DATA GROUP

Full-stack

GEN AI

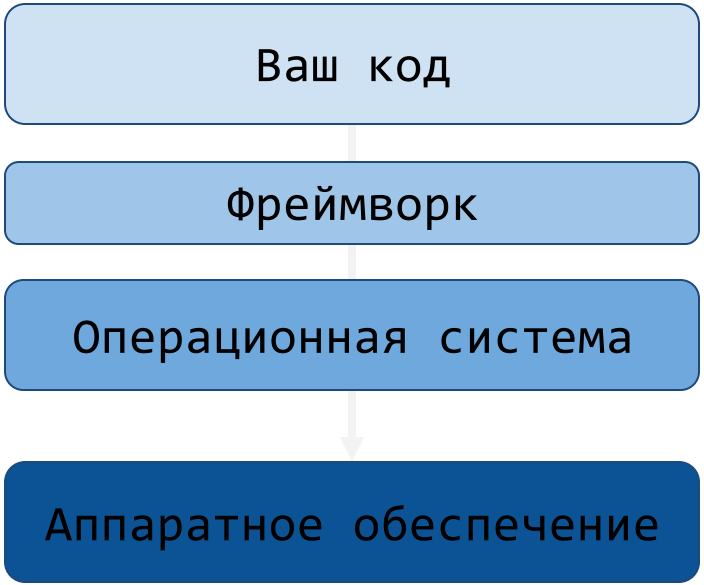
План:

- Введение
- Инструменты
- Технологии
- Основные правила HTML
- Основные теги HTML

Введение

Когда вы решаете разработать приложение с использованием любого языка программирования, одной из первых проблем, с которой вы сталкиваетесь, является то, что языки программирования не включают библиотеку для создания пользовательских интерфейсов.

Вам необходимо использовать какой-либо фреймворк для доступа к уровню ОС. Каждый язык программирования имеет как минимум один, но вам необходимо сначала его выбрать.

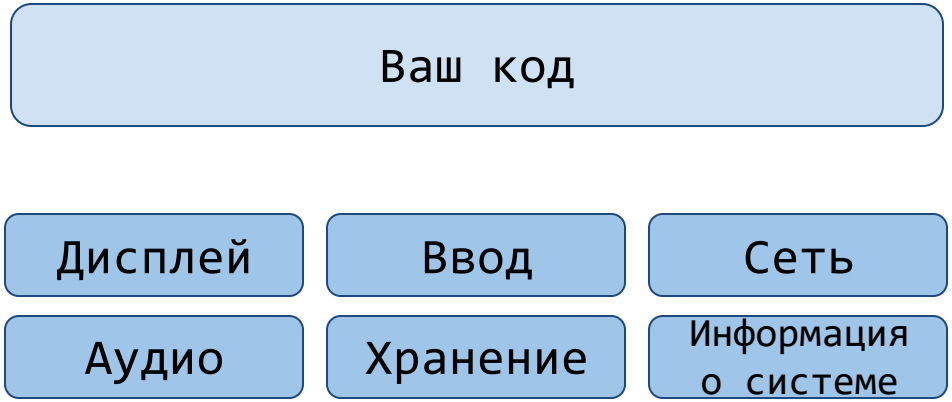


Фреймворк и ОС

Обычно от фреймворка для доступа к ОС ожидают трех вещей:

- Отображение информации на экране
- Получение ввода от пользователя
- Запрос данных из Интернета.
- Воспроизведение аудио
- Хранение данных
- Для получения системной информации, такой как дата, разрешение экрана и т. д.

