

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



1797

Выпускная квалификационная работа

на тему

«Разработка web-редактора CSS анимаций»

Обучающегося 4 курса очной формы
обучения направление подготовки:

09.03.01 «Информатика и
вычислительная техника»

направленность (профиль): Технологии
разработки программного обеспечения

Шимко Никиты Вячеславовича

Руководитель выпускной квалификационной работы:

Кандидат педагогических наук, доцент, доцент

Государев Илья Борисович

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
Введение	2
Глава 1. Теоретические и практические основы создания web-редактора CSS	5
1.1 Анализ основных типов web-редакторов CSS	5
1.2 Анализ актуальности создания собственного web-редактора CSS анимации	8
1.3 Обоснование выбора языка программирования для разработки web-редактора CSS анимации	10
Выводы по главе 1	14
Глава 2. Разработка web-редактора CSS анимации	15
2.1. Техническое задание	15
2.2. Принципы работы приложения	19
2.3. Архитектура приложения	20
2.4. База данных	28
2.5. Пользовательский интерфейс	30
Выводы по главе 2	31
ЗАКЛЮЧЕНИЕ	32
ГЛОССАРИЙ	33
СПИСОК ЛИТЕРАТУРЫ	Ошибка! Закладка не определена.

Введение

21 век, это век технологий, большинство людей погружены в этот прекрасный мир. Он необъятен, состоит из огромного множества сайтов для монтажа видео, создания аудио и изображений, редактирования, и много другого. По своей сути, это все программное обеспечение и веб приложения, позволяющие людям создавать и творить. Каждый старается создать неповторимый продукт, однако одних стараний мало и люди прибегают к использованию дизайнерских решений. Это может быть как дополнительная анимированная картинка в стандартизированном шаблоне, так и целая структура анимации. Что по итогу является небольшой строчкой в коде. Эту проблему все чаще решают при помощи web-редакторов CSS анимации.

Web-редактор CSS анимации – это специальная программа или же сайт, который многократно упрощает верстку нового сайта и автоматизирует уже существующий процесс создания анимации на языке описания внешнего вида HTML-документа.

Целью данной работы является анализ и исследование возможностей существующих web-редакторов CSS анимации, а также разработка собственного web-редактора, позволяющего облегчить процесс разработки CSS анимации.

Для достижения этой цели необходимо выполнение определенного ряда задач:

1. Найти и рассмотреть существующие web-редакторы CSS анимации.
2. Проанализировать самые популярные редакторы.
3. Выбрать функционал web-редактора
4. Определить язык программирования, который будет использоваться при написании данного проекта.

5. Обосновать выбранный язык программирования.
6. Разработать web-редактор CSS анимации с возможностью интеграции созданной анимации в верстку, посредством копирования готового кода CSS

Основным предметом исследования будет непосредственная разработка web-редактора и возможность интеграции полученного кода.

Новизна данного проекта проявляется в анализе приложений для разработки, а также демонстрация работы web-редактора CSS при создании анимации.

Основой теории и методики данного исследования, являются концепции современного программирования и научные труды ведущих IT-специалистов с описанием создания программ, в частности web-редакторов.

Работа имеет следующую структуру: введение, две главы, заключение, список литературы. Выпускная квалификационная работа содержит - страниц. В работе представлено - рисунков, - таблиц.

Глава 1. Теоретические и практические основы создания web-редактора CSS

1.1 Анализ основных типов web-редакторов CSS

Возможности web-редакторов охватывают практически все возможности самого языка CSS, основные из них:

- Помогают создать градиент при помощи визуального редактора;
- Реализовывают изменение заднего фона;
- Позволяют изменять изображение для последующей интеграции;
- Изменение свойств текста;
- Собирают отзывы клиентов;
- Создание анимации;

Редакторы можно поделить на две категории: теоретические(демо) и практические. Теоретические(демо), позволяют ознакомиться с определённой функцией, без последующей интеграции и настройки. В свою очередь практические web-редакторы позволяют тонко настраивать нужную функцию, а также интегрировать полученный результат в свой проект. Кроме того, web-редакторы делятся на текстовые и WYSIWYG (what you see is what you get), Второй тип все чаще применяются новичками и опытными специалистами для автоматизации и ускорения процесса создания проекта.

Популярными на данный момент web-редакторами CSS анимации являются «Animista» и «WebCodeTools». Их возможности не ограничиваются созданием шаблонной анимации, присущие всем web-редакторам CSS анимации. В их функционал входит тонкая настройка анимации: форма, вид, время, эффект. Рассмотрим подробнее каждый из них:

- **«Animista»**
«Animista» - бесплатное веб приложение, позволяющее создавать css-анимации любой сложности. Данное приложение не требует больших

познаний в CSS от пользователя, все делается по нажатию одной кнопки, это видно на рисунке 1.

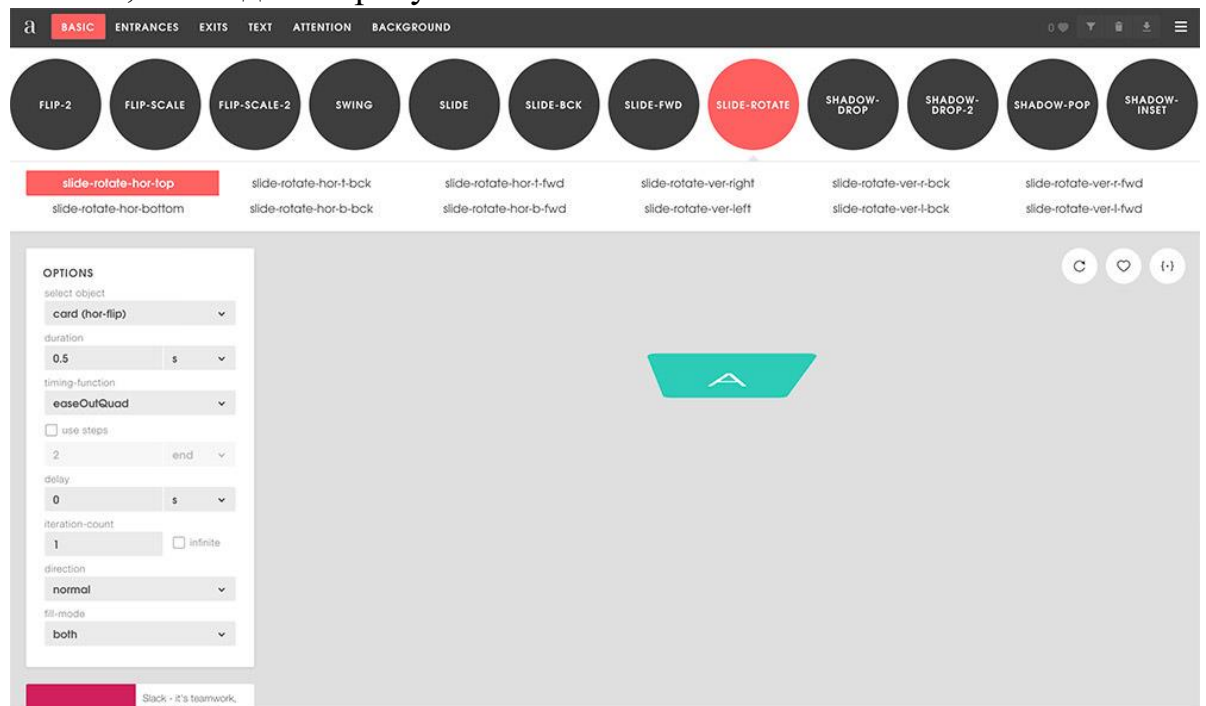


Рисунок 1

Сначала кажется, что интерфейс интуитивно непонятный, однако это не так, данный сервис предоставляет огромное множество вариантов редактирования CSS-анимации. Создание анимации происходит в несколько этапов:

