**Проектная работа на тему:**

«Создание и работа сайта»

Выполнил: ученик 11 «А» класса

МБОУ СОШ №88 г. Воронеж

Баранов Александр Владимирович

Воронеж, 2022

**Содержание**

1. Введение……………………………………………………………………3
2. История возникновения Интернета………………………………………4
3. Прототип Интернета
4. Интернет нашего времени
5. Работа сайта…………………………………………………………..……5
   1. Взаимодействие с пользователем
   2. Хостинг
   3. Браузеры
   4. Домен
   5. Протоколы
6. Введение в создание сайтов…………………..………………………..…9
   1. Состав
   2. Верстка
   3. Конструкторы
7. Список литературы…………………………………………………….…14
8. Приложения……………………………………………………………….

***Введение***

В современном мире сайты стали обыденностью. Крупные и даже малые бизнесы имеют свой сайт, множество интернет-магазинов — это сайт, а также существуют визитки в виде сайта. Интернет быстро развивается, появляются новые возможности использования интернет-ресурсов. Но большинство пользователей не знают, как работают и появляются сайты. Как задачу данного проекта, я ставлю создать 2 сайта разными способами.

***История возникновения Интернета***

**Прототип Всемирной паутины**

История интернета начинается в 60-годы XX века. Впервые определение описал Джозеф Карл Робнетт Ликлайдер – американский ученый с названием «Галактическая сеть».

В 1969 агентство DARPA начало создавать экспериментальную сеть «с коммутацией пакетов» и назвали ее ARPANET.

Коммутация пакетов - способ передачи данных по сети, принцип работы очень простой: делим информацию на маленькие пакеты и отправляем их независимо друг от друга. Это нужно для надёжности, скорости и эффективности.

В декабре 1970 года Network Working Group придумала протокол управления сетью, а в 1971 - 1972 его реализовали в ARPANET. Из-за этого появилась возможность создавать сетевые приложения. Первым приложением стала электронная почта, её сделали в 1972-м.

**Всемирная паутина нашего времени**

Паутину, какую мы ее знаем, придумал Тим Бернерс-Ли. Он изобрёл технологии URI/URL, HTTP, и HTML.

Первая важная технология, которая позволила появиться интернету - URL. Она ﻿применяется для обозначения адресов почти всех ресурсов Интернета: сайты и ﻿сервера ﻿приложений вроде ﻿электронной почты. ﻿URL сочетает в себе две технологии:

URI - стандарт записи уникального адреса. Например, адрес сайта это URI: http://google.com

DNS - система доменных имён. У любого устройства в сети есть свой числовой адрес, он называется IP-адрес. Но людям удобнее запоминать буквы - доменные имена. DNS просто помогает найти IP-адрес по доменному имени. Пример: 173.194.44.40 это IP-адрес, а google.com - доменное имя, которое ему соответствует.

***Работа сайта***

**Среда сайта**

World Wide Web (WWW, Всемирная паутина) – одна из информационных служб Интернета. Всемирную паутину образуют миллионы веб-серверов сети Интернет, расположенных по всему миру.

**Веб-сервер**

Веб-сервер — компьютерная программа, запускаемая на подключённом к сети компьютере и использующая протокол HTTP для передачи данных. Более сложные веб-серверы способны в ответ на HTTP-запрос динамически генерировать документы с помощью шаблонов и сценариев.

Для просмотра информации, полученной от веб-сервера, на клиентском компьютере применяется специальная программа — веб-браузер. Всемирная паутина неразрывно связана с понятиями гипертекста и гиперссылки. Большая часть информации в Вебе представляет собой именно гипертекст.

**Наполняющие сайта**

Гиперссылка – ключевое слово или объект в документе, с которым связан указатель для перехода на другую страницу в паутине.

Гипертекст – тест, в котором используются гиперссылки.

**Что такое сайт**

Сайт —несколько логически связанных между собой веб-страниц.

Веб-страница — документ или информационный ресурс Всемирной паутины, доступ к которому осуществляется с помощью веб-браузера.

Типичная веб-страница представляет собой текстовый файл в формате HTML, который может содержать ссылки на файлы в других форматах (текст, графические изображения, видео, аудио, мультимедиа, прикладные программы, базы данных, веб-службы и прочее), а также гиперссылки для быстрого перехода на другие веб-страницы или доступа к ссылочным файлам.