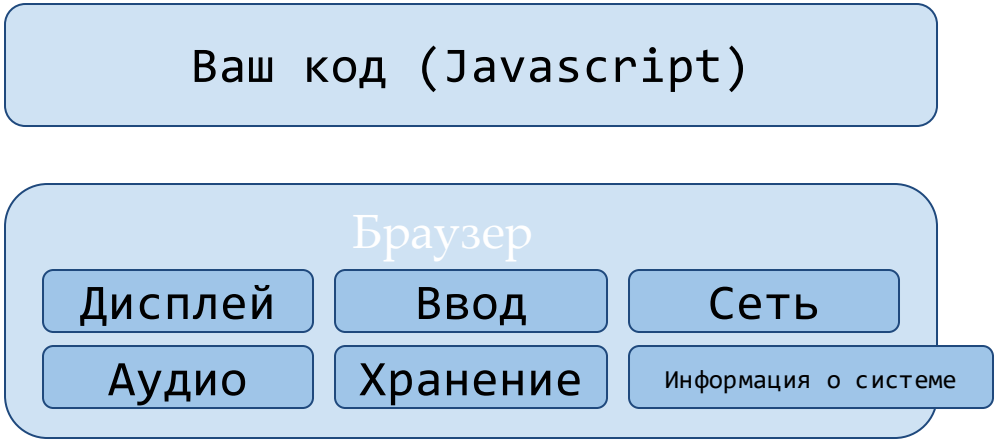
Каждый язык программирования имеет свой набор библиотек для выполнения всех этих задач, но иногда их настройка может быть



Веб-фреймворк

Одним из преимуществ разработки для веб-приложений является то, что веб предоставляет очень богатый и простой фреймворк для создания приложений, включающих множество функций, не только интерфейс, но и доступ к периферийным устройствам (аудио, ввод, геймпады и т. д.), и этот API очень прост в использовании.

А поскольку этот API является универсальным, он работает в любом браузере на любой платформе.

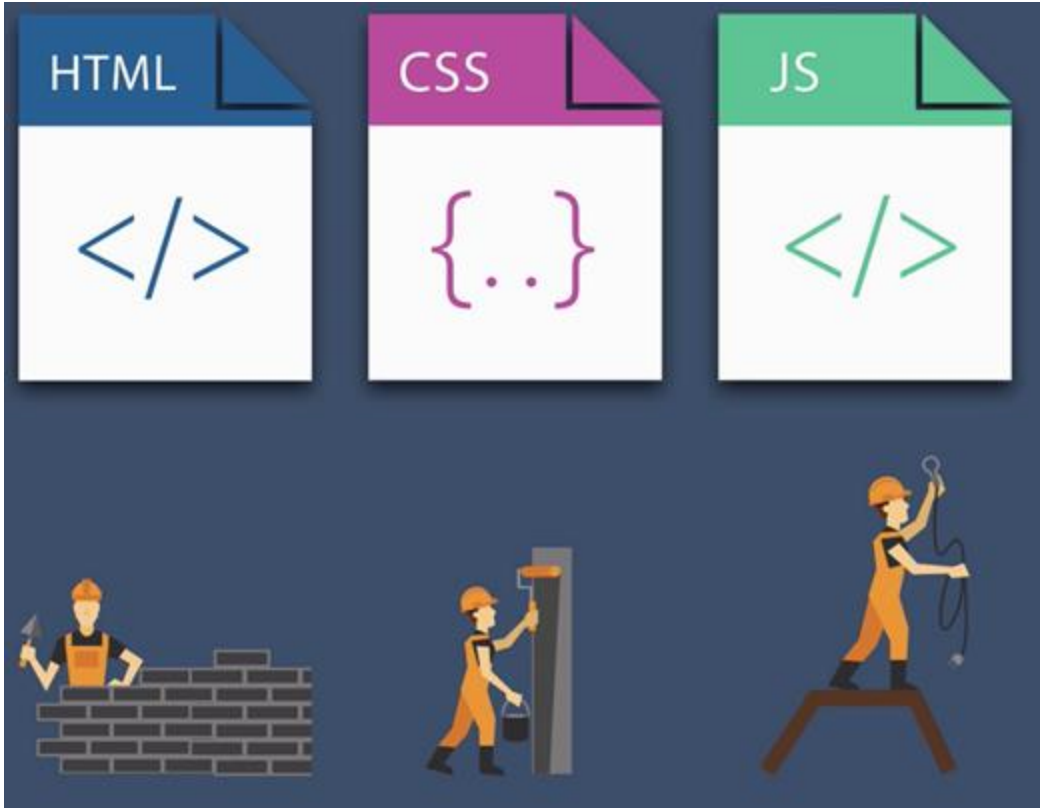


Общая картина

Введение в веб-технологии:

- HTML для создания структуры и содержания документа
- CSS для управления визуальным аспектом
- Javascript для обеспечения интерактивности

Как работают сайты



Как работают сайты (2)

