|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н. Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н. Э. Баумана)** |

ФАКУЛЬТЕТ\_\_\_\_\_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА\_\_\_\_\_\_\_\_\_\_\_ЗАЩИТА ИНФОРМАЦИИ \_ \_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА*К КУРСОВОМУ ПРОЕКТУ***

***НА ТЕМУ:***

***“Сайт для репетитора математики.”***

Студент \_\_ИУ10-24\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_**Д.В.Серебряков**\_\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсового проекта **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_**Е.А.Ефремов**\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2023 г.*

**Оглавление**

[Введение 4](#_Toc131980651)

[1. Теоретическая часть 6](#_Toc131980652)

[1.1 Цель создания Веб-сайта 6](#_Toc131980653)

[1.2 Виды сайтов 7](#_Toc131980654)

[1.2.1 Landing Page 7](#_Toc131980655)

[1.2.1.1 Заголовок 8](#_Toc131980656)

[1.2.1.2 Картинка 8](#_Toc131980657)

[1.2.1.3 Описание предложения 8](#_Toc131980658)

[1.2.1.4 Отзывы 8](#_Toc131980659)

[1.2.1.5 Логотип 9](#_Toc131980660)

[1.2.1.6 Иконки социальных сетей в подвале 9](#_Toc131980661)

[1.2.1.7 Форма для заполнения информации 9](#_Toc131980662)

[1.2.1.8 Кнопка 9](#_Toc131980663)

[1.2.1.9 Форма после отправки заявки клиентом 9](#_Toc131980664)

[1.2.2 Типы Landing Page: 10](#_Toc131980665)

[1.2.2.1 Лидогенерационный (Lead Generation Landing Page) 10](#_Toc131980666)

[1.2.2.2 Landing Page для клика (Click-through page) 10](#_Toc131980667)

[1.2.2.3 Сжатая страница (Squeeze Page) 11](#_Toc131980668)

[1.2.2.5 Яркие страницы (Splash page) 12](#_Toc131980669)

[1.2.3 Сайт-визитка 13](#_Toc131980670)

[1.2.4 Коммерческий сайт 14](#_Toc131980671)

[1.2.5 Интернет магазин 14](#_Toc131980672)

[2. Технологии создания сайта 17](#_Toc131980673)

[2.1 Производство контента 17](#_Toc131980674)

[2.2 HTML -язык развёртки 18](#_Toc131980675)

[3. Инструменты для разработки сайта 22](#_Toc131980676)

[3.1 Онлайн-конструкторы 22](#_Toc131980677)

[3.2 VS Code 23](#_Toc131980678)

[4. Практическая часть 26](#_Toc131980679)

[4.1 Дизайн 26](#_Toc131980680)

[4.2 HTML – код 28](#_Toc131980681)

[4.2.1 Структура HTML-документа 29](#_Toc131980682)

[**4.2.1.1. Элемент <html> 31**](#_Toc131980683)

[**4.2.1.2. Элемент <head> 32**](#_Toc131980684)

[*4.2.*1.3. Элемент <body> 39](#_Toc131980685)

[4.3 CSS – код 40](#_Toc131980686)

[4.3.1 Виды таблиц стилей 41](#_Toc131980687)

[**4.3.2 Виды селекторов 43**](#_Toc131980688)

[**4.3.3 Комбинация селекторов 50**](#_Toc131980689)

[**4.3.4 Группировка селекторов 50**](#_Toc131980690)

[**4.3.5 Наследование и каскад 50**](#_Toc131980691)

[5. Заключение 54](#_Toc131980692)

[6. Список используемой литературы 56](#_Toc131980693)

## Введение

В современном мире происходит цифровизация всех сфер, в последнее время сеть Интернет стала одним из важнейших средств маркетинга, которое имеет значительные преимущества по сравнению с традиционными маркетинговыми каналами, а следовательно, поиск репетитора математики сводится к написанию запроса в поисковике или профильном сайте. В своей работе, рассмотрев множество предложений, я решил создать сайт репетитора математики. Это первый опыт написания полнофункционального сайта. По завершению проекта, мы получаем не только функциональный сайт репетитора математики, но и шаблон для последующих работ и возможности перепрофилировать сайт под другие предметы для преподавания. Однако возможности, предоставляемые сетью Интернет, настолько новы и необычны, что большинство фирм плохо представляют, как их правильно использовать. Одним из аспектов сети Интернет, требующим маркетингового подхода, является разработка Веб-страниц.

Создание Веб-сайтов является одной из важнейших технологий разработки ресурсов Интернет. Веб-сайт – это информационный ресурс, состоящий из связанных между собой гипертекстовых документов (Веб-страниц), размещенный на Веб-сервере и имеющий индивидуальный адрес. Веб-страница представляет собой текстовый файл с расширением \*.html, который содержит текстовую информацию и специальные команды – HTML-коды, определяющие в каком виде эта информация будет отображаться в окне браузера. Хороший сайт, вбирая в себя всю полезную информацию, является лучшей визитной карточкой и коммерческой фирмы, и образовательного учреждения, работая на них в любое время суток.

С точки зрения маркетинга, Веб-сайт – это набор информационных блоков и инструментов для взаимодействия с одним или несколькими сегментами рынка. Какая информация будет представлена на сайте, какие инструменты будут задействованы и как они будут взаимодействовать между собой – зависит от целей создания сайта, от краткосрочных и долгосрочных задач, а также от типа сегментов целевой аудитории и наличия возможности взаимодействовать с ней тем или иным образом. Грамотно подготовленный Веб-сайт в состоянии решать вопросы привлечения клиентов и увеличения продаж, создания имиджа и изучения потребительского спроса и т. д. Однако даже незначительные промашки при реализации сайта могут нанести существенный ущерб деятельности фирмы.

Для создания Веб-страниц нужен любой браузер для просмотра и контроля сделанной Веб-страницы, а также любой текстовый редактор для создания Веб-страниц. Чтобы создать Веб-страницу, первым делом создают структуру сайта. Сайт – это сервер, имеющий уникальное имя и постоянный адрес, физически подключенный к WWW, или виртуально размещенный на другом, более крупном сервере.

Задачи:

1. Проектирование создания Веб-сайта
2. Технологии разработки Веб-сайта
3. Инструменты для разработки Веб-сайта

Объект исследования – создание веб-сайтов.

Предмет исследования – способы и средства разработки веб-сайтов.

Методами для исследования практическое создание веб-сайтов. Информационная база взята из книжной и учебной литературы.

## Теоретическая часть

## Цель создания Веб-сайта

В основе эффективного использования сети Интернет в деятельности фирмы лежат стратегические решения компании, среди которых должно присутствовать и решение о создании корпоративного Веб-сайта. Такое решение должно базироваться на результатах бизнес-планирования и содержать оценку эффективности Веб-сайтов, а также четко сформулированные концепцию бизнеса и цели создания Веб-сайта и критерии их достижения, характеристику целевой аудитории, описание этапов реализации проекта с указанием ответственных исполнителей. Определение целей создания Веб-сайта и критериев их достижения играют основополагающее значение для определения как структуры Веб-сайта, так и необходимого программного обеспечения. Распространены следующие варианты:

1. Создание дополнительного информационно-рекламного канала, совершенствование коммуникации с действующими и потенциальными партнерами, формирования положительного имиджа компании как предприятия, использующего в своей деятельности современные технологии.

2. Использование сети Интернет для рекламы и сбыта и продаж.

3. Создание портала, обеспечивающего работу с клиентами, партнерами, поставщиками (онлайн заказы, каталог товаров и услуг, прайс-листы, информационная поддержка), подключение платежным системам, создание и развитие систем складирования и доставки.

На принятие решений по реализации Интернет-проекта существенное влияние оказывает определение целевой аудитории. Целевой сегмент может не совпадать с сегментом, на который фирма ориентируется в реальном бизнесе, поскольку сеть Интернет предоставляет гораздо более широкие возможности, например, в плане географического охвата, а может оказаться значительно уже из-за технических проблем в том или ином регионе. На этом этапе следует определить характеристики, которые описывают целевой сегмент. Кроме традиционных факторов, таких как демографические, поведенческие, психологические и другие, необходимо учитывать характеристики, связанные с особенностями компьютерного обеспечения потребителей в сети (быстродействие компьютера, объем памяти, вид модема, скорость и качество передачи данных по линии связи, качество монитор и т. д.).

Знание характеристик потенциальных потребителей позволит привлечь на Веб-страницу максимальное число посетителей. В зависимости от выбранной структуры Веб-сайта, будь то визитка или портал, меняются затраты на реализацию каждого из этапов, но общий подход к проекту остается одинаковым.

## Виды сайтов

## 1.2.1 Landing Page

Landing Page — это всегда одностраничный сайт, который призывает к какому-либо одному действию***.*** На Landing Page вы не в целом рекламируете себя, свою компанию, товары и услуги, а конкретный продукт с конкретной целью. У Landing Page есть своя структура и определенный стандартный набор элементов. Рассмотрим подробно все составляющие Landing Page, и на какие нюансы необходимо обратить внимание при разработке:



##### 

Рис.1 Landing Page

## 1.2.1.1 Заголовок

##### Заголовок должен содержать максимально сжатую информацию, которая ясно выражает суть вашего предложения. Это первое, что будет читать пользователь на вашем Landing Page. Можно назвать это рекламным слоганом. Если в заголовок не вмещается вся необходимая информация, напишите подзаголовок.

## 1.2.1.2 Картинка

Картинка стоит тысячи слов. Она нужна для привлечения внимания посетителя, для создания визуального положительного образа вашего продукта.

На этом изображении, если мы говорим о программе, должны быть наиболее удачные скриншоты, например, мобильная версия, Ipad-версия и компьютер. Этой картинкой мы говорим посетителю, что программа поддерживается на различных устройствах.

## 1.2.1.3 Описание предложения

Конечно, Landing Page должен содержать текстовую информацию о вашем предложении. Это может быть один абзац, или детальное описание, все зависит от ваших целей. Здесь главное — не увлекаться описанием товара. Ваша задача не вдаваться во все тонкости продукта, а склонить покупателя к действию различными приемами (будь то скидка, пробный продукт и т. д.)

## 1.2.1.4 Отзывы

Отзывы являются одним из обязательных элементов Landing Page. Эта информация необходима для вызова доверия у потенциального клиента. Если посетитель хочет купить ваш продукт и видит, что другие клиенты уже воспользовались им и оказались довольны — повышается уровень его доверия, и возрастает вероятность того, что он купит.

## 1.2.1.5 Логотип

Логотип вашей компании должен присутствовать на Landing Page обязательно, но он не должен “вести” на главный сайт. Здесь должна быть просто картинка, а не ссылка, это важно понимать.

## 1.2.1.6 Иконки социальных сетей в подвале

Эти иконки также не должны быть ссылками на ваши профили в социальных сетях. Это должны быть формы добавления посетителя в друзья к вам на этих ресурсах. Почему это должно быть так? Для того, чтобы человек не уходил со страницы вашего Landing Page фокусировался только на вашем конкретном продукте и предложении.

## 1.2.1.7 Форма для заполнения информации

Форма записи содержит поля для ввода информации, которую вы хотите получить от клиента, для дальнейшей ее обработки и хранения в вашей базе данных.