- Для начала нужно выбрать формат анмации из предложенных сверху
 - Следом настроить его тип шаблонного движения: swing, slide, fade, flip
 - Последним пунктом является настройка параметров на панели слева
- Через данные параметры можно изменить продолжительность анимации, задержку, направление движения, практически все что

удобно, причем работать это будет на чистом CSS без использования сторонних языков программирования.

Закончив настройку и нажав на значок скобки, открывается новая страница с CSS-кодом представленном на рисунке 2.

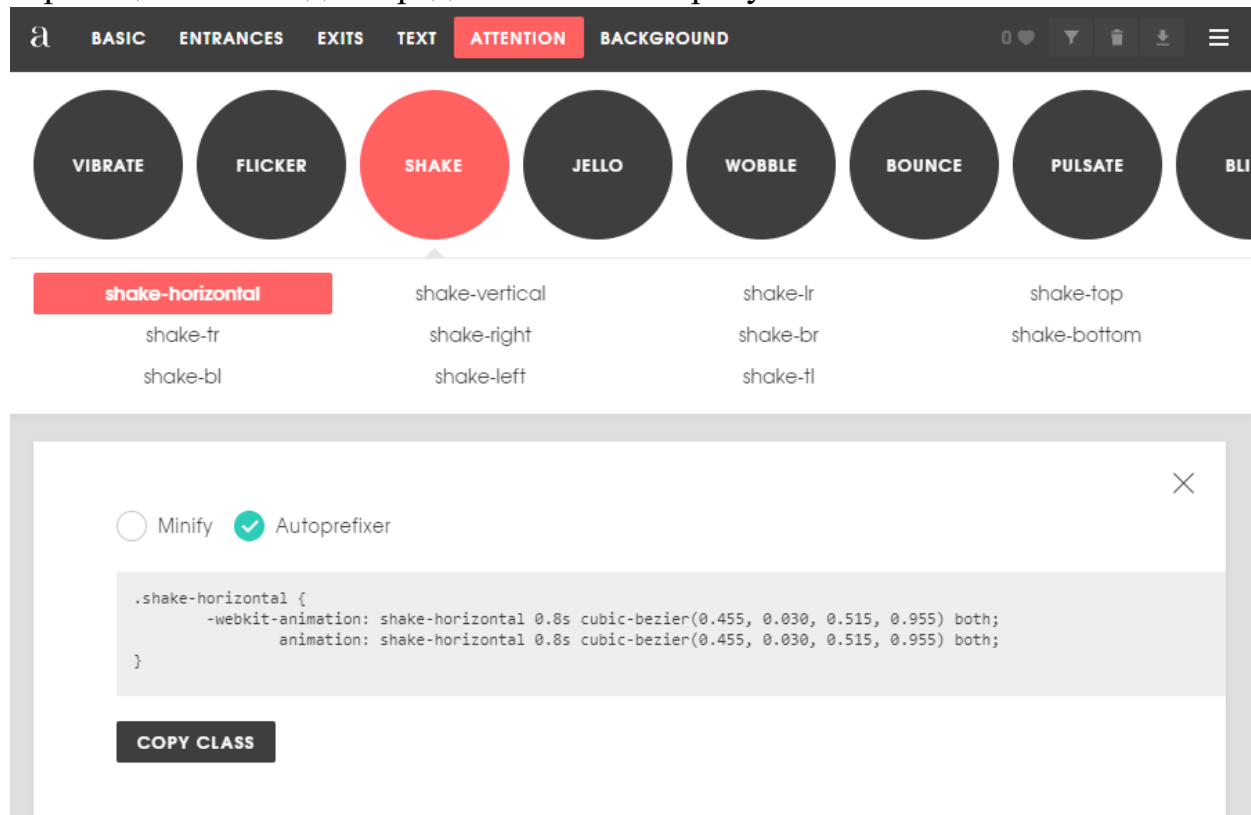


Рисунок 2

В заключение хочется добавить, что самый главный плюс данного редактора, он полностью бесплатен и не требует дополнительную плату для разблокировки функционала

- **«WebCodeTools»**

«WebCodeTools» - еще одно бесплатное web-приложение, позволяющее не просто редактировать CSS-анимации.

На рисунке 3 видно, что данный редактор не предлагает нам создать свою анимацию, предлагается выбрать один из подготовленных шаблонов для дальнейшего редактирования из разных категорий: Attention, Background, Basic. Выбрав анимацию, можно приступить к ее изменению, под блоком категорий представлено меню настроек. Закончив настройку анимации и посмотрев на ее визуализированную версию, можно скопировать готовый CSS-код

