**Цели проекта:**

* Цель нашего проекта: Реализация генератора задач по выполнению и анализу простых алгоритмов ЕГЭ по информатике на языке JavaScript.

**Задачи:**

1. Изучить HTML и CSS.
2. Ознакомиться с языком программирования JavaScript.
3. Перебрать и выделить различные подвиды задач.
4. Научиться решать данные задачи.
5. Написать генератор задач с использованием скриптов JavaScript.

**Теоретический материал.**

**HTML:**

**HTML** (от английского **HyperText Markup Language**) — это язык разметки документов для просмотра веб-страниц. Он используется для того, чтобы дать браузеру понять, как нужно отображать загруженный сайт. Каждый сайт в сети интернет использует для отображения информации язык HTML.

В этом языке всё состоит это **тегов** — это команды, которые браузер преобразует в визуальные объекты. Огромный плюс данного языка в том, что для реализации простых задач не нужны какие-либо программы, достаточно создать блокнот файл на рабочем столе и можно приступать к работе.

Основной задачей данного языка является структурирование информации на странице браузера. Для этого существуют разные способы:

* Создание таблиц;
* Различные виды списков;
* Параграфы;
* Добавление в документ изображения;
* Использование заголовков;
* Создание различных форм;

Но не всё можно изменить во внешнем виде страницы, например:

* Нельзя поменять цвет текста;
* Нельзя установить фоновое изображение;
* Нельзя менять шрифт;

**CSS**

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

Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы от описания внешнего вида этой веб-страницы. При подобном разделении работа над версткой и дизайном сайта может вестись параллельно.

**Благодаря использованию CSS:**

* увеличивается скорость загрузки сайта и уменьшается размер самого кода сайта
* более понятное, удобное и быстрое управление внешним видом сайта (достаточно отредактировать один файл со стилем и оформление нужных документов сразу же поменяется, вместо того чтобы модифицировать десятки html)
* стили CSS имеют гораздо больше возможностей по оформлению элементов веб-страниц.
* можно разнообразить внешний вид сайта для разных устройств вывода: монитора, принтера, смартфона, КПК и др.

Для добавления стилей на веб-страницу существует 4 способа, которые различаются своими возможностями и назначением:

1. Связанные стили. При данном методе создается отдельный файл как правило, с расширением css, в котором и описываются все параметры внешнего вида вашего сайта.

2. Глобальные стили. При использовании глобальных стилей свойства CSS описываются в самом документе и располагаются между тегами <style> и </style>, которые, в свою очередь, располагаются в этом документе между тегами <head> и </head>. То есть здесь не создается отдельного файла с расширением css, а все его содержимое храниться между тегами <head> и </head> в коде сайта.

3. Внутренние стили. В данном способе свойства стилей располагаться в теле какого-то отдельного тега (посредством его атрибута style). Все правила этой таблицы стилей действуют только на содержимое этого тега.

4. Импорт CSS. В данном способе свойства стилей подключены к веб-документу посредством директивы [@import](https://vk.com/id14455831), располагающейся в этом документе между тегами <style> и </style> , которые, в свою очередь, располагаются в этом документе между тегами <head> и </head>. Этот метод допускается использовать совместно со связанными или глобальными стилями, но никак не с внутренними стилями.

**JavaScript**

**JavaScript** — язык сценариев, или скриптов. Скрипт представляет собой программный код — набор инструкций, который не требует предварительной обработки перед запуском. Код JavaScript интерпретируется движком браузера во время загрузки веб-страницы.

Скрипты распространяются и выполняются, как простой текст. Им не нужна специальная подготовка или компиляция для запуска.

Действия браузера, при получении скрипта:

* Движок читает текст скрипта.
* Затем он преобразует скрипт в машинный язык.
* После этого машинный код запускается и работает достаточно быстро.

Движок применяет оптимизации на каждом этапе. Он даже просматривает скомпилированный скрипт во время его работы, анализируя проходящие через него данные, и применяет оптимизации к машинному коду, полагаясь на полученные знания. В результате скрипты работают очень быстро.

В браузере JavaScript может манипулировать веб-страницами, взаимодействовать с пользователем и веб-сервером.

Так, например, JavaScript позволяет:

* Добавлять новый HTML-код на страницу, изменять существующее содержимое, модифицировать стили.
* Реагировать на действия пользователя, щелчки мыши, перемещения указателя, нажатия клавиш.
* Отправлять сетевые запросы на удалённые сервера, скачивать и загружать файлы.
* Получать и устанавливать куки, задавать вопросы посетителю, показывать сообщения.
* Запоминать данные на стороне клиента.

Но нужно пронимать, что возможности JavaScript ограничены ради безопасности пользователя. Цель заключается в предотвращении доступа недобросовестной веб-страницы к личной информации или нанесения ущерба данным пользователя.

Примеры таких ограничений включают в себя:

* JavaScript на веб-странице не может читать/записывать произвольные файлы на жёстком диске, копировать их или запускать программы. Он не имеет прямого доступа к системным функциям ОС.
* Различные окна/вкладки не знают друг о друге. Иногда одно окно, используя JavaScript, открывает другое окно. Но даже в этом случае JavaScript с одной страницы не имеет доступа к другой, если они пришли с разных сайтов
* JavaScript может легко взаимодействовать с сервером, с которого пришла текущая страница. Но его способность получать данные с других сайтов/доменов ограничена. Хотя это возможно в принципе, для чего требуется явное согласие с удалённой стороной. Опять же, это ограничение безопасности.