## 1.2.1.8 Кнопка

Это то, чем заканчивается работа в системе — посетитель по заполнении формы должен нажать на кнопку Одобрить, отправить, записаться, купить и т. д. Только после нажатия вы получите информацию, заполненную клиентом в форме.

## 1.2.1.9 Форма после отправки заявки клиентом

После завершения основных действий клиентом, он должен увидеть форму оповещения.

## 1.2.2 Типы Landing Page:

## 1.2.2.1 Лидогенерационный (Lead Generation Landing Page)

Цель данного типа Landing Page — собирать информацию о лидах — потенциальных клиентах. Например, ФИО, телефон, email — та минимально необходимая информация, по которой можно в дальнейшем связаться с лидом (позвонить, выслать письмо, исходя из контекста предложения).



Рис.2 Landing Page

## 1.2.2.2 Landing Page для клика (Click-through page)

Цель данного типа страниц — сбор кликов. Он используется для того, чтобы человек перешел на страницу покупки товаров. Вы рекламируете товар или услугу на Landing Page и у вас есть всего одна кнопка Купить или перейти на страницу товара.

Ваш Landing Page не имеет корзины и возможности оплатить, но может направить посетителя на страницу товара, где он уже может заказать его.



Рис.3 Click-through page

## 1.2.2.3 Сжатая страница (Squeeze Page)

Основная цель данного типа страниц — сбор email-адресов для будущих рассылок.

Как правило, такие страницы содержат большую картинку, заголовок, форму для сбора email и краткую информацию о тех компетенциях, которыми ваша компания может быть полезна своим потенциальным клиентам. Поэтому данный тип и называется Сжатая страница — здесь могут быть пропущены некоторые элементы Landing Page.

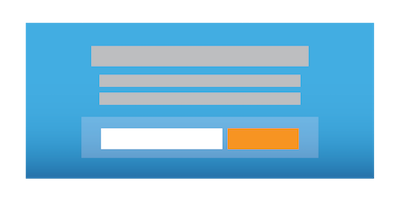
****

Рис. 4 Squeeze Page

##### **1.2.2.4 Страница продажи (Sales page)**

##### Данный тип страницы используется для продажи товаров. Чем она отличается от остальных типов? На этой странице всегда присутствует подробная информация о товаре, услуге, необходимая для принятия клиентом решения. Обыкновенно это большие страницы.



Рис. 5 Sales page

## 1.2.2.5 Яркие страницы (Splash page)

Это страницы с “яркой” подачей. Здесь упор делается на продукте, а не на заполнении формы. Если остальные страницы сосредоточены на форме (Купите, Зарегистрируйтесь и т. д.), то здесь продукт и его преимущества — в центре внимания. В данном случае важно, чтобы клиент узнал о товаре, запомнил его. Он может уйти со страницы несколькими путями, но узнав о вашем продукте, о том, какой он замечательный.

****

Рис. 6 Splash page

## 1.2.3 Сайт-визитка

Сайт-визитка это – одна или несколько страниц в интернете, расположенных на одном доменном имени, предоставляющая общую информацию о владельце, основной перечень услуг, товаров, контактные данные и раздел с ответами на основные вопросы. Главная задача такого сайта – предоставить пользователю основной перечень информации о компании или частном лице.

Стандартная структура проста и состоит всего из двух уровней:

1. Главная

2. Остальные страницы

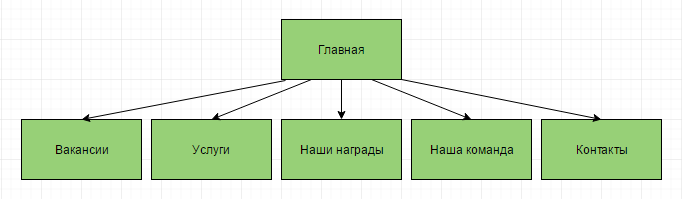


Рис. 7 Структура сайта-визитки

## 1.2.4 Коммерческий сайт

Основными страницами будут главные направления вашей деятельности, страницами второго уровня – их разновидности. Такое решение идеально для сайтов услуг и любых других коммерческих сайтов без функций интернет-магазина.

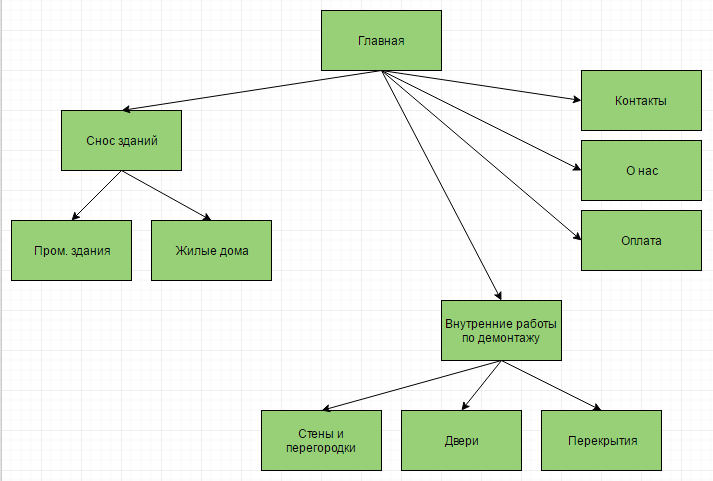


Рис. 8 Структура коммерческого сайта

## 1.2.5 Интернет магазин

Интернет-магазин позволяет перевести продажи товаров непосредственно в онлайн-пространство, и многие компании сегодня используют именно такой тип сайтов для получения прибыли через интернет.

Это уже совершенно иной уровень организации сайта. Чтобы охватить все возможные запросы, которые могут ввести в поисковой строке ваши потенциальные покупатели, вам нужно будет внедрить не только систему разделов и подразделов, но и добавить на сайт фильтры. Главное, правильно определить, что пойдет в фильтры, а что будет основным свойством продаваемого товара.

Один из ключевых моментов – грамотная структура каталога товаров. Структура сайта интернет-магазина — это определенная система взаимного расположения веб-страниц и разделов, логичный способ их связи. От того, насколько она удобна, зависит непосредственное перемещение пользователя по сайту.

Задача структуры - быстро сориентировать клиента в каталоге, чтобы он легко мог найти нужный товар, выбрать его по своим критериям, сравнить с другими. Существует два инструмента для создания структуры: категории и фильтр.

Так, например, для магазина диванов логично будет разделить диваны на кожаные и тканевые (материал), а не на синие и красные (цвет). Если фишка – это разноцветные диваны, то будет лучше разделить их по цветам, сделав материал второстепенным признаком и вынести его в фильтр. Фильтры нужны тогда, когда одному товару присуще сразу несколько качеств, которые интересуют людей в запросах, например, «цвет + форма» (кровать белая круглая) или «размер + форма + материал» (большой угловой кожаный диван).

Если не получается собственноручно создать страницы со всевозможными вариантами сочетаний параметров товаров – нужны фильтры. Если у товаров всего один-два параметра (например, если продается конкретный вид продукции - бамбуковые одеяла, и единственный параметр, который изменяется, это размер), можно обойтись без фильтров. Фильтры могут быть одинаковыми для каждого уровня и раздела каталога, могут разрабатываться отдельно – все зависит от специфики.

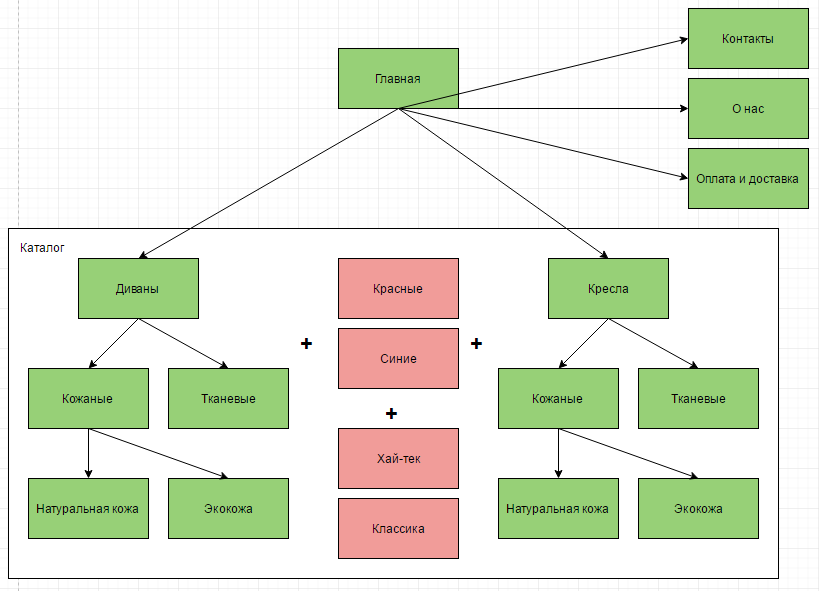


Рис. 9 Структура интернет-магазина

## 2. Технологии создания сайта

## 2.1 Производство контента

Контент – информационное наполнение сайта. Тип и формат представления информации.

Информация, представленная на Веб-странице, должна удовлетворять следующим критериям:

* должна соответствовать целям создания Веб-страницы;
* должна учитывать особенности целевого сегмента потребителей;
* должна быть в определенной степени уникальной, чтобы привлечь внимание посетителей. Тем более, что в сети Интернет существует множество похожих Веб-страниц, и конкуренция между ними достаточно сильная;
* должна быть оперативной. Для поддержания интереса к Веб-странице ее необходимо постоянно обновлять и модернизировать. Можно установить, как часто будут обновляться данные. Веб-страницы, информация на которых обновляется раз в месяц, могут рассчитывать на повторные посещения пользователей только через несколько месяцев. Если же новые сведения появляются каждый день, то повторные визиты будут происходить раз в несколько дней. Для того чтобы показать пользователям, что Веб-страница обновляется, можно указать на ней дату ее создания и последнего обновления;
* должна быть объективной и достоверной. Посетители, которые обнаружат ошибочную или неточную информацию, вряд ли захотят повторно обратиться к этой странице;
* не должна носить излишне рекламный, навязчивый характер.

На Веб-странице можно представить разнообразные материалы: информацию о фирме и ее продукции, каталог продуктов и услуг, списки дилеров, адреса торговых представителей, справочные сведения, новости, сетевой магазин, раздел обратной связи, вакансии и т. д. Следует размещать только ту информацию, которая является наиболее интересной и полезной.

Одна из основных задач, которую следует ставить перед собой при составлении текстов для Веб-страниц – это использование технической и развлекательной информации для выделения особенностей фирмы. Низкая стоимость системы WWW приводит к тому, что у любой Веб-страницы существует множество конкурентов. Поэтому тексты на Веб-странице должны четко отвечать на вопросы: чем отличается компания от всех остальных? Как можно обосновать и подчеркнуть эти отличия?

Некоторые сведения следует указывать на каждой Веб-странице. К ним относятся, в частности, электронный адрес и имя администратора сервера, а также название компании и сведения об авторском праве.

