**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И науки Самарской области**

**государственное Бюджетное профессиональное   
образовательное учреждение самарской области**

**«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ»**

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ**

**ПО ПРОХОЖДЕНИЮ**

**ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**

**ПМ.04 СОПРОВОЖДЕНИЕ И ОБСЛУЖИВАНИЕ   
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ   
КОМПЬЮТЕРНЫХ СИСТЕМ**

***09.02.07 Информационные системы и программирование***

***технический профиль***

**ДЛЯ СТУДЕНТОВ ОЧНОЙ ФОРМЫ ОБУЧЕНИЯ**

**Самара, 2023 г.**

|  |  |  |
| --- | --- | --- |
| ОДОБРЕНО  Предметно - цикловой  (методической) комиссией  Председатель:  \_\_\_\_\_\_\_\_\_\_\_ Е.В. Третьякова «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023г. |  | Составлено в соответствии  с требованиями ФГОС СПО по  специальности 09.02.07 Информационные системы и программирование  Рекомендовано к изданию решением методического совета №\_\_\_\_\_\_\_  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |
| СОГЛАСОВАНО  Заместитель директора по учебной работе  \_\_\_\_\_\_\_\_\_\_\_\_ Е.М. Садыкова  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |  | Председатель совета Заместитель директора по учебной работе и научно-исследовательской деятельности  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.Ю. Нисман  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |

|  |  |
| --- | --- |
| **Составитель:** | Нуриддинов С.Х., преподаватель ГБПОУ «ПГК». |
| **Рецензенты:** | Синева О.В., методист ГБПОУ «ПГК»;  Третьякова Е.В., преподаватель ГБПОУ «ПГК». |

Методические рекомендации по организации и прохождению производственной практики являются частью учебно-методического комплекса (УМК) ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем.

Методические рекомендации определяют цели и задачи, конкретное содержание, особенности организации и порядок прохождения производственной практики студентами, а также содержат требования по подготовке отчета о практике.

Методические рекомендации адресованы студентам очной формы обучения.

В электронном виде методические рекомендации размещены на файловом сервере колледжа, а также на сайте колледжа по адресу: [www.pgk63.ru](http://www.pgk63.ru).

|  |  |  |
| --- | --- | --- |
| МП.1130.2019 |  | © ГБПОУ «Поволжский  государственный колледж» |

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ 4

1 ЦЕЛИ И ЗАДАЧИ ПРАКТИКИ 6

2 Знакомство с VS Code 13

3 Что такое HTML 16

4 Что такое CSS 17

5 Что такое Prettier 19

6 Изучение JS 20

7 О проекте 21

ПРИЛОЖЕНИЕ А Инициализация проекта 24

ПРИЛОЖЕНИЕ Б Знакомство с DOM структурой 25

ПРИЛОЖЕНИЕ В Настройка prettier 26

ПРИЛОЖЕНИЕ Г Изучение тегов в html 28

ПРИЛОЖЕНИЕ Д Написание полной структуры html 29

ПРИЛОЖЕНИЕ Е Селекторы css 36

ПРИЛОЖЕНИЕ Ж Застилизовать весь проект 39

ПРИЛОЖЕНИЕ И Что такое пользовательская часть сайта 40

ПРИЛОЖЕНИЕ К Написать все javascript функции 41

# ВВЕДЕНИЕ

**Сайт** — это взаимосвязанные веб-страницы с общим доступом, имеющие общий домен. Веб-сайты могут быть созданы разными людьми, группами лиц, компаниями или организациями для самых разных целей. Все сайты составляют так называемую Всемирную паутину.  
В этом материале мы рассмотрим назначение сайта, его виды и то, что нужно для создания и поддержки личного веб-сайта.

**Для чего нужен сайт**

Веб-сайт это то место, где компания может опубликовать подробную информацию о себе. Такой подход обеспечивает рост доверия со стороны потенциальных клиентов. Сайт информирует людей в любое время дня и ночи. На нем легко опубликовать информацию, которую сложно реализовать с помощью традиционных рекламных методов.

Также не стоит забывать, что сайт привлекает новых клиентов, потому что охват аудитории намного больше. По сути, владельцу доступен рынок, численность которого равна количеству пользователей интернета. Небольшим плюсом является то, что наличие сайта сообщает, что компания продвинута в технических вопросах.

**Виды сайтов**

Обычно выделяется 3 вида веб-сайтов. Рассмотрим каждый из них подробнее.

**Сайт-визитка**

Данный термин стал широко использоваться на личных сайтах дизайнеров и веб-разработчиков. По сути, сайт-визитка представляет собой простой одностраничник, который отображает личную информацию, контактные данные и ссылки на другие социальные сети, подобно тому, как традиционная тактильная визитная карточка используется для отображения основной информации. Основная функция такого сайта — ознакомление потенциальных клиентов. Такой площадке иногда достаточно одной страниц, чтобы у будущих пользователей сложилось правильное первичное впечатление.

**Лендинг**

Это страница, на которой вы оказываетесь при переходе по ссылке. Если смотреть на термин так, то лендингом можно назвать любые сайты. Но обычно — это страница, созданная для получения и преобразования трафика. Это вносит ясность в определение. Лендинг — не просто одностраничный сайт. Это страница, созданная специально для маркетинга.

**Интернет-магазин**

Этот вид сайта важен для современного человека. Люди предпочитают совершать покупки онлайн, не выходя из дома. Наличие на сайте иконок MasterCard/Visa — признак страницы, вызывающей доверие. Поэтому нынешним бизнес-проектам, которые ориентированы на прибыль, нельзя игнорировать интернет-пространство. Сегодня все серьезные организации имеют собственный интернет-магазин: от крупных сетевых гипермаркетов до аптек.

