



**МИНОБРНАУКИ РОССИИ**  
**федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**«Московский государственный технологический университет**  
**«СТАНКИН»**  
**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

---

Институт автоматизации и робототехники  
Кафедра компьютерных систем управления

Образовательная программа 15.03.04  
«Автоматизация технологических процессов и производств»

Дисциплина «Основы системного программного  
обеспечения»

**Отчет**  
**по лабораторной работе №3**

Выполнил:  
студент  
гр.АДБ-20-08

\_\_\_\_\_  
(дата)

\_\_\_\_\_  
(подпись)

Денисенко М.М.

Принял:  
к.т.н., доцент

\_\_\_\_\_  
(дата)

\_\_\_\_\_  
(подпись)

Ковалев И.А.

**Москва 2023**

## Цель работы:

Научиться создавать простые страницы сайтов используя Язык HTML JavaScript и CSS. Разобраться как происходит процесс разработки простых диалогов.

## Теоретические сведения

### Язык HTML

Язык гипертекстовой разметки (HyperText Markup Language — HTML, рис.1), основной строительный блок веб-страниц, используется для создания и визуального представления веб-страниц. Он определяет содержание страницы, но не её функциональность.



Рис. 1. – HyperText Markup Language.

HTML добавляет разметку в обычный текст. Гипертекст содержит ссылки, которыми веб-страницы связываются друг с другом, делая Всемирную паутину тем, чем она является сегодня. HTML поддерживает как изображения, так и другой медиаконтент. С помощью HTML каждый может создать статический, а также динамический сайт. HTML является языком, описывающим структуру и семантику содержимого веб-документа. Контент веб-страницы размечен с помощью тегов, представляющих HTML-элементы. Примерами таких элементов являются `<img>`, `<title>`, `<p>`, `<div>`, `<picture>` и так далее. Эти элементы формируют строительные блоки для любого веб-сайта.

Большая часть элементов HTML представляет собой элементы, сформированные тегами открытия и тегами закрытия ("`<p>text here</p>`") этих элементов, как и в других языках форматирования. Не каждый элемент должен сопровождаться тегом закрытия, например – есть теги переноса строки `<br>` и тому подобные.

### Объектная Модель Документа (DOM)

Объектная Модель Документа (DOM, рис.2) – это программный интерфейс (API) для HTML и XML документов. DOM предоставляет структурированное представление документа и определяет то, как эта структура может быть доступна из программ, которые могут изменять содержимое, стиль и структуру документа. Представление DOM состоит из структурированной группы узлов и объектов, которые имеют свойства и методы. По существу DOM соединяет веб-страницу с языками описания сценариев либо языками программирования.

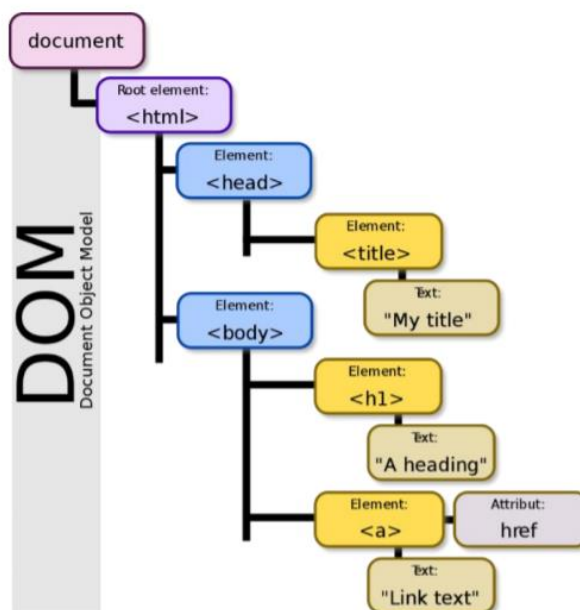


Рис. 2. Модель DOM.

## JavaScript

JavaScript изначально создавался для того, чтобы сделать web-странички «живыми». Программы на этом языке называются скриптами. В браузере они подключаются напрямую к HTML и, как только загружается страничка – тут же выполняются.