## 2.2 HTML -язык развёртки

**HTML получил широкую популярность. На данный момент – это единственный язык, с помощью которого создается разметка сайта.**

**Происхождение языка HTML начинается с GML, стандарт которого был принят в 80-х годах прошлого века с названием SGML (Standard Generalized Markup Language – стандартный обобщенный язык разметки). Предложенный в 1974 году Чарльзом Голдфарбом, SGML представляет собой метаязык – систему для описания других языков. При всех своих возможностях он слишком сложен для большинства браузеров. Только спецификация SGML занимает свыше 500 страниц. Поэтому на основе SGML был создан HTML – специализированный, гораздо более простой язык. года сообществом WHATWG ведется разработка пятой версии HTML. Теги языка HTML представляют собой служебные слова языка, заключенные в треугольные скобки. Например, многие теги используются парами. Тогда различают открывающий и закрывающий тег. Например, закрывающий тег начинается с косой черты, за которой идет название открывающего тега. Внутри пары обычно располагается информация и другие теги. Поэтому парные теги называют контейнерами.**

**Так как HTML-документ может представлять собой целый набор файлов, то лучше всего поместить их все в отдельную папку. А файл-запуск назвать стандартным именем “index.html”, известным любому браузеру. После того как документ сохранен, его можно просмотреть в браузере. Браузер «понимает» и исполняет html-код. Через меню «Вид/Источник» можно снова вернуться к Блокноту с html-кодом, внести изменения, сохранить их в файле. Если в HTML коде содержатся ошибки, то браузер просто проигнорирует их и не выдаст Вам или посетителю страницы сообщений об ошибках. Это делает процедуру отладки более трудоемкой. Любой браузер “понимает” язык тегов. Веб-страница – это еще одно название HTML-документа. Оно дается тем документам, которые размещаются в сети Интернет. Набор веб-страниц, объединенных общей темой и гиперсвязями и расположенных на одном сервере, называют сайтом. Как правило, сайт содержит гиперссылки на другие сайты. Огромное множество взаимосвязанных веб-сайтов и составляют Всемирную паутину сети Интернет. Еще одно применение HTML-документы нашли в сфере разработки электронных мультимедийных учебников и обучающих программ. Одними из самых распространенных HTML-редакторов являются: VisualStudio Code, текстовый редактор.**

**2.3 CSS – табдица стилей**

**Таблицы стилей предназначены в первую очередь для визуального оформления структурных элементов страницы. Структурные же элементы создают все те теги, которые рассматривались ранее. Правда некоторые теги (<FRON>, <B> и т. д.) из рассмотренных использовались для стилевого форматирования элементов, но в версии HTML 4.0 они считаются устаревшими и не рекомендуются для дальнейшего использования на вновь создаваемых страницах.**

**Каскадные таблицы стилей позволяют задать уже известным тегам авторский стиль оформления. Например, возможно всем заголовкам первого уровня придать зеленый цвет и больший размер или у всех элементов маркированного списка изменить маркер на красивое графическое изображение.**

**Таким образом, таблицы стилей снимают ограничения старых версий HTML на набор свойств тех или иных структурных элементов документа. Применение CSS начинается с помещения в раздел заголовка HTML-документа следующего тега. Существует несколько способов определения стиля элемента.**

****

**Способ 1. Атрибут style**

**Стиль задается непосредственно для какого-либо элемента в основном разделе документа с помощью параметра STYLE. Например:**

**<p style=”font-size: 20px; font-style: italic; color: blue”>**

**Здесь слово STYLE выступает в роли атрибута тега. Из примера видно, что значение параметра STYLE представляет собой последовательность свойств и их значений, разделенных точкой с запятой. Очередное свойство и его значение в этой последовательности разделяются двоеточием.**

**Способ 2. Таблица стилей с тегом <style>**

**Атрибут STYLE в предыдущем способе задает свойства только одного элемента, а с помощью тега <STYLE> можно задать свойства всех одноименных элементом. Контейнер <STYLE> размещается в заголовке <HEAD></HEAD>.**

**Таблица стилей может состоять из одного или нескольких правил, определяющих внешний вид описываемых элементов. Правило состоит из 2-х частей: селектора и описания.**

**Стиль для вложенных элементов P B будет использоваться браузером только тогда, когда элемент B окажется внутри элемента P. При этом свойство font-size будет унаследовано от предыдущего описания стиля тега B.**

**Последняя строка тега STYLE показывает, как можно одно описание свойства применить сразу к нескольким элементам, перечисленным через запятую.**

**Способ 3. Классы**

**Предыдущий способ позволяет присвоить любому элементу только один стиль. Иногда удобнее назначать для стилей имена и указывать их вместе с элементами. Такие универсальные стили называют классами.**

**При желании можно создать любое количество классов, каждый из которых будет иметь свой вариант форматирования для одного элемента.**

**Способ 4. Селекторы id и контекстные селекторы id**

**Этим способом можно создавать классы, связанные с нестандартными html-элементами, то есть элементами веб-разработчика.**

**С атрибутом id тесно связан структурный элемент <DIV> (от англ. Division – часть, раздел). Тег <DIV> не описывает базовых элементов html-документа, таких как заголовок, абзац, список, таблица и т. д. Он предназначен для описания какой-либо структуры, созданной самим дизайнером. Типичным примером такого элемента является навигационная панель имя такой структуре дает тоже разработчик через атрибут id. Таблица стилей используется для придания внешнего вида структурному элементу.**

## 3. Инструменты для разработки сайта

## 3.1 Онлайн-конструкторы

**Для того, чтобы создать сайт быстро и качественно, можно воспользоваться готовыми конструкторами сайтов, которые рассчитаны на работу неподготовленного человека. Таким образом, этот человек сможет освоить создание сайта совершенно бесплатно и без проблем, оформить свой сайт и в дальнейшем поддерживать его.**

**С помощью конструктора легко менять оформление сайта и наполняемую информацию. На каждом блоке располагаются кнопки для редактирования, настройки раздела, с их помощью становится возможным редактирование текста и настройка необходимых функциональных параметров.**

**Конструктор сайтов – это специализированный онлайн сервис, позволяющий создавать и объединять веб-страницы в целостную структуру – сайт, а также управлять ими, не обладая специальными техническими знаниями и навыками. Файлы сайта, созданного в конструкторе, размещены в облаке – удалённом сервере-хостинге, сохранность и работоспособность которого поддерживается командой администраторов конструктора без вмешательства пользователя. Конструкторов для создания сайтов существует огромное количество,**

* **Wix. Известный сервис, работает уже почти 10 лет. Основное достоинство сервиса – язык HTML 5, т. е. поддержка современных стандартов разметки страниц. Имеет большой выбор шаблонов, разбитых по категориям. Сайт собирается не из готовых блоков, а из элементов, каждый из которых можно настраивать. Есть и дополнительные возможности: подключение домена, поисковая оптимизация.**

**Изображение выглядит как Веб-сайт

Автоматически созданное описание**

Рис. 10 Онлайн-конструктор **Wix**

* **uKit. Хороший вариант, если нужен сайт для бизнеса. Имеется возможность подключить хостинг, купить домен привязать платёжные сервисы, добавить виджеты, подготовить мобильную версию и изучать статистику.**

Изображение выглядит как Веб-сайт

Автоматически созданное описание

Рис. 11 Онлайн-конструктор **uKit**

## VS Code

Visual Studio Code (VS Code) — текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSenseи средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией.

Имеет многоязычный интерфейс пользователя и поддерживает ряд языков программирования, подсветку синтаксиса, IntelliSense, рефакторинг, отладку, навигацию по коду, поддержку Git и другие возможности. Многие возможности Visual Studio Code недоступны через графический интерфейс, зачастую они используются через палитру команд или JSON-файлы (например, пользовательские настройки). Палитра команд представляет собой подобие командной строки, которая вызывается сочетанием клавиш.

VS Code также позволяет заменять кодовую страницу при сохранении документа, символы перевода строки и язык программирования текущего документа.

С 2018 года появилось расширение Python для Visual Studio Code с открытым исходным кодом. Оно предоставляет разработчикам широкие возможности для редактирования, отладки и тестирования кода.

Изображение выглядит как текст, монитор, черный, снимок экрана

Автоматически созданное описание

Рис. 12 Редактор Visual Studio Code

## Практическая часть

## Дизайн

Дизайн данного сайта выполнен в простом стиле, синем цвете. Синий цвет выбран из-за того, что математика ассоциируется с синим цветом, большая часть учебников по математике оформлены в синем цвете. Простой стиль даёт показать, что математики как предмет не сложная.

В верней части сайта представлен логотип в виде софы, так как сова символ мудрости, с именем и отчеством преподавателя.

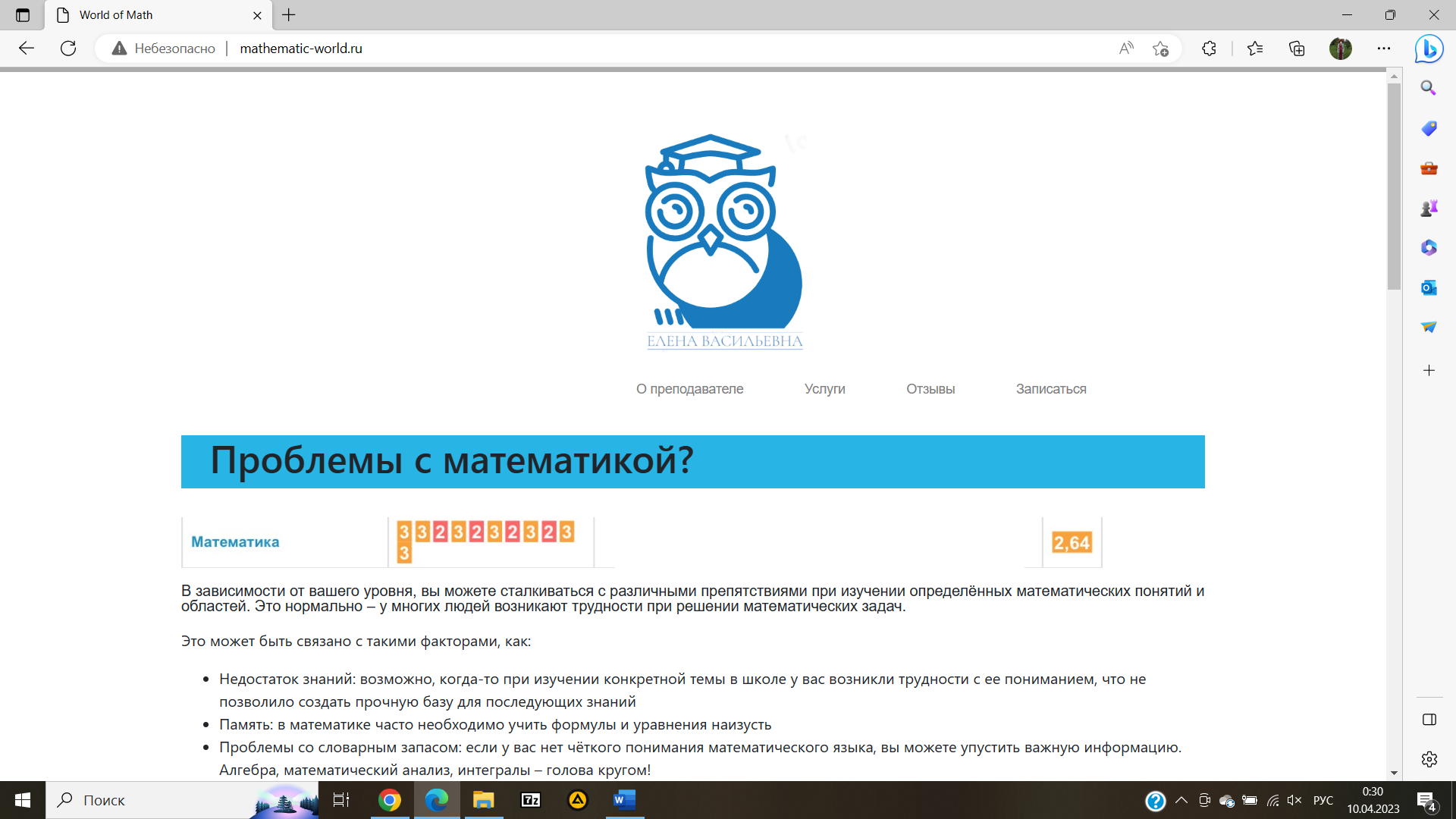


Рис. 13 Логотип сайта

Под логотипом расположено меню, которое позволяет переместиться в интересующий раздел.

