Министерство образован­­ия Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

Отчет

по лабораторной работе №3

«*JavaScript*»

Вариант № 18

|  |  |
| --- | --- |
| Выполнил ст. гр. 820604  Климович Н.В.  Проверил преп. каф. ИТАС  Гончаревич А. Л. |  |

Минск 2021

СОДЕРЖАНИЕ

[1 Цель работы 3](#_Toc26004)

[2 Теоретическая часть 4](#_Toc4500)

[3 Ход работы 7](#_Toc30027)

[Заключение 22](#_Toc30066)

# ЦЕЛЬ РАБОТЫ

Изучение структуры языка *JavaScript*, работа с операторами *for* и *while*, использование объектов *JavaScript*: *Math*, *Date*, *String*, работа с формами, исследование методов для вывода информации, введение в *DHTML*.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

*JavaScript* является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, *JavaScript* имеет ряд свойств, присущих функциональным языкам – функции как объекты 1-го класса, объекты как списки, анонимные функции, замыкания – что придаёт языку дополнительную гибкость.

Простейшие элементы, которыми оперируют остальные, более сложные элементы, называют литералами. В языке *JavaScript* предусмотрены следующие типы литералов: целого типа, вещественные, логического типа, строковые литералы.

К объектам относятся данные основных системных типов, например, числовые значения и массивы. Свойствами являются значения, принадлежащие объектам, например, *Math*.*PI* – значение математической константы π объекта *Math*.

Объект – это какой-либо предмет, в том числе и составляющие компьютерного виртуального мира. Что касается *JavaScript*, его объекты находятся внутри браузера. Кроме этого, непосредственно в языке *JavaScript* имеется возможность создавать новые объекты и потом их использовать в каждом подходящем случае. При объектно-ориентированном программировании первоочередное внимание уделяется объектам, которые могут производить некоторые манипуляции, а не логическим правилам и вычислительным процедурам, необходимым для таких манипуляций. Примером объекта служит объект *Window* – окно при запуске браузера.

Метод – это действия, которые выполняет объект. В реальном мире каждый объект тоже обладает некоторым набором методов: машины ездят, собаки лают, доллар покупается и т.д. Примерами методов объекта являются *open*(), *close*() и *click*(): открытие окон, закрытие окон и нажатие кнопок.

У каждого объекта могут иметься какие-то свойства (*properties*): машины можно отличить по колесам, а у собаки имеется шерсть. Что касается *JavaScript*, то, например, у такого объекта языка, как браузер, имеется название и номер версии.

Операторы – мощный инструмент языков программирования и языка *JavaScript* в том числе. Любое выражение состоит из операторов. Существуют различные виды операторов: (оператор *if* – оператор организации ветвлений, операторы организации цикла (оператор *do* … *while*, оператор *while*, оператор *for*, оператор *for* … *in*) и другие.

Объекты *Math*, *Date* и *String* являются стандартными встроенными объектами языка *JavaScript*, поддерживаемыми всеми браузерами.

Объект *Math* содержит набор стандартных математических функций. Для использования свойств и методов этого объекта не требуется предварительного создания объекта. События в этом объекте не используются. В состав объекта *Math* входят шесть свойств-констант: *Math*.*E*, *Math*.*PI*, *Math*.*LN*10, *Math*.*LN*2, *Math*.*SQRT*1*\_*2, *Math*.*SQRT*2. Эти константы можно использовать без предварительного определения в программе. *JavaScript* выдает для них приближенные значения, выдаваемые используемым браузером.

Объект *Date* также, как и *Math* является стандартным объектом языка *JavaScript*. Этот объект обеспечивает программиста большим количеством методов, связанных с датами и временем (свойств у этого объекта нет, события не используются). В языке *JavaScript* дата исчисляется количеством миллисекунд, прошедших с 1-го января 1970 года. Для работы с методами объекта *Date* необходимо помнить, что основой всех операций с датами и временем являются системные дата и время, установленные на вашем компьютере.

Объект *String* представляет последовательность символов в строке и позволяет изменять и форматировать текстовую строку, выделять внутри неё часть строки, разбить строку на отдельные строки и занести их в массив и т.д. Строка – это конечная последовательность символов некоторого алфавита. Алфавит – это конечное множество символов. Пустая строка – строка, не содержащая ни одного символа. В *JavaScript* строковым литералом (строковой переменной, просто строкой или, наконец, объектом *String*) считается любой текст (набор символов используемого алфавита), заключенный в одинарные или двойные кавычки.

Формы используются для ввода пользователем данных, в том числе в интерактивном режиме, для последующей обработки этих данных программой-приложением. Для интерактивного взаимодействия с компьютером посредством формы используются понятия события и обработчика события, а обработка данных осуществляется с помощью понятия функции, в том числе встроенной в качестве метода в тот или иной объект. В рамках объектной модели документа форме в языке *HTML* соответствует дескриптор <*FORM*> … </*FORM*> с некоторыми атрибутами, а в языке *JavaScript* – объект *window*.*document*.*form*.