Программы на JavaScript – обычный текст. Они не требуют какой-то специальной подготовки. В этом плане JavaScript сильно отличается от другого языка, который называется Java.

Для выполнения программ, не важно на каком языке, существуют два способа:

«компиляция» и «интерпретация»:

- Компиляция – это когда исходный код программы, при помощи специального инструмента, другой программы, которая называется «компилятор», преобразуется в другой язык, как правило – в машинный код. Этот машинный код затем распространяется и запускается. При этом исходный код программы остаётся у разработчика.
- Интерпретация – это когда исходный код программы получает другой инструмент, который называют «интерпретатор», и выполняет его «как есть». При этом распространяется именно сам исходный код (скрипт). Этот подход применяется в браузерах для JavaScript.

Современные интерпретаторы перед выполнением преобразуют JavaScript в машинный код или близко к нему, оптимизируют, а уже затем выполняют. И даже во время выполнения стараются оптимизировать. Поэтому JavaScript работает очень быстро.

Во все основные браузеры встроен интерпретатор JavaScript, именно поэтому они могут выполнять скрипты на странице. Но, разумеется, JavaScript можно использовать не только в браузере. Это полноценный язык, программы на котором можно запускать и на сервере, и даже в стиральной машинке, если в ней установлен соответствующий интерпретатор.

Современный JavaScript – это «безопасный» язык программирования общего назначения. Он не предоставляет низкоуровневых средств работы с памятью, процессором, так как изначально был ориентирован на браузеры, в которых это не требуется. Что же касается остальных возможностей – они зависят от окружения, в котором запущен JavaScript. В браузере JavaScript умеет делать всё, что относится к манипуляции со страницей, взаимодействию с посетителем и, в какой-то мере, с сервером:

- Создавать новые HTML-теги, удалять существующие, менять стили элементов, прятать, показывать элементы и т.п.
- Реагировать на действия посетителя, обрабатывать клики мыши, перемещения курсора, нажатия на клавиатуру и т.п.
- Посылать запросы на сервер и загружать данные без перезагрузки страницы (эта технология называется "AJAX").
- Получать и устанавливать cookie, запрашивать данные, выводить сообщения и др.

JavaScript – быстрый и мощный язык, но браузер накладывает на его исполнение некоторые ограничения. Это сделано для безопасности пользователей, чтобы злоумышленник не мог с помощью JavaScript получить личные данные или как-то навредить компьютеру пользователя.

Этих ограничений нет там, где JavaScript используется вне браузера, например на сервере. Кроме того, современные браузеры предоставляют свои механизмы по установке плагинов и расширений, которые обладают расширенными возможностями, но требуют специальных действий по установке от пользователя.

Большинство возможностей JavaScript в браузере ограничено текущим окном и страницей. Список ограничений JavaScript представлен ниже:

- JavaScript не может читать/записывать произвольные файлы на жесткий диск, копировать их или вызывать программы. Он не имеет прямого доступа к операционной системе (рис.3).
- Современные браузеры могут работать с файлами, но эта возможность ограничена специально выделенной директорией – «песочницей». Возможности по доступу к устройствам также прорабатываются в современных стандартах и частично доступны в некоторых браузерах.

- JavaScript, работающий в одной вкладке, не может общаться с другими вкладками и окнами, за исключением случая, когда он сам открыл это окно или несколько вкладок из одного источника (одинаковый домен, порт, протокол).
- Есть способы это обойти, и они раскрыты в учебнике, но они требуют специального кода на оба документа, которые находятся в разных вкладках или окнах. Без него, из соображений безопасности, залезть из одной вкладки в другую при помощи JavaScript нельзя.
- Из JavaScript можно легко посылать запросы на сервер, с которого пришла страница. Запрос на другой домен тоже возможен, но менее удобен, т. к. и здесь есть ограничения безопасности.

