# МБОУ Средняя общеобразовательная школа № 15 г. Йошкар-Олы

Создание Web-сайта:

Первые механические устройства вычисления

По предмету: Информатика

Работа ученика 9 ”Г” класса

Быстров Алексей Павлович

Руководитель проекта: Короткова Екатерина Михайловна

Работа допущена к защите «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 201\_\_\_\_г.

Подпись руководителя проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

Оценка проекта «\_\_\_\_\_»

Город Йошкар-Ола- 2021г

Содержание

Введение \*Страницы\*

Часть 1.HTML 5-8

Часть 2. CSS

Список литературы \*Страницы\*

Приложение \*Страницы

1

Введение

Влияние глобальной компьютерной сети Интернет на современный мир

не имеет аналогов в истории. На сегодняшний день происходит расцвет Интернета, он проникает во все сферы человеческой жизни, это нечто большее, чем просто маркетинговая кампания, это основа новой философии и новой деловой стратегии.

Интернет и WWW прочно вошли в нашу жизнь, и нам уже трудно представить свою деятельность без них. WWW – это огромный набор гипертекстовых документов, которые благодаря Интернету доступны всем у кого есть какой либо выход в него.

Создание Web- сайтов является одной из важнейших технологий разработки ресурсов Интернет. Web-сайт - это информационный ресурс, состоящий из связанных между собой гипертекстовых документов (Web-страниц), размещенный на Web-сервере и имеющий индивидуальный адрес.

Однако перечисленные достоинства приобретаются автоматически, а появляются лишь в результате обдуманного, обоснованного подхода к их созданию.

Актуальность темы

Интернет оказывает огромное влияния на формирование духовно-нравственных ценностей у молодежи, вовлекая их в активную деятельность.

Тема нашего проекты «Первые механические устройства вычесления», в форме Web страницы, о семи первых механических устройствах вычесления, которые были изобретены или найдены.

Цель проекта – создать сайт, рассказать принципы его создания и с помощью бесплатных хостингов зарегистрировать его в Интернет.

Задачи проекта:

3

Основная часть

Для начала мы должны понять что такое сайт.

Если говорить человеческим, а не техническим языком, то сайт – это место во всемирной паутине интернет, в котором содержатся данные, предоставляемые для всеобщего обозрения владельцем, естественно, с определенной целью. Любой информационный ресурс доступен по конкретному адресу в интернете, который мы вводим в адресную строку – www.vknew.com или ru или net и тому подобное.

В современном мире уже не обойтись без интернета, и сложно представить себе поиск любой информации другим способом.

Сайт – это представительство человека в интернете, посетителями которого могут стать люди всей планеты, у которых есть доступ в интернет. Как известно, каждый человек пришел в этот мир, чтобы принести пользу людям (как бы громко это не звучало). С помощью сайта можно принести пользу большему количеству людей, которые заинтересованы в получении информации, чем без его использования.

И так , нами был разработан сайт в стиле и простоте сравнимый с Википедией ([https://ru.wikipedia.org](https://ru.wikipedia.org/)) и этот сайт относительно прост для людей занимающихся разработкой сайтов, но для меня, новичка, который не разу не имел с этим дело – было не просто. Для создания даже такого простого сайта пришлось изучить большое количество информации и обучающих видеороликов по данной теме.

Дальше мы углубиться уже в само строение сайта.

HTML

Первый и пожалуй основной язык это- HTML( *HyperText Markup Language*  — «язык [гипертекстовой](https://ru.wikipedia.org/wiki/Гипертекст) разметки»*)* стандартизированный язык разметки веб-страниц во [Всемирной паутине](https://ru.wikipedia.org/wiki/Всемирная_паутина). Код HTML интерпретируется  [браузерами](https://ru.wikipedia.org/wiki/Браузер), полученная в результате интерпретации страница отображается на экране монитора компьютера или мобильного устройства.

Итак, HTML состоит из тегов.

Тег, теги - именованная метка, читается более правильное название — дескриптор. В SGML (HTML, WML, AmigaGuide, языках семейства XML) — элемент языка разметки гипертекста.

Любая HTML страничка начинается с тега <html> и заканчивается тегом </html> .

Тег <html> является контейнером, который заключает в себе все содержимое веб-страницы, включая теги <head> и <body>. Открывающий и закрывающий теги <html> в документе необязательны, но хороший стиль диктует непременное их использование.

Тег <head> элемент представляет собой контейнер для метаданных (данных о данных) . Метаданные - это данные о документе HTML. Метаданные не отображаются. Метаданные обычно определяют заголовок документа, набор символов, стили, сценарии и другую метаинформацию.