Элементы формы отображаются на вэб-странице посредством специальных *HTML*-тэгов, называемых элементами управления. Каждый элемент управления предназначен для передачи полученных от пользователя данных соответствующему приложению (возможно и обратное). Современные языки программирования, также, как и *JavaScript*, имеют встроенные механизмы (интерфейс) передачи данных между элементами управления и приложением.

*DHTML* (динамический *HTML*) – это набор средств, которые позволяют создавать более интерактивные *Web*-страницы без увеличения загрузки сервера. Другими словами, определенные действия посетителя ведут к изменениям внешнего вида и содержания страницы без обращения к серверу.

*DHTML* построен на объектной модели документа (*Document* *Object* *Model*, *DOM*), которая расширяет традиционный статический *HTML*-документ. *DOM* обеспечивает динамический доступ к содержимому документа, его структуре и стилям. В *DOM* каждый элемент *Web*-страницы является объектом, который можно изменять. *DOM* не определяет новых тегов и атрибутов, а просто обеспечивает возможность программного управления всеми тегами, атрибутами и каскадными листами стилей (*CSS*).

# 3 ХОД РАБОТЫ

Рассмотрим практическую реализацию инструментов, приведенной в цели данной работы. На рисунке 1 приведена начальная страница, на которой возможна демонстрация работы этих инструментов.

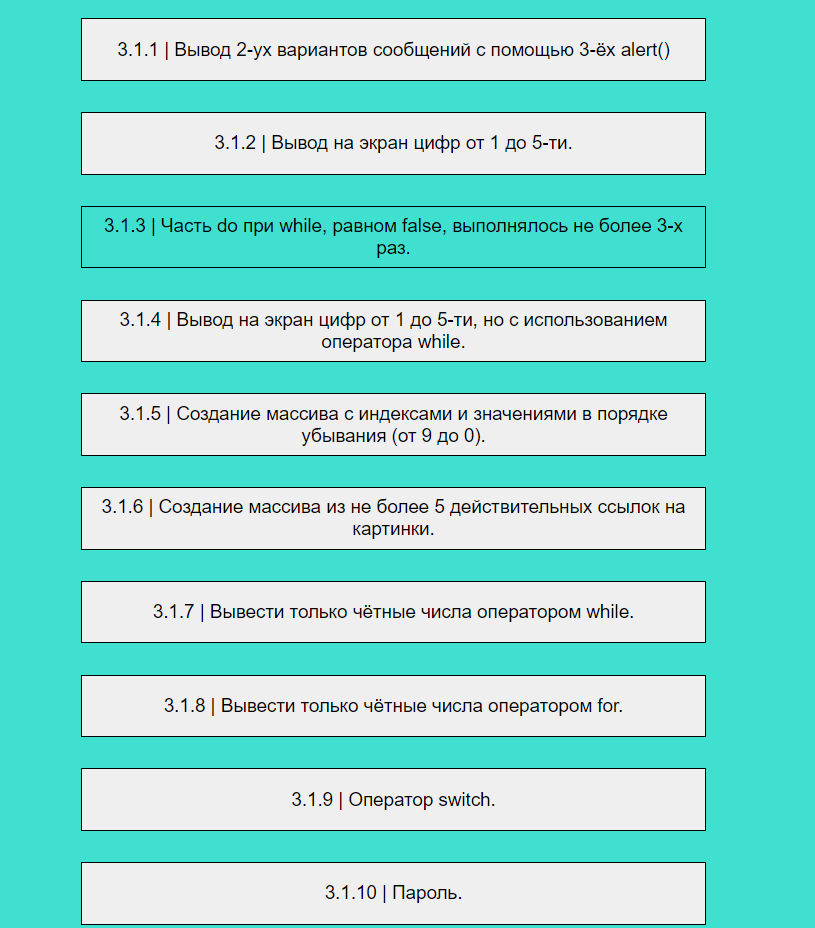


Рисунок 3.1 – Начальная страница

При нажатии на кнопку 3.1.1 происходит вывод двух вариантов сообщений с использованием трех операторов *alert*, что показано на рисунках 3.2, 3.3.