# 1. ЦЕЛИ И ЗАДАЧИ ПРАКТИКИ

**Цели практики:**

* + - 1. Создание **Сайт-Лендинг**
      2. Работа с DOM структурой
      3. Изучение нужных нам программ (табл. 1)

Таблица №1

Программы & Фреймворки

|  |
| --- |
| **Программы** |
| VS Code |
| HTML |
| CSS |
| ESLint |
| Prettier |
| JS |
| Emmet |
| Git, git-baush |

# 2. Знакомство с VS Code

**Visual Studio Code** (VS Code) — [текстовый редактор](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%B8%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D0%B0), разработанный [Microsoft](https://ru.wikipedia.org/wiki/Microsoft" \o "Microsoft) для [Windows](https://ru.wikipedia.org/wiki/Windows), [Linux](https://ru.wikipedia.org/wiki/Linux" \o "Linux) и [macOS](https://ru.wikipedia.org/wiki/MacOS" \o "MacOS). Позиционируется как «лёгкий» редактор кода для [кроссплатформенной](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D1%8C) разработки [веб](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5)- и [облачных](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BB%D0%B0%D1%87%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F) приложений. Включает в себя [отладчик](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BB%D0%B0%D0%B4%D1%87%D0%B8%D0%BA), инструменты для работы с [Git](https://ru.wikipedia.org/wiki/Git" \o "Git), [подсветку синтаксиса](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D1%81%D0%B2%D0%B5%D1%82%D0%BA%D0%B0_%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81%D0%B0), [IntelliSense](https://ru.wikipedia.org/wiki/IntelliSense" \o "IntelliSense) и средства для [рефакторинга](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3" \o "Рефакторинг). Имеет широкие возможности для [кастомизации](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%81%D1%82%D0%BE%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F" \o "Кастомизация): пользовательские темы, [сочетания клавиш](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%87%D0%B5%D1%82%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BA%D0%BB%D0%B0%D0%B2%D0%B8%D1%88) и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с [открытым исходным кодом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), но готовые сборки распространяются под [проприетарной лицензией](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BF%D1%80%D0%B8%D0%B5%D1%82%D0%B0%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5" \o "Проприетарное программное обеспечение).

Visual Studio Code основан на [Electron](https://ru.wikipedia.org/wiki/Electron" \o "Electron) и реализуется через веб-редактор Monaco, разработанный для Visual Studio Online.

Visual Studio Code был анонсирован 29 апреля 2015 года компанией [Microsoft](https://ru.wikipedia.org/wiki/Microsoft" \o "Microsoft) на [конференции Build](https://ru.wikipedia.org/wiki/Build_(%D0%BA%D0%BE%D0%BD%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D0%B8%D1%8F_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%87%D0%B8%D0%BA%D0%BE%D0%B2)), и вскоре была выпущена [бета-версия](https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D1%82%D0%B0-%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F).

18 ноября 2015 года Visual Studio Code был выпущен под [лицензией MIT](https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D1%86%D0%B5%D0%BD%D0%B7%D0%B8%D1%8F_MIT), а исходный код был опубликован на [GitHub](https://ru.wikipedia.org/wiki/GitHub" \o "GitHub). Анонсирована поддержка расширений.

14 апреля 2016 года Visual Studio Code вышел из стадии [бета-тестирования](https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D1%82%D0%B0-%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5). Visual Studio Code — это [редактор исходного кода](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%B8%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D0%B0). Он имеет многоязычный интерфейс пользователя и поддерживает ряд [языков программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), [подсветку синтаксиса](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D1%81%D0%B2%D0%B5%D1%82%D0%BA%D0%B0_%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81%D0%B0), [IntelliSense](https://ru.wikipedia.org/wiki/IntelliSense" \o "IntelliSense), [рефакторинг](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3" \o "Рефакторинг), [отладку](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BB%D0%B0%D0%B4%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D1%8B), навигацию по коду, поддержку [Git](https://ru.wikipedia.org/wiki/Git" \o "Git) и другие возможности. Многие возможности Visual Studio Code недоступны через [графический интерфейс](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F), зачастую они используются через палитру команд или [JSON](https://ru.wikipedia.org/wiki/JSON)-файлы (например, пользовательские настройки). Палитра команд представляет собой подобие [командной строки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%BE%D0%B9_%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B8), которая вызывается [сочетанием клавиш](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%87%D0%B5%D1%82%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BA%D0%BB%D0%B0%D0%B2%D0%B8%D1%88).

Также VS Code поддерживает редактирование и выполнение файлов типа «Блокнот Jupyter» ([Jupyter Notebook](https://ru.wikipedia.org/w/index.php?title=Jupyter_Notebook&action=edit&redlink=1" \o "Jupyter Notebook (страница отсутствует))) напрямую «из коробки» без установки внешнего модуля в режиме визуального редактирования и в режиме редактирования исходного кода.

На март 2019 года посредством встроенного в продукт пользовательского интерфейса можно загрузить и установить несколько тысяч расширений только в категории «programming languages» (языки программирования).

Также расширения позволяют получить более удобный доступ к программам, таким как [Docker](https://ru.wikipedia.org/wiki/Docker" \o "Docker), [Git](https://ru.wikipedia.org/wiki/Git" \o "Git) и другие. В расширениях можно найти линтеры кода, темы для редактора и поддержку синтаксиса отдельных языков.

# 3. Что такое HTML

**HTML** (от английского HyperText Markup Language) — это язык гипертекстовой разметки текста. Он нужен, чтобы размещать на веб-странице элементы: текст, картинки, таблицы и видео.

Когда вы заходите на сайт, браузер подгружает HTML-файл с информацией о структуре и контенте веб-страницы. HTML как бы выстраивает визуальный фундамент сайта, но не «запускает» сайт самостоятельно. Он всего лишь указывает, где располагаются элементы, какой у них будет базовый дизайн, откуда брать [стили для элементов](https://skillbox.ru/media/code/chto_takoe_css/) и скрипты (обычно их пишут на [JavaScript](https://skillbox.ru/media/code/chto-takoe-javascript-i-zachem-on-nuzhen/)).