Тег <head> является одним из главных тегов HTML документа на ряду с тегом <body>

Тег <body> предназначен для хранения содержания 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 и т.д.

Все теги делятся на родительские и дочерние, так два данных тега являются дочерними для тега <html> и родителями для других тегов, о который щас и пойдет речь. (смотри приложение 1)

<Meta>

Тег <meta> определяет метатеги, которые используются для хранения информации предназначенной для браузеров и поисковых систем. Размещен в контейнере <head>.

В наших страницах этот тег используется по одному разу, для каждой отдельной страницы, чтобы задать кодировку символов.

<title>

Определяет заголовок документа. Элемент <title> не является частью документа и не показывается напрямую на веб-странице. В операционной системе Windows текст заголовка отображается в левом верхнем углу окна браузера. Допускается использовать только один тег <title> на документ и размещать его в контейнере <head>.

<link>

Устанавливает связь с внешним документом вроде файла со стилями или со шрифтами. В отличие от тега <a>(Сылка), тег <link> размещается всегда внутри контейнера <head> и не создает ссылку.

В нашем случае с помощью этого тега мы подключили стили в файл HTML. Плюс подключения стилей с помощью данного тега, его можно внедрить сразу в несколько файлов и изменение в одном исходном файле стилей отобразиться на каждой странице, куда этот файл стилей подключен. Что экономит время при работе с несколькими HTML страницами.

<header>

Тег <header> задает «шапку» сайта или раздела, в которой обычно располагается заголовок. В нашем сайте в такой «шапке» находится название и ссылка на переход на исходную страницу.

<a>

Тег <a> является одним из важных элементов HTML и предназначен для создания ссылок. В зависимости от присутствия атрибутов name или href тег <a> устанавливает ссылку или якорь.

Якорем называется закладка внутри страницы, которую можно указать в качестве цели ссылки. При использовании ссылки, которая указывает на якорь, происходит переход к закладке внутри веб-страницы.

В нашем сайте тег используется в основном на странице локации по сайту. А в станицах статей он используется только в ссылке назад.

<h1,2,3,4,5,6>

В HTML сущевствует 6 заголовком разного уровня от 1 до 6. <h1> (заголовок первого уровня) обычно обозначается важнейший текст на странице, 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. Теги <h1>,...,<h6> относятся к блочным элементам, они всегда начинаются с новой строки, а после них другие элементы отображаются на следующей строке. Кроме того, перед заголовком и после него добавляется пустое пространство.

<p>

Определяет текстовый абзац. Тег <p> является блочным элементом, всегда начинается с новой строки, абзацы текста идущие друг за другом разделяются между собой. Разделение можно управлять с помощью стилей. Если закрывающего тега нет, считается, что конец абзаца совпадает с началом следующего блочного элемента.

Весь текст в HTML сайта выделен тегом <p> , для отделение его от текста между собой .

<Img>

Тег <img> предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG. Адрес файла с картинкой задаётся через атрибут src , пишется путь от картинки к HTML файул. Если необходимо, то рисунок можно сделать ссылкой на другой файл, поместив тег <img> в контейнер <a>.

Также в внутри тега задаются атрибуты width(ширина), height(высота) и align (выравнивание картинки, например, align=center)

<footer>

Тег <footer> задаёт «подвал» сайта или раздела, в нём может располагаться имя автора, дата документа, контактная и правовая информация.

В нашем случае, в подвале указано кому принадлежит авторство работы.

CSS

Второй используемый язык это- CSS ( Cascading Style Sheets «каскадные таблицы стилей»)  [формальный язык](https://ru.wikipedia.org/wiki/Формальный_язык) описания внешнего вида документа, написанного с использованием [языка разметки](https://ru.wikipedia.org/wiki/Язык_разметки). В нашем случае языке HTML.

Простыми словами с помощью CSS задается внешний вид текста, т.е. задается шрифт текста, размер,  расположения отдельных блоков.

CSS может подключаются к HTML документам четырьмя различными способами.

CSS может находиться, как в самом текстовом документе, внешний вид которого он описывают, так и во внешние файлах , имеющих расширение  .css (index.css).

* когда описание стилей находится в отдельном файле, оно может быть подключено к документу посредством элемента <link>, включённого в элемент <head> (смотри приложение рис.2)
* когда файл стилей размещается отдельно от родительского документа, он может быть подключён к документу инструкцией @import в элементе<style>(смотри приложение 3 )
* когда стили описаны внутри документа, они могут быть включены в элемент <style>, который, включается в элемент <head>(смотри в приложении 4)
* когда стили описаны в теле документа, они могут располагаться в атрибутах отдельного элемента>(смотри в приложении 5)

Но использовался в нашем сайте один , подлючение с помощью ссылки <link>, т.е. первый способ.