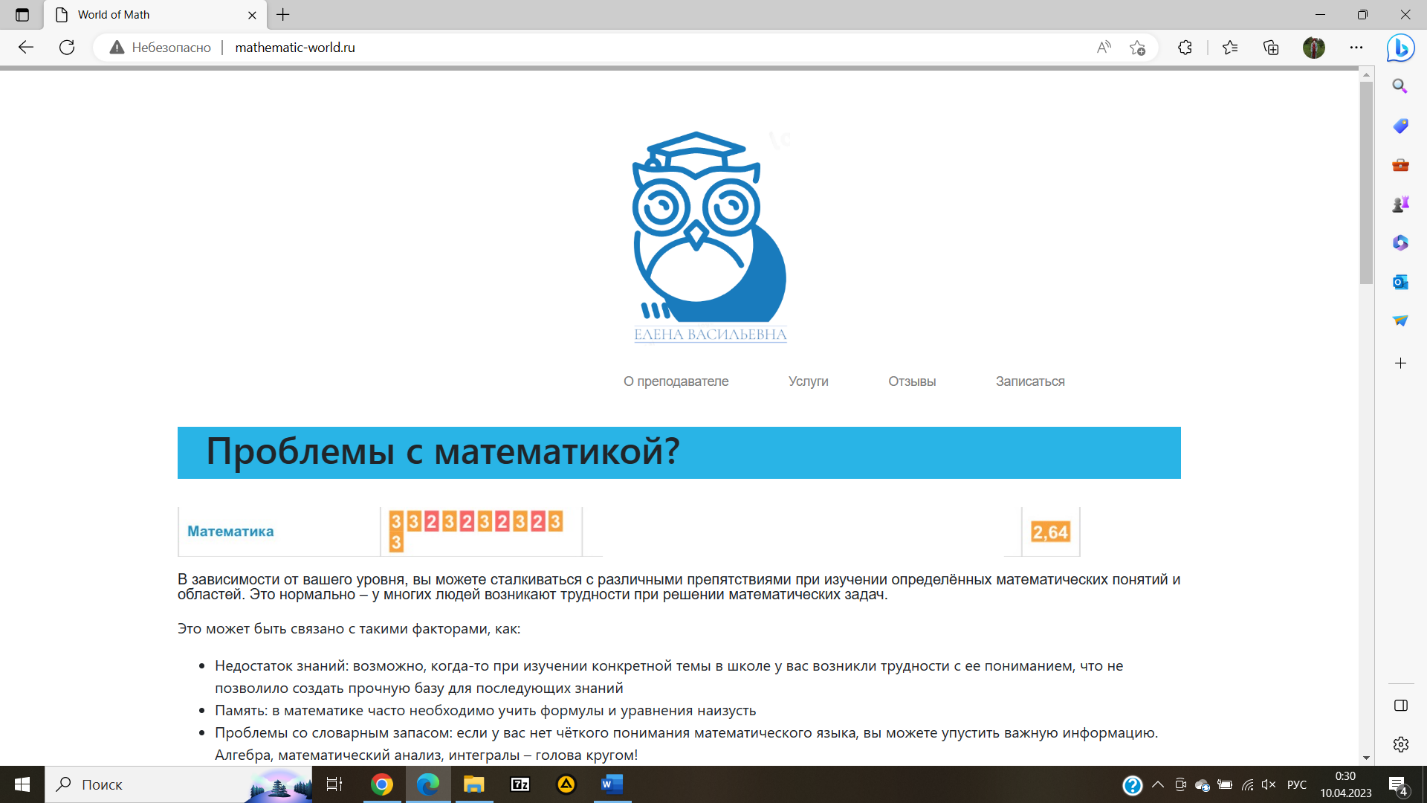


Рис. 14 Меню сайта

После меню расположена часть для привлечения клиента в её начале расположено изображение с двойками по математике, что для некоторых будет ассоциацией с собой. Далее рассказывается о важности математики в жизни и плюсы репетитора. В завершении находится изображение с пятёрками по математике, что даёт намёк, что после занятий с репетитором оценки существенно улучшаться, как и понимание предмета.

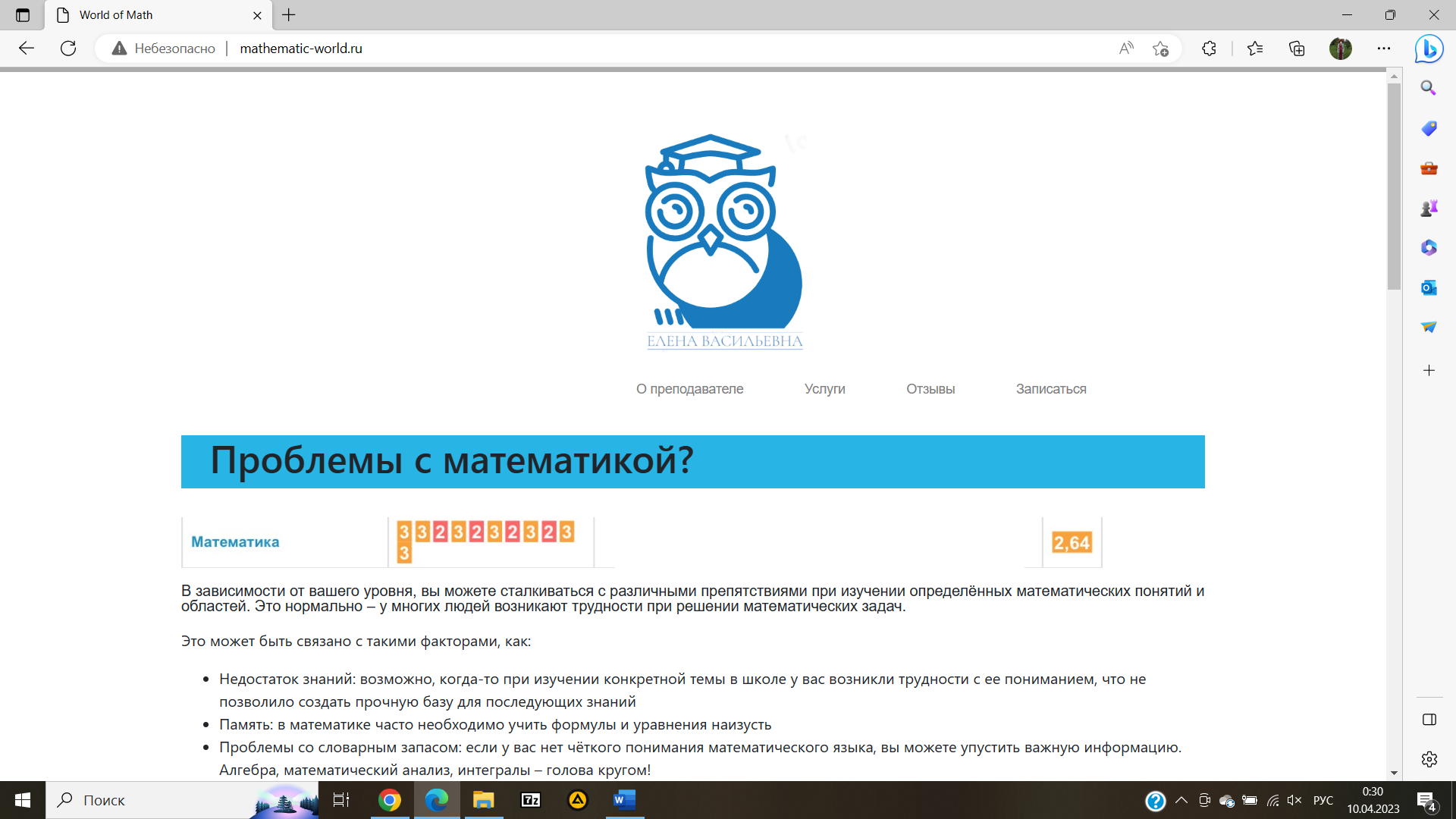


Рис. 15 Раздел привлечения аудитории

После части привлечения расположена краткая информация о преподавателе, для того чтобы познакомить посетителя сайта с педагогом.