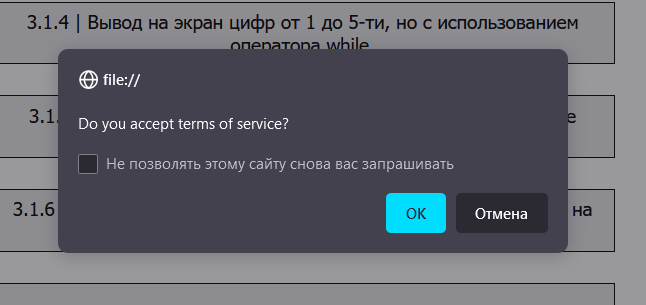


Рисунок 3.2 – Первое сообщение

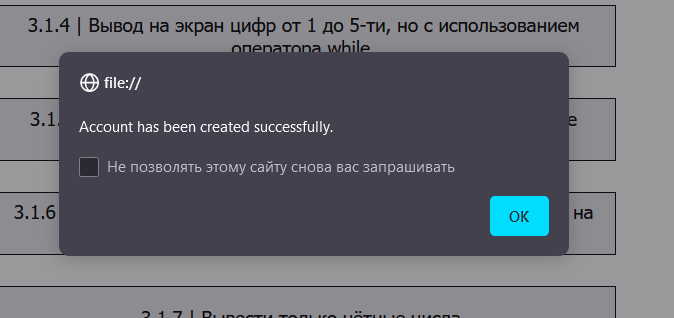


Рисунок 3.3 – Второе сообщение

При нажатии на кнопку 3.1.2 происходит последовательный вывод цифр от одного до пяти на экран с использованием оператора *for*. Фрагмент вывода показан на рисунке 3.4.

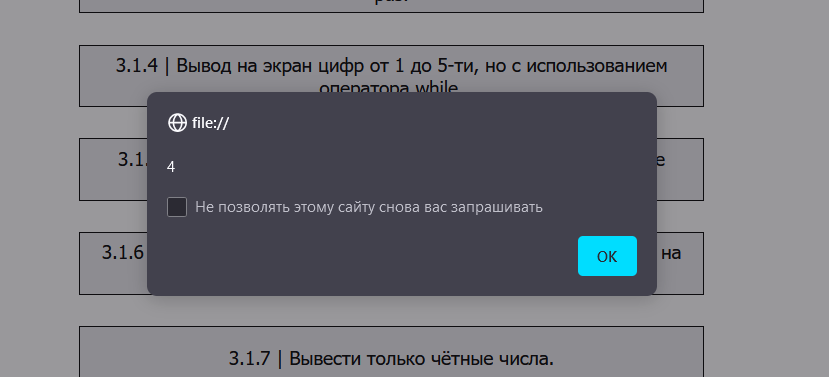


Рисунок 3.4 – Фрагмент вывода цифр от одного до пяти

с использованием оператора *for*

При нажатии на кнопку 3.1.3 будет запрошен ввод имени и фамилии, при этом будет дано три попытки, чтобы ввести данные. После окончания попыток появится сообщение о том, что попытки закончились. Процесс показан на рисунках 3.5, 3.6.

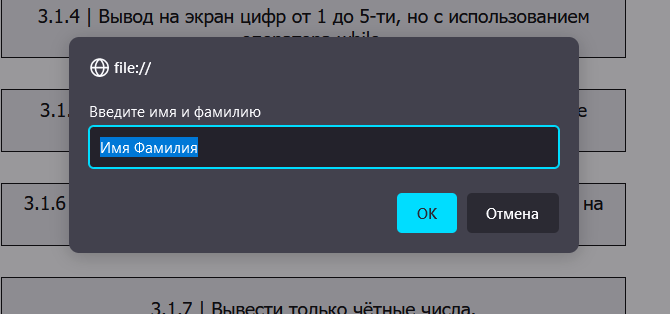


Рисунок 3.5 – Ввод имени и фамилии

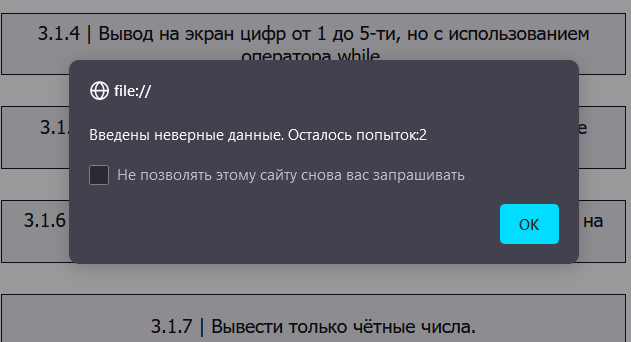


Рисунок 3.6 – Вывод количества оставшихся попыток

При нажатии на кнопку 3.1.4 происходит последовательный вывод цифр от одного до пяти на экран с использованием оператора *while*. Фрагмент вывода показан на рисунке 3.7.