Правила и свойства CSS отличаются от правил HTML.

Стили задаются с помощью двух частей: селектора и блок объявлений, в котором заключаются свойство селектора и его значение

**{**

**селектор**, **селектор** :

**свойство: значение;**

**свойство: значение;**

**свойство: значение;**

}

#### Виды селекторов

Селектор - это часть CSS-правила, которая сообщает браузеру, к какому элементу (или элементам) веб-страницы применить стиль.

Различных видов селекторов сущевствует более 10 штук, но мы вам расскажем только об 3 основных и более применимых.

##### Универсальный селектор

\* {

**margin**: 0;

**padding**: 0;

}

Универсальный селектор задаются знаком \* и он подразумевает ,что указанные в нем свойства задаются на все элементы HTML файла.

##### Селектор тегов

**p** {

**font-family**: arial, helvetica, **sans-serif**;

}

Данный селектор применяет свойства и значения только для заданного в нем тега.

##### Селектор классов

.**note** {

**color**: **red**;

**background-color**: **yellow**;

**font-weight**: **bold**;

}

C его помощью задают свойства только одному, из множества таких же тегов документа ,например, если мы зададим селектор тега <h1>, то он применится на все теги <h1> этого документа, но если нам понадобится изменить стиль только одного из тегов <h1>.  На помощь приходят селекторы классов. Всё что вам необходимо сделать, это создать стиль, где селектор – придуманное вами имя класса, но также нам надо вписать это имя в тег который мы хотим изменить нашего HTML файла.

Основные свойства CSS

display

display-многоцелевое свойство, которое определяет, как элемент должен быть показан в документе.

Значения данного свойства очень обширны и насчитывают более пятнадцати видов, но мы вам покажем только пару основных.

## 1.[Значение none](https://learn.javascript.ru/display" \l "znachenie-none)

Самое простое значение. Элемент не показывается, вообще. Как будто его и нет.

## 2. [Значение block](https://learn.javascript.ru/display" \l "znachenie-block)

* Блочные элементы располагаются один над другим, вертикально (если нет особых свойств позиционирования, например float).
* Блок стремится расшириться на всю доступную ширину. Можно указать ширину и высоту явно.
* Это значение display многие элементы имеют по умолчанию: <div>, заголовок <h1>, параграф <p>.

## 3. [Значение inline](https://learn.javascript.ru/display" \l "znachenie-inline)

* Элементы располагаются на той же строке, последовательно.
* Ширина и высота элемента определяются по содержимому. Поменять их нельзя.

Wight и height

Такие же названия мы уже встречали в HTML, данные свойства отвечают за ширину и длину выбранного нами блока

В этих свойствах задается всего одно значения это px, то есть расстояние отведенное для элемента в пикселях. Используется, например, к тегу img или к любому тексту, чтобы определить его границы.

Margin и padding

Padding - yстанавливает значение полей вокруг содержимого элемента. Полем называется расстояние от внутреннего края рамки элемента до воображаемого прямоугольника, ограничивающего его содержимое. Свойство padding позволяет задать величину поля сразу для всех сторон элемента или определить ее только для указанных сторон.

**Отступы:**

* **padding-top:** *значение;*
* **padding-right:** *значение;*
* **padding-bottom:** *значение;*
* **padding-left:** *значение;*

Mahgin- yстанавливает величину отступа от каждого края элемента. Отступом является пространство от границы текущего элемента до внутренней границы его родительского элемента. Свойство margin позволяет задать величину отступа сразу для всех сторон элемента или определить ее только для указанных сторон.

**Поля:**

* **margin-top:** *значение;*
* **margin-right:** *значение;*
* **margin-bottom:** *значение;*
* **margin-left:** *значение;*

Color

Исходя из названия можно понять за, что отвечает данное свойство Color - это сокращение английского слова colour, что в переводе означает цвет. Так и само свойство Color определяет цвет элемента, например:

**p** {

color:red

}

Здесь задается селектор абзаца, тега <p>, красный. Это один из способов задать цвет элемента, всего их сущевствует четыре.

