчный элемент`</span>`

`<div>`Блочный элемент`</div>`

`<span>`Строчный элемент`</span>`

`<span>`Строчный элемент`</span>`

## Теги <h1> - <h6>

Это теги заголовков разного уровня

Цифра показывает уровень заголовка

Чем выше уровень, тем меньше размер шрифта

На странице не должно быть больше одного <h1>

# Пример

`<h1>Главный заголовок страницы</h1>`

`<p>`

...Текст рыба

`</p>`

`<h2>Подзаголовок</h2>`

`<p>`

...Текст рыба

`</p>`

`<h2>Подзаголовок</h2>`

`<p>`

...Текст рыба

`</p>`



## Теги <p>, <br>, <hr>

**<p>** – блочный, семантическое значение абзац

**<br>** – строчный, но семантическое  
значение перенос строки

**<hr>** – блочный элемент, семантическое значение  
разделитель

## Теги форматирования текста

Теги **<b>** и **<strong>** делают текст внутри них **жирным**, с одним отличием: тег `<strong>` означает важный текст, а тег `<b>` просто выделяет его жирными

Теги **<i>** и **<em>** выделяют текст *курсивом(italic)*, тег **<em>** – определяет текст как акцентированный

**Для форматирования текста внутри блоков,  
так же используются следующие теги:**

**<small>** – делает текст в параграфе меньше

**<mark>** – выделяет текст, делает его подсвеченным

**<code>** – с помощью данного тега мы можем вставить  
пример некоторого кода на сайт

**<del>** – для удаленного текста

**<s>** – для некорректного текста, стиль как у <del>

**<u>** – делает текст подчеркнутым, был удален

в HTML 4.01, но заново введен в стандарт в HTML 5

# Другие теги, связанные с форматированием

**<blockquote>** – Тег обозначающий цитату. Есть его упрощенная версия **<q>**, которая отличается в стандартной стилизации. В атрибуте cite=‘АВТОР\_ЦИТАТЫ’ мы можем сослаться на автора или произведение

**<abbr>** – Тег обозначающий аббревиатуру. В атрибуте title=“указываем расшифровку аббревиатуры

**<bdo>** – используется с обязательными атрибутами dir=‘ltr’ или dir=‘rtl’  
Позволяет определить направление текста внутри. Удобно использовать, когда нам нужны вставки текста из арабских языков.

**<pre>** – Позволяет сохранить исходные переносы строк в html

# Практическое задание

Преобразовать данный текст в валидную HTML верстку:

## История Front-End разработчика

Я пришёл на курсы, чтобы изучить основы создания сайтов. От преподавателя я узнал, что семантика в HTML, это очень **важно**.  
Есть теги, на которых *стоит акцентировать свое внимание*.  
Я думаю будет не лишним их подчеркнуть. Это теги input и img  
Они опасны тем, что ~~я их не знаю~~ их необязательно закрывать.  
Особенно хочу выделить тег `code` с помощью него, мы можем вставлять примеры нашего кода на сайт, и стилизовать их.

## Бонус задание

- 1) Оформить текст, согласно семантическому значению, все переносы строк, сохранить как в исходном варианте, используя <pre>
- 2) Записать высказывание от имени УМО(Украинское Математическое Сообщество, это аббревиатура):

*Высказывание УМО, касательно научной работы*

*Проводя следующий анализ полученных нами формул, мы пришли к выводу, что цитата арабского ученого Аль-Кинди: «Истиной никого нельзя унижить, наоборот, истина облагораживает всякого» как нельзя лучше, подходит для нашей работы.*

- 3) Цитату оформить на арабском(перевести в Google Translate) и задать правильное направление текста.

# Добавляем динамики. Ссылки в HTML

Ссылки в html задаются с помощью тега  
**<a href='http://ссылка'>Текст ссылки</a>**

- 1) **Якорные ссылки** – ссылаются на элемент на текущей странице
- 2) **Внешние ссылки** – ссылаются на внешние сайты
- 3) **Внутренние ссылки(относительные)** – ссылаются на страницы текущего сайта

# Пример ссылок в тексте

<p>

Товарищи! сложившаяся структура организации требуют определения и уточнения форм развития. Равным образом

<a href="/new-model/">новая модель организационной</a>

<!-- Пример внутренней ссылки в тексте-->

деятельности представляет собой интересный эксперимент проверки модели развития. Равным образом

<a href="https://google.com?q=Укрепление">укрепление</a>

<!-- Пример внешней ссылки в тексте-->

и развитие структуры способствует подготовки и реализации существенных финансовых и административных условий.

<a href="#borders">рамки</a>

<!-- Пример якорной ссылки в тексте-->

и место обучения кадров позволяет выполнять важные задания по разработке существенных финансовых и административных условий.