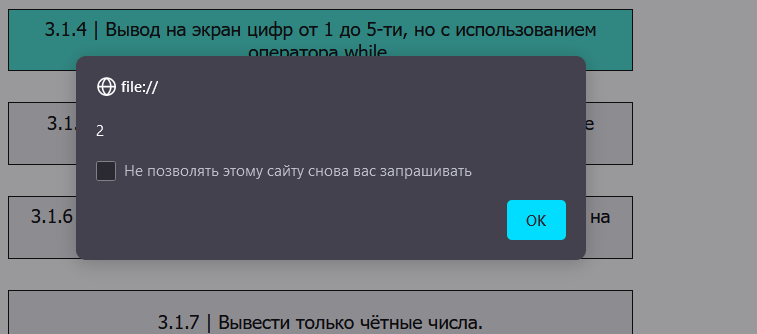


Рисунок 3.7 – Фрагмент вывода цифр от одного до пяти

с использованием оператора *while*

При нажатии на кнопку 3.1.5 происходит создание массива с индексами и значениями в порядке убывания от девяти до нуля. Вывод показан на рисунке 3.8.

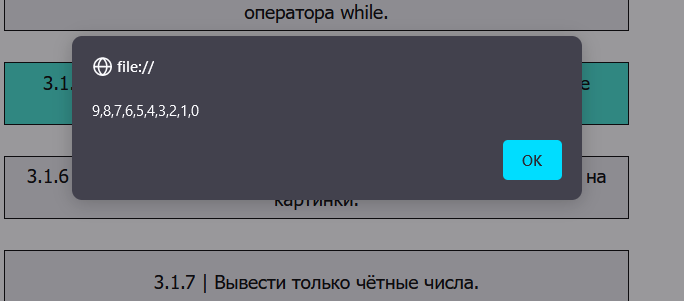


Рисунок 3.8 – Вывод массива с индексами и значениями

в порядке убывания от девяти до нуля

При нажатии на кнопку 3.1.6 происходит создание массива из пяти ссылок на существующие картинки, вывод которого можно увидеть на рисунке 3.9.

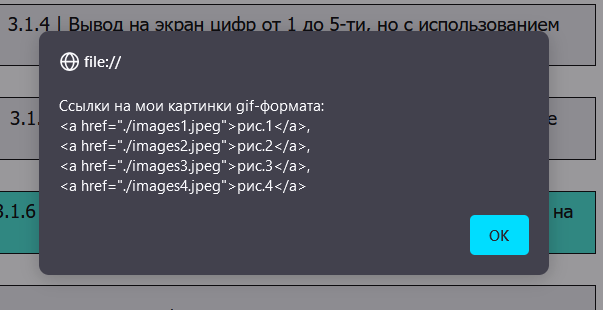


Рисунок 3.9 – Вывод массива из пяти ссылок

на существующие картинки

При нажатии на кнопку 3.1.7 происходит последовательный вывод четных чисел от одного до 10 с использованием оператора *whil*e, фрагмент которого продемонстрирован на рисунке 3.10.

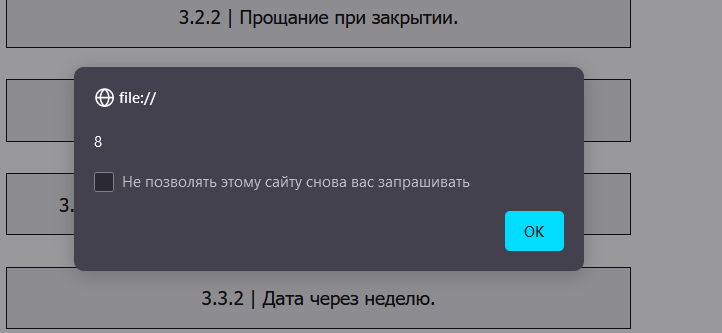


Рисунок 3.10 – Фрагмент вывода четных чисел от одного

до 10 с использованием оператора *whil*e

При нажатии на кнопку 3.1.8 происходит последовательный вывод четных чисел от одного до 10 с использованием оператора *for*, фрагмент которого продемонстрирован на рисунке 3.11.

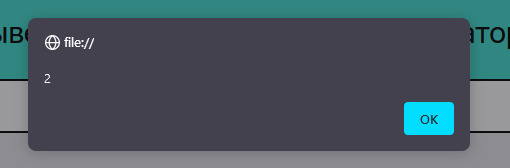


Рисунок 3.11 – Фрагмент вывода четных чисел от одного

до 10 с использованием оператора *for*

При нажатии на кнопку 3.1.9 диалоговое окно запрашивает у пользователя ввод латинской буквы от *a* до *e*. В случае правильного ввода появляется сообщение о введенной букве, а в случае неправильного выводиться сообщение о том, что буква введена не из того диапазона. Процесс показан на рисунках 3.12 – 3.15.