HTML состоит из тегов — команд, которые указывают браузеру, как отображать помещённый в них текст. Это и есть элементы веб-страницы. У каждого тега есть имя, которое заключается в угловые скобки < и >.

**HTML** — это каркас сайта. По нему браузер отрисовывает веб-страницу, наводит «красоту» с помощью CSS и добавляет логику через JavaScript. Например, в HTML-файле можно прописать:

* гиперссылки;
* таблицы;
* изображения;
* блоки;
* абзацы;
* формы;
* заголовки.

Можно даже задавать простой дизайн напрямую в HTML — например, устанавливать цвет и шрифт текста или фоновый цвет блока. Однако профессионалы не рекомендуют так делать — лучше указывать стили в файле CSS. Там-то ваша фантазия ничем не будет ограничена: задавайте любые отступы, размещайте элементы нестандартными способами, играйтесь с прозрачностью, тенями и анимацией. Всё что угодно — но в CSS. Договорились? :)

## 4. Что такое CSS

CSS (Cascading Style Sheets) — это код, который вы используете для стилизации вашей веб-страницы. *Основы CSS* помогут вам понять, что вам нужно для начала работы. Мы ответим на такие вопросы как: Как сделать мой текст черным или красным? Как сделать так, чтобы контент появлялся в определённом месте на экране? Как украсить мою веб-страницу с помощью фоновых изображений и цветов? Как и HTML, CSS на самом деле не является языком программирования. Это не язык разметки - это *язык таблицы стилей.* Это означает, что он позволяет применять стили выборочно к элементам в документах HTML. Например, чтобы выбрать **все** элементы абзаца на HTML странице и изменить текст внутри них с чёрного на красный, вы должны написать этот CSS:

p {

color: red;

}

Значение свойства (Property value)

Справа от свойства, после двоеточия, у нас есть значение свойства, которое выбирает одно из множества возможных признаков для данного свойства (существует множество значений color, помимо red).

Обратите внимание на важные части синтаксиса:

* Каждый набор правил (кроме селектора) должен быть обёрнут в фигурные скобки ({}).
* В каждом объявлении необходимо использовать двоеточие (:), чтобы отделить свойство от его значений.
* В каждом наборе правил вы должны использовать точку с запятой (;), чтобы отделить каждое объявление от следующего.

Существует множество различных типов селектора. Выше мы рассматривали только селектор элементов, который выбирает все элементы определённого типа в HTML документе. Но мы можем сделать выбор более конкретным.

Выше я уже сказал, что CSS используется для всего визуального оформления документов. Но чтобы стало яснее, посмотрим на примеры использования разметки в коде. Так вы поймете, что возможностей у CSS куда больше, чем кажется.

Также можно добавлять контент или CSS-разметку на новые объекты, созданные вне HTML. Для этого в CSS есть псевдоселекторы **:after** и **:before** (они позволяют разместить новые элементы до или после выбранного объекта).

По умолчанию стили прописываются в текстовом файле, но это не единственный вариант. Базовый CSS не обладает преимуществами сложных языков программирования, переменными или вложением свойств.

Поэтому разработчики создали препроцессоры наподобие LESS. Это механизм на базе JavaScript, позволяющий внедрить в работу со стилями синтаксис и некоторые примитивные функции из JavaScript. Если вам кажется, что у CSS недостаточно функциональности, то следует ознакомиться с препроцессорами.

## 5. Что такое Prettier

[**Prettier**](https://prettier.io/) — это средство для форматирования кода, которое нацелено на использование жёстко заданных правил по оформлению программ. Оно форматирует код автоматически. Вот как это выглядит.

Вот какие возможности и особенности Prettier позволяют говорить о полезности этого инструмента:

* Приведение в порядок существующей кодовой базы. Подобное, с помощью Prettier, можно выполнить буквально одной командой. Ручная обработка больших объёмов кода займёт гораздо больше времени. Представьте себе, например, затраты труда, необходимые для того, чтобы вручную отформатировать 20000 строк кода.
* Prettier легко внедрить. Prettier использует «усреднённый», наименее спорный подход к стилю при форматировании кода. Так как проект это опенсорсный, многие внесли в него вклад, улучшая его и сглаживая острые углы.
* Prettier позволяет сосредоточиться на написании кода, а не на его форматировании. Многие просто не осознают того, как много времени и сил тратится на форматирование кода. Использование Prettier позволяет не думать о форматировании, а заниматься вместо этого программированием. В моём случае, например, эффективность работы, благодаря Prettier, выросла на 10%.
* Prettier помогает начинающим. Если вы — начинающий программист, работающий в одной команде с серьёзными профессионалами, и вы хотите достойно смотреться на их фоне, в этом вам поможет Prettier.

## 6. Изучение JS

**JavaScript** — это язык программирования, который используют для написания frontend- и backend-частей сайтов, а также мобильных приложений. Часто в текстах и обучающих материалах название языка сокращают до JS. Это язык программирования высокого уровня, то есть код на нем понятный и хорошо читается.

JS поддерживают все популярные браузеры. Во frontend-части сайтов язык используют для создания интерактива (анимаций, всплывающих форм, автозаполнения), так как он связан с [HTML](https://blog.skillfactory.ru/glossary/html/) и [CSS](https://blog.skillfactory.ru/glossary/css/) и может ими манипулировать. В backend-части с языком JavaScript работают на платформе Node.js. С ее помощью, например, разрабатывают серверные веб-приложения и подключают библиотеки. В поисковике Google на JavaScript работает строка автозаполнения, а Netflix, Uber, eBay используют его в своем backend. Уже 6 лет JS — [самый популярный](https://incrussia.ru/news/yazyki-programmirovaniya/) язык среди разработчиков по версии [GitHub](https://blog.skillfactory.ru/glossary/github-2/" \t "_blank).