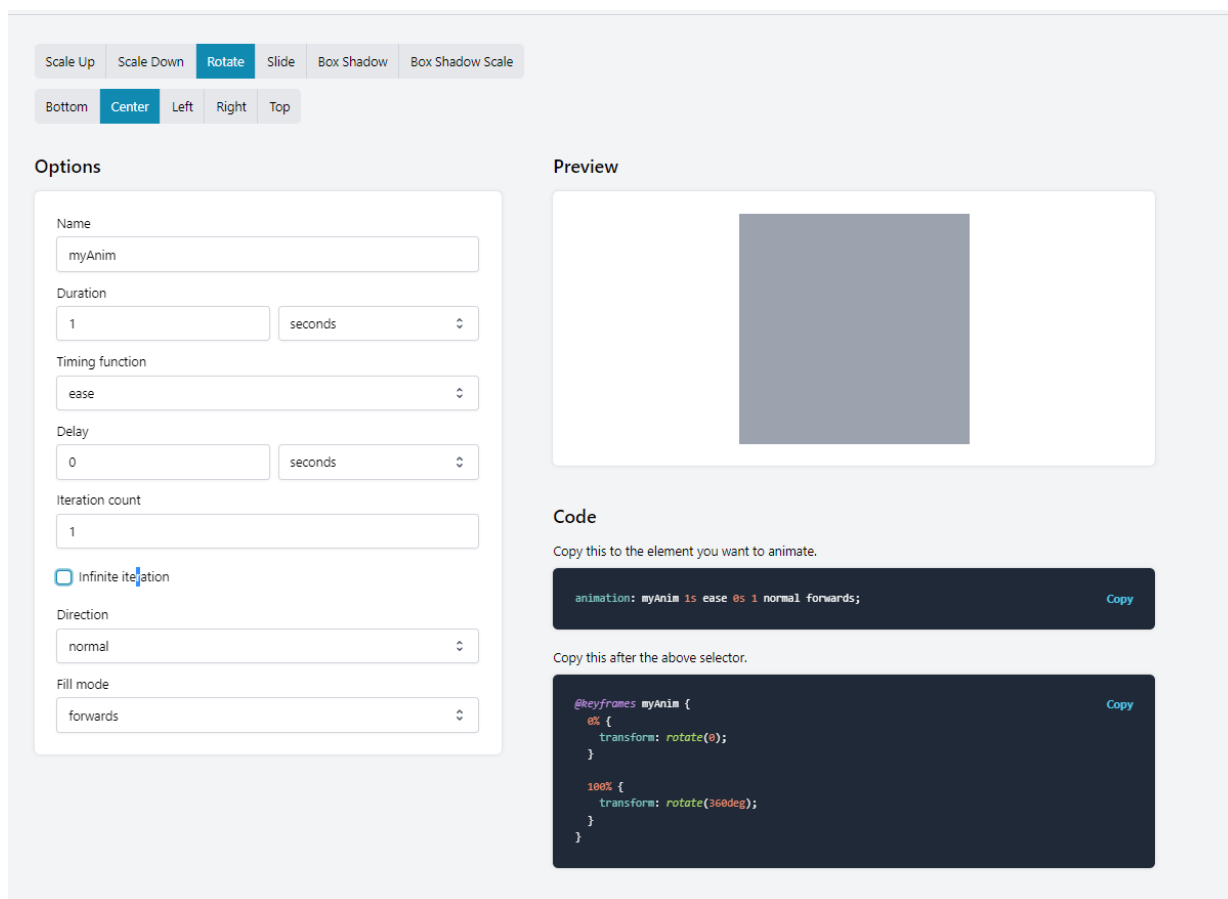


Рисунок 3

Затрагивая тему популярности web-редакторов CSS анимации, нельзя с точностью определить наиболее часто используемый, ведь практически любой редактор дает функционал, подходящий под навыки специалиста.

1.2 Анализ актуальности создания собственного web-редактора CSS анимации

Актуальность темы обусловлена отсутствием на просторах интернет-ресурсов, позволяющих генерировать CSS анимации, с возможностью их последующего редактирования непосредственно в формате онлайн.

Созданный редактор является необходимым для молодых специалистов и студентов, до этого не сталкивавшихся с web-редакторами CSS анимации, так как обладает интуитивно понятным интерфейсом, высокой скоростью работы и удобной интеграцией.

В интернете практически web-редакторы с понятным интерфейсом и быстрой скоростью работы, но даже те, которые удалось найти, не подходят под критерии, описанные выше.

Единственным таким редактором оказался «Magic CSS», у него присутствует удобный интерфейс, тонкая настройка анимации, хорошая интеграция,

однако данный редактор уже является программным обеспечением и требует установки, а это не удобно

Был проведен опрос, подтверждающий, что люди предпочитают онлайн редактор. Сам опрос присутствует по данному qr-коду:



Рисунок 4

Наиболее предпочитаемым редактором по результатам опроса, оказался онлайн редактор (более 69%)

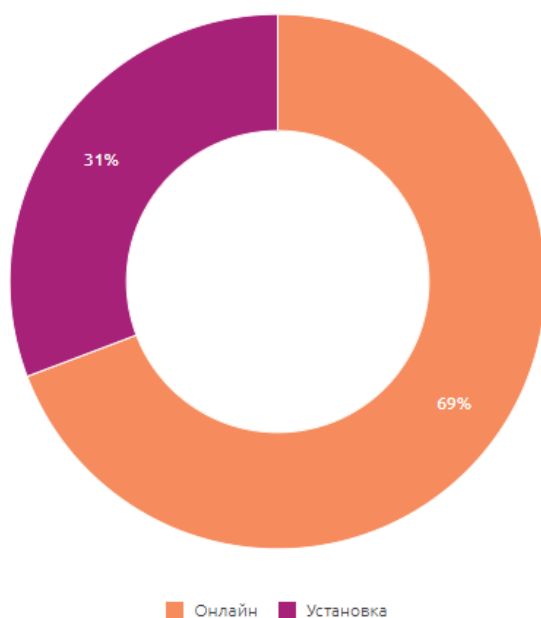


Рисунок 5

Из проведенного анализа можно сделать вывод о том, что создание web-редактора CSS анимации целесообразно.

Кроме того, в опросе был задан вопрос о важности дополнительного функционала у редактора. По результатам исследования было выявлено, что

для большинства специалистов и разработчиков на первом месте находится user-friendly интерфейс и скорость работы.

Данное исследование стало основанием актуальности проблематика данной работы: у специалистов и разработчиков, особенно молодых, остро стоит вопрос об удобстве и скорости использования web-редактора CSS анимации. Исходя из проведенного исследования, можно сделать вывод об актуальности и необходимости создания web-редактора CSS анимации, опираясь на полученные данные.

1.3 Обоснование выбора языка программирования для разработки web-редактора CSS анимации

Язык программирования – это набор текста в определённом формате, определяющего как выглядит программа и ее функционал, а также позволяющего верстать web-приложение.

Верстка – описание визуальной части сайта с помощью гипертекстового документа на основе HTML-разметки, что позволяет соединять и располагать на странице разные элементы web-сайта: текст, картинки, таблицы, видеоролики и тому подобное.

HTML – язык программирования или же язык гипертекстовой разметки для просмотра и редактирования web-страниц. Позволяющего менять визуальную часть при помощи CSS, а создавать любые рабочие функции при помощи интеграции JavaScript.