**Взаимодействие с пользователем**

В простейшем виде веб-сервер получает по сети HTTP-запрос на определённый ресурс, находит соответствующий файл на локальном жёстком диске и отправляет его по сети запросившему компьютеру.

**Хостинг**

Хостинг сайтов – это онлайн услуга, которая позволяет публиковать ваш веб-сайт или веб-приложение в интернете. Когда вы подписываетесь на услугу хостинга, вы обычно арендуете пространство на сервере, на котором вы можете хранить все файлы и данные, необходимые для правильного функционирования вашего сайта. Сервер – это физический компьютер, который работает без перерывов.

**Браузеры**

Что делает браузер:

* Разрешение DNS (процесс понятия браузером к какому серверу подключиться)
* HTTP-обмен (установление с сервером TCP-соединения и начало HTTP-обмена)
* Рендеринг (Браузер считывает и анализирует HTML-код, загружает дополнительные ресурсы, включенные в него (например, там могут быть указаны для погрузки JavaScript-файлы или CSS-документы) и представляет их пользователю как можно скорее.)
* Сброс и повтор (Браузер сбрасывает все настройки, применившиеся с последнего запроса)

**Домены**

Домены бывают нескольких уровней:

1. Корневой домен — домен самого верхнего уровня в любой системе доменных имён. Корневой домен Интернета (называемый иногда доменом нулевого уровня) обслуживается корневыми серверами системы доменных имен (DNS), которые располагаются в различных странах мира.

В системе DNS корневой домен обозначается пустым (т. е. не содержащим никаких символов) именем. Если эта точка есть (например «www.example.com.»), то доменное имя считается полным (абсолютным), включая в себя имена всех родительских доменов иерархии DNS. Если точки в конце имени нет («www.example» или «www.example.com»), то имя считается относительным.

1. Домены верхнего (первого уровня). Их еще называют доменными зонами

К доменам этого уровня относят доменные имена типа .ru, .рф, .su, .com, .net, .club.

Домены верхнего уровня разделяют на несколько групп:

Национальные домены верхнего уровня (cctld). На территории РФ к таким доменам относят доменные зоны .RU, .РФ, .SU. Так же к национальным доменам можно отнести доменные зоны: .ua (Украина), .de (Германия), .eu (Европейский Союз).

Международные (общие) домены верхнего уровня (gTLD). Это такие доменные зоны как: .com, .info, .biz, .name, .org.

Новые международные домены верхнего уровня (new gTLD). Эти доменные зоны тематически ориентированы на различные сферы жизни и бизнеса: хобби и быта (.photography, .gallery, .garden), бизнеса и финансов (.finance, .money, .business), спорта и развлечений (.bike, .fitness, .bar), стиля и красоты (.style, .best, .fashion), профессиональной деятельности (.guru, .expert).

1. Домены второго уровня.

На доменах первого уровня регистрируются домены второго уровня. Именно домены второго уровня распространены для коммерческого использования. К примеру, доменами второго уровня является сайты dnevnik.ru, wikipedia.org, google.com.

1. Домены 3 уровня

С их помощью выносятся части сайта, и они становятся отдельными ресурсами, например ru.wikipedia.com, где ru. – домен 3 уровня.

**Протоколы**

Протокол сайта, он же протокол передачи данных - набор правил, которые описывают очередность, особенности взаимодействия двух и более устройств, подключенных к одной сети и осуществляющих коммуникацию. Без протокола устройства, подключенные к сети, просто не могли бы договориться между собой, кто и в каком порядке отсылает запросы, кто и как их обрабатывает. Протокол сайта помогает все это урегулировать и дает возможность всем пользователям взаимодействовать с серверами без каких-либо проблем.

В основном сайты используют такие протоколы как HTTP (Hyper Text Transfer Protocol) и HTTPS (Hyper Text Transfer Protocol Secure).

Протокол HTTP:

Работа через данный протокол осуществляется по схеме client-server. Существует сервер, который в пассивном режиме постоянно ожидает, когда с ним будет установлено соединение. Это соединение с ним рано или поздно установит клиент, то есть машинный интерфейс пользователя интернетом. Клиент хочет что-то получить от сервера: получить страницу, открыть картинку, скачать песню. Чтобы сообщить о том, что именно хочет клиент, пользователь отправляет запросы, которые сервер умело обрабатывает. Сервер умеет обрабатывать запросы юзера благодаря инструкции, которой его снабдил HTTP протокол.