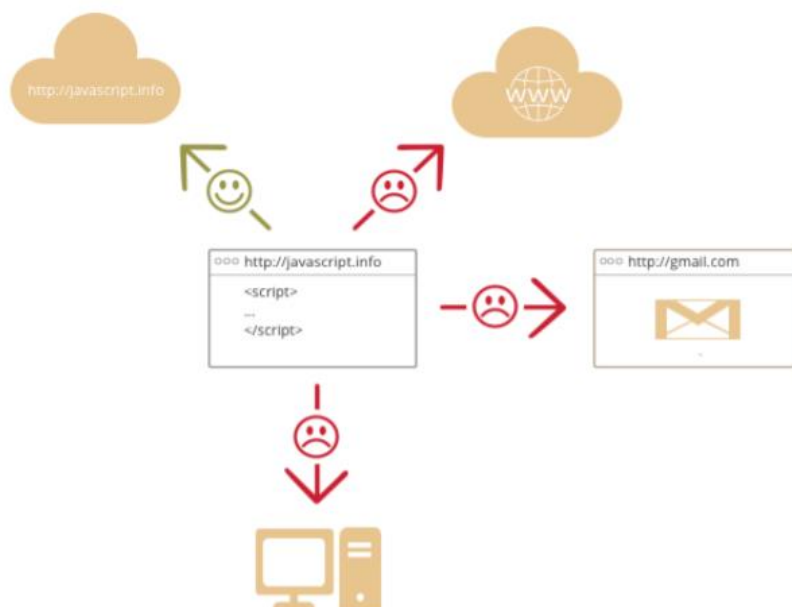


Рис. 3. Ограничения JavaScript.

## CSS

CSS (Каскадные Таблицы Стилей, рис.4) позволяют создавать стилизованные и отформатированные веб-страницы, состоящие из слоев, блоков и так далее.

Рассмотрим, что такое CSS и как браузеры превращают HTML в (DOM), как CSS применяется к различным частям DOM, некоторые базовые примеры синтаксиса и подключение CSS к нашей веб-странице.



Рис. 4. – CSS.

Документ обычно является текстовым файлом, который структурирован при помощи языка разметки. HTML это самый распространенный язык разметки, но вы можете столкнуться и с другими (SVG или XML).

Представление документа пользователю значит его конвертацию в понятную для пользователя форму. Браузеры, такие как Firefox, Chrome или Internet Explorer, предназначены для представления документов визуальнo, например, на экране компьютера, проектор или принтер.

Рассмотрим влияние CSS на HTML. Браузер применяет CSS правила к документу чтобы описать, как он будет отображаться. CSS-правила формируются из:

- Набора свойств, которые имеют значения, устанавливающие, как будет отображаться содержимое (HTML). Например, можно сделать, чтобы ширина элемента равнялась 50% ширины родительского элемента и его фон был красным.
- Селектор, который выбирает (англ. selects) элемент/элементы, к нужно применить измененные значения. Например, можно применить это CSS-правило ко всем параграфам в HTML-документе.

Набор правил CSS, содержащихся в таблице стилей (stylesheet), определяет, как должна выглядеть веб-страница.

## Практическая часть

### Индивидуальное задание 1

Функция - Нахождение максимума двух чисел.  $\max(a,b)$

```
<script>
function findMax(){
    var n1 = document.getElementById('n1').value;
    var n2 = document.getElementById('n2').value;

    var maxN = Math.max(n1, n2);

    document.getElementById('otvet').textContent = "Максимальное число: " + maxN;
}
</script>
```

file:///C:/Users/maksi/Desktop/lr%203.1.html

Какое из чисел максимально?

Найдем максимум двух чисел

300 -первое число  
500 -второе число

Вычислить Максимальное число

**Максимальное число: 500**

## Индивидуальное задание 2

Объявите две переменные: admin и name.  
Запишите в name строку свое имя.  
Скопируйте значение из name в admin.  
Выведите admin.