### Шестнадцатеричные цвета (hex)

### Шестнадцатеричная система счисления (hexadecimal, hex) основывается на числе 16. Для записи шестнадцатеричного значения используется 16 символов: арабские цифры от 0 до 9 и первые буквы латинского алфавита (A, B, C, D, E, F). Цвет в шестнадцатеричном формате записывается в виде трех двузначных чисел от 00 до FF (перед ними обязательно ставится символ решетки #), что соответствует трем компонентам: Red, Green, Blue (цветовая модель RGB). Иными словами, запись цвета можно представить как #RRGGBB, то есть цвет указывается в виде чисел и символов, где первая пара символов определяет уровень красного, вторая – уровень зеленого, третья – уровень синего цвета. Результирующий цвет является сочетанием этих трех цветов.

#### Сокращенная запись hex-цветов

Некоторые шестнадцатеричные значения цветов из Шестнадцаричной системы(hex) можно записать сокращенно. Для этого следует превратить запись вида #RRGGBB в #RGB. Это можно сделать в том случае, когда в hex-числе имеется три пары одинаковых символов.

Примеры сокращенной записи hex-цветов: (смотри приложение 5)

.

*3.*Цветовая модель RGB

Второй способ указания цвета в CSS – использовать десятичные значения RGB (Red, Blue, Green). Для этого необходимо записать после свойства color ключевое слово rgb, а затем в скобках и через запятую – три числа в диапазоне от 0 до 255, каждое из которых означает интенсивность красного, зеленого и синего цветов (r, g, b). Чем больше число, тем более интенсивен цвет.

p {

color: rgb(0, 255, 0);

}

### *4*.Цветовая модель RGBA

Задать цвет в формате RGBA можно почти так же, как и в RGB. Отличие RGBA от RGB заключается в наличии альфа-канала, который отвечает за прозрачность цвета. Задается прозрачность с помощью числа в диапазоне от 0 до 1, где 0 соответствует полной прозрачности, а 1, наоборот, означает полную непрозрачность. Промежуточные значения вроде 0.5 позволяют задать полупрозрачный вид (допускается сокращенная запись, без нуля, но с точкой – .5). Например, чтобы сделать текст цветным и слегка прозрачным, нужно записать после свойства color ключевое слово rgba и значение цвета:

p {

color: rgba(94, 81, 3, .9);

}

Font-size

Определяет размер шрифта элемента. Размер может быть установлен несколькими способами. Набор значений (xx-small, x-small, small, medium, large, x-large, xx-large) задает размер, который называется абсолютным. По правде говоря, они не совсем абсолютны, поскольку зависят от настроек браузера и операционной системы.

Другой набор значений (larger, smaller) устанавливает относительные размеры шрифта. Поскольку размер унаследован от родительского элемента, эти относительные размеры применяются к родительскому элементу, чтобы определить размер шрифта текущего элемента.

В конечном итоге, размер шрифта сильно зависит от значения свойства font-size у родителя элемента.

Также разрешается использовать любые допустимые единицы CSS: em (высота шрифта элемента), ex (высота символа х), пункты (pt), пикселы (px), проценты (%) и др. За 100% берется размер шрифта родительского элемента. Отрицательные значения не допускаются.

text-align

Определяет горизонтальное выравнивание текста. У данного тега есть большое количество значений:

*center*

Выравнивание текста по центру. Текст помещается по центру горизонтали окна браузера или контейнера, где расположен текстовый блок. Строки текста словно нанизываются на невидимую ось, которая проходит по центру веб-страницы. Подобный способ выравнивания активно используется в заголовках и различных подписях, вроде подрисуночных, он придает официальный и солидный вид оформлению текста. Во всех других случаях выравнивание по центру применяется редко по той причине, что читать большой объем такого текста неудобно.

*justify*

Выравнивание по ширине, что означает одновременное выравнивание по левому и правому краю. Чтобы произвести это действие браузер в этом случае добавляет пробелы между словами.

*left*

Выравнивание текста по левому краю. В этом случае строки текста выравнивается по левому краю, а правый край располагается «лесенкой». Такой способ выравнивания является наиболее популярным на сайтах, поскольку позволяет пользователю легко отыскивать взглядом новую строку и комфортно читать большой текст.

*right*

Выравнивание текста по правому краю. Этот способ выравнивания выступает в роли антагониста предыдущему типу. А именно, строки текста равняются по правому краю, а левый остается «рваным». Из-за того, что левый край не выровнен, а именно с него начинается чтение новых строк, такой текст читать труднее, чем, если бы он был выровнен по левому краю. Поэтому выравнивание по правому краю применяется обычно для коротких заголовков объемом не более трех строк. Мы не рассматриваем специфичные сайты, где текст приходится читать справа налево, там возможно подобный способ выравнивания и пригодится.