Протокол HTTPS:

Он является расширенной версией HTTP. Главное отличие в том, что теперь запросы от клиента отправляются не в голом виде, а в зашифрованном благодаря криптографическим механизмам SSL и TLS (асимметричная криптография для аутентификации ключей обмена, симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений). Использование этого протокола позволяет добиться такого результата, при котором запрос от пользователя может быть действительно прочтен только на стороне сервера, и никак не может быть перехвачен третьей стороной где-то по середине.

***Создание сайта***

**Введение в создание**

**Состав**

Как работают сайты мы разобрались, но откуда они берутся? Их можно делать двумя способами: писать вручную и использовать конструкторы.

В программировании отрасль создания сайтов называется WebDev. Она разделяется на внешнюю и внутреннюю структуру сайта. Внутренняя часть состоит из скелета, скриптов, серверной части и управлением содержимым. Внешняя часть отвечает только за оформление сайта.

**Верстка**

Верстка сайта — это структурированное сочетание изображений, заголовков, подзаголовков, таблиц, инфографик, текста и других элементов на странице c помощью языка разметки HTML и языка описания внешнего вида страницы CSS.

**Ручная верстка**

**HTML**

Hyper Text Markup Language - язык разметки гипертекста. Опять разбираемся в сложных терминах. В этой аббревиатуре нам интересно слово "разметка".

Разметка - что это вообще такое? Представь, что ты передаёшь текст по сети. Как сделать в тексте заголовок? Выделить абзац? Подчеркнуть слово? Самый простой вариант - пометить начало и конец выделяемого фрагмента условными метками.

HTML позволил создавать документы, в которых были заголовки, абзацы, ссылки, таблицы, картинки и куча всего прочего.

**Теги**

Как ты уже догадался, в HTML для разметки используется особый набор символов, называющийся тег.

Тег — это синтаксическая единица языка HTML, которая выделяет или создаёт элемент. Это набор символов, с помощью которого браузер понимает, где элемент создается, начинается и заканчивается. Есть 2 вида тегов: двойные и одинарные.

Двойные теги показывает начало и конец элемента. Начало элемента обозначается открывающим тегом <…>, а конец - закрывающим </…>.

Одинарные теги просто не имеют пары. Примеры: тег переноса строки <br> или горизонтальной линии <hr>.

Теги можно вкладывать друг в друга.

**Атрибуты**

Атрибуты — это свойства тега. С помощью них задаются параметры тега. У атрибута может не быть значения, тогда наличие атрибута включает какой-то параметр, а отсутствие - отключает

**Скрипты**

Для создания скриптов (последовательности действий) используют JS, реже Python. Их используют для приложений, анимации, внедрении рекламы.

**Структура HTML документа**

<!DOCTYPE>

Первым тегом в любом HTML документе должен идти тег <!DOCTYPE>. Он говорит браузеру, по какому стандарту написана страница

<html>

Вторым тегом идет <html> - контейнер, корневой тег, в котором все остальные элементы должны быть его потомками.

<head>

В теге <head> хранится информация о странице. Здесь указывают кодировку <meta charset="...">, имя страницы <title>...</title>, специальную информацию для поисковиков, а ещё тут подключаются стилевые файлы и скрипты. Тег <head> не отображается. Его цель — сказать браузеру информацию о странице.

<body>

В теге <body> размещается весь контент страницы, который пользователь увидит в браузере.

Еще один важный тег – div. Он является блочным элементом и предназначен для выделения фрагмента документа с целью изменения вида содержимого

**Серверная часть**

Большинство крупных веб-сайтов используют программирование серверной части чтобы динамично отображать различные данные при необходимости, в основном взятые из базы данных, располагающейся на сервере и отправляемые клиенту для отображения. Динамические сайты могут выделять контент, который более актуален в зависимости от предпочтений и привычек пользователя. Это также может упростить использование сайтов за счёт сохранения личных предпочтений и информации, например, повторного использования сохранённых данных кредитной карты для оптимизации последующих платежей.