</p>



# Атрибуты у ссылок

- 1) **href** указывает куда попадет пользователь при клике на ссылку. Является обязательным
- 2) **title** позволяет назвать ссылку, будет отображен при наведении на неё
- 3) **target** указывает в каком окне(вкладке) нужно открывать ссылку
- 4) В **hreflang** можно указать язык открываемой ссылки
- 5) **type** показывает тип контента по данной ссылке(изображение, html страница, etc.)

# Ссылки можно(и нужно) помещать внутри других тегов.

Пример заголовка-ссылки

```
<h1>  
  <a href="https://google.com">  
    Учимся работать с google  
  </a>  
</h1>
```

# Изображения

А что в HTML можно вставлять картинки?



# Тег <img>

```

```

## Атрибуты тега <img>

- 1) src – ссылка на картинку(относительная или абсолютная)
- 2) alt – текст, который будет отображен, если картинка недоступна.  
Очень любят поисковики.
- 3) width, height – ширина и высота изображения
- 4) srcset – альтернативный путь к картинке(для ретины например)
- 5) usemap – имя карты(map) изображений

# Что может быть картинкой?

Файлы .png, .jpg, .gif, .svg

Строка в формате base64.

# Создание изображения-ссылки

```
<a href="/link/to/img">  
      
</a>
```

# Специальные символы в HTML

Для отображения символов, которых нет на клавиатуре, применяются специальные знаки, начинающиеся с амперсанда (&) и заканчивающиеся точкой с запятой (;)

Найти список таких СИМВОЛОВ МОЖНО ТУТ:

[https://www.w3schools.com/html/html\\_symbols.asp](https://www.w3schools.com/html/html_symbols.asp)

$$a_2 \in \mathbb{R}$$

`<p>a<sub>2</sub> &#8712; R</p>`

# Разбираемся со списками

Списки бывают двух видов:

Маркированные

- Вода
- Соль
- Сода
- Мука

`<ul>`

`<li>Вода</li>`

`<li>Соль</li>`

`<li>Сода</li>`

`<li>Мука</li>`

`</ul>`

Нумерованные

1. Вода
2. Соль
3. Сода
4. Мука

`<ol>`

`<li>Вода</li>`

`<li>Соль</li>`

`<li>Сода</li>`

`<li>Мука</li>`

`</ol>`



# Вложенные списки

Для того, чтобы сделать вложенный список, мы просто помещаем в элемент списка – еще один список

```
<ul>
  <li>
    Сливочный соус
    <ul>
      <li>Масло</li>
      <li>Яица</li>
      <li>Растительное масло</li>
    </ul>
  </li>
  <li>Соль</li>
  ...
</ul>
```

Чтобы сделать списки отличными друг от друга, мы можем изменить стиль маркеров у этих списков:

**list-style-type:** circle | disc | square | armenian | decimal | decimal-leading-zero | georgian | lower-alpha | lower-greek | lower-latin | lower-roman | upper-alpha | upper-latin | upper-roman

# **Инструменты отладки Chrome Dev Tools**

# Валидатор HTML кода

<https://validator.w3.org/>

# Структура вашего проекта

`index.html` //Главный файл, он же точка входа

`css/` // Файлы стилей, css-файлы

`js/` // Javascript файлы нашего сайта

`/lib` // Опционально, библиотеки JS, которые мы подключаем

`fonts/` // Шрифты для нашего сайта

`img/` // Изображения

С помощью структуры – мы упрощаем читаемость больших проектов. Найти какой-либо файл, становится легче.

# Оформление HTML кода

- Вложенный блок должен визуально быть внутри родительского (это реализуется за счет увеличения отступа от края документа). Для отступов у вложенных элементов используются пробелы (2 или 4).
- Теги и их атрибуты пишутся строчными буквами. Для значений атрибутов всегда используются двойные кавычки.
- Атрибут класса у HTML-элементов пишется первым. Единообразное написание помогает легче считывать код и быстрее разбираться в назначении блоков по их классам. Остальные атрибуты могут быть расставлены в любом порядке, но тоже единообразно для одинаковых элементов.
- Кодировка символов на странице всегда должна быть явно указана, чтобы обеспечить корректное отображение текста. Кодировка utf-8 предпочтительна.
- Для элемента `html` в атрибуте `lang` должен указываться соответствующий язык документа. Это помогает инструментам синтеза речи определить, какое использовать произношение или системам перевода, какие использовать языковые правила.
- Код должен содержать комментарии. Комментарий должен отвечать на вопрос "Зачем этот код", а не "Что этот код делает". Со временем вы будете легко понимать что происходит в комментируемом фрагменте. Но даже Вы сами, через 2-3 недели, уже не вспомните для чего предназначен именно этот участок кода и подсказки в виде комментариев будут очень кстати.
- Все имена собственные должны быть "говорящими" и только на английском языке, т.е. должно быть понятно содержимое элемента по его названию (в HTML этот принцип относится, в первую очередь, к названиям файлов и изображений).