HTML

HTML + CSS

HTML + CSS + JavaScript

Инструменты

Что нужно для начала:

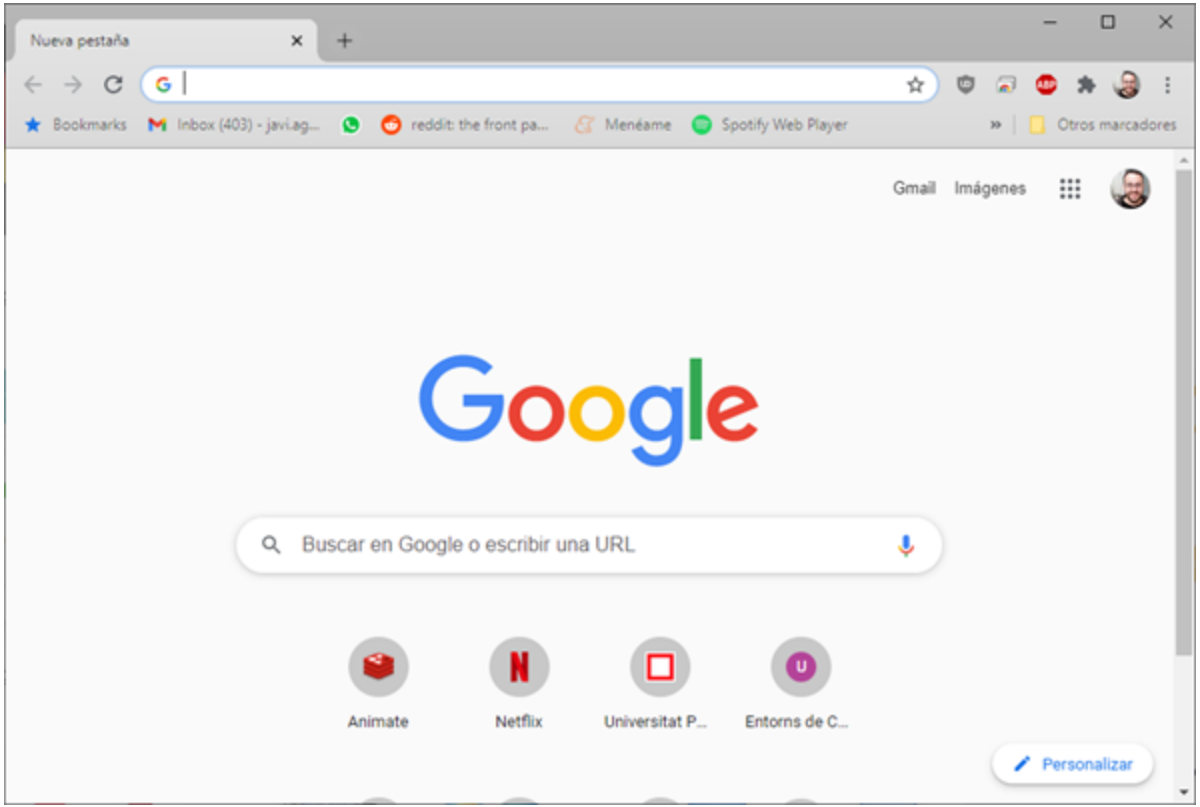
- хороший веб-браузер (Chrome или Firefox)
- хороший текстовый редактор, например:
  - VSCode (кроссплатформенный)
  - Notepad++ (Win)
  - textWrangler (osx)
  - sublime text (кроссплатформенный)
  - ecode (кроссплатформенный)
- пример HTML-кода для начала

Как я могу протестировать свой код?

Просто откройте файл index.html из шаблона в текстовом редакторе и в браузере.
Когда вы вносите какие-либо изменения в код, проверьте их в браузере, нажав F5 (обновить сайт).
Чтобы открыть инструменты разработчика, нажмите:
Windows: `Control + Shift + I` или 