Это также даёт возможность взаимодействовать с пользователем сайта, посылая уведомления и обновления по электронной почте или по другим каналам.

**CSS**

За внешнюю составляющую сайта отвечает CSS (CascadingStyleSheets) CSS – каскадные таблицы стилей — формальный язык описания внешнего вида документа, написанного с использованием языка разметки (HTML или XHTML). CSS на самом деле не является языком программирования. Это не язык разметки — это язык таблицы стилей. Это означает, что он позволяет применять стили выборочно к элементам в документах HTML.

Он представляет из себя ссылку на элемент их HTML с определенными значениями вида: шрифт, цвет, размер, границы и тому подобное.

Но необходимо связать наш документ с таблицей стилей. Это можно сделать несколькими способами:

Связанные стили

При использовании связанных стилей описание элементов и их значений располагается в отдельном файле, как правило, с расширением .css, а для связывания документа с этим файлом применяется тег <link>. Данный тег помещается в контейнер <head>

Глобальные стили

При использовании глобальных стилей свойства CSS описываются в самом документе и располагаются в заголовке веб-страницы.

Внутренние стили

Внутренний или встроенный стиль является по существу расширением для одиночного тега используемого на текущей веб-странице

Импорт CSS

В текущую стилевую таблицу можно импортировать содержимое CSS-файла(Табл. 1)

**Структура CSS**

Селектор

Имя HTML-элемента в начале набора правил. Он выбирает элемент(ы) для применения стиля. Для стилизации другого элемента, просто измените селектор.

Свойства

Способы, которыми вы можете стилизовать определённый HTML-элемент. В CSS вы выбираете, какие свойства вы хотите затронуть в вашем правиле.

Значение свойства

Справа от свойства, после двоеточия, у нас есть значение свойства, которое выбирает одно из множества возможных признаков для данного свойства

Вид кода:

Селектор {объявление}, где объявление описывается так: свойство: значение свойства и, в общем и целом, у нас получается селектор свойство: значение свойства}

Правила синтаксиса

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

**Селекторы**

Существует множество различных типов селектора. Рассмотрим их.

**Комментарии**

В CSS комментарии описываются с помощью /\* и \*/.

**Конструктор**

Конструктор сайтов — программно-реализованная сложная система для создания веб-страниц. Обычно является отдельным сервисом, но может и предоставляться как дополнительная услуга хостинг-компаний. Существует возможность создания сайтов, как с помощью CMS, так и с помощью и SaaS-платформ.

Основа конструктора – создание страниц путем создания блоков элементов.

Созданный мною сайт с помощью конструктора: 2356442.mya5.ru

**CMS**

 Из всей огромной массы веб-сайтов, наполняющих интернет, около 31% используют CMS. CMS (content management system) – это ПО на базе скриптов, которое позволяет управлять содержимым ресурса, менять его, просматривать и контролировать. Сайт созданный на CMS: https://pojectmy.wordpress.com/

Существует 4 типа CMS:

* С открытым кодом (движок может модифицировать любой желающий)
* Коробочные (изменения могут внести только официальные разработчики)
* Самописные (разрабатывают на заказ под конкретный проект)
* Фреймворки (надстройка над ЯП, набор библиотек)

Из 4 960 632 допрошенных доменов зоны RU CMS обнаружена на 26,1% (1294725), где 6,76% (8752341­) — это платаные CMS, а 1,7487% (22640) узкоспециализированные.

**Создание сайта**

***HTML разметка***

Сначала прописываем версию HTML:

*<DOCTYPE html>*

Дальше необходимо начать наш код:

*<html>*

*</html>*

В этих тегах будет находится весь остальной код.

Создаем тег, для работы с данными документа:

*<head>*

*<head>*

В нем делаем для начала несколько тегов:

*<meta charset=”utf-8”>* - он сообщает браузеру стандарт кодирования символов нашего документа

*<title>Главная</title>* - называет вкладку документа для браузера

Воспользуемся методом связывания css файлов с нашими документами, предварительно создав папку, где будет наш css файл:

*<link rel=”stylesheet” href=”Css\style.css”>*

Далее сделаем иконку, которая будет находиться рядом с названием вкладки, а иконку положим в созданную папку:

*<link rel=”icon” href=”Img\logo.svg.svg” type image/x-icon>*

Также создаем тег для шапки нашего сайта:

*<header>*

*</header>*

В нем мы делаем навигационное меню:

*<nav>*

*<ul class=”hr”> -задаем класс для css, для дальнейших изменений*

*<li><img class="logo" src="Img\logo.svg.svg" width="350px" height="350px"></li> - наш*

*логотип*

*<li><a href="index.html">Главная</a></li>*

*<li><a href="about.html">О проекте</a></li> - другие страницы*

*<li><a href="Code.html">О вебразработке</a></li>*

*</ul>*

*</nav>*

И сделаем разделяющий знак в виде линии:

*<hr/>*

Необходим тег, где будет находится все содержимое документа:

*<body>*

*</body>*

Сделаем еще подвал для нашего сайта, а перед ним пропишем разделяющую линию:

*<hr/>*

*<footer>*

*</footer>*

Весь его контент возьмем в блок:

*<div>*

*<p class="Download">Скачать проект с разрешением .docx:<a href="Files\Проект А. В. Баранова 'Работа и создание сайта'.docx " download=""> Проект А. В. Баранова 'Работа и создание сайта'</a> </p>*

*<p class="Me">Автор проекта и разработчик сайта: Баранов Александр Владимирович</p>*

*</div>*

Создадим два отдельных класса для двух параграфов, чтобы они были отцентрированы по левому и правому краю страницы.

И так у нас получилась общая составляющая для наших документов:

*<!DOCTYPE html>*

*<html>*

*<head>*

*<meta charset="utf-8">*

*<title>Главная</title>*

*<link rel="stylesheet" href="Css\style.css">*

*<link rel="icon" href="Img\logo.svg.svg" type="image/x-icon">*

*</head>*

*<header>*

*<nav>*

*<ul class="hr">*

*<li><img class="logo" src="Img\logo.svg.svg" width="350px" height="350px"></li>*

*<li><a href="index.html">Главная</a></li>*

*<li><a href="about.html">О проекте</a></li>*

*<li><a href="Code.html">О вебразработке</a></li>*

*<li><a href="Site.html">О сайте</a></li>*

*</ul>*

*</nav><hr/>*

*</header>*

*<body>*

*</body>*

*<hr/>*

*<footer>*

*<div>*

*<p class="Download">Скачать проект с разрешением .docx:<a href="Files\Проект А. В. Баранова 'Работа и создание сайта'.docx " download=""> Проект А. В. Баранова 'Работа и создание сайта'</a> </p>*

*<p class="Me">Автор проекта и разработчик сайта: Баранов Александр Владимирович</p>*

*</div>*

*</footer>*

*</html>*

Также стоит объяснить, что:

*<p>текст</p>* - создает текст в параграфе

*<nav></nav>* - определяющий тег для навигации

Атрибуты width и height – задают ширину и высоту для элемента соответственно

*<a href=”ссылка”></a>* - создает ссылку, а текст между этим тегом является гипертекстом

Далее будут рассмотрены теги и атрибуты, использующиеся на всех документах этого сайта:

*<h3></h3>* или *<h4></h4>* - заголовочные теги, которые идут со значениями от 1 до 6, означающие важность заголовка, где 1-самый важный, а 6-наимение важный уровень, относительно этого тега, данном случаем они имеют значения 3 и4.

*</br>* -переход на следующую строку

*<ul>*

*<li></li>* - создает маркированный список

*</ul>*

*<ol>*

*<li></li>*  - создает нумерованный список

*</ol>*

*&lt;* и *&gt;* - заменяют для HTML символы < и > соответственно

*<table>*

*<tr>*

*<th></th>- с*оздает таблицу, где tr – строка, а th это заголовок

*<td></td>*  столбца, и td это отдельная ячейка

*</tr>*

*</table>*

***CSS файл***

Рассмотрим свойства для тегов нашего сайта:

*background-color* – задает цвет для заднего фона

*border* – задает границы для элемента

*border-collapse* – задает стиль для границ, например двойные линии

*border-left* – задает границу слева для элемента

*border-left-style* – задает стиль для границы слева

*border-radius* – определяет скругление границ элемента

*color* – задает цвет для элемента

*display* – определяет тип отображения, на нашем примере это inline в навигационном меню, являющимся списком

*float* - указывает, что текущий элемент должен быть изъят из обычного потока и прижат к левой или правой стороне родительского элемента, причем текс и *inline* будут обтекать данный элемент