**Особенности JavaScript**

JavaScript называют языком скриптов или сценариев. Скрипты — это набор инструкций, которые выполняются при загрузке страницы. Браузер самостоятельно интерпретирует код на JavaScript, для этого даже не требуется компиляция (перевод языка программирования в машинный код).

Скрипты можно прописать внутри кода страницы или подключить к HTML отдельным файлом. Например в стандартной разметке index.html JS-код прописывают внутри тега script, помещенного в тег body:<!DOCTYPE html> <html lang=”en”> <head> <meta charset=”UTF-8”> <meta name=”viewport” content=”width=device-width, initial-scale=1.0”> <title>Document</title> </head> <body> <h1>Привет!</h1> <script> console.log(‘Hello world!’) </script> </body> </html>

Тогда в браузере благодаря инструкции console.log появится слово «Привет!».

JS — мультипарадигменный язык, так как он поддерживает разные парадигмы (стили) программирования:

* Объектно-ориентированный — представляет программу как систему отношений между объектами — наборами данных и функций.
* Императивный — в этом стиле программирования используют инструкции, которые выполняются последовательно, а для доступа к данным используются переменные, значение которых может быть переписано в ходе вычисления.
* Функциональный — в нем вычисление основано на математических функциях. Так как функциональное программирование не опирается на переменные, которые могут менять значение, результаты запросов будут всегда одинаковыми.
* Веб-страницы частично обрабатываются с помощью JavaScript на компьютере пользователя. Это снижает нагрузку на сервер: часть операций выполняется без запросов к нему, что экономит время и трафик.

**Где применяется JavaScript**

* **Клиентская часть веб**—приложений (frontend). Это интерфейс страницы, то есть всё, что видит пользователь: контент, кнопки, формы обратной связи, меню. С помощью JS интерфейс реагирует на действия пользователя (клики мыши, нажатия клавиш), также язык отвечает за запоминание данных и автозаполнение форм.
* **Серверная часть веб**—приложений (backend). Серверный код пишут на платформе Node.js. На JS работают, например, запросы AJAX (asynchronous javascript and XML), которые отправляются на сервер в фоновом режиме, без перезагрузки веб-страницы, и push-уведомления — всплывающие сообщения в браузере, которые реализуются с помощью технологии Comet. Такие уведомления приходят со специального Comet-сервера, который постоянно поддерживает соединение с браузером. Как раз с помощью JavaScript устанавливается это соединение.
* **Мобильные приложения на Android, iOS, Windows Mobile** — когда нужно кросс-платформенное приложение или адаптация веб-приложения, а языков [Kotlin](https://blog.skillfactory.ru/glossary/kotlin/" \t "_blank) (для Android) и Swift (для iOS) недостаточно, то используется JavaScript.

# 7. О проекте

В этом проекте мы создаём сайт **Лендинг** на котором будет реклама и продажа товара.

**Лендинг** - **пейдж** (от англ. landing page), или просто лендинг, — это посадочная страница, на которую пользователи попадают через рекламу в социальных сетях, благодаря релевантно настроенному таргетингу или баннерной рекламе.

**Наш сайт.**

Веб-сайт в честь гонки состоявшего в Sitio do Escampadinho по названию Lamborghini Super Trofeo Гранд-финал 2022.



Рисунок 1

.

# ПРИЛОЖЕНИЕ А Инициализация проекта

**ПОДГОТОВКА VS CODE К ПРОЕКТУ**

Для начало мы должны установить нужные нам расширение такие как:

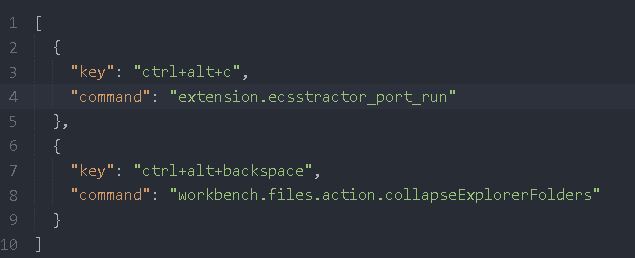
* CSS Peek
* eCSStractor for VSCode
* Gist
* Sass
* HTML to CSS autocompletion
* IntelliSense for CSS class names in HTML
* Live Server
* One Dark Pro

**Пресет настроек settings.json**



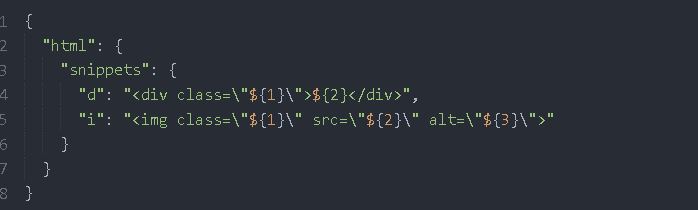
Рисунок

**Горячие клавиши keybindings.json**

****

Рисунок

**Пример сниппета emmet/snippets.json**

****

Рисунок

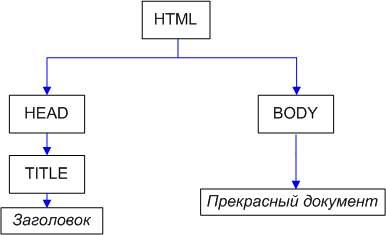
# ПРИЛОЖЕНИЕ Б Знакомство с DOM структурой

**ЧТО ТАКОЕ DOM СТРУКТУРА ?**

**DOM** (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Document Object Model — «объектная модель документа») — это независящий от платформы и языка [программный интерфейс](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9), позволяющий [программам](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) и [скриптам](https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82) получить доступ к содержимому [HTML](https://ru.wikipedia.org/wiki/HTML)-, [XHTML](https://ru.wikipedia.org/wiki/XHTML)- и [XML](https://ru.wikipedia.org/wiki/XML)-документов, а также изменять содержимое, структуру и оформление таких документов.