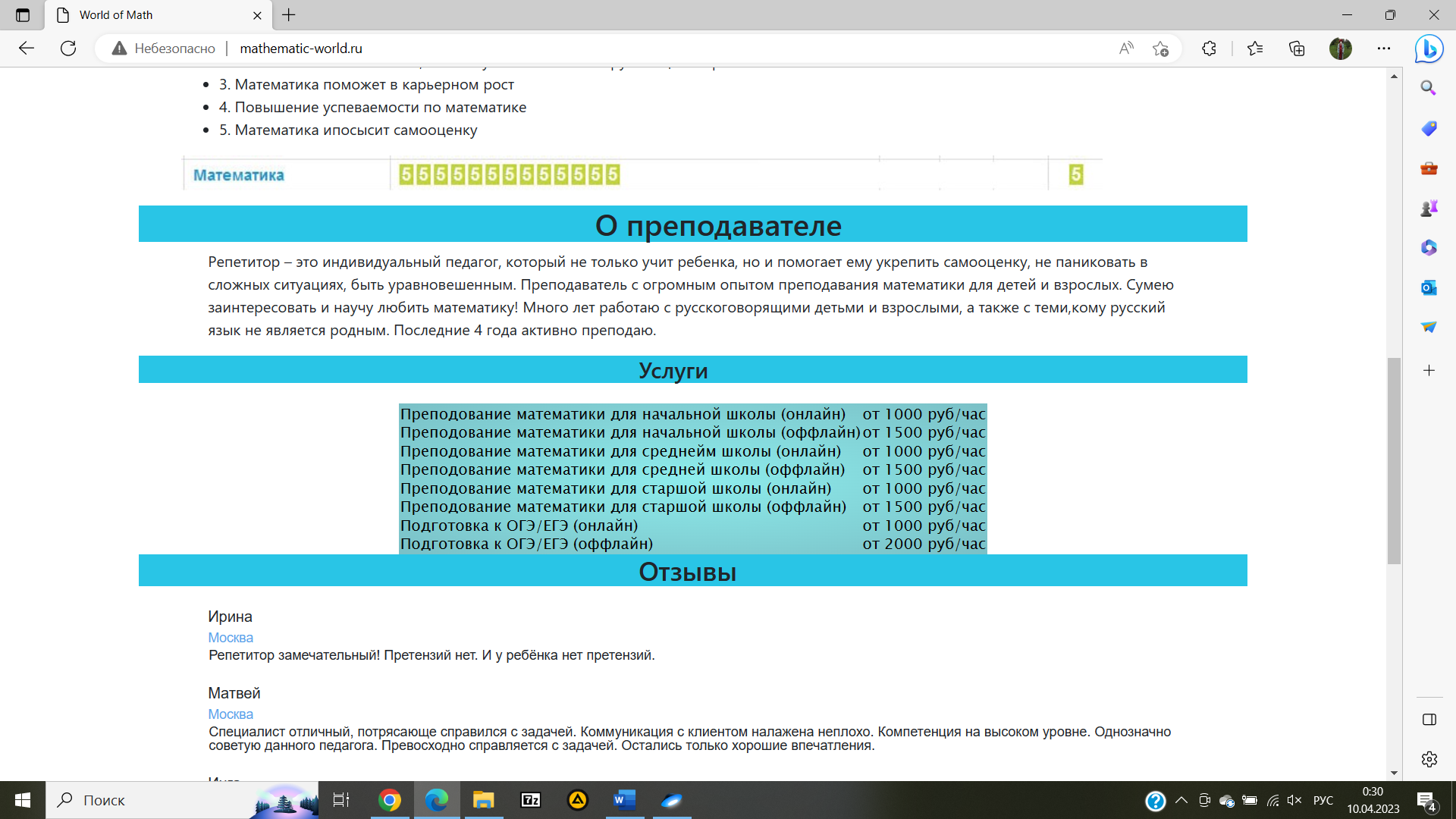


Рис. 16 Раздел информации о преподавателе

После информации о педагоге представлены цены на услуги. Изображение выглядит как текст

Автоматически созданное описание

Рис. 17 Раздел цен

Далее расположены отзывы людей, получивших уже услуги репетитора.

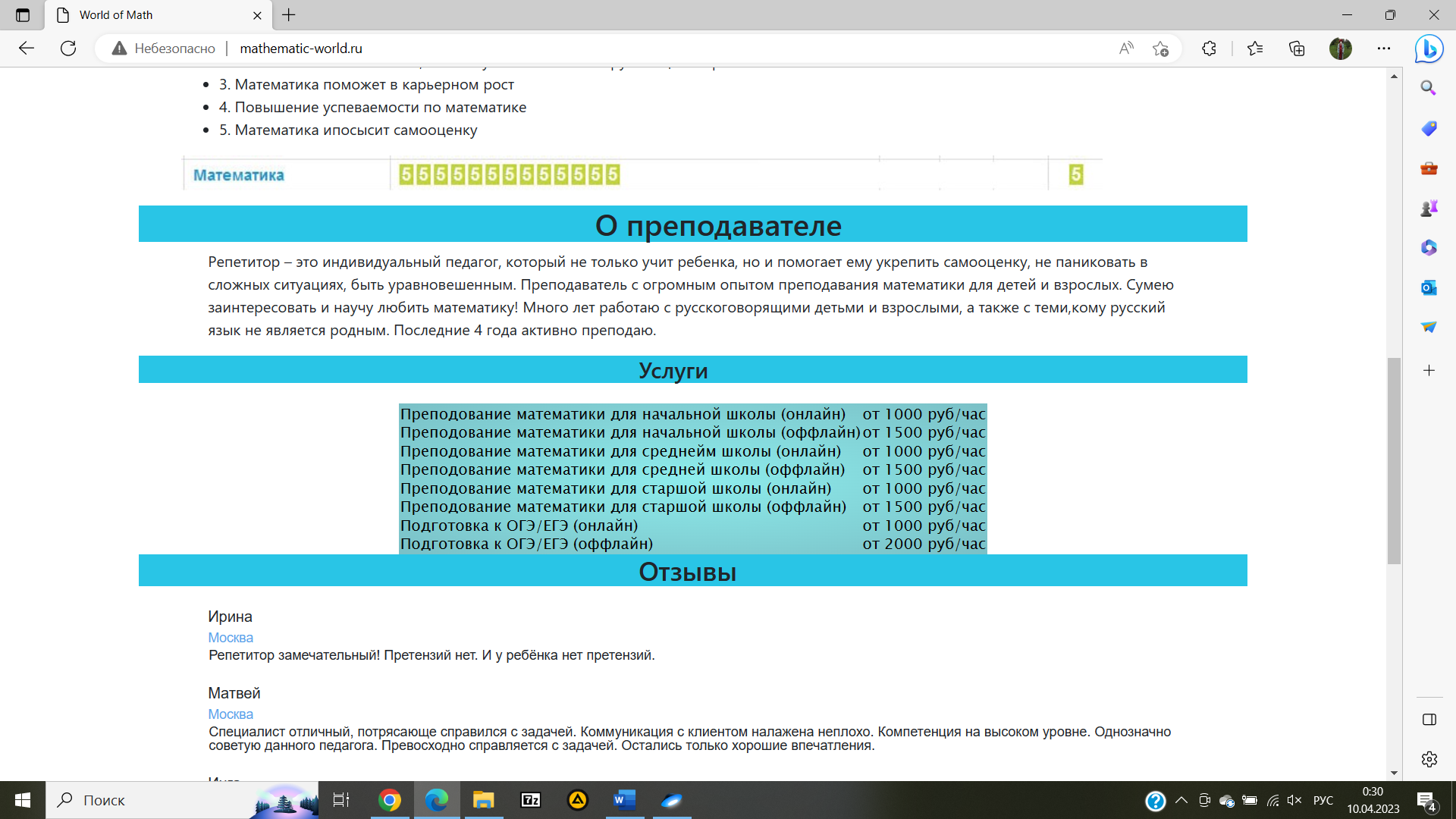


Рис. 18 Отзывы

В конце сайта расположена форма обратной записи, в которой указываются контактные данные и пожелания по уроку.

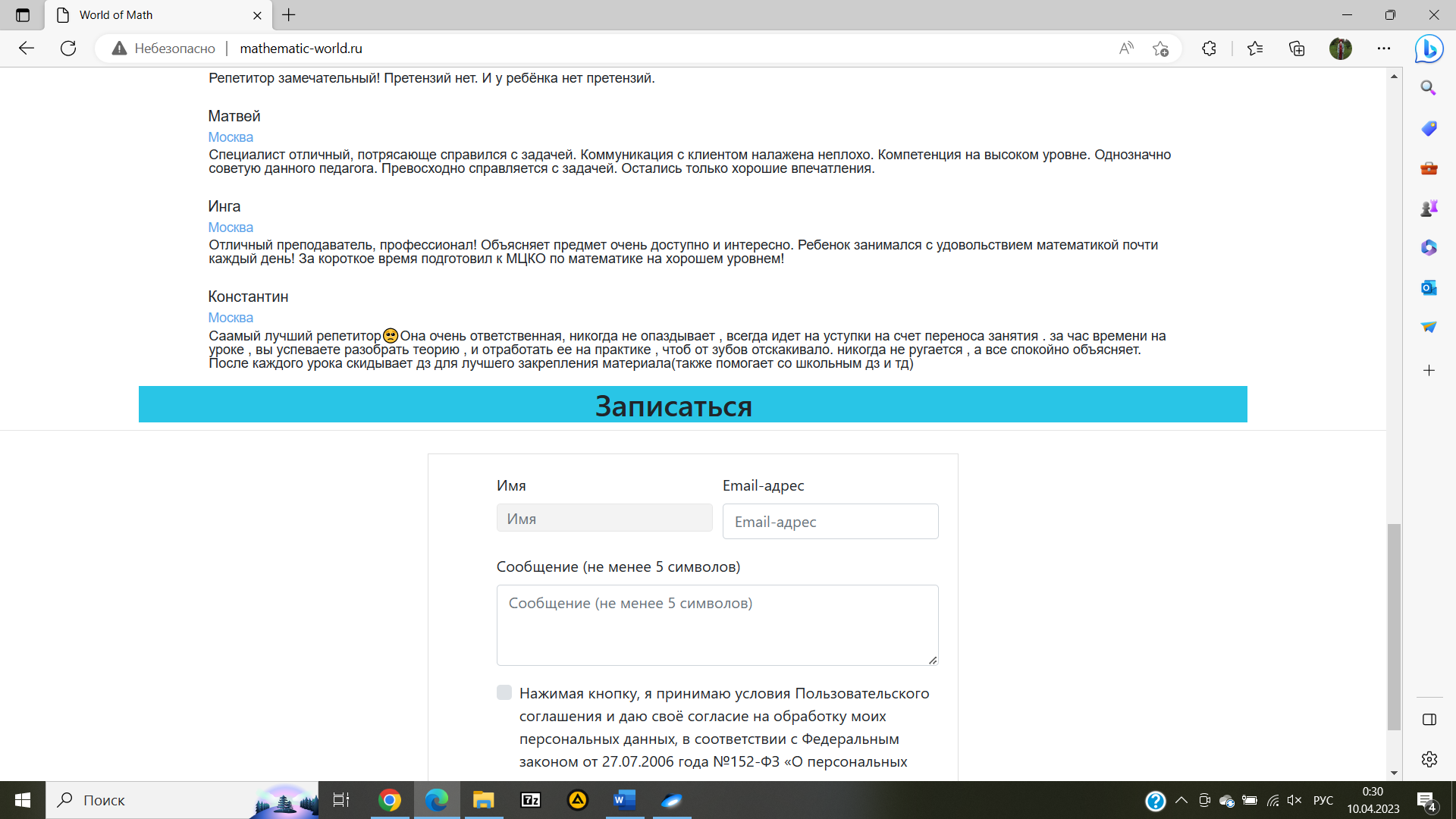


Рис. 19 Форма обратной связи

## HTML – код

**Основы HTML** содержат основные правила языка HTML, описание структуры HTML-страницы, отношения в структуре HTML-документа между HTML-элементами.

HTML-документ — это обычный текстовый документ, может быть создан как в обычном текстовом редакторе **(Блокнот)**, так и в специализированном, с подсветкой кода **(Notepad++, Visual Studio Code и т. п.)**. HTML-документ имеет расширение .html.

HTML-документ состоит из дерева HTML-элементов и текста. Каждый элемент обозначается в исходном документе начальным (открывающим) и конечным (закрывающим) тегом (за редким исключением).

**Начальный тег** показывает, где начинается элемент, конечный — где заканчивается. **Закрывающий тег** образуется путем добавления слэша / перед именем элемента: <имя элемента> … </имя элемента>. Между начальным и закрывающим тегами находится содержимое элемента — контент.

Элементы, представленные одиночными тегами, не могут хранить в себе содержимого напрямую, оно прописывается как значение атрибута, например, элемент <input type="button" value="Кнопка"> создаст кнопку с текстом **Кнопка** внутри.