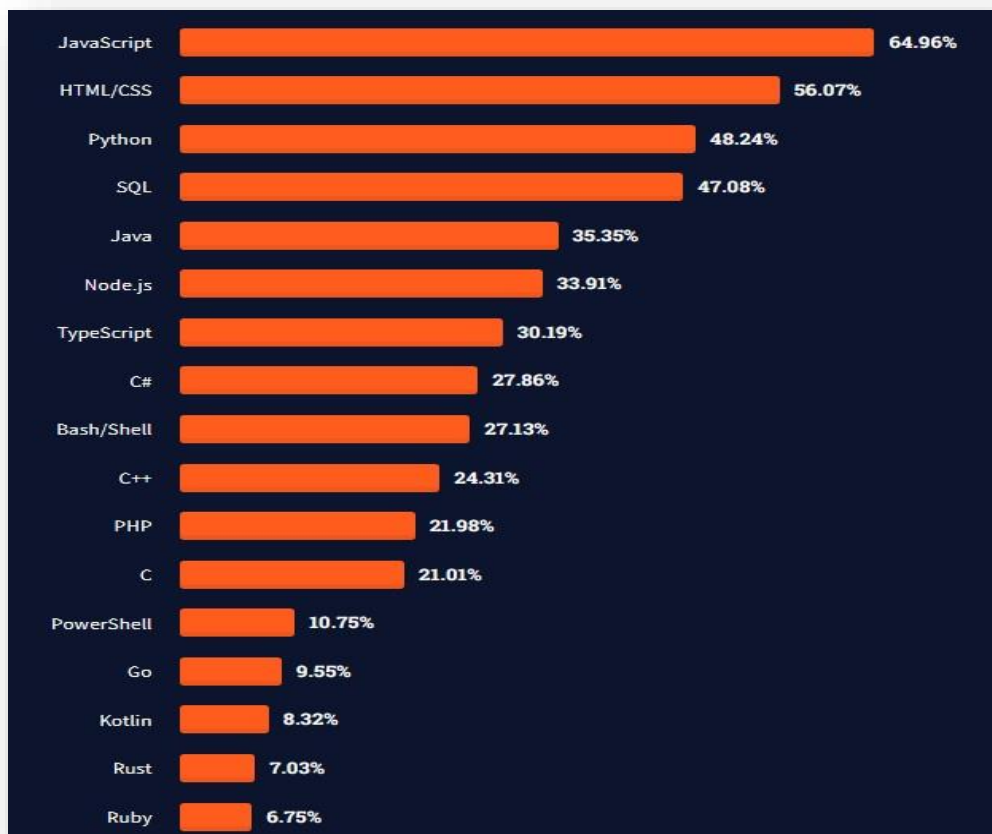
CSS – это таблица каскадных стилей, что означает язык, который отвечает за описание внешнего вида HTML-документа.

JavaScript – язык программирования высокого уровня, с собственным синтаксисом, позволяющий применять функции разной сложности на web-странице. Каждый раз, когда на странице происходит, что-то помимо статичного отображения и некоторых анимации — это JavaScript.

Синтаксис – система правил толкования каждого языка программирования, позволяющая обрабатывать данные.

В наше время конечно огромное количество языков программирования. У каждого своя специфика, сфера применения и особенности. На рисунке 3,

показан рейтинг самых популярных языков программирования:



Из данного рисунка можно сделать вывод, что JavaScript является одним из самых популярных языком программирования на данный момент, кроме того, он лучше всего подходит для разработки функционала web-редактора CSS анимации.

А для создания онлайн web-редактора CSS анимации лучше всего подойдет связка HTML/CSS.

Языки разметки HML и CSS используются при создании web-страниц. При этом основным языком является HTML, определяющий, что, как и где будет находиться на странице, CSS же описывает объекты, их цвет, шрифт, отступы, свойства изображений и пунктов меню.

Язык разметки HyperText Markup Language (HTML), появился в далеком 1991 году, являясь наследием SGML. Язык создавался для удобного использования людьми-неспециалистами в области верстки, уже с момента разработки у HTML были такие плюсы как:

- Простота – у HTML не так много структурных элементов, тегов, что позволяет легко понять их смысл
- Так как HTML текстовый редактор, то он позволяет форматировать документ без использования определённого визуального формата отображения.

Уже в версии HTML 3.0 в 1995 году, создатели решили добавить поддержку нового языка, разработанного специально для оформления web-страниц.

CSS (Cascading Style Sheets) – что в переводе является иерархической стилевой спецификацией, которая подключается к документу HTML и служит для оформления определённых частей документа. Данная система позволяет не зависеть от HTML, имеет свой синтаксис, независимо от тега, позволяет изменять внешние параметры представления на web-странице. После создания CSS стало возможным менять шрифт и размер чего-бы то ни было. Что самое важное, это стало возможно без создания новых тегов в HTML, механизм влияет на уже существующие.

В 2014 году вышел HTML 5, позволил еще более тонко использовать мультимедиа-технологии, язык стал больше внимания уделять поддержке скриптов, JavaScript стало возможно интегрировать непосредственно в код HTML.

В настоящее время 90% web-сайтов созданы при помощи HTML/CSS. Ниже приведены примеры преимуществ и недостатков HTML и CSS:

Преимущества HTML:

- Сайты на HTML мало весят.
- Проекты разработанные на HTML, намного быстрее своих конкурентов, кроме того, расходуют меньше ресурсов.
- HTML-версии не требуют обновлений.
- Легкость в верстке проекта.
- Низкая стоимость аренды хостинга.
- Низкая стоимость разработки.
- Стабильность каждой отдельной страницы, ведь каждая страница верстается отдельно
- Простота и дешевизна поддержки проекта, любой человек может заниматься обслуживанием

Недостатки HTML:

- Сложность внесения правок в формате онлайн, кроме того сложность изменения элементов, которые используются на каждой странице сайта.

- Отсутствие органов управления для админа, в таких случаях приходится использовать CMS
- Для верстки и изменения сайта, необходимо обладать навыками HTML/CSS

Преимущества CSS:

- Простота языка вместе с отделением от содержания, позволяет упростить разработку и поддержку сайта.
- Язык позволяет настраивать несколько вариантов страницы для просмотра на разных форматах устройств.
- Страница загружается быстрее, за счет переноса данных о оформлении в отдельный CSS-файл. Браузер сначала загружает “скелет” сайта, после чего данные об оформлении, которые в последствие хранятся в кэше браузера.
- CSS построенный на web-стандартах, повышает совместимость с разными платформами.

Недостатки CSS:

- Зависимость от версии браузера, один и тот же код, на браузере разной версии может отображаться по-разному.
- Иногда приходится изменять не только CSS-файл и структуру документа HTML.

Также для разработки web-редактора, нас понадобится библиотека jQuery. Данная библиотека позволяет повысить функционал кода, при его меньшем объеме.

Сейчас jQuery является проектом с открытым исходным кодом, который поддерживают разработчики со всего мира.

Основными преимуществами jQuery является:

- Написание кода в компактном виде, что занимает меньшее количество строк, чем чистый JavaScript.
- Упрощенный синтаксис, добавляющий удобства для обработки событий, добавления эффектов анимации и прочего.
- jQuery позволяет написать код, который работает во всех поддерживаемых браузерах без проверки, в отличие от JavaScript/
- Открытый исходный код, jQuery полностью бесплатен и используется разных проектов.

Затрагивая выбор типа web-редактора, то по проведенному исследованию видно, что пользователям требуется онлайн формат и нет необходимости в разработке программного обеспечения.

В качестве основных языков программирования для разработки web-редактора CSS анимации будут использоваться: CSS, HTML и JavaScript совместно с библиотекой jQuery.

Выводы по главе 1

Проанализировав рассмотренные инструменты, был сделан вывод, что на данный момент не существует web-ресурса, позволяющего максимально просто воспользоваться редактором CSS-анимации. Из чего было принято решение о разработке такого сервиса, пользуясь языками программирования HTML/CSS/JavaScript

Глава 2. Разработка web-редактора CSS анимации

Для успешной разработки данного приложения необходимо пройти через ряд устоявшихся шагов:

2.1. Техническое задание

2.1.1. ОБЩИЕ ПОЛОЖЕНИЯ

2.1.1.1. Наименование системы

2.1.1.1.1. Полное наименование системы

Полное наименование системы: Web-редактор CSS анимации

2.1.1.2. Наименование организаций – Заказчика и *Разработчика*

2.1.1.2.1. Разработчик

Разработчик: Шимко Никита Вячеславович

2.1.1.3. Плановые сроки начала и окончания работы

2.1.1.3.1. Общие даты

Плановый срок начала работ по созданию web-редактора – 9 апреля 2022 года.