Модель DOM не накладывает ограничений на структуру документа. Любой документ известной структуры с помощью DOM может быть представлен в виде дерева узлов, каждый узел которого представляет собой элемент, атрибут, текстовый, графический или любой другой объект. Узлы связаны между собой отношениями «родительский-дочерний».

ПРИМЕР DOM СТРУКТУРЫ:



Рисунок

# ПРИЛОЖЕНИЕ В Настройка prettier

# ЧТО ТАКОЕ PRETTIER И КАК ЕГО НАСТРОИТЬ

У Prettier есть огромные преимущества:

* Чистит весь код проекта. В качестве альтернативы можете самостоятельно пересмотреть 20 тысяч строк кода.
* Без проблем настраивается под нужды конкретного проекта.
* Позволяет сконцентрироваться на главном. Вы даже не представляете, сколько времени и энергии тратите на форматирование. Займитесь логикой приложения, а его красотой займется Prettier.
* Облегчает жизнь новичкам. Благодаря настроенным правилам молодые программисты могут работать вместе с профессиональными разработчиками.

### Настройка

Создайте директорию app и инициализируйте проект с помощью команды npm init -y. Дальше в примерах будет использоваться yarn, но вы можете продолжать управлять пакетами через npm.

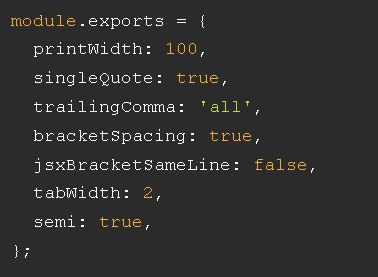
Установите первый пакет: yarn add --dev prettier

package.json теперь выглядит так:



Рисунок

У нас уже почти все готово для реализации второго варианта, осталось лишь создать файл конфигурации для Prettier. Он называется prettier.config.js.



Рисунок

Пробежимся по настройкам:

* printWidth следит, чтобы строчки кода были не длиннее 100 символов.
* singleQuote меняет двойные кавычки на одинарные.
* trailingComma расставляет [висящие запятые](https://medium.com/@nikgraf/why-you-should-enforce-dangling-commas-for-multiline-statements-d034c98e36f8) после последнего свойства объектов.
* bracketSpacing ставит пробелы после открывающей и перед закрывающей скобками объектных литералов.

# ПРИЛОЖЕНИЕ Г Изучение тегов в html

**ЧТО ТАКОЕ ТЕГИ В HTML**

Тег (html-тег, тег разметки) — управляющая символьная последовательность, которая задает способ отображения гипертекстовой информации.

HTML-тег состоит из имени, за которым может следовать необязательный список атрибутов. Весь тег (вместе с атрибутами) заключается в угловые скобки<>:< имя\_тега [атрибуты]>. Как правило, теги являются парными и состоят из начального и конечного тегов, между которыми и помещается информация.

Основными частями элемента являются:

Открывающий тег: Он состоит из названия (обозначения) элемента (в нашем случае, p), помещённого внутри угловых скобок. Данный тег служит признаком начала элемента, с этого момента тег начинает влиять на следующее после него содержимое.

Закрывающий тег: выглядит как и открывающий, но содержит слеш перед названием тега. Он служит признаком конца элемента. Пропуски закрывающих тегов — типичная ошибка новичков, которая может приводить к неопределённым результатам — в лучшем случае всё сработает правильно, в других страница может вовсе не прорисоваться или прорисоваться не как ожидалось.

Содержимое: Как видно, в нашем случае содержимым является простой текст.

Элемент: открывающий тег + закрывающий тег + содержимое = элемент.



Рисунок

[**Вложенные элементы**](https://developer.mozilla.org/ru/docs/Learn/HTML/Introduction_to_HTML/Getting_started#%D0%B2%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5_%D1%8D%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D1%8B)

Вы также можете вкладывать элементы внутрь других элементов — это называется **вложенностью**. Если мы хотим подчеркнуть, что наш кот **очень** сердитый, мы можем заключить слово "очень" в элемент [<strong>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/strong) , который означает, что это слово крайне важно в данном контексте:

**<p>Мой кот <strong>очень</strong> сердитый.</p>**

[**Блочные и строчные элементы**](https://developer.mozilla.org/ru/docs/Learn/HTML/Introduction_to_HTML/Getting_started#%D0%B1%D0%BB%D0%BE%D1%87%D0%BD%D1%8B%D0%B5_%D0%B8_%D1%81%D1%82%D1%80%D0%BE%D1%87%D0%BD%D1%8B%D0%B5_%D1%8D%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D1%8B)

Существует две важных категории элементов в HTML, которые вам стоит знать — элементы блочного уровня и строчные элементы.

* Элементы блочного уровня формируют видимый блок на странице — они окажутся на новой строке после любого контента, который шёл до них, и любой контент после них также окажется на новой строке. Чаще всего элементами блочного уровня бывают структурные элементы страницы, представляющие собой, например, параграфы (абзацы), списки, меню навигации, футеры, или подвалы, и т. п. Элементы блочного уровня не вкладываются в строчные элементы, но иногда могут вкладываться в другие элементы блочного уровня.
* **Строчные элементы** — это те, которые содержатся в элементах блочного уровня и окружают только малые части содержимого документа, не целые абзацы и группировки контента. Строчные элементы не приводят к появлению новой строки в документе: они обычно встречаются внутри абзаца текста, например, элемент [<a>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/a) (ссылка) или акцентирующие элементы вроде [<em>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/em) или [<strong>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/strong).