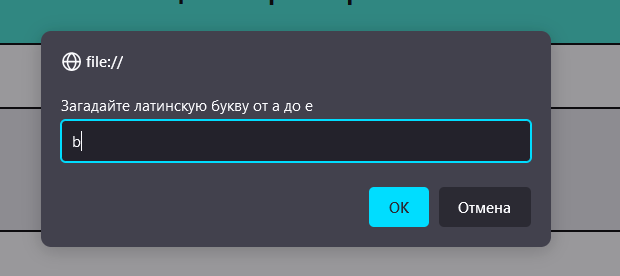


Рисунок 3.12 – Ввод правильной буквы

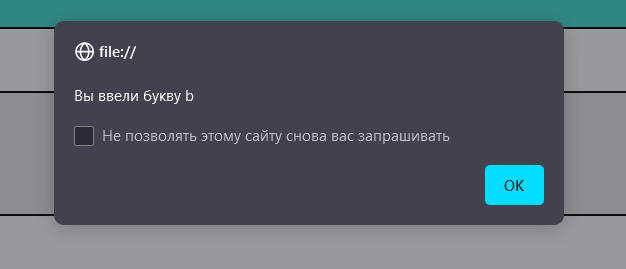


Рисунок 3.13 – Сообщение при вводе правильной буквы

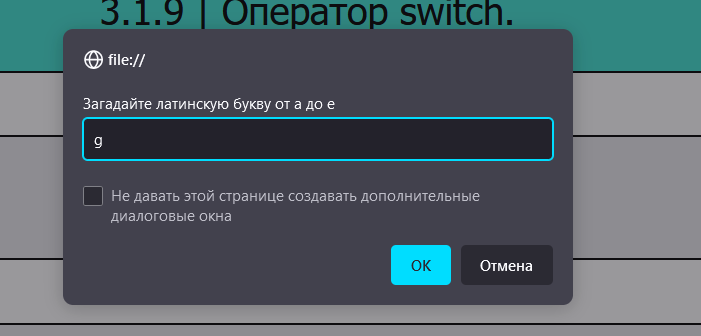


Рисунок 3.14 – Ввод неправильной буквы

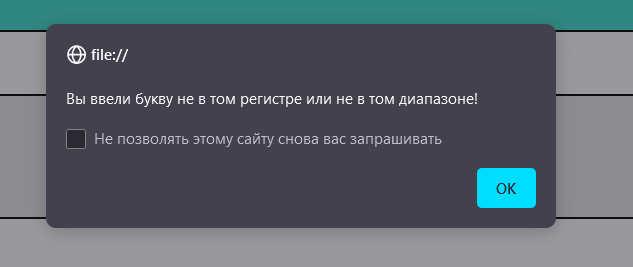


Рисунок 3.15 – Сообщение при вводе неправильной буквы

При нажатии на кнопку 3.1.10 диалоговое окно запрашивает у пользователя пароль. В случае трех неудачных попыток ввода появляется сообщение об ошибке, в случае успеха – сообщение о том, что пароль введен верно. Фрагменты ввода показаны на рисунках 3.16, 3.17.

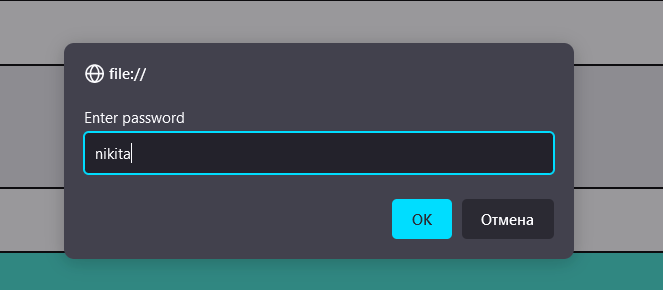


Рисунок 3.16 – Ввод пароля

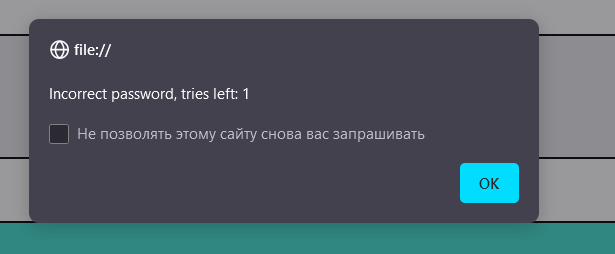


Рисунок 3.17 – Сообщение о количестве оставшихся

попыток

При наведении на кнопку 3.2.1 происходит вывод зарезервированного сообщения с помощью свойства кнопки *onMouseout*, что показано на рисунке 3.18.