*font-family* - определяет шрифт для текста

*font-size* – определяет размер текста

*font-style* – форматирует текст

*font-weight* – задает «жирность» для теста

*height* – задает высоту элемента

*line-height* – задает межстроковой отступ

*margin* – внешний отступ со всех четырех сторон

*margin-right* – внешний отступ с права

*padding* – внутренний отступ со всех четырех сторон

*text-align* – задает центрирование текста

*text-decoration* – задает параметры подчеркивания текста

*transform* - позволяет сдвигать, масштабировать и наклонять элемент

*width* – задает ширину для элемента

Также посмотрим на псевдокласс *hover*, он принимает определенные свойства при наведении на элемент мышкой

**Результат**

Используя основы веб-разработки и справочные данные, любой человек может написать собственный сайт. Как пример хочу предоставить два сайта, первый разработан на CMS «WordPress», а второй на языках программирования HTML и CSS.

***Список литературы***

a5.ru

developer.mozilla.org

dimox.name

domenus.ru

habr.com

hexlet.io

hostinger.com.ua

htmlbook.ru

linkbuilder.su

mozzila.org

reg.ru

vepp.ru

wikipedia.org

wordpress.com

«Как работает DNS» Статья 28.10.21 Марины Суворовой

«Как работает браузер» Блог Алекса Надалина

**Приложения**

Основные международные домены

.biz — сайты для бизнеса

.com — сайты коммерческих организаций

.edu — образовательные ресурсы

.gov — сайты правительственных учреждений

.info — информационные сайты

.mil — сайты военных организаций

.movie — официальные сайты фильмов

.museum — сайты музеев

.net — провайдеры и сетевые организации

.org — некоммерческие организации

.pro — для профессионалов

.tv — сайты телевизионных компаний

.name — персональный домен

.club — домен для клуба по общим интересам

.xyz — универсальный домен

.center — чем-то схоже с зоной .club, но шире

Стандарт HTML

*<заголовок>HTML</заголовок>*

*<полужирный>HyperText Markup Language</полужирный> <курсив>(* *язык разметки гипертекста)</курсив> - язык разметки гипертекста.*

Структура HTML-документа

*<!DOCTYPE html>*

*<html>*

*<head>*

*<meta charset="utf-8">*

*<title>Страница<title>*

*</head>*

*<body>*

*<h1>...</h1>*

*<p>...</p>*

*</body>*

*</html>*

Связанные стили

*<link rel=”stylesheet” href=”mysite.css”>*

Глобальные стили

*<style>*

*H1 {*

*font-size: 120%;*

*font-family: Verdana, Arial, Helvetica, sans-serif;*

*color: #333366;*

*}*

*</style>*

Внутренние стили

*<p style=”font-size: 120%; font-family: monospace; color: #cd66cc”>Пример текста</p>*

Импорт CSS

*<style>*

*@import url(“style/header.css”);*

*</style>*

|  |  |  |
| --- | --- | --- |
| Имя селектора | Что выбирает | Пример |
| *Селектор элемента (иногда называемый селектором тега или типа)* | *Все HTML-элемент(ы) указанного типа.* | *p Выбирает <p>* |
| *ID селектор* | *Элемент на странице с указанным ID на данной HTML. Лучше всего использовать один элемент для каждого ID (и конечно один ID для каждого элемента), даже если вам разрешено использовать один и тот же ID для нескольких элементов.* | *#my-id Выбирает <p id="my-id"> или <a id="my-id">* |
| *Селектор класса* | *Элемент(ы) на странице с указанным классом (множество экземпляров класса может объявляться на странице).* | *.my-class Выбирает <p class="my-class"> и <a class="my-class">* |
| *Селектор атрибута* | *Элемент(ы) на странице с указанным атрибутом.* | *img[src] Выбирает <img src="myimage.png"> но не <img>* |
| *Селектор псевдокласса* | *Указанные элемент(ы), но только в случае определённого состояния, например, при наведении курсора.* | *a:hover Выбирает <a>, но только тогда, когда указатель мыши наведён на ссылку.* |

Сайт на HTML + CSS

https://sord-zeus.github.io/Project-School/

Сайт на CMS

https://pojectmy.wordpress.com/