Плановый срок окончания работ по созданию веб-сервиса – 10 сентября 2022 года.

2.1.1.3.2. Стадии и этапы разработки

- Разработка документации - до 09.04.2022
- Разработка тестовой версии с ограниченным функционалом - до

01.04.2022

- Тестирование и сбор обратной связи - до 22.05.2022
- Разработка тестовой версии ПО с расширенным функционалом - до 10.06.2022
- Тестирование и сбор обратной связи - до 01.08.2022
- Разработка финальной версии ПО - до 22.09.2022
- Начало процесса введения в эксплуатацию - до 31.12.2022

2.1.1.4. Основания для разработки

Главным обоснованием для разработки является отсутствие удобного web-редактора с таким функционалом, а также возможная научная ценность данного проекта и применение специалистами при верстке сайтов.

2.1.1.5. Порядок оформления и предъявления заказчику результатов работ по созданию системы

Тестирование сервиса происходит путем общего доступа к web-редактору и на основании сбора данных через опрос

Работа считается выполненной, если она была завершена в установленный срок и у заказчика не осталось замечаний к исправлению, относящихся к требованиям в данном ТЗ. В обратном случае разрабатывается новое ТЗ с новыми сроками работы и перечисленными исправлениями.

2.1.1.6. Перечень нормативно-технических документов,

При разработке web-сервиса и создании проектноэксплуатационной документации Исполнитель должен руководствоваться требованиями следующих нормативных документов:

– ГОСТ 19.201-78. ТЕХНИЧЕСКОЕ ЗАДАНИЕ. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОФОРМЛЕНИЮ;

– ГОСТ 34.602-89. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания;

2.1.2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

2.1.2.1. Назначение системы

ПО предназначено для помощи в создании CSS-анимации при верстке сайта на языке HTML

2.1.2.2. Цели создания системы

Основными целями создания веб-сервиса являются:

- Упрощение интеграции CSS-анимации при вёрстке сайта за счет копирования созданного кода;
- Улучшение производительности CSS-анимации по средством написания кода на чистом CSS;
- Ускорение верстки сайта, за счет удобного web-интерфейса.

2.1.3. ТРЕБОВАНИЯ К СИСТЕМЕ

2.1.3.1. Требования к функциям выполняемым системой

- ПО должно быть выполнено с использованием клиент-серверной архитектуры для возможности использования с помощью браузера;
- ПО должно обеспечивать функции ввода параметров CSS-анимации;
- ПО должно обеспечивать возможность копирования созданной CSS-анимации;
- ПО должно обеспечивать возможность выбора готовых шаблонов CSS-анимации;
- ПО должно быть разработано с учетом различных параметров пользователя

2.1.3.2. Требования к надежности

Система должна сохранять работоспособность и обеспечивать выполнение своих функций при возникновении следующих внештатных ситуаций:

- При ошибках в наличии Интернет-соединения;
- при ошибках в работе аппаратных средств, восстановление функции системы возлагается на ОС;
- при ошибках, связанных с программным обеспечением (ОС и драйверы устройств), восстановление работоспособности возлагается на ОС. Для защиты аппаратуры от бросков напряжения и коммутационных помех должны применяться сетевые фильтры.

2.2. Принципы работы приложения

Для разработки web-редактора будет использована клиент-серверная архитектура. Это заключается в наличии одного вычислительного сервера, являющегося центром приема запросов от клиента



Рисунок 8. Обобщенная схема работы сервиса

Первое что возвращает сервер при новом запросе – главная страница (*шаг 1*), она необходима для того, чтобы была возможность визуально контактировать с существующими органами управления, а также выбирать готовые шаблоны анимации (*шаг 2*). После выбора анимации, пользователь сразу переходит к форме ввода параметров анимации (*шаг 3*)

Форма ввода параметров представляет собой html-форму, в которой пользователь может ввести свои параметры для анимации и после отправки их на сервер, будет выполнена запись их в временный css-файл, для возможности дальнейшего копирования.

Весь функционал веб-сервиса, фактически, представлен одной HTML страницей, на которой находится форма ввода данных,

написанная на языке HTML, меню выбора анимации, проигрываемая анимация и окно копирования кода.

После отправки заполненной формы на сервер, будет выполнена обработка данных при помощи JavaScript, позволяющая вычлениь нужные параметры из объекта формы. После чего нужные данные интегрируются во временный css-файл, на который ссылается полученная анимация, позволяя изменить, а после скопировать измененный код.

Подобные меры помогают не только удобно работать с параметрами анимации, но и позволяют изменять параметры анимации, представленной в шаблонах, после чего можно получить измененный

После копирования пользователь может снова вернуться к выбору новой анимации, для дальнейшего изменения (шаг 4)

2.3. Архитектура приложения

Приложение состоит из нескольких основных частей:

- Меню выбора
- Меню копирования кода
- Форма ввода
- Блок с анимацией

1. Меню выбора представляет собой список категорий, внутри которых можно выбрать анимацию со стандартным шаблоном.

Реализация происходит посредством HTML и CSS.

```

<div class="cssAnimationContainer">
  <div class="leftBlock">

    <h1 class="mainHeading">Kiwunaka</h1>
    <div class="typeAnimation">
      <div class="headingTypeAnimation">Slide Effect</div>
      <div class="listAnimation">

        <div class="elemAnimation" data-name-animation="slideOutDown">
          <div class="nameAnimation">slideOutDown</div>
<pre class="codeAnimation"...>
        </div>
        <div class="elemAnimation" data-name-animation="slideOutRight">

          <div class="nameAnimation">slideOutRight</div>
<pre class="codeAnimation"...>
        </div>

        <div class="elemAnimation" data-name-animation="slideOutLeft">

          <div class="nameAnimation">slideOutLeft</div>
<pre class="codeAnimation"...>
        </div>

        <div class="elemAnimation" data-name-animation="slideOutUp">
          <div class="nameAnimation">slideOutUp</div>

```

Рисунок 6

Получается голый скелет, для удобного отображения на сайте, обратимся к файлу style.css, где формат списка приведётся к нужному

```
35     .mainHeading
36     {
37         color: #000;
38         font-size: 20px;
39         font-weight: 500;
40         margin-bottom: 10px;
41         padding: 10px;
42         text-align: center;
43         text-decoration: none;
44     }
45     .leftBlock
46     {
47         order: 1;
48         background: #F3F3F3;
49         background: #fff;
50         box-shadow: 0 0 3px #000;
51         display: flex;
52         flex-direction: column;
53         height: 100%;
54         width: 20%;
55         z-index: 999;
56     }
57     .headingTypeAnimation
58     {
59         background: #e97e7e;
60         font-weight: 500;
61         padding: 10px;
62     }
```

состоянию.