Элементы могут вкладываться друг в друга, например, <p><i> Текст</i> </p>. При вложении следует соблюдать порядок их закрытия **(принцип «матрёшки»)**, например, следующая запись будет неверной: <p><i> Текст</p> </i>.

HTML-элементы могут иметь атрибуты (глобальные, применяемые для всех HTML-элементов, и собственные). Атрибуты прописываются в открывающем теге элемента и содержат имя и значение, указываемые в формате имя атрибута="значение". Атрибуты позволяют изменять свойства и поведение элемента, для которого они заданы.

Каждому элементу можно присвоить несколько значений class и только одно значение id. Множественные значения class записываются через пробел, <div class="nav top">. Значения class и id должны состоять только из букв, цифр, дефисов и нижних подчеркиваний и должны начинаться только с букв или цифр.

Браузер просматривает (интерпретирует) HTML-документ, выстраивая его структуру (DOM) и отображая ее в соответствии с инструкциями, включенными в этот файл (таблицы стилей, скрипты). Если разметка правильная, то в окне браузера будет отображена HTML-страница, содержащая HTML-элементы — заголовки, таблицы, изображения и т. д.

Процесс интерпретации **(парсинг)** начинается прежде, чем веб-страница полностью загружена в браузер. Браузеры обрабатывают HTML-документы последовательно, с самого начала, при этом обрабатывая CSS и соотнося таблицы стилей с элементами страницы.

HTML-документ состоит из двух разделов — заголовка — содержимое элемента <head> и содержательной части — содержимое <body>.

## 4.2.1 Структура HTML-документа

Язык HTML следует правилам, которые содержатся в файле объявления типа документа (Document Type Definition, или DTD). DTD представляет собой XML-документ, определяющий, какие элементы, атрибуты и их значения действительны для конкретного типа HTML. Для каждой версии HTML есть свой DTD.

**DOCTYPE** отвечает за корректное отображение веб-страницы браузером. DOCTYPE определяет не только версию HTML (например, html), но и соответствующий DTD-файл в Интернете.

Элементы, находящиеся внутри элемента <html>, образуют дерево документа, так называемую **объектную модель документа, DOM (document object model)**. При этом элемент <html> является корневым элементом.

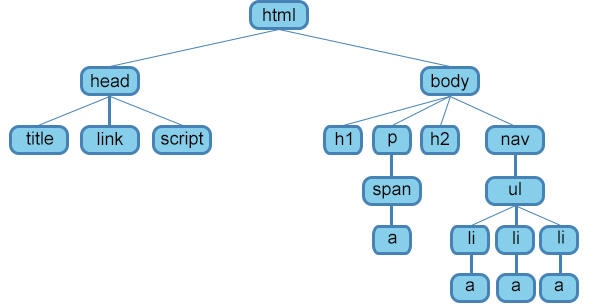


Рис. 20 Структура веб-страницы

Чтобы разобраться во взаимодействии элементов веб-страницы, необходимо рассмотреть так называемые **«родственные отношения»** между элементами. Отношения между множественными вложенными элементами подразделяются на родительские, дочерние и сестринские.

* **Предок** — элемент, который заключает в себе другие элементы. На рисунке 1 предком для всех элементов является <html>. В то же время элемент <body> является предком для всех содержащихся в нем элементов: <h1>, <p>, <span>, <nav> и т. д.
* **Потомок** — элемент, расположенный внутри одного или более типов элементов. Например, <body> является потомком <html>, а элемент <p> является потомком одновременно для <body> и <html>.
* **Родительский элемент** — элемент, связанный с другими элементами более низкого уровня, и находящийся на дереве выше их. На рисунке 1 <html> является родительским только для <head> и <body>. Элемент <p> является родительским только для <span>.
* **Дочерний элемент** — элемент, непосредственно подчиненный другому элементу более высокого уровня. На рисунке 1 только элементы <h1>, <h2>, <p> и <nav> являются дочерними по отношению к <body>.
* **Сестринский элемент** — элемент, имеющий общий родительский элемент с рассматриваемым, так называемые элементы одного уровня. На рисунке 1 <head> и <body> — элементы одного уровня, так же, как и элементы <h1>, <h2> и <p> являются между собой сестринскими.

### 4.2.1.1. Элемент <html>

Является корневым элементом документа. Все остальные элементы содержатся внутри <html>...</html>. Все, что находится за пределами элемента, не воспринимается браузером как HTML-код и никак им не обрабатывается.

### 4.2.1.2. Элемент <head>

Раздел <head>...</head> содержит техническую информацию о странице: заголовок, описание, ключевые слова для поисковых машин, кодировку и т. д. Введенная в нем информация не отображается в окне браузера, однако содержит данные, которые указывают браузеру, как следует обрабатывать страницу.

#### **1. Элемент <title>**

Обязательным элементом раздела <head> является <title>. Текст, размещенный внутри элемента <title>, отображается в строке заголовка веб-браузера.

Длина заголовка должна быть не более 60 символов, чтобы полностью поместиться в заголовке. Текст заголовка должен содержать максимально полное описание содержимого веб-страницы.**2. Элемент <meta>**

Необязательным элементом раздела <head> является элемент <meta>. С его помощью можно задать описание содержимого страницы и ключевые слова для поисковых машин, автора HTML-документа и прочие свойства метаданных.

Элемент <head> может содержать несколько элементов <meta>, потому что в зависимости от используемых атрибутов они несут различную информацию.



Описание содержимого страницы и ключевые слова одновременно можно указывать на нескольких языках, например, на русском и английском:



#### **3. Элемент <style>**

Внутри этого элемента задаются стили, которые используются на странице. Для задания стилей в HTML-документе используется язык CSS. Таких элементов на странице может быть несколько.

|  |  |
| --- | --- |
| **Атрибут** | **Описание, принимаемое значение** |
| media | Определяет, для какого типа устройства предназначены те или иные каскадные таблицы стилей. Внутри таблицы стилей CSS специфичные правила для определенных устройств прописываются с помощью media query. |
| nonce | Генерируемая случайным образом на сервере строковая переменная, которая устанавливает правила использования встроенных стилей с целью защиты контента. Значение атрибута — строка текста. |
| type | Определяет MIME-тип подключаемого файла со стилями, в данном случае type="text/css". |

Таблица 1 Атрибуты элемента <style>

Элемент может содержать код форматирования как самих элементов веб-страницы, так и веб-страницы целиком.



Чтобы подключить к элементу заданный стиль, необходимо через атрибут class (или id) присвоить элементу соответствующее название:



CSS-код можно встраивать непосредственно в элемент разметки в виде значение атрибута style, например:



#### **4. Элемент <link>**

Задать стили для документа можно также при помощи другого способа — записать их в отдельный файл с расширением .css, например, style.css.

Подключить файл со стилями к веб-странице можно двумя способами:

* через директиву @import url
* с использованием элемента <link>. Данный способ является предпочтительным.



Элемент <link> определяет отношение между текущей страницей и другими документами. Таких элементов на странице может быть несколько.



|  |  |
| --- | --- |
| **Атрибут** | **Описание, принимаемое значение** |
| crossorigin | Указывает, должен ли использоваться CORS (технология браузеров, которая позволяет предоставить веб-странице доступ к ресурсам другого домена) при извлечении изображения с сайта. anonymous — в кросс-доменный запрос браузер автоматически добавляет заголовок Origin, содержащий имя домена, с которого осуществлён запрос. Если сервер не ответит CORS-заголовком Access-Control-Allow-Origin: \* (или имя домена вместо звёздочки), то загрузка изображения будет заблокирована. use-credentials — если сервер не предоставляет учётные данные с помощью Access-Control-Allow-Credentials: true, то загрузка изображения будет заблокирована. |
| href | Основной атрибут элемента, в качестве значения выступает путь к файлу со стилями. |
| hreflang | Определяет язык текста в документе, на который идет ссылка. |
| media | Определяет тип устройства, к которым должен быть применен ресурс ссылки. |
| nonce | Генерируемая случайным образом на сервере строковая переменная, которая устанавливает правила использования встроенных стилей с целью защиты контента. Значение атрибута — строка текста. |
| rel | Атрибут определяет отношения между текущим документом и документом, на который идет ссылка. alternate — ссылка на тот же документ, но в другом формате (например, страницы для печати, перевод, зеркало, лента в формате RSS или Atom), <link rel="alternate stylesheet" type="text/css" title="small\_font" href="small-font.css">. <link rel="alternate" type="application/rss+xml" title="my\_RSS" href="index.xml"> <link rel="alternate" type="application/atom+xml" title="News" href="/atom.xml?type=news"> archives — указывает на то, что документ по ссылке представляет исторический интерес. Ссылка может указывать на коллекцию записей, документов и других материалов. author ссылка на страницу об авторе документа или на страницу с контактными данными автора. bookmark ссылка на ближайшего предка статьи, являющегося связующим звеном, или на раздел статьи, наиболее тесно связанных с элементом, если нет предка. external используется для указания того, что страница на которую ведет ссылка не являются частью данного сайта. first указывает ссылку, ведущую на первый документ из последовательности документов. help ссылка на документ со справкой. icon определяет путь к иконке, которая будет использована для текущего документа. last указывает ссылку, ведущую на последний документ в последовательности документов. license ссылка на сведения об авторских правах для документа. next указывает, что этот документ является частью серии, и что ссылка ведет на следующий документ в этой серии. <link rel="next" href="/next.html" type="text/html" title="Следующая страница"> nofollow указывает на то, что ссылка не одобрена автором страницы или что ссылка носит коммерческий характер. noreferrer указывает на то, что заголовок запроса клиента, содержащий url источника запроса, не должен передаваться при переходе по ссылке. pingback указывает адрес пингбэк-сервера, что дает возможность для блога автоматически оповещать сайты, ссылающиеся на него. prefetch указывает, что следует заранее кэшировать файл, на который ведет ссылка. prev указывает, что этот документ является частью серии, и что ссылка ведет на предыдущий документ в этой серии. <link rel="prev" href="/next.html" type="text/html" title="Предыдущая страница"> search указывает, что ссылаемый документ содержит интерфейс поиска и связанных с ним ресурсов. sidebar указывает, что ссылаемый документ, если это возможно, будет показан в дополнительном контексте браузера, и некоторые браузеры при щелчке по гиперссылке открывают окно для добавления ссылки в панель закладок. stylesheet ссылка на внешний файл, который будет использоваться в качестве таблицы стилей для данного документа. tag указывает на то, что метка, на которую ведет гиперссылка, относится к данному документу. up указывает, что страница является частью иерархической структуры, и что гиперссылка ведет на более высокий уровень ресурса в структуре. |
| sizes | Указывает размер иконок для визуального отображения. Атрибут sizes используется только совместно с rel="icon", и может принимать следующий значения: ширинахвысота — определяет список размеров, разделенных пробелами, каждый размер должен быть в формате — ширинахвысота (размеры иконки задаются в пикселях), например: <link rel="icon" href="favicon.png" sizes="16x16 32х32" type="image/png">; any — иконка может масштабироваться до любого размера. |
| title | Определяет заголовок ссылки или название набора альтернативных таблиц стилей. Значение атрибута — текст. |
| type | Определяет MIME-тип документа, на который идет ссылка. В данном случае он принимает значение "text/css". |

#### Таблица 2 Атрибуты элемента <link>

#### **5. Элемент <script>**

Элемент <script> позволяет присоединять к документу различные сценарии. Текст сценария может располагаться либо внутри этого элемента, либо во внешнем файле. Если текст сценария расположен во внешнем файле, то он подключается с помощью атрибутов элемента.