*auto*

Не изменяет положение элемента.

*inherit*

Наследует значение родителя.

*start*

Аналогично значению *left*, если текст идёт слева направо и *right*, когда текст идёт справа налево.

*end*

Аналогично значению *right*, если текст идёт слева направо и *left*, когда текст идёт справа налево.

Заключение

В данной работе рассмотрены актуальные вопросы разработки и создания современного Web-сайта на платформе бесплатного хостинга Hostinger.ru.

При этом мною были решены следующие задачи:

– изучил с современные Интернет-технологии и их использование в настоящей разработке;

– изучил программный инструментарий, применяемый для разработки и

создания Web-сайтов;

– ознакомился с основными правилами и рекомендациями по разработке и созданию Web-сайтов и неукоснительно следовал им на практике;

– научился определять структуру Web-страниц;

– научился выбирать стратегии разработки и создания Web-сайта.

В результате проведенных работ на базе выбранных технологий был создан действующий, пополняемый современный Web-сайт.

Список литературы:

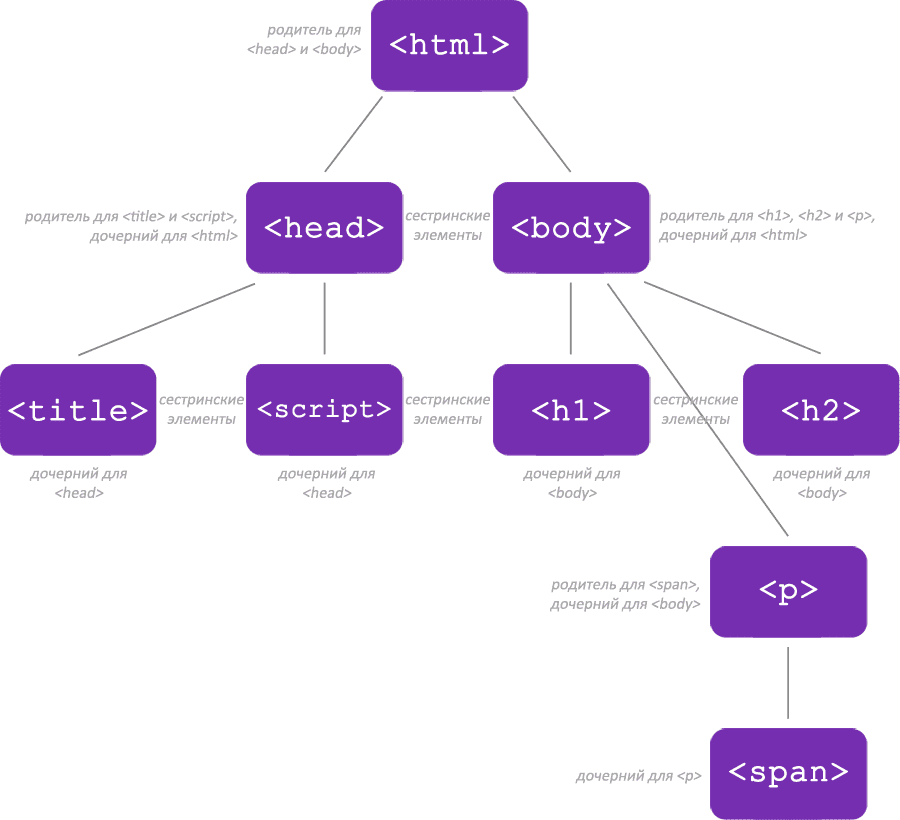
Википедия (https://ru.wikipedia.org/wiki/)

Техническая документация по HTML (http://htmlbook.ru/html)

Техническая документация по CSS (http://htmlbook.ru/css)

16

Приложение

1. 

2.

<!DOCTYPE html>

<**html**>

<**head**>

.....

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

</**head**>

<**body**>

.....

</**body**>

</**html**>

3.

<!DOCTYPE html>

<**html**>

<**head**>

.....

<**style** media="all">

@**import** **url**(**style**.**css**);

</**style**>

</**head**>

</**html**>

4.

<!DOCTYPE html>

<**html**>

<**head**>

.....

<**style**>

**body** {

**color**: **red**;

}

</**style**>

</**head**>

<**body**>

.....

</**body**>

</**html**>

5.

<!DOCTYPE>

<**html**>

<**head**>

.....

</**head**>

<**body**>

<**p** style="font-size: 20px; color: green; font-family: arial, helvetica, sans-serif">

.....

</**p**>

</**body**>

</**html**>

6.