OSX: `Command + Opt + I`

Анатомия браузера

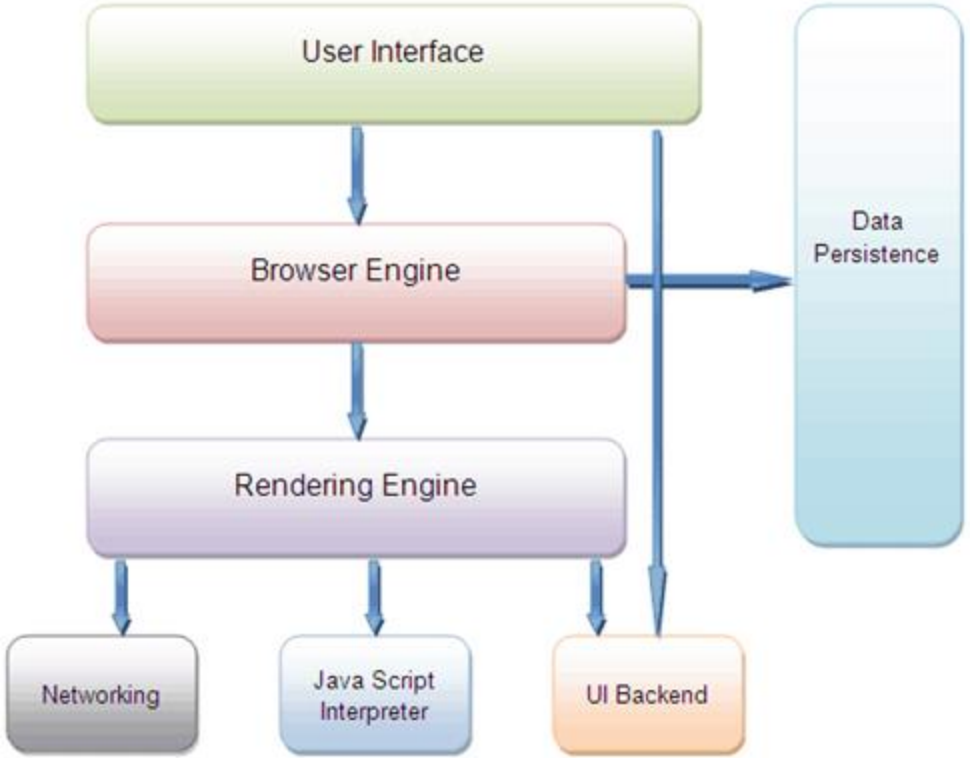


Внутри браузера

Браузеры состоят из нескольких различных частей.

Нас интересуют две из них:

- движок рендеринга (отвечает за преобразование HTML+CSS в визуальное изображение).
- Интерпретатор Javascript (также известный как VM), отвечающий за выполнение кода Javascript.



Технологии

- HTML
- CSS
- Javascript

Браузеры как рендереры

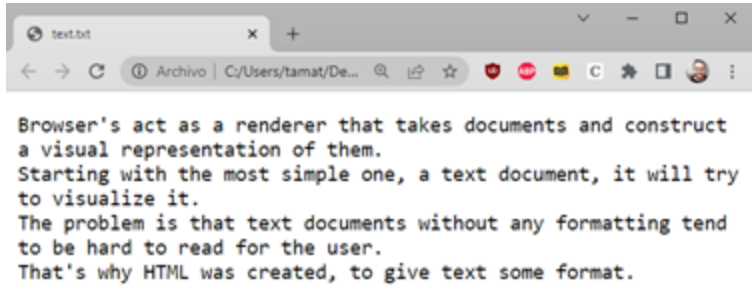
Браузеры действуют как рендереры, которые принимают документы и создают их визуальное представление.

Начиная с самого простого, текстового документа, он попытается визуализировать его.

Вы можете попробовать перетащить любой файл .txt в браузер, чтобы визуализировать его.

Проблема заключается в том, что текстовые документы без форматирования, как правило, трудно читать пользователю (и довольно скучно).

Именно для этого и был создан HTML — чтобы придать тексту некоторое форматирование.



Язык разметки

Существует множество языков разметки, которые добавляют в текст специальные теги, которые рендерер не отображает, но использует для определения способа отображения текста.

В HTML эти теги используют следующую нотацию:

```
Меня зовут <b>Javi</b>
```

HTML

HTML означает «язык гипертекстовой разметки».

HTML позволяет нам определять структуру документа или веб-сайта.

HTML — это **НЕ** язык программирования, а язык разметки, то есть его цель — придать структуру содержанию веб-сайта, а не определять алгоритм.