## *4.2.*1.3. Элемент <body>

Элемент <body> предназначен для хранения содержимого веб-страницы (контента), отображаемого в окне браузера. Информацию, которую следует выводить в документе, следует располагать именно внутри контейнера <body>. К такой информации относится текст, изображения, теги, скрипты JavaScript и т. д.

<body> также применяется для определения цветов ссылок и текста на веб-странице. Подобная практика в HTML осуждается и взамен для указания цветовой схемы рекомендуется использовать стили, применяя их к селектору body.

Часто <body> используется для размещения обработчика событий, например, [onload](https://webref.ru/html/attr/onload), которое выполняется после того, как документ завершил загрузку в текущее окно или фрейм.

Открывающий и закрывающий теги <body> на веб-странице не являются обязательными, однако хорошим стилем считается их использование, чтобы определить начало и конец HTML-документа.

## CSS – код

**CSS (Cascading Style Sheets)** — язык таблиц стилей, который позволяет прикреплять стиль (например, шрифты и цвет) к структурированным документам (например, документам HTML и приложениям XML).

Обычно CSS-стили используются для создания и изменения стиля элементов веб-страниц и пользовательских интерфейсов, написанных на языках HTML и XHTML, но также могут быть применены к любому виду XML-документа, в том числе XML, SVG и XUL.

Отделяя стиль представления документов от содержимого документов, CSS упрощает создание веб-страниц и обслуживание сайтов.

CSS поддерживает таблицы стилей для конкретных носителей, поэтому авторы могут адаптировать представление своих документов к визуальным браузерам, слуховым устройствам, принтерам, брайлевским устройствам, карманным устройствам и т. д.

Каскадные таблицы стилей описывают правила форматирования элементов с помощью свойств и допустимых значений этих свойств. Для каждого элемента можно использовать ограниченный набор свойств, остальные свойства не будут оказывать на него никакого влияния.

Объявление стиля состоит из двух частей: **селектора** и **объявления**. В HTML имена элементов нечувствительны к регистру, поэтому «h1» работает так же, как и «H1». Объявление состоит из двух частей: имя свойства (например, color) и значение свойства (grey). Селектор сообщает браузеру, какой именно элемент форматировать, а в блоке объявления (код в фигурных скобках) перечисляются форматирующие команды — свойства и их значения.

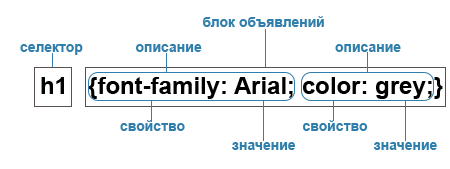


Рис. 21 Структура объявления стилей

## 4.3.1 Виды таблиц стилей

#### **1. Внешняя таблица стилей**

**Внешняя таблица стилей** представляет собой текстовый файл с расширением .css, в котором находится набор CSS-стилей элементов. Файл создаётся в редакторе кода, так же, как и HTML-страница. Внутри файла могут содержаться только стили, без HTML-разметки. Внешняя таблица стилей подключается к веб-странице с помощью элемента <link>, расположенного внутри раздела <head></head>. Такие стили работают для всех страниц сайта.

К каждой веб-странице можно присоединить несколько таблиц стилей, добавляя последовательно несколько элементов <link>, указав в атрибуте media назначение данной таблицы стилей. rel="stylesheet" указывает тип ссылки (ссылка на таблицу стилей).



Атрибут type="text/css" не является обязательным по стандарту HTML5, поэтому его можно не указывать. Если атрибут отсутствует, по умолчанию используется значение type="text/css".

#### **2. Внутренние стили**

**Внутренние стили** встраиваются в раздел <head></head> HTML-документа и определяются внутри элемента <style></style>. Внутренние стили имеют приоритет над внешними, но уступают встроенным стилям (заданным через атрибут style).



#### **3. Встроенные стили**

Когда мы пишем **встроенные стили**, мы пишем CSS-код в HTML-файл, непосредственно внутри элемента с помощью атрибута style:



Такие стили действуют только на тот элемент, для которого они заданы.

#### **4. Правило @import**

**Правило @import** позволяет загружать внешние таблицы стилей. Чтобы директива @import работала, она должна располагаться в таблице стилей (внешней или внутренней) перед всеми остальными правилами:



Правило @import также используется для подключения веб-шрифтов:



### 4.3.2 Виды селекторов

**Селекторы** представляют структуру веб-страницы. С их помощью создаются правила для форматирования элементов веб-страницы. Селекторами могут быть элементы, их классы и идентификаторы, а также псевдоклассы и псевдоэлементы.

#### **1. Универсальный селектор**

Соответствует любому HTML-элементу. Например, \* {margin: 0;} обнулит внешние отступы для всех элементов сайта. Также селектор может использоваться в комбинации с псевдоклассом или псевдоэлементом: \*: after {CSS-стили}, \*: checked {CSS-стили}.

#### **2. Селектор элемента**

Селекторы элементов позволяют форматировать все элементы данного типа на всех страницах сайта. Например, h1 {font-family: Lobster, cursive;} задаст общий стиль форматирования всех заголовков h1.

#### **3. Селектор класса**

Селекторы класса позволяют задавать стили для одного и более элементов с одинаковым именем класса, размещенных в разных местах страницы или на разных страницах сайта. Например, для создания заголовка с классом headline необходимо добавить атрибут class со значением headline в открывающий тег <h1> и задать стиль для указанного класса. Стили, созданные с помощью класса, можно применять к другим элементам, не обязательно данного типа.

Html:



CSS:



Если элемент имеет несколько атрибутов класса, их значения объединяются с пробелами.



#### **4. Селектор идентификатора**

Селектор идентификатора позволяет форматировать **один** конкретный элемент. Значение id должно быть уникальным, на одной странице может встречаться только один раз и должно содержать хотя бы один символ. Значение не должно содержать пробелов.

Нет никаких других ограничений на то, какую форму может принимать id, в частности, идентификаторы могут состоять только из цифр, начинаться с цифры, начинаться с подчеркивания, состоять только из знаков препинания и т. д.

Уникальный идентификатор элемента может использоваться для различных целей, в частности, как способ ссылки на конкретные части документа с использованием идентификаторов фрагментов, как способ нацеливания на элемент при создании сценариев и как способ стилизации конкретного элемента из CSS.

Html:

 CSS:



#### **5. Селектор потомка**

Селекторы потомков применяют стили к элементам, расположенным внутри элемента-контейнера. Например, ul li {text-transform: uppercase;} — выберет все элементы li, являющиеся потомками всех элементов ul.

Если нужно отформатировать потомки определенного элемента, этому элементу нужно задать стилевой класс:

* p. first a {color: green;} — данный стиль применится ко всем ссылкам, потомкам абзаца с классом first;
* p. first a {color: green;} — если добавить пробел, то будут стилизованы ссылки, расположенные внутри любого элемента класса. first, который является потомком элемента <p>;
* . first a {color: green;} — данный стиль применится к любой ссылке, расположенной внутри другого элемента, обозначенного классом. first.

#### **6. Дочерний селектор**

Дочерний элемент является прямым потомком содержащего его элемента. У одного элемента, может быть, несколько дочерних элементов, а родительский элемент у каждого элемента может быть только один. Дочерний селектор позволяет применить стили только если дочерний элемент идёт сразу за родительским элементом и между ними нет других элементов, то есть дочерний элемент больше ни во что не вложен.

Например, p> strong — выберет все элементы strong, являющиеся дочерними по отношению к элементу p.

#### **7. Сестринский селектор**

Сестринские отношения возникают между элементами, имеющими общего родителя. Селекторы сестринских элементов позволяют выбрать элементы из группы элементов одного уровня:

* h1 + p — выберет все первые абзацы, идущие непосредственно за любым элементом <h1>, не затрагивая остальные абзацы;
* h1 ~ p — выберет все абзацы, являющиеся сестринскими по отношению к любому заголовку h1 и идущие сразу после него.

#### **8. Селектор атрибута**

Селекторы атрибутов выбирают элементы на основе имени атрибута или значения атрибута:

* [атрибут] — все элементы, содержащие указанный атрибут, [alt] — все элементы, для которых задан атрибут alt;
* селектор[атрибут] — элементы данного типа, содержащие указанный атрибут, img[alt] — только картинки, для которых задан атрибут alt;
* селектор[атрибут="значение"] — элементы данного типа, содержащие указанный атрибут с конкретным значением, img[title="flower"] — все картинки, название которых содержит слово flower;
* селектор[атрибут~="значение"] — элементы, частично содержащие данное значение, например, если для элемента задано несколько классов через пробел, p[class~="feature"] — абзацы, имя класса которых содержит feature;
* селектор[атрибут|="значение"] — элементы, список значений атрибута которых начинается с указанного слова, p[class|="feature"] — абзацы, имя класса которых feature или начинается на feature;
* селектор[атрибут^="значение"] — элементы, значение атрибута которых начинается с указанного значения, a[href^="http://"] — все ссылки, начинающиеся на http://;
* селектор[атрибут$="значение"] — элементы, значение атрибута которых заканчивается указанным значением, img[src$=".png"] — все картинки в формате png;
* селектор[атрибут\*="значение"] — элементы, значение атрибута которых содержит в любом месте указанное слово, a[href\*="book"] — все ссылки, название которых содержит book.

#### **9. Селектор псевдокласса**

Псевдоклассы — это классы, фактически не прикрепленные к HTML-элементам. Они позволяют применить CSS-правила к элементам при совершении события или подчиняющимся определенному правилу.

Псевдоклассы характеризуют элементы со следующими свойствами:

* : link — не посещенная ссылка;
* : visited — посещенная ссылка;
* : hover — любой элемент, по которому проводят курсором мыши;
* : focus — интерактивный элемент, к которому перешли с помощью клавиатуры или активировали посредством мыши;
* : active — элемент, который был активизирован пользователем;
* : valid — поля формы, содержимое которых прошло проверку в браузере на соответствие указанному типу данных;
* : invalid — поля формы, содержимое которых не соответствует указанному типу данных;
* : enabled — все активные поля форм;
* : disabled — заблокированные поля форм, т. е., находящиеся в неактивном состоянии;
* :in-range — поля формы, значения которых находятся в заданном диапазоне;
* :out-of-range — поля формы, значения которых не входят в установленный диапазон;
* : lang () — элементы с текстом на указанном языке;
* : not(селектор) — элементы, которые не содержат указанный селектор — класс, идентификатор, название или тип поля формы —: not([type="submit"]);
* : target — элемент с символом #, на который ссылаются в документе;
* : checked — выделенные (выбранные пользователем) элементы формы.