Рисунок 7

Конечный результат виден на картинке ниже:

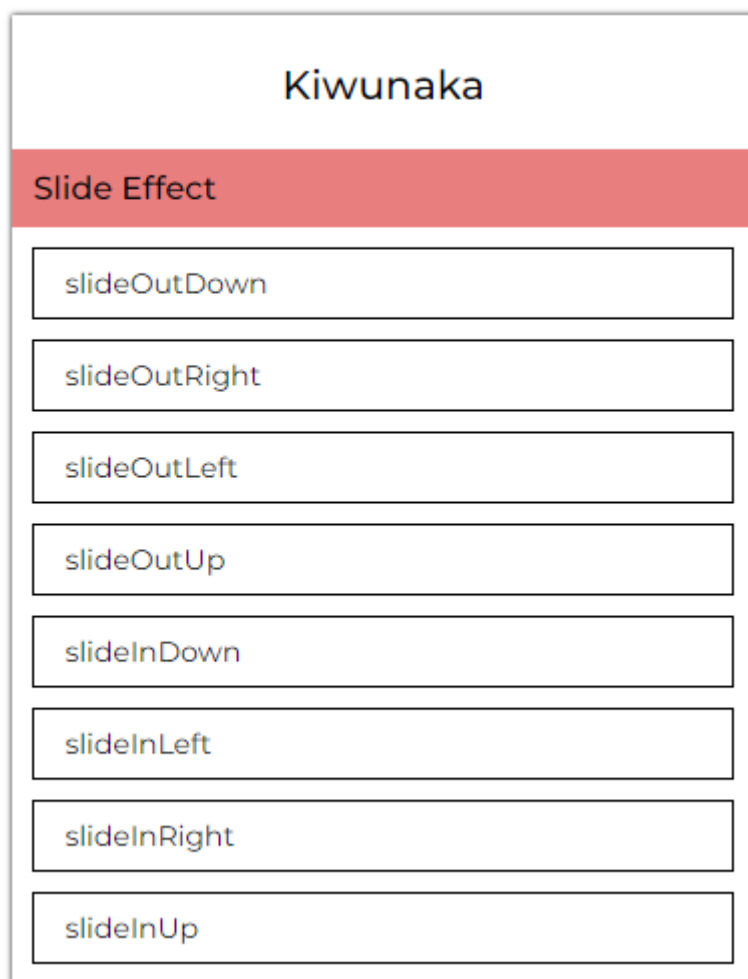


Рисунок 8

2. Меню копирования кода состоит из блока, появляющегося после нажатия, в котором размещен текст, заранее описывающий выбранную анимацию, реализация происходит посредством HTML. Картинка7

```

        </ul>
        <div class="butGetResult">Получить код</div>
        <div class="resetCode">Сбросить</div>
    </div>

    <div class="modelViewCode">
        <div class="modelContent">
            <div class="headerModal">
                <div class="textBlock">
                    Код для копирования
                </div>
                <div class="butExit">×</div>
            </div>
            <pre class="codeBlock"></pre>
            <div class="copyCode">Копировать код</div>
        </div>
    </div>
</div>
</div>

```

Рисунок 9

В ходе обработки каскадными стилями, получается итоговая демонстрация, представленная ниже.

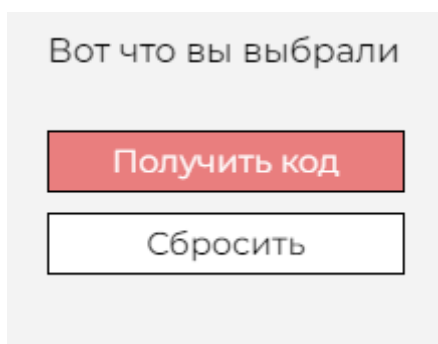


Рисунок 10

Чтобы окно вывода состояло из кода анимации, создана функция JavaScript, позволяющая отправить данные о выбранной анимации по нажатию.


```

/*при выборе анимации*/
$('.elemAnimation').on('click', function() {
    nameAnimation = $(this).attr('data-name-animation');

    if($(this).hasClass('active')) {
        $(this).removeClass('active');

        $('.blockResultSelect .listResultSelect').find("selectElem_"+nameAnimation+"").remove();
    }
    else {
        $(this).addClass('active');

        $('.blockResultSelect .listResultSelect').append("<li class='selectElem selectElem_"+nameAnimation+"'>"+nameAnimation+"</li>")
    }

    if($('.elemAnimation.active').length > 0) {
        $('.blockResultSelect').css('display', 'flex')
    }
})

```

Рисунок 11

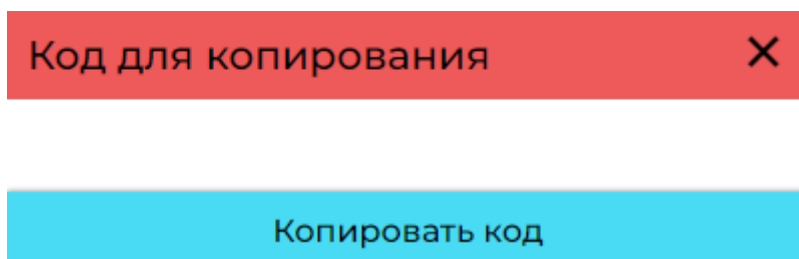


Рисунок 12

3. Форма ввода реализована посредством команд form и input, которые позволяют ввести данные для последующей обработки JavaScript, рисунок 11

```

<form id="option">
<legend>Options</legend>
    <div class="form-control duration">
<!--      все NAME (name="width") для инпутов делаешь такие же какие должны быть в css свойствах
    <label>
        Ширина:
        <input class="input" name="width" type="text" value="100" />
        <select name="size" class="size-unit-select" id="anim-size-unit">
            <option value="px">px</option>
        </select>
    </label>
    <label>
        Итерации:
        <input class="input" name="animation-iteration-count" type="text" value="infinite" />
        <select name="iteration" class="iteration-unit-select" id="anim-iteration-unit">

        </select>
    </label>
    <label>
        Длительность анимации:
        <input class="input" name="animation-duration" type="text" value="1" />
        <select name="duration" class="time-unit-select" id="anim-time-unit">
            <option value="s">s</option>
        </select>
    </label>
    </div>
    <button type="submit">Apply</button>
</form>

```

Рисунок 13

Полученные данные обрабатываются отдельной функцией option в JavaScript, после чего создается временный css файл testElem.css со значениями из формы ввода:

```

$('#option')
.on('submit', function(event) {
    event.preventDefault();
    const form = event.target;
    const body = new FormData(form);
    let optionValue = {};
    body.forEach((value : FormDataEntryValue, key : string) => {
        optionValue = {
            ...optionValue,
            [key]: value
        }
        console.log(key+" "+value)
    });

    // тут добавляем css элементу
    // в optionValue лежат все значения css, которые и применяются
    // со значениями в px - работает отлично, если нужно значение в ms или s, например, animation-duration: 5s, то еще нужно добавить как раз эти s или ms
    // например у в body пришли такие значения = {width: 100, animation-duration: 5}, чтобы свойства css применились нужно чтобы было так {width: 100, animation-duration: 5s}
    testElem.css(optionValue);