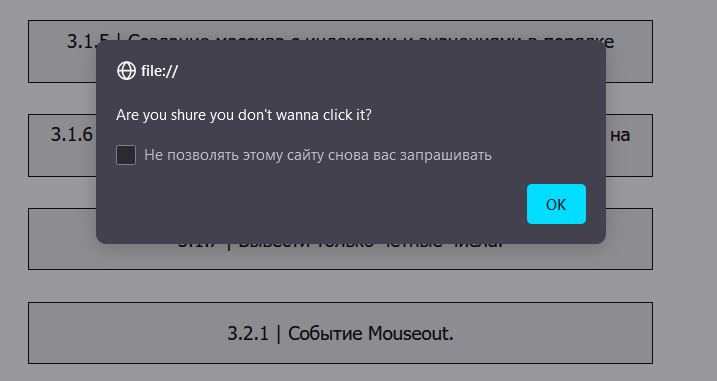


Рисунок 3.18 – Вывод зарезервированного сообщения

при наведении на кнопку

При попытке выхода происходит активация кнопки 3.2.2, которая запрашивает подтверждение выхода. Реализация показана на рисунке 3.19.

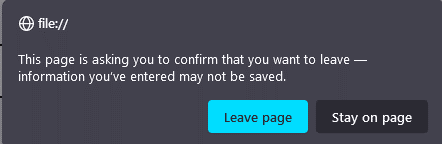


Рисунок 3.19 – Подтверждение выхода

Кнопка 3.2.3 предназначена для конвертации температуры из одной шкалы в другую. Сперва вводятся начальная и конечная шкалы, затем значение. В результате получаем новое значение температуры. Пример использования показан на рисунках 3.20 – 3.23.

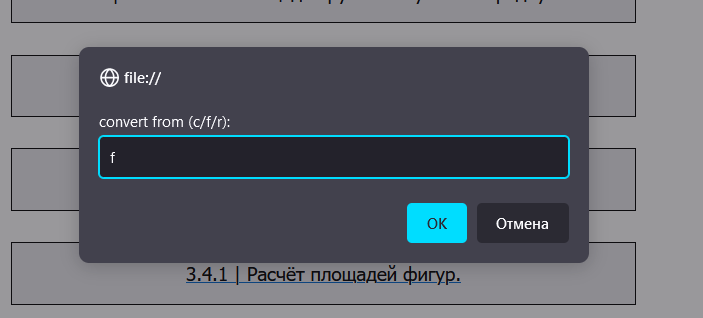


Рисунок 3.20 – Выбор шкалы для входных

данных

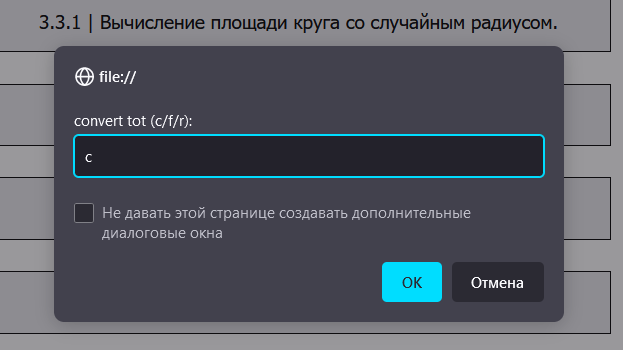


Рисунок 3.21 – Выбор шкалы для

выходных данных

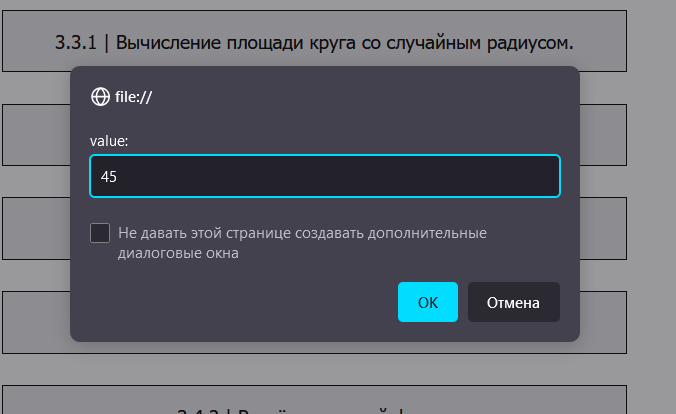


Рисунок 3.22 – Ввод значения

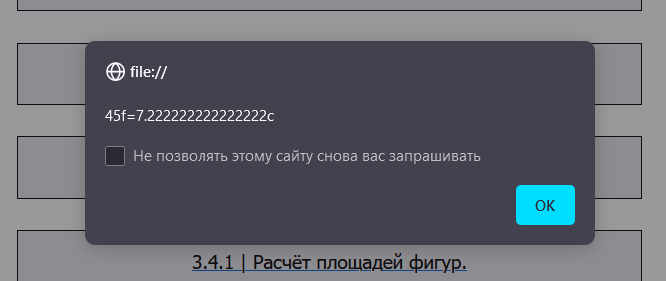


Рисунок 3.23 – Вывод результата

Кнопка 3.3.1 предназначена для вычисления случайного числа в диапазоне от одного до 100, которое затем округляется до целого и используется в качестве радиуса круга. Также происходит вычисление площади круга с этим радиусом. Пример использования продемонстрирован на рисунке 3.24.