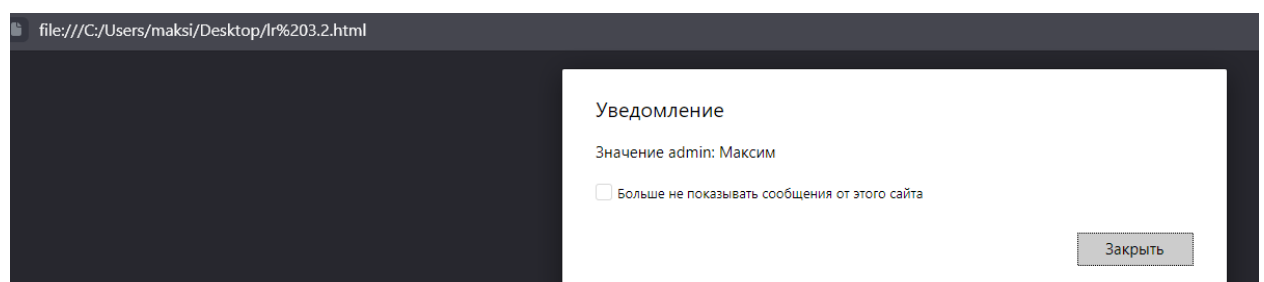
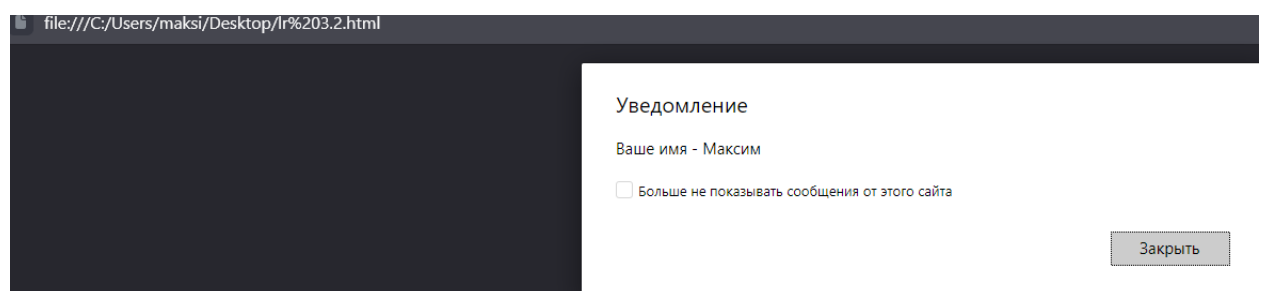
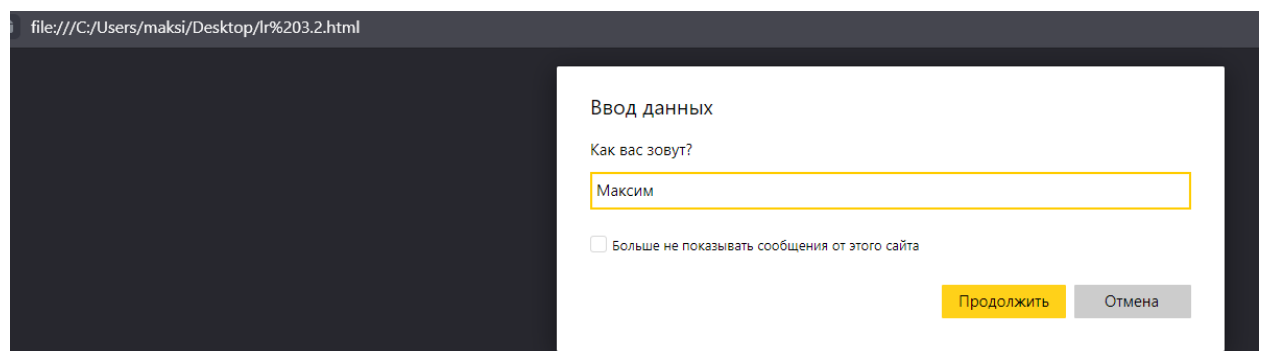
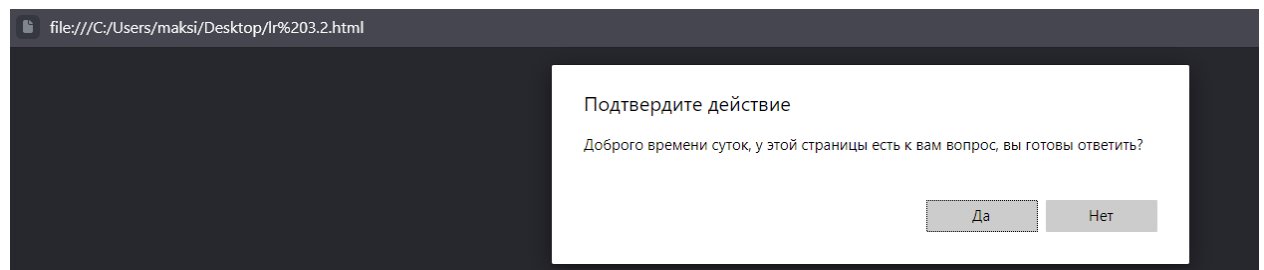
```
<script>
  if (confirm('Доброго времени суток, у этой страницы есть к вам вопрос, вы готовы ответить?')){
    var name = prompt('Как вас зовут?', 'Введите сюда ваше имя');
    alert("Ваше имя - " + name);
  }
  alert("Спасибо, что уделите время");

  var admin, name;

  //Записываем в name строку с именем
  name = "Максим";

  //Копируем значение из name в admin
  admin = name;

  //Выводим admin
  alert("Значение admin: " + admin);
</script>
```