```

Рисунок 14

4. Блок с анимацией, фактически и является анимацией из примера, на странице размещен блок testEffect:

```
<div class="testEffect">  
  <picture></picture>  
</div>
```

Рисунок 15

В данном блоке находится изображение, уже к которому применяются анимации. Применение происходит посредством изменения каскадных стилей в файле style.css

```
.bounce  
{  
  animation-duration: 1s;  
  animation-fill-mode: both;  
  animation-name: bounce;  
  ms-transform-origin: center bottom;  
  transform-origin: center bottom;  
  webkit-animation-duration: 1s;  
  webkit-animation-fill-mode: both;  
  webkit-animation-name: bounce;  
  webkit-transform-origin: center bottom;  
}
```

Рисунок 16

Чтобы же выбрать нужную анимацию, реализована функция на JavaScript, применяющая определенные стили из style.css, при наведении мышью, которую видно на рисунке 17

```

/*при наведение на анимацию*/
$(".elemAnimation")
    .mouseenter(function() {
        nameAnimation = $(this).attr('data-name-animation');

        $(testElem).attr('class', '');
        $(testElem).addClass(nameAnimation);
    })

let fullCode;
$(".butGetResult").on( e: 'click', t: function() {

    fullCode = '';

    $(".elemAnimation.active").each(function() {
        fullCode = fullCode + $(this).find('.codeAnimation').html();
    })

    $(".modelViewCode .codeBlock").html(fullCode)

    $(".modelViewCode").css("display", "flex")

})

$('.butExit').on( e: 'click', t: function() {
    $(".modelViewCode").css("display", "none")
})

```

Рисунок 17

2.4. База данных

В данном web-редакторе не нужна сложная структура базы данных, по причине существования всего пары параметров: Длительность анимации, размер и количество итераций.

Каждый из этих параметров, полученных с формы ввода, имеют уникальный ID, он позволяет отслеживать их нахождение в основном CSS-файле, рисунок 8.

```

<form id="option">
<legend>Options</legend>
    <div class="form-control duration">
<!--      все NAME (name="width") для инпутов делаешь такие же какие должны быть в css свойства-->
        <label>
            Ширина:
            <input class="input" name="width" type="text" value="100" />
            <select name="size" class="size-unit-select" id="anim-size-unit">
                <option value="px">px</option>
            </select>
        </label>
        <label>
            Итерации:
            <input class="input" name="animation-iteration-count" type="text" value="infinite" />
            <select name="iteration" class="iteration-unit-select" id="anim-iteration-unit">
                </select>
        </label>
        <label>
            Длительность анимации:
            <input class="input" name="animation-duration" type="text" value="1" />
            <select name="duration" class="time-unit-select" id="anim-time-unit">
                <option value="s">s</option>
            </select>
        </label>
    </div>
    <button type="submit">Apply</button>
</form>

```

Рисунок 18

Обрабатываются данные при помощи функции FormData на языке JavaScript, которая позволяет создать объект, состоящий из введенных значений присвоив им уникальный ID.

```

$('#option')
    .on( e: "submit", t: function(event) {
        event.preventDefault()
        const form = event.target;
        const body = new FormData(form);
        let optionValue = {};
        body.forEach( callbackfn: (value : FormDataEntryValue ,key : string ) => {
            optionValue = {
                ...optionValue,
                [key]: value
            }
            console.log(key+" "+value)
        });
    });

```

2.5. Пользовательский интерфейс

Для начала создадим прототип страницы со всеми нужными элементами:



Рисунок 19

Теперь настроим через css, каждый из элементов, для удобства поделим страницу на два условных блока LeftBlock и RightBlock. В одном будет располагаться Меню выбора анимации, в другом сама анимация. Последним блоком является BlockResult, в котором выводится меню копирования полученного кода анимации. В итоге получается готовая страница, показана ниже.



Рисунок 20

Последний создаваемый элемент интерфейса, это `modelViewCode`. Данный элемент появляется уже фактически перед копированием код и выглядит следующим образом.

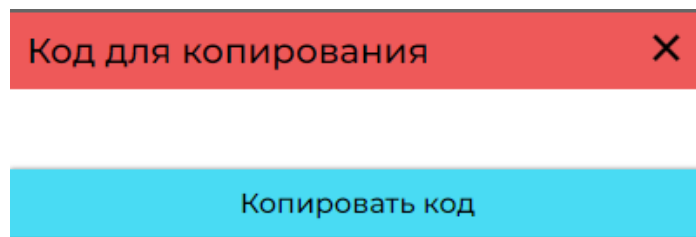


Рисунок 21

Выводы по главе 2

Во время разработки web-редактора были выполнены следующие шаги:

1. Разработано техническое задание
2. Описаны принципы работы приложения
3. Определена архитектура ресурса
4. Спроектирована база данных
5. Осуществлено создание пользовательского интерфейса
6. Разработан web-редактор CSS-анимации, работающий по принципу клиент-сервер, на основе HTML, CSS и JavaScript.

ЗАКЛЮЧЕНИЕ

В ходе работы были рассмотрены основные существующие типы web-редакторов CSS-анимации. Также был проведен анализ web-редакторов CSS-анимации, что позволило выяснить актуальность разработки собственного web-редактора CSS-анимации. Из проведенного анализа было решено выбрать тип продукта, набравшего большее количество откликов – онлайн web-редактор. В ходе данной работы был выполнен выбор языка программирования, соответствующий заявленному функционалу редактора. Наиболее подходящим оказался язык программирования JavaScript в связке с HTML и CSS. Для продолжения работы были проанализированы существующие библиотеки позволяющих работать с языками, после чего были выбраны наиболее подходящие

Для обоснования актуальности, потребовалось провести исследования со сбором обратной связи, с помощью которых было выявлено, что создание данного продукта необходимо. Проанализировав все полученные данные, был разработан web-редактор CSS анимации, позволяющий:

- Выбирать готовый шаблон анимации CSS
- Редактировать анимацию внутри выбранного шаблона
- Копировать как код шаблонного варианта анимации, так и изменной версии.

Таким образом в ходе данной работы были исполнены все задачи поставленный в главе 1, кроме того, разработан продукт, позволяющий облегчить специалистам верстки интернет-сайтов.

ГЛОССАРИЙ

JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript.

ECMAScript — это встраиваемый расширяемый не имеющий средств ввода-вывода язык программирования, используемый в качестве основы для построения других скриптовых языков.

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.

IDE — (англ. Integrated development environment — IDE), также единая среда разработки, ЕСР — комплекс программных средств, используемый программистами для разработки программного обеспечения (ПО). Среда разработки включает в себя: текстовый редактор, Транслятор (компилятор и/или интерпретатор), средства автоматизации сборки, отладчик.