Это серия вложенных тегов (подмножество XML), которые содержат всю информацию веб-сайта (например, тексты, изображения и видео). Вот пример тегов:

```
<title>    заголовок</title>
```

HTML определяет структуру страницы. Веб-сайт может иметь несколько HTML-кодов для разных страниц.

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

        <div>

        &lt;p&gt;Приве
т&lt;/p&gt;

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

HTML: основные правила

Некоторые правила HTML:

- Используется синтаксис XML (теги с атрибутами, могут содержать другие теги).

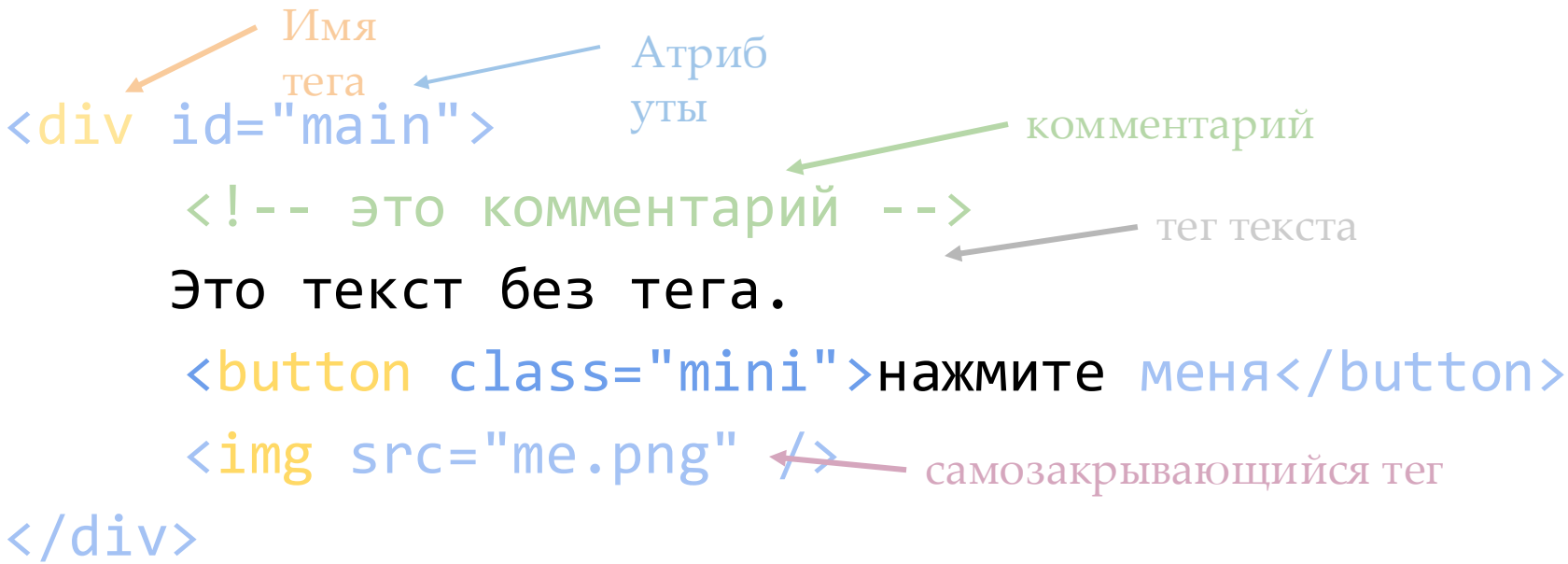
```
<tag_name attribute="value"> content </tag_name>
```

- Он хранит всю информацию, которая должна быть показана пользователю.
- Существуют различные элементы HTML для разных типов информации и поведения.
- Информация хранится в древовидной структуре (узлы, содержащие узлы внутри), называемой DOM (Document Object Model).
- Он придает документу некоторую семантическую структуру (например, это заголовок, это раздел, это форма), которая помогает компьютерам понимать содержание веб-сайтов.
- Она не должна содержать информацию о том, как она должна отображаться (эта информация принадлежит CSS), поэтому не должно быть информации о цвете,