Посмотрите на следующий пример:

**<em>Первый</em><em>второй</em><em>третий</em>**

**<p>четвёртый</p><p>пятый</p><p>шестой</p>**

[**Пустые элементы**](https://developer.mozilla.org/ru/docs/Learn/HTML/Introduction_to_HTML/Getting_started#%D0%BF%D1%83%D1%81%D1%82%D1%8B%D0%B5_%D1%8D%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D1%8B)

Не все элементы соответствуют вышеупомянутому шаблону: открывающий тег, контент, закрывающий тег. Некоторые элементы состоят из одного тега и обычно используются для вставки чего-либо в то место документа, где размещены. Например, элемент [<img>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/img) вставляет картинку на страницу в том самом месте, где он расположен:

**<img src="https://raw.githubusercontent.com/mdn/beginner-html-site/gh-pages/images/firefox-icon.png">**

Это выведет на вашу страницу следующее:

****

Рисунок

# ПРИЛОЖЕНИЕ Д Написание полной структуры html

**ЧТО ТАКОЕ СТРУКТУРА HTML**

В дополнение к определению отдельных частей вашей страницы (таких как «абзац» или «изображение»), [HTML](https://developer.mozilla.org/ru/docs/Glossary/HTML) также содержит ряд элементов блочного уровня, используемых для определения областей вашего веб-сайта (такие как «заголовок», «навигационное меню», «колонка основного содержимого»). В этой статье рассматривается, как планировать базовую структуру сайта и писать HTML для представления этой структуры.

[**Основные составляющие документа**](https://developer.mozilla.org/ru/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure#%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D0%BD%D1%8B%D0%B5_%D1%81%D0%BE%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D1%8F%D1%8E%D1%89%D0%B8%D0%B5_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0)

Веб-страницы могут и будут отличаться друг от друга, но все они, преимущественно, состоят из аналогичных стандартных компонентов, если только страница не отображает полноэкранное видео или игру, не является частью какого-либо художественного проекта или просто плохо структурирована:

**Заголовок (колонтитул)**

Обычно это большая полоса вверху страницы, с крупным заголовком и / или логотипом. Здесь указывается общая информация о веб-сайте, не меняющаяся от страницы к странице.

**Навигационное меню**

Ссылки на основные разделы сайта; обычно в виде кнопок, ссылок или вкладок. Также как и заголовок, навигация остаётся неизменной на всех страницах сайта — наличие непоследовательной навигации на вашем сайте запутает и разочарует пользователей. Многие веб-дизайнеры считают панель навигации частью заголовка, а не отдельным компонентом, но это не является обязательным требованием; на самом деле, некоторые также утверждают, что их разделение на отдельные компоненты улучшает [доступность](https://developer.mozilla.org/ru/docs/Learn/Accessibility), поскольку раздельная структура будет понятнее для людей, пользующихся считывателями экрана.

**Основное содержимое**

Большая область в центре страницы, содержащая, в основном, уникальный контент данной веб-страницы, например видео, которое вы хотите посмотреть, или рассказ, который вы читаете, или карту, которую вы хотите просмотреть, или заголовки новостей и т. д. Это одна из частей сайта, которая определённо будет меняться от страницы к странице!

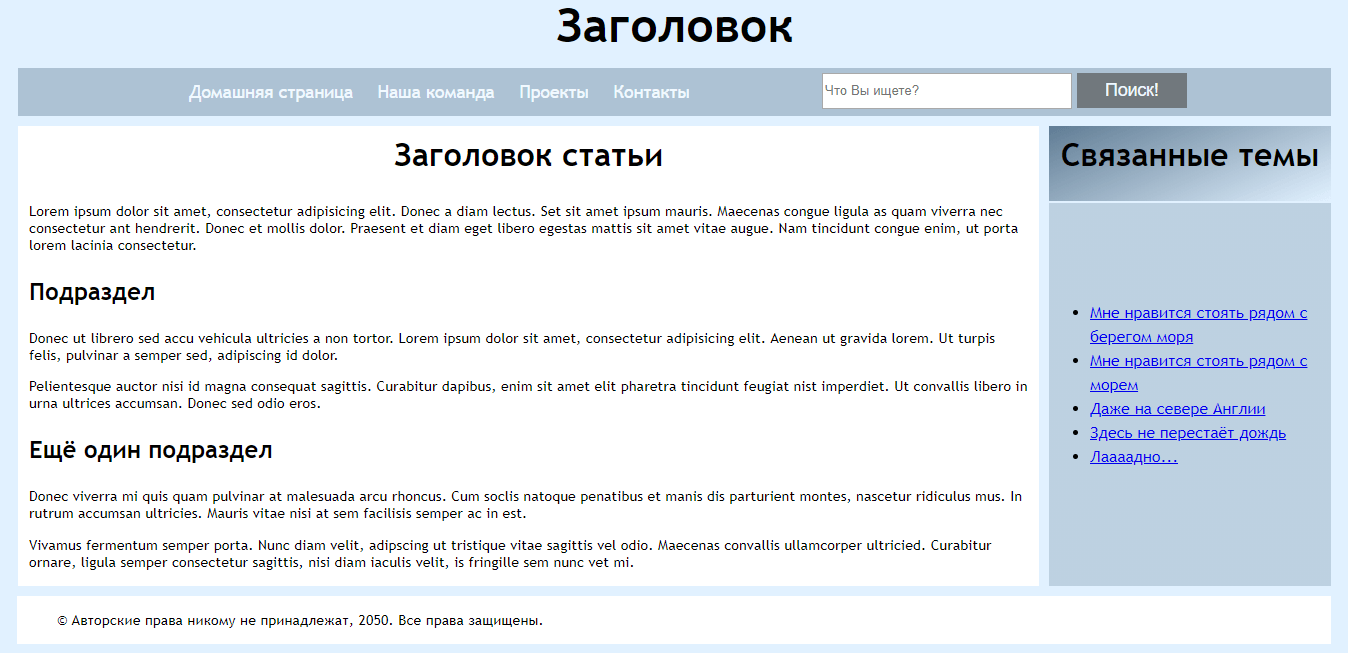
**Боковая панель**

Как правило, содержит некоторую второстепенную информацию, ссылки, цитаты, рекламу и т.д. Обычно она относится к содержимому в основном контенте (например, на странице со статьёй, боковая панель может содержать биографию автора или ссылки на связанные статьи), но в некоторых случаях здесь размещают и другие элементы, например, вторичную навигационную систему.

