Министерство образования Республики Беларусь

Учреждение высшего образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

Лабораторная работа № 3 по дисциплине

ТЕХНОЛОГИИ ИНТЕРНЕТ-ПРОГРАММИРОВАНИЯ

на тему

«JavaScript»

Вариант задания 10

Выполнил студент группы 620603 Е.О. Новик

Проверил преп. каф. ИТАС А.Л. Гончаревич

Минск 2019

# 1 Цель работы

## 1.1 Описание цели работы

Целью данной лабораторной работы является продолжение знакомства с JavaScript и его синтаксисом. Внедрение JavaScript в HTML-документ, ввод/вывод информации; объекты, методы, свойства. Работа с циклами, условными операторами, математическими функциями и тегами.

1. **ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

## Описание индивидуальных заданий

* + 1. Разработать страницу, чтобы при выборе альтернативы, выводилось два варианта сообщений, но только с помощью трех операторов *alert()*.
    2. Разработать программу на языке *JavaScript* для вывода цифр от одного до пяти.
    3. Разработать программу на языке *JavaScript*, чтобы часть кода *do* при *while* равном *false* выполнялось не более трёх раз.
    4. Разработать программу на языке *JavaScript*, чтобы на экран выводились цифры от одного до пяти, но с использованием оператора ***while***.
    5. Разработать программу на языке *JavaScript*, чтобы создавался массив с индексами и значениями в порядке убывания (от девяти до нуля).
    6. Создать массив на языке *JavaScript,* чтобы он содержал до пяти действующих ссылок на картинки и (или) фотографии в *JPEG*-формате (с расширением *jpg*).
    7. Разработать программу на языке *JavaScript*, чтобы на экран выводились одни чётные числа.
    8. Разработать программу, которая принимает число от одного до 10 и прерывает цикл на соответствующем числе, следует прервать выполнение операторов *while* и *for*.
    9. Разработать программу на языке *JavaScript* с использованием оператора *switch*, в котором будут выполняться различные операторы в зависимости от того, какая буква введена в окне запроса: a, b, c, d или e.
    10. С использованием любого оператора разработать программу на языке *JavaScript*, чтобы форма с требованием ввода пароля выводилась три раза и лишь после этого загружалась бы дополнительная страница.
    11. Создать страницу с функцией, которая будет выполняться после того, как пользователь уберет курсор с гиперссылки. При этом должна загружаться страница со списком  ваших примеров и заданий. Изменить также соответствующие подсказки в операторе *Alert* и тексте ссылки.
    12. Создать страницу, при закрытии которой пользователь получит прощальное сообщение.
    13. Создать свой сценарий, с четырьмя  функциями и  соответствующими кнопками переводов: Цельсий-Реомюр, Фаренгейт-Реомюр, Реомюр-Цельсий, Реомюр-Фаренгейт.
    14. Разработать программу на языке *JavaScript* для вычисления псевдослучайного числа в промежутке от нуля до 100, которое затем, предварительно округлённое до целого, будет использовано в качестве радиуса для вычисления площади круга.
    15. Разработать программу на языке *JavaScript* , которая выполнит расчет и вывод на экран даты, которая наступит через неделю.
    16. Написать собственный сценарий для определения среднемесячного дохода за шестимесячный период с указанием данных о себе.
    17. Усложнить программу вычисления площадей фигур, введя еще одну фигуру - равносторонний треугольник, а также возможность расчета площадей в различных единицах: миллиметрах, сантиметрах, метрах, дюймах. Единица для входного параметра – метр.
    18. Добавить вычисление и вывод максимального и минимального месячных доходов и соответствующих им названий месяцев с использованием данных за год.
    19. Создать анимационный фрагмент, в котором вокруг перемещающегося справа налево изображения-центра вращается по окружности другое изображение-спутник.
    20. Разработать программу на языке *JavaScript*, чтобы на экран выдавалась информация о тэгах, имеющихся в коде примера.
    21. Разработать программу на языке *JavaScript*, чтобы сменялись изображения при наведении и отведении курсора в заданной области.
    22. Разработать программу на языке *JavaScript*, чтобы сменялись изображения через определённый промежуток времени.