HTML: пример синтаксиса

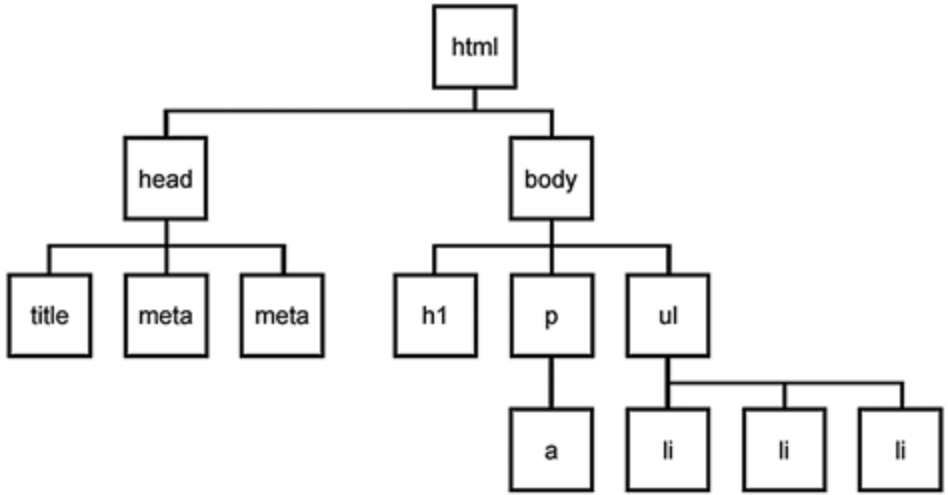
```
<div id="main">
      <!-- это комментарий -->
      Это текст без тега.
      <button class="mini">нажмите меня</button>
      <img src="me.png" />
</div>
```

HTML: пример синтаксиса



DOM представляет собой дерево

Каждый узел может иметь только одного родителя, а каждый узел может иметь несколько детей, поэтому структура выглядит как дерево.



HTML: основные теги

Хотя в спецификации HTML есть много тегов, 99% веб-сайтов используют поднабор HTML-тегов, состоящий менее чем из 10 тегов, наиболее важными из которых являются:

- : контейнер, обычно представляет собой прямоугольную область с информацией внутри.
- <img/>: изображение
- <a>: кликабельная ссылка для перехода по другому URL
- <p>: абзац текста
- <h1>: заголовок (h2, h3, h4 — заголовки меньшей важности)
- <input>: виджет, позволяющий пользователю вводить информацию
- <style> и <link>: для вставки правил CSS
- <script>: для выполнения Javascript
- : нулевой тег (ничего не делает), подходит для маркировки информации

HTML: другие интересные теги

Есть несколько тегов, которые иногда могут быть полезны:

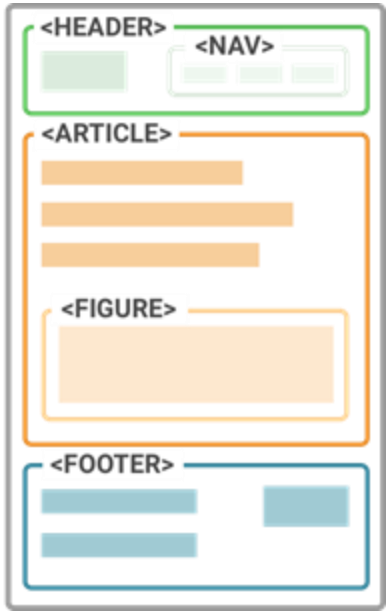
- <button> для создания кнопки
- <audio>: для воспроизведения аудио
- <video>: для воспроизведения видео
- <canvas>: для рисования графики с помощью javascript
- <iframe>: для размещения другого веб-сайта внутри нашего

HTML: обозначение информации

Мы используем HTML-теги для обертывания различной информации на нашем сайте.

Чем более структурирована информация, тем проще будет получить к ней доступ и представить ее.

Мы можем изменять способ представления информации на экране в зависимости от тегов, в которых она содержится, поэтому не стоит беспокоиться о том, что мы используем слишком много тегов.



Правильное использование HTML

Хорошо, когда вся информация правильно обернута в теги, которые придают ей некоторую семантику.

Мы также можем расширить семантику кода, добавив дополнительные атрибуты к тегам:

- id: указывает уникальный идентификатор для этого тега
- class: указывает общий идентификатор для этого тега
- hidden: указывает браузеру не показывать этот элемент

```
<div id="profile-picture" class="mini-image">...</div>
```