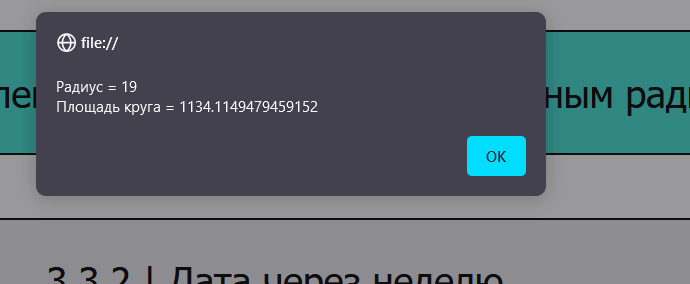


Рисунок 3.24 – Вывод радиуса и

площади круга

Кнопка 3.3.2 предназначена для расчета даты через неделю. На рисунке 3.25 показан вывод при нажатии на кнопку 6 декабря 2021 года.

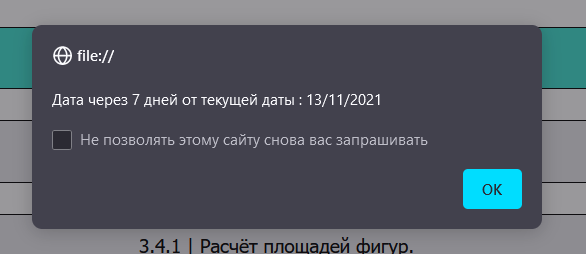


Рисунок 3.25 – Дата через неделю

Кнопка 3.3.3 предназначена для перехода на новую страницу и расчета среднемесячного дохода по введенным данным. Вводится зарплата за шесть месяцев, и при нажатии кнопки «Вычислить» происходит вывод в соответствующее поле, показанный на рисунке 3.26. Кнопка «Обновить» используется для очистки всех полей.

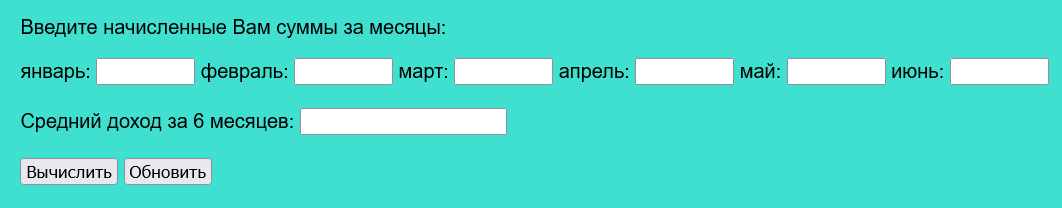


Рисунок 3.26 – Расчет среднемесячного

дохода за шесть месяцев

Кнопка 3.4.1 предназначена для перехода на новую страницу и расчета площади фигур. Из выпадающего списка происходит выбор фигуры как показано на рисунке 3.27.

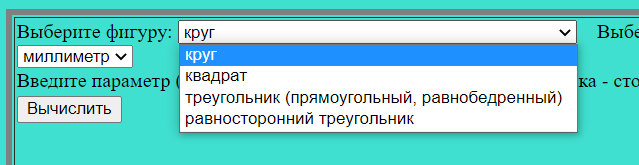


Рисунок 3.27 – Выбор фигуры

Затем следующий выпадающий список предлагает нам выбрать единицу измерения, что можно видеть на рисунке 3.28.

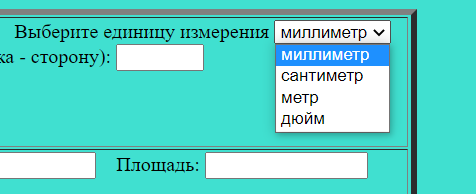


Рисунок 3.28 – Выбор единицы измерения

Затем происходит ввод стороны треугольника или квадрата, либо радиуса, если был выбран круг. Для равнобедренного треугольника также будет запрошен ввод высоты. После нажатия кнопки «Вычислить» происходит расчет площади и вывод в соответствующее поле. Пример показан на рисунке 3.29.