## Индивидуальное задание 3

Напишите скрипт, который будет спрашивать ваш возраст и после ввода возраста уточнять его.

```
<script>
var year;
while (true){
    year = prompt('Сколько вам лет?', '');

    if (year != null)
    {
        if (confirm('Вам точно ' + year + ' лет?'))
        {
            break;
        }
    }
    else
    {
        alert('Пожалуйста, введите ваш возраст.');
```

```
    }
    alert("Спасибо, что уделите время");
}
//script>
```

file:///C:/Users/maksi/Desktop/lr%203.3.html

Ввод данных

Сколько вам лет?

Продолжить

Отмена

file:///C:/Users/maksi/Desktop/lr%203.3.html

Подтвердите действие

Вам точно 20 лет?

☐ Больше не показывать сообщения от этого сайта

Да

Нет

file:///C:/Users/maksi/Desktop/lr%203.3.html

Уведомление

Спасибо, что уделите время

☐ Больше не показывать сообщения от этого сайта

Закреть



## Индивидуальное задание 4

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styles/style.css">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@300&family=Raleway:wght@600&display=swap" rel="stylesheet">
  <script src="script.js"></script>
  <title>Max Denisenko</title>
</head>
<body>
  <header>
    <h5>Макс Денисенко</h5>
  </header>
  <main>
    <div class="info">
      <h1>Здравствуй, студент.</h1>
      
    </div>
    <div class="aboutYou">
      <h2>Введи фамилию ниже</h2>
      <input type="text" placeholder="Твоя фамилия">
      <button onclick="setSurname()" id="b1">сохранить</button>
      <h2>Поменяй цвет фона</h2>
      <button onclick="setWallColor()" id="b2">поменять</button>
    </div>
    <div class="anketa">
      <div class="getInfo">
        <h2>Твое резюме</h2>
        <input type="text" placeholder="Введи имя">
        <input type="text" placeholder="Введи фамилию">
        <input type="text" placeholder="Введи возраст">
        <select>
          <option>Выбери пол</option>
          <option>Мужской</option>
          <option>Женский</option>
        </select>
        <textarea name="" id="" cols="15" rows="5" placeholder="Напиши о себе"></textarea>
        <button onclick="createResume()"><b>Отправить</b></button>
        <button onclick="deleteResume()"><b>Сброс</b></button>
      </div>
      <div class="resume"></div>
    </div>
  </main>
</body>
</html>
```

Вывод: Я научился создавать простые страницы сайтов используя Язык HTML JavaScript и CSS и разобрался как происходит процесс разработки простых диалогов.