#### **10. Селектор структурных псевдоклассов**

Структурные псевдоклассы отбирают дочерние элементы в соответствии с параметром, указанным в круглых скобках:

* : nth-child(odd) — нечётные дочерние элементы;
* : nth-child(even) — чётные дочерние элементы;
* : nth-child(3n) — каждый третий элемент среди дочерних;
* : nth-child(3n+2) — выбирает каждый третий элемент, начиная со второго дочернего элемента (+2);
* : nth-child(n+2) — выбирает все элементы, начиная со второго;
* : nth-child (3) — выбирает третий дочерний элемент;
* : nth-last-child () — в списке дочерних элементов выбирает элемент с указанным местоположением, аналогично с: nth-child (), но начиная с последнего, в обратную сторону;
* : first-child — позволяет оформить только самый первый дочерний элемент;
* : last-child — позволяет форматировать последний дочерний элемент;
* : only-child — выбирает элемент, являющийся единственным дочерним элементом;
* : empty — выбирает элементы, у которых нет дочерних элементов;
* : root — выбирает элемент, являющийся корневым в документе — элемент html.

#### **11. Селектор структурных псевдоклассов типа**

Указывают на конкретный тип дочернего элемента:

* : nth-of-type () — выбирает элементы по аналогии с: nth-child (), при этом берёт во внимание только тип элемента;
* : first-of-type — выбирает первый дочерний элемент данного типа;
* : last-of-type — выбирает последний элемент данного типа;
* : nth-last-of-type () — выбирает элемент заданного типа в списке элементов в соответствии с указанным местоположением, начиная с конца;
* : only-of-type — выбирает единственный элемент указанного типа среди дочерних элементов родительского элемента.

#### **12. Селектор псевдоэлемента**

Псевдоэлементы используются для добавления содержимого, которое генерируется с помощью свойства content:

* : first-letter — выбирает первую букву каждого абзаца, применяется только к блочным элементам;
* : first-line — выбирает первую строку текста элемента, применяется только к блочным элементам;
* : before — вставляет генерируемое содержимое перед элементом;
* : after — добавляет генерируемое содержимое после элемента.

### 4.3.3 Комбинация селекторов

Для более точного отбора элементов для форматирования можно использовать комбинации селекторов:

* a[href][title] — выберет все ссылки, для которых заданы атрибуты href и title;
* img[alt\*="css"]: nth-of-type(even) — выберет все четные картинки, альтернативный текст которых содержит слово css.

### 4.3.4 Группировка селекторов

Один и тот же стиль можно одновременно применить к нескольким элементам. Для этого необходимо в левой части объявления перечислить через запятую нужные селекторы:



### 4.3.5 Наследование и каскад

Наследование и каскад — два фундаментальных понятия в CSS, которые тесно связаны между собой.

**Наследование** заключается в том, что элементы наследуют свойства от своего родителя (элемента, их содержащего).

**Каскад** проявляется в том, как разные виды таблиц стилей применяются к документу, и как конфликтующие правила переопределяют друг друга.

#### **1. Наследование**

**Наследование** является механизмом, с помощью которого определенные свойства передаются от предка к его потомкам. Спецификацией CSS предусмотрено наследование свойств, относящихся к текстовому содержимому страницы, таких как color, font, letter-spacing, line-height, list-style, text-align, text-indent, text-transform, visibility, white-space и word-spacing. Во многих случаях это удобно, так как не нужно задавать размер шрифта и семейство шрифтов для каждого элемента веб-страницы.

Свойства, относящиеся к форматированию блоков, не наследуются. Это background, border, display, float и clear, height и width, margin, min-max-height и -width, outline, overflow, padding, position, text-decoration, vertical-align и z-index.

##### Принудительное наследование

С помощью ключевого слова inherit можно принудить элемент наследовать любое значение свойства родительского элемента. Это работает даже для тех свойств, которые не наследуются по умолчанию.

##### Как задаются и работают CSS-стили

Стили могут наследоваться от родительского элемента (наследуемые свойства или с помощью значения inherit).

Стили, расположенные в таблице стилей ниже, отменяют стили, расположенные в таблице выше.

К одному элементу могут применяться стили из разных источников. Проверить, какие стили применяются, можно в режиме разработчика браузера. Для этого над элементом нужно щёлкнуть правой кнопкой мыши и выбрать пункт «Посмотреть код» (или что-то аналогичное). В правом столбце будут перечислены все свойства, которые заданы для этого элемента или наследуются от родительского элемента, а также файлы стилей, в которых они указаны, и порядковый номер строки кода.

При определении стиля можно использовать любую комбинацию селекторов — селектор элемента, псевдокласса элемента, класса или идентификатора элемента.

Html:



CSS:



#### **2. Каскад**

**Каскадирование** — это механизм, который управляет конечным результатом в ситуации, когда к одному элементу применяются разные CSS-правила. Существует три критерия, которые определяют порядок применения свойств — правило! important, специфичность и порядок, в котором подключены таблицы стилей.

##### Правило! important

Вес правила можно задать с помощью ключевого слова! important, которое добавляется сразу после значения свойства, например, span {font-weight: bold! important;}. Правило необходимо размещать в конец объявления перед закрывающей скобкой, без пробела. Такое объявление будет иметь приоритет над всеми остальными правилами. Это правило позволяет отменить значение свойства и установить новое для элемента из группы элементов в случае, когда нет прямого доступа к файлу со стилями.

##### Специфичность

Для каждого правила браузер вычисляет **специфичность селектора**, и, если у элемента имеются конфликтующие объявления свойств, во внимание принимается правило, имеющее наибольшую специфичность. Значение специфичности состоит из четырех частей: 0, 0, 0, 0. Специфичность селектора определяется следующим образом:

* для id добавляется 0, 1, 0, 0;
* для class добавляется 0, 0, 1, 0;
* для каждого элемента и псевдоэлемента добавляется 0, 0, 0, 1;
* для встроенного стиля, добавленного непосредственно к элементу — 1, 0, 0, 0;
* универсальный селектор не имеет специфичности.



В результате к элементу применятся те правила, специфичность которых больше. Например, если на элемент действуют две специфичности со значениями 0, 0, 0, 2 и 0, 1, 0, 1, то выиграет второе правило.

##### Порядок подключённых таблиц

Вы можете создать несколько внешних таблиц стилей и подключить их к одной веб-странице. Если в разных таблицах будут встречаться разные значения свойств одного элемента, то в результате к элементу применится правило, находящееся в таблице стилей, идущей в списке ниже.

## UML-схема

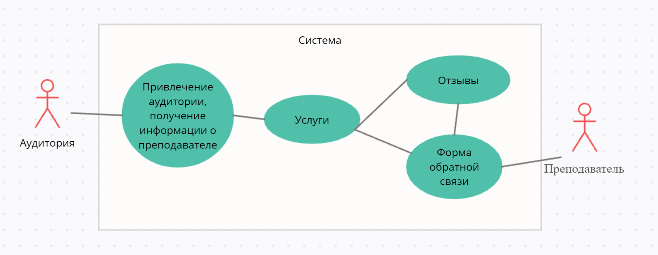


Рис. 22 UML-схема

## Заключение

В данной курсовой работе были изучены основные принципы при разработке Веб-сайта. Узнали, что прежде, чем приступить к разработке сайта необходимо определить цель его создания. Обычно их бывает три:

* создание дополнительного информационно-рекламного канала, совершенствование коммуникации с действующими и потенциальными партнерами, формирования положительного имиджа компании как предприятия, использующего в своей деятельности современные технологии;
* использование сети Интернет для рекламы и сбыта и продаж;
* создание портала, обеспечивающего работу с клиентами, партнерами, поставщиками (онлайн заказы, каталог товаров и услуг, прайс-листы, информационная поддержка), подключение.

В зависимости от цели выбирается структура сайта, их бывает много, но были выделены три соответственно: сайт-визитка, коммерческий сайт, интернет-магазин. Где сайт-визитка самый простой в построении сайт, в то время как интернет-магазин самый многофункциональный и труднореализуемый вид сайта.

После того, как определена структура сайта предстоит выбрать каталоги, подкаталоги и их наполнение контентом. Для лучшей работы сайта контент должен быть отфильтрован согласно рекомендациям. Текст должен быть понятным и хорошо заметным, изображения необходимо доводить до минимального размера без потери видимого качества. А также не перебарщивать с наполнением веб-страниц, чтобы не отпугивать посетителей сайта.

Познакомились с языком разметки HTML, который и позволяет написать код веб-страницы. Дизайн веб-страниц позволяет настроить таблица стилей CSS. Код стилей CSS может быть написан как внутри HTML-документа, так и в отдельном файле \*.css формата.

Для редактирования кода можно использовать любой текстовый редактор, например, VisualStudio Code. Либо специализированную среду разработки, самой популярной является – Adobe Dreamweaver.

В случае, когда у разработчика недостаточно знаний или мало времени можно использовать специальные онлайн-конструкторы для создания веб-сайтов. Они позволяют быстро и легко создавать веб-страницы. Конструкторы так же можно использовать для наброски веб-страницы, чтобы определить дизайн будущего сайта прежде, чем приступить к полноценному его созданию.

## Список используемой литературы

1. Ukit – ресурс с шаблонами сайтов, онлайн конструктор [Электронный ресурс] — шаблон сайта: <https://ukit.com/neowizard#template_preview/all/6e65775f74656163686572/g-theme-site-1> (дата обращения: 15.02.2023)
2. Хабр – информация [Электронный ресурс] —схема вёрстки веб-страницы: <https://habr.com/ru/articles/202408/> (дата обращения: 17.02.2023)
3. WebforMyaself – информация [Электронный ресурс] — пример формы обратной связи в HTML CSS: <https://webformyself.com/sdelajte-sovremennye-formy-s-pomoshhyu-css3-i-validacii-html5/> (дата обращения: 21.03.2023)
4. HTML5BOOK – информация [Электронный ресурс] — описание создания html-таблицы: <https://html5book.ru/html-table/> (дата обращения: 7.03.2023)
5. ИТФЕШ – информация [Электронный ресурс] — виды форм и примеры их создания: <https://itchief.ru/bootstrap/feedback-form#nastroyka-processphp-pochta-kapcha-i-dr> (дата обращения: 21.03.2023)
6. Хабр – информация [Электронный ресурс] — структура лендинга: <https://habr.com/ru/companies/trinion/articles/273917/> (дата обращения: 16.02.2023)
7. Micrisoft – информация [Электронный ресурс] — текстовый редактор Visual Studio Code: <https://learn.microsoft.com/ru-ru/training/modules/get-started-with-web-development/> (дата обращения: 13.02.2023)
8. HTML5BOOK – информация [Электронный ресурс] — структура CSS: <https://html5book.ru/osnovy-css/> (дата обращения: 01.03.2023)
9. HTML5BOOK – информация [Электронный ресурс] — структура HTML: <https://html5book.ru/html-html5/> (дата обращения: 20.02.2023)
10. Loftschool – информация [Электронный ресурс] — структура и синтаксис CSS: <https://loftschool.com/blog/posts/css-chto-eto> (дата обращения: 02.032023)
11. Хабр – информация [Электронный ресурс] — структура HTML: <https://habr.com/ru/companies/vdsina/articles/500190/> (дата обращения: 26.02.2023)