**Нижний колонтитул (футер)**

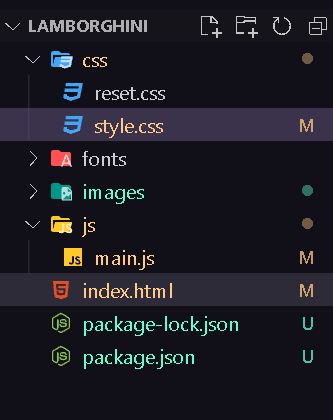
Полоса в нижней части страницы, которая обычно содержит уведомления об авторских правах или контактную информацию. Это место для размещения общей информации (например, заголовка), но обычно эта информация не является критичной или вторична для самого веб-сайта. Нижний колонтитул также иногда используется для [SEO](https://developer.mozilla.org/ru/docs/Glossary/SEO) целей, предоставляя ссылки для быстрого доступа к популярному контенту.

"Типичный веб-сайт" может быть структурирован примерно так:

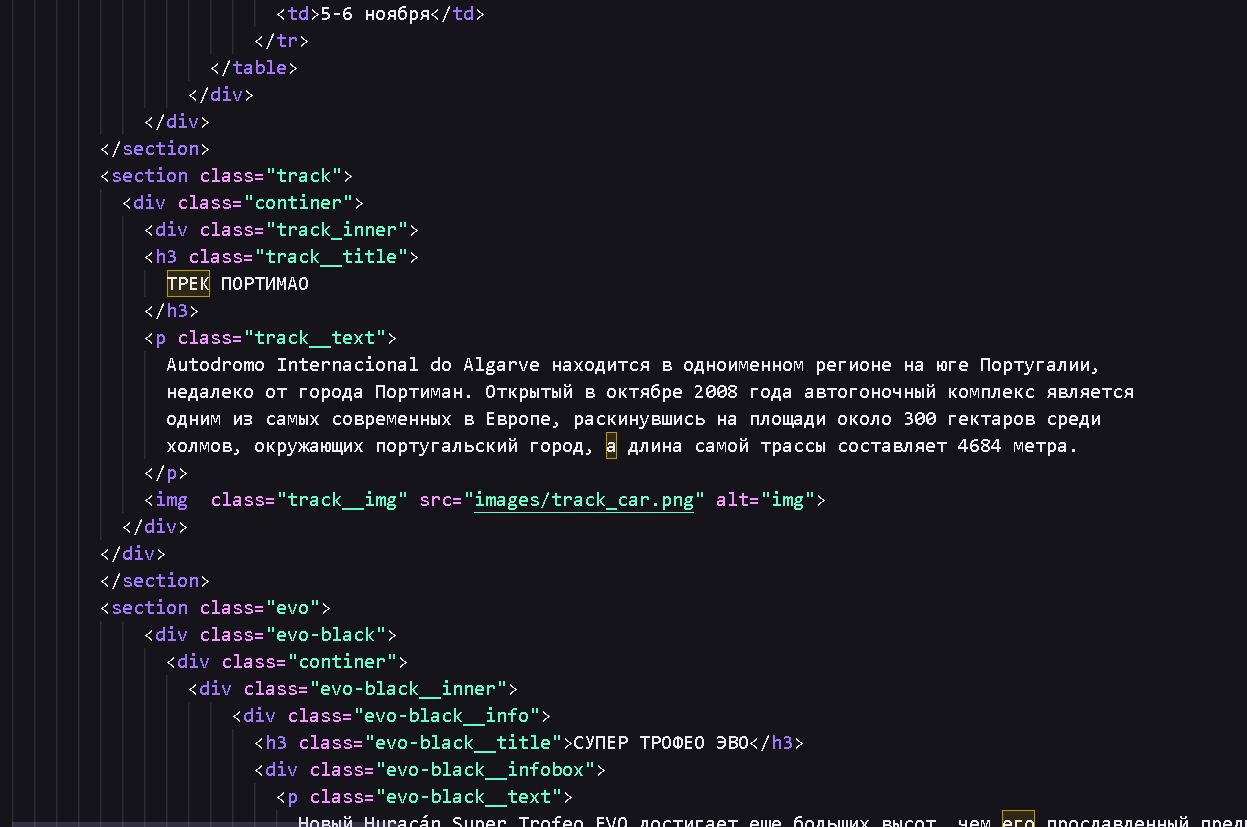


Рисунок