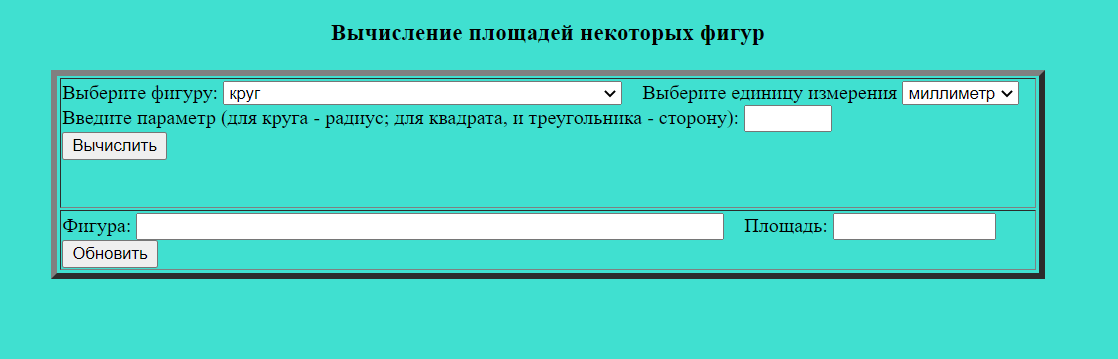


Рисунок 3.29 – Расчет площади

Кнопка 3.4.2 предназначена для перехода на новую страницу и расчета среднемесячного дохода студента Петрова по введенным данным. Вводится доход за 12 месяцев, и при нажатии кнопки «Вычислить» происходит вывод в соответствующие поля, показанный на рисунке 3.30. Кнопка «Обновить» используется для очистки всех полей.



Рисунок 3.30 – Расчет среднемесячного дохода

студента Петрова

Кнопка 3.4.3 предназначена для перехода на страницу с демонстрацией работы анимации движения слоёв, имитирующей солнечную систему. Нажав на кнопку *Faster,* можно ускорить движение планеты Земли и Луны.Результат можно увидеть на рисунке 3.31.

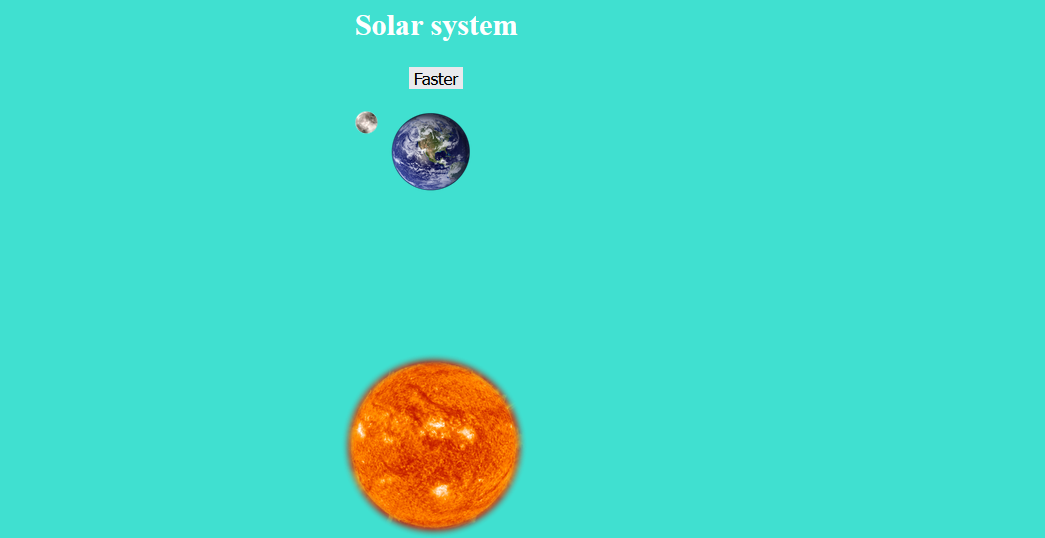


Рисунок 3.31 – Фрагмент работы анимации,

имитирующей солнечную систему

Анимированная игра реализована на рисунке 3.32. Игра представляет собой раннер, где игрок-динозавр уклоняется от птиц и кактусов. В случае столкновения игрока с одним из препятствий появляется сообщение о завершении игры.

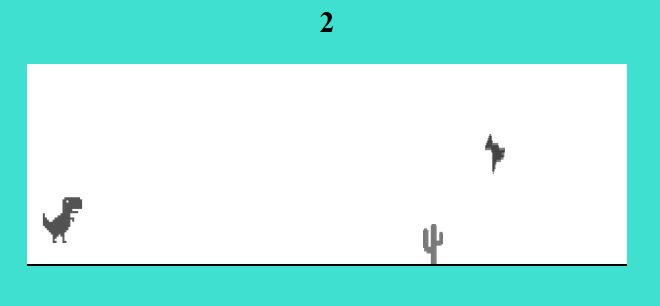


Рисунок 3.32 – Фрагмент работы анимированной игры

# ЗАКЛЮЧЕНИЕ

В результате мною была изучена структура языка *JavaScript*, работа с операторами *for* и *while*, использование объектов *JavaScript*: *Math*, *Date*, *String*, работа с формами, исследование методов для вывода информации, и введение в *DHTML*.