WebStorm — интегрированная среда разработки на JavaScript, CSS & HTML от компании JetBrains, разработанная на основе платформы IntelliJ IDEA. WebStorm обеспечивает авто дополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями.

Visual Studio Code — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений.

<!DOCTYPE> - предназначен для указания типа текущего документа

`<html>` - Тег `<html>` является контейнером, который заключает в себе все содержимое веб-страницы, включая теги `<head>` и `<body>`

`<body>` - Элемент `<body>` предназначен для хранения содержания веб-страницы (контента), отображаемого в окне браузера. Информацию, которую следует выводить в документе, следует располагать именно внутри контейнера `<body>`

`<h1>` to `<h6>` - HTML предлагает шесть заголовков разного уровня, которые показывают относительную важность секции, расположенной после заголовка. Так, тег `<h1>` представляет собой наиболее важный заголовок первого уровня, а тег `<h6>` служит для обозначения заголовка шестого уровня и является наименее значительным. По умолчанию, заголовок первого уровня отображается самым крупным шрифтом жирного начертания, заголовки последующего уровня по размеру меньше.

`<p>` - Определяет текстовый абзац. Тег `<p>` является блочным элементом, всегда начинается с новой строки, абзацы текста идущие друг за другом разделяются между собой отбивкой.

`
` Тег `
` - устанавливает перевод строки в том месте, где этот тег находится.

`<hr>` - Рисует горизонтальную линию, которая по своему виду зависит от используемых параметров, а также браузера

`<!--...-->` - Тег добавляет комментарий в код документа. Текст комментария не отображается на странице.

СПИСОК ЛИТЕРАТУРЫ

1. Александрович, Д. В. (2016). *PHP, MySQL, HTML 5 и CSS 3. Разработка современных динамических Web-сайтов*. БХВ-Петербург. - 2017 - 160с.
2. Вячеславович, Д. В. (2015). *HTML, скрипты и стили. 4-е изд.* БХВ-Петербург., 1990.— 224 С.
3. Бизли Д. JavaScript. Подробный справочник. - Пер. с англ. - СПб.: СимволПлюс, 2010. - 864 с, ил.
4. Брукс Ф., Мифический человеко-месяц, или Как создаются программные системы, М.: Символ-Плюс, 2010.
5. Буйначев, С. К. Основы программирования на языке HTML [Электронный ресурс]: учебное пособие / С. К. Буйначев, Боклаг Н. Ю. – Электрон. текстовые дан. – Издательство Уральского университета, 2014.
6. Гэддис Т. Начинаем программировать на JavaScript. – 4-е изд.: Пер. с англ. СПб.: БХВ-Петербург, 2019. – 768 с.
7. Долинский М.С. Решение сложных и олимпиадных задач по программированию - Учебное пособие - М.: – 2006.
8. Доусен М. Програмируем на HTML(HTML Programming for the Absolute Beginner)//изд. «Питер», серия Бестселлеры O'Reilly, 2016,- 416с.
9. ДГрег, С. (2021). *Наглядный CSS*. “Издательский дом Питер.” – 416 с.
- 10.Златопольский Д.М. Основы программирования на языке JavaScript. – М.: ДМК Пресс, 2017. – 284 с.

- 11.Сапожников, А. П. Самоучитель HTML / А. П. Сапожников. — Санкт-Петербург : Питер, 2011. — 413 с. — Текст : непосредственный.
- 12.Элизабет, Р., & Эрик, Ф. (2018). *Изучаем HTML, XHTML и CSS 2-е изд.* “Издательский дом Питер., 2011. – 992 с.
- 13.Лутц М. Программирование на JavaScript, том II, 4-е издание. — Пер. с англ. СПб.: Символ-Плюс, 2011. – 992 с.
- 14.Кит Г. CSS для профи. : “Издательский дом “Питер,”” 2019. 768 с.
- 15.Александрович Д. В. JavaScript в Web-дизайне. : БХВ-Петербург, 2005. - 480 с.
- 16.Майк МакГрат «Программирование на Python для начинающих» Эксмо, 2015.
- 17.Александрович Д. В. HTML 5 и CSS 3: Современный Web-дизайн. : БХВ-Петербург, 2011. Вып. 1.
- 18.Прохоренок Н., Дронов В. Python 3 и PyQt 5. Разработка приложений //изд. «БХВ-Петербург», 2016,- 832с.
1. Прохоренок Н.А. JavaScript. Разработка приложений. – СПб.: БХВПетербург, 2012. – 704 с.
2. Прохоренок Н.А. Самое необходимое. — СПб.: БХВ-Петербург, 2011. — 416 с.
1. Рейтц К., Шлюссер Т. Автостопом по HTML. – СПб.: Питер, 2017. – 336 с.: ил. – (Серия «Бестселлеры O’Reilly»).
1. Россум Г., Дж. Дрейк Ф.Л., Откидач Д.С. Язык программирования CSS. 2011.
2. Саммерфилд М. JavaScript на практике, пер. А. Слинкин //изд. «ДМКПресс», 2014,-338с.

1. Свейгарт, Эл. Автоматизация рутинных задач с помощью JavaScript: практическое руководство для начинающих. Пер. с англ. — М.: Вильямс, 2016. — 592 с.
1. Северенс Ч. - Введение в программирование на Python - Национальный Открытый Университет "ИНТУИТ" - 2016 - 231с.
2. Робин, Н. (2016). *Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 4-е изд.* “Издательский дом Питер.” Символ –Плюс, 2008. -368 с, ил.
3. Сузи, Р. А. Язык программирования HTML[Электронный ресурс]: учеб. пособие/ Р. А. Сузи. – Электрон. текстовые дан. – ИнтернетУниверситет Информационных Технологий, 2007.
4. Сузи Р.А. - Язык программирования CSS- Национальный Открытый Университет "ИНТУИТ" - 2016 - 350с.
5. Сэнд У., Сэнд К. «console.log() Занимательное программирование на языке JavaScript» - М.: – 2016.
6. Федоров Д. Ю. Основы программирования на примере языка РЕБД. //Учебное пособие. – Санкт-Петербург: 2016.
1. Федоров, Д. Ю. Программирование на языке высокого уровня JavaScript: учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 161 с. – (Бакалавр. Прикладной курс).
2. Форсайт Р. Паскаль для всех / Пер. с англ. М.В.Сергиевского, А. В.Шалашова; Под ред. Ю.И. Топчиева. — М.: Машиностроение, 1986.- 288 с: ил.
3. Шелудько, В. М. Основы программирования на языке высокого уровня JavaScript: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 146 с.

1. Шелудько, В. М. Язык программирования высокого уровня JavaScript. Функции, структуры данных, дополнительные модули: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 107 с.
2. Шмпандар, Алекс Дж. CSS в верстке сайтов. : Пер. с англ. — М.: ООО "ИД. Вильямс", 2007. — 768 с.: ил.
3. Щерба А.В. Изучение языка программирования JavaScript на основе задач УМК авторов И.А. Калинин и Н.Н. Самылкина. //Учебное пособие. —М.: МПГУ, 2015.