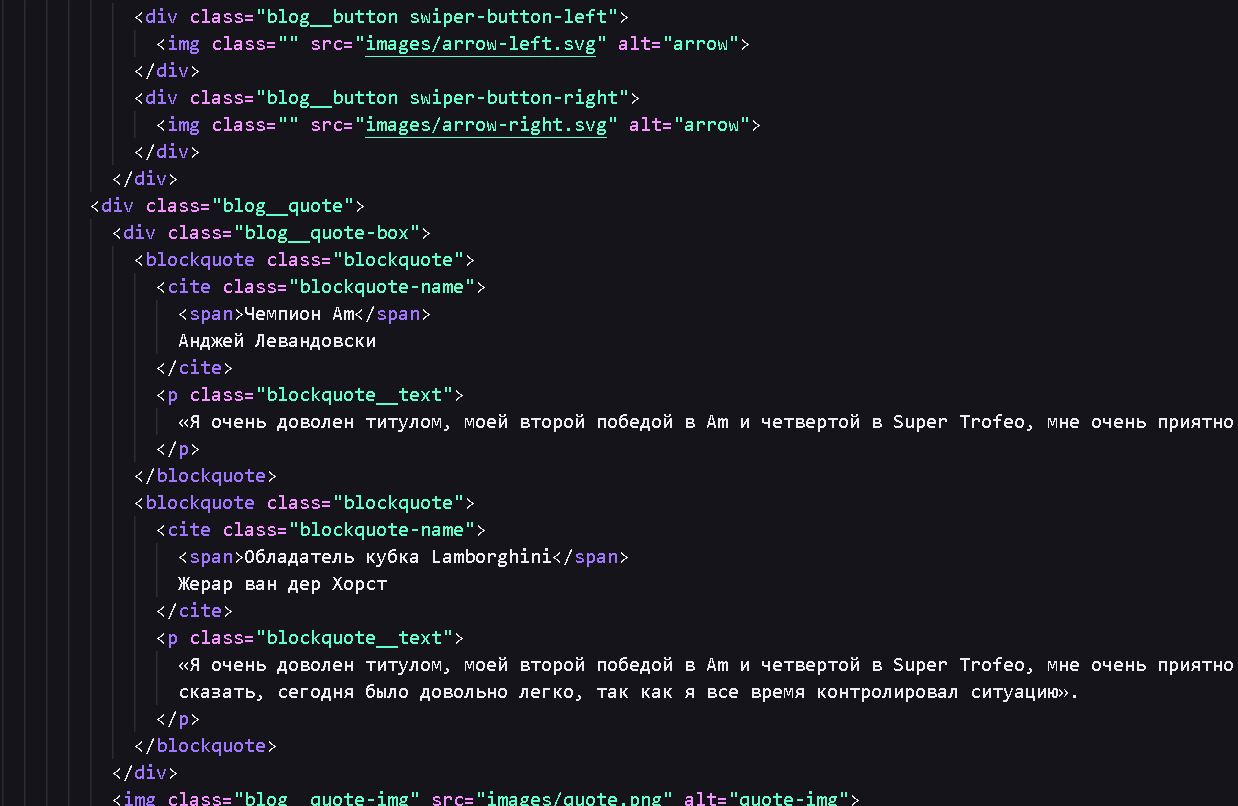
# В нашем сайте структура HTML будет гораздо сложнее, но будет интересно исходник кода будет в формате фотографии:



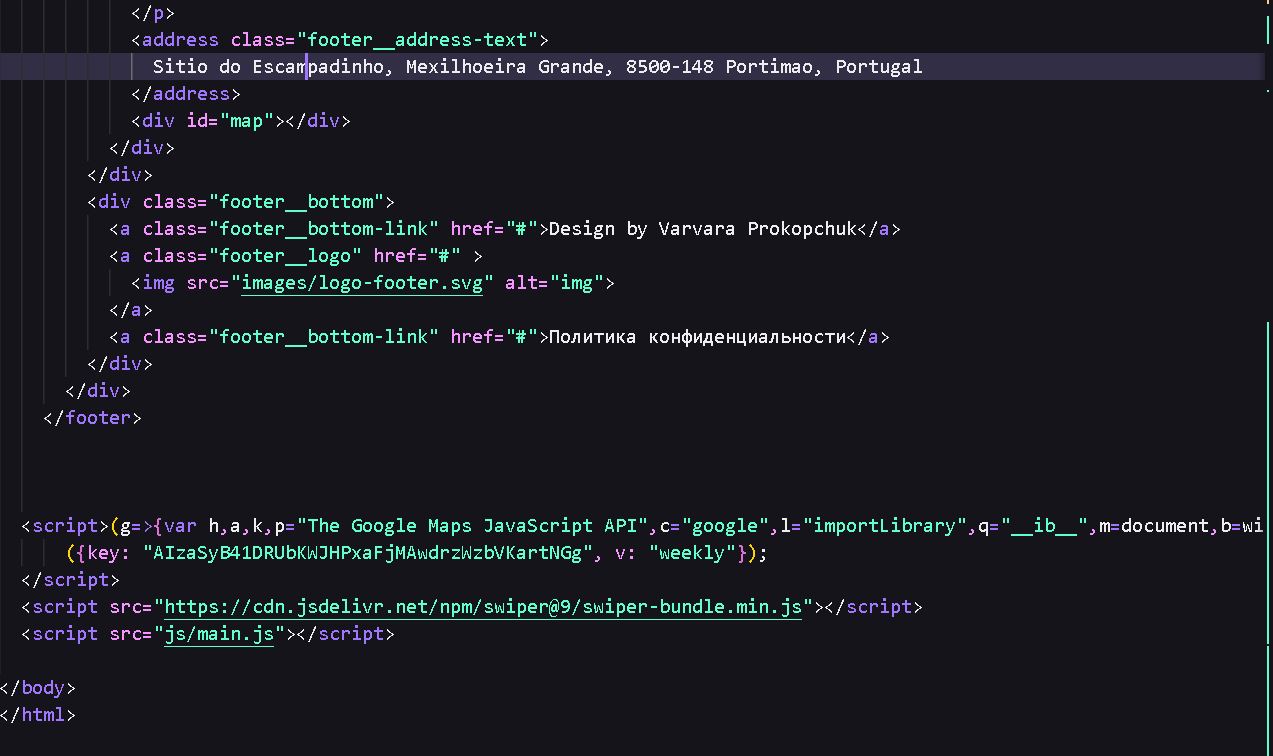
Рисунок



Рисунок



Рисунок



Рисунок

# ПРИЛОЖЕНИЕ Е

Селекторы css

**Министерство образования и науки Самарской области**

# ПРИЛОЖЕНИЕ Ж Застилизовать весь проект

**ЛИЧНАЯ КАРТОЧКА ИНСТРУКТАЖА ПО БЕЗОПАСНЫМ МЕТОДАМ РАБОТЫ И ПРОТИВОПОЖАРНОЙ БЕЗОПАСНОСТИ**

# ПРИЛОЖЕНИЕ И Что такое пользовательская часть сайта

# ПРИЛОЖЕНИЕ К Написать все javascript функции

**РЕЕСТР КОМПЬЮТЕРНОЙ ТЕХНИКИ И**

**ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**