3 Разработка программного продукта

3.1

Инструментальные и программные средства разработки

Создание web-приложений, их поддержка и развитие осуществляется с помощью

специализированного ПО

Программно-инструментальные средства - это программные продукты,

предназначенные для разработки программного обеспечения. К ним относят системы

программирования, которые включают систему команд процессора и периферийных

устройств, трансляторы с различных языков программирования.

Для разработки проекта были использованы следующие программные средства

разработки:

**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.

Хотя приведенный пример пытается влиять только на пару свойств, необходимых для рендеринга HTML-документа, он сам по себе квалифицируется как таблица стилей. В сочетании с другими таблицами стилей (одна фундаментальная особенность CSS заключается в том, что таблицы стилей объединяются), правило будет определять окончательное представление документа.

## Виды каскадных таблиц стилей и их специфика

* **СОДЕРЖАНИЕ:**
* [1. Виды таблиц стилей](https://html5book.ru/osnovy-css/" \l "part1)
* [2. Виды селекторов](https://html5book.ru/osnovy-css/" \l "part2)
* [3. Комбинация селекторов](https://html5book.ru/osnovy-css/" \l "part3)
* [4. Группировка селекторов](https://html5book.ru/osnovy-css/" \l "part4)
* [5. Наследование и каскад](https://html5book.ru/osnovy-css/" \l "part5)

### 1. Виды таблиц стилей

#### 1.1. 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 содержатся 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).

<head>

<link rel="stylesheet" href="css/style.css">

<link rel="stylesheet" href="css/assets.css" media="all">

</head>

HTML

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".

#### 1.2. 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).

<head>

<style>

h1,

h2 {

color: red;

font-family: "Times New Roman", Georgia, Serif;

line-height: 1.3em;

}

</style>

</head>

<body>

...

</body>

HTML

#### 1.3. Встроенные стили

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

<p style="font-weight: bold; color: red;">Обратите внимание на этот текст.</p>

HTML

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

#### 1.4. 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:

<style>

@import url(mobile.css);

p {

font-size: 0.9em;

color: grey;

}

</style>

HTML

Правило @import также используется для подключения веб-шрифтов:

@import url(https://fonts.googleapis.com/css?family=Open+Sans&subset=latin,cyrillic);

CSS

**JavaScript** – это язык программирования, который добавляет интерактивность на ваш веб-сайт (например: игры, отклик при нажатии кнопок или при вводе данных в формы, динамические стили, анимация). Эта статья поможет вам начать работать с этим захватывающим языком и даст вам представление о том, на что он способен.

[JavaScript](https://developer.mozilla.org/ru/docs/Glossary/JavaScript) ("JS" для краткости) — это полноценный [динамический язык программирования](https://developer.mozilla.org/ru/docs/orphaned/Glossary/Dynamic_programming_language), который применяется к [HTML](https://developer.mozilla.org/ru/docs/Glossary/HTML) документу, и может обеспечить динамическую интерактивность на веб-сайтах. Его разработал Brendan Eich, сооснователь проекта Mozilla, Mozilla Foundation и Mozilla Corporation.

JavaScript невероятно универсален и дружелюбен к новичкам. Обладая большим опытом, вы сможете создавать игры, анимированную 2D и 3D графику, полномасштабные приложения с базами данных и многое другое!

JavaScript сам по себе довольно компактный, но очень гибкий. Разработчиками написано большое количество инструментов поверх основного языка JavaScript, которые разблокируют огромное количество дополнительных функций с очень небольшим усилием. К ним относятся:

Программные интерфейсы приложения ([API](https://developer.mozilla.org/ru/docs/Glossary/API)), встроенные в браузеры, обеспечивающие различные функциональные возможности, такие как динамическое создание HTML и установку CSS стилей, захват и манипуляция видеопотоком, работа с веб-камерой пользователя или генерация 3D графики и аудио сэмплов.

Сторонние API позволяют разработчикам внедрять функциональность в свои сайты от других разработчиков, таких как Twitter или Facebook.

Также вы можете применить к вашему HTML сторонние фреймворки и библиотеки, что позволит вам ускорить создание сайтов и приложений.

Поскольку эта статья должна быть только лёгким введением в JavaScript, мы не собираемся путать вас на этом этапе, подробно рассказывая о том, какая разница между основным языком JavaScript и различными инструментами, перечисленными выше. Вы можете подробно изучить все это позже, в нашей [учебной области JavaScript](https://developer.mozilla.org/ru/docs/Learn/JavaScript) и в остальной части MDN.

Ниже мы познакомим вас с некоторыми аспектами основного языка, и вы также будете играть с несколькими функциями API браузера. Веселитесь!

Вместо рутинного вывода HTML-кода командами языка (как это происходит, например, в Perl или C), скрипт PHP содержит HTML с встроенным кодом (в нашем случае, это вывод текста "Привет, я - скрипт PHP!"). Код PHP отделяется специальными [начальным и конечным тегами <?php и ?>](https://www.php.net/manual/ru/language.basic-syntax.phpmode.php), которые позволяют "переключаться" в "PHP-режим" и выходить из него.

**PHP** отличается от JavaScript тем, что PHP-скрипты выполняются на сервере и генерируют HTML, который посылается клиенту. Если бы у вас на сервере был размещён скрипт, подобный вышеприведённому, клиент получил бы только результат его выполнения, но не смог бы выяснить, какой именно код его произвёл. Вы даже можете настроить свой сервер таким образом, чтобы обычные HTML-файлы обрабатывались процессором PHP, так что клиенты даже не смогут узнать, получают ли они обычный HTML-файл или результат выполнения скрипта.

PHP крайне прост для освоения, но вместе с тем способен удовлетворить запросы профессиональных программистов. Не пугайтесь длинного списка возможностей PHP. Вы можете быстро начать, и уже в течение нескольких часов сможете создавать простые PHP-скрипты.

Хотя PHP, главным образом, предназначен для работы в среде веб-серверов, область его применения не ограничивается только этим. Читайте дальше и не пропустите главу [Возможности PHP](https://www.php.net/manual/ru/intro-whatcando.php) либо, начните непосредственно с [Вводного руководства](https://www.php.net/manual/ru/tutorial.php), если вас интересует исключительно веб-программирование.

Также использовались следующие инструментальные средства:

GitHub - сервис онлайн-хостинга репозиториев, обладающий всеми функциями

распределённого контроля версий и функциональностью управления исходным кодом —

всё, что поддерживает Git и даже больше. Также GitHub может похвастаться контролем

доступа, багтрекингом, управлением задачами и вики для каждого проекта.

Рисунок 7 – репозиторий GitHub

**OpenServer** — это дистрибутив для развертывания локального сервера на ПК с предустановленной ОС Windows (WAMP). Его устанавливают для разработки сайта (например, WordPress) без выхода в интернет. Такой проект полноценно функционирует, идентичен размещенному в internet-сети, отображается в браузере. Пользуются им вебмастера и разработчики для тестирования настроек Тем, плагинов, кода, функций сайта.

Visual Studio Code – это один из наиболее популярных редакторов кода,

разработанный корпорацией Microsoft. Он распространяется в бесплатном доступе и

поддерживается всеми актуальными операционными системами: Windows, Linux и macOS.

VS Code представляет собой обычный текстовый редактор с возможностью подключения

различных плагинов, что дает возможность работать со всевозможными языками

программирования для разработки любого IT-продукта.