1. **РЕШЕНИЕ ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ**

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.1

Для начала создадим простую стартовую страницу с гиперссылками на выполненные задания. Результат продемонстрирован на рисунке 1. С ней и будет связан файл «*index.js*», в котором будут храниться скрипты, вызываемые на этой странице.

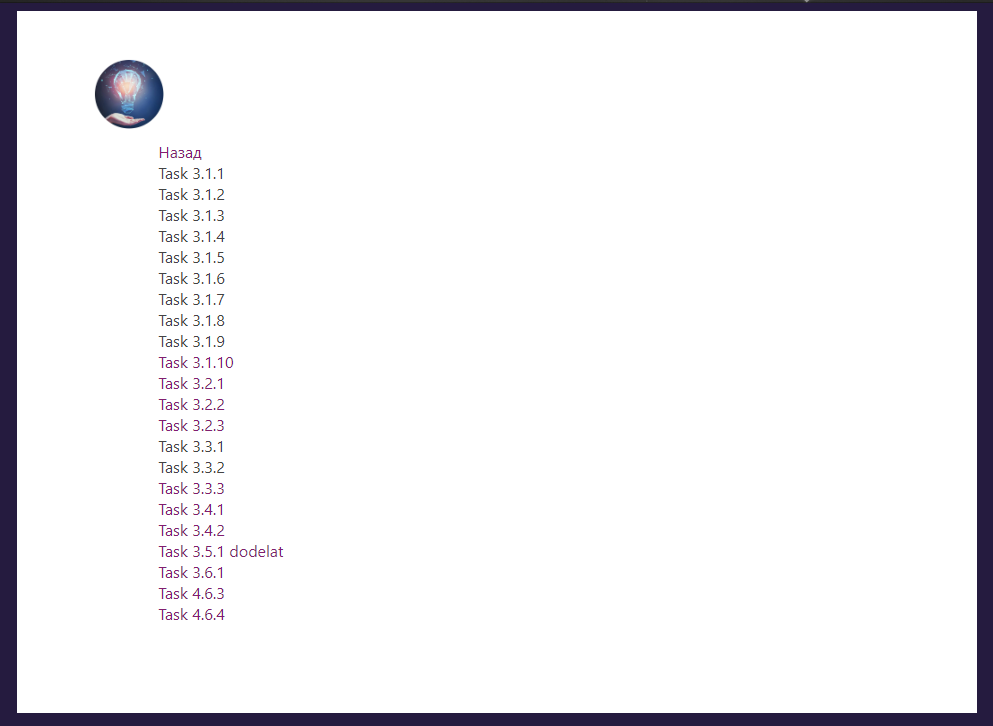


Рисунок 1 – Главная страница

В результате доработки примера из лекции получили следующую функцию:

*function task311() {*

*var response = confirm("Вы выбираете JavaScript? - Нажмите OK. А если нет - Cancel.");*

*alert('Сейчас вы получите ответ на основе вашего выбора')*

*if (response == true) {*

*alert("Отличный выбор!")*

*}*

*else {*

*alert("Может, подумаете? Просмотрите конспект на сайте КМТИ.")*

*}*

*}*

Как видно из кода, кроме двух окон из исходного примера добавлено промежуточное окно с имитацией обработки выбора. Результат работы представлен на рисунках 2-5.

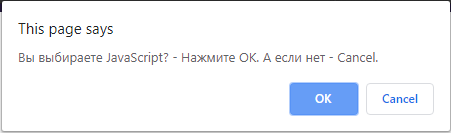


Рисунок 2 – Первое диалоговое окно

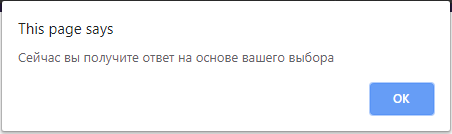


Рисунок 3 – Промежуточное диалоговое окно



Рисунок 4 – Окно после нажатия кнопки «*ОК*»

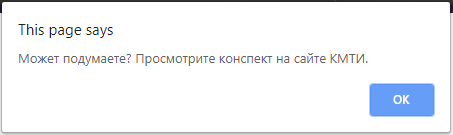


Рисунок 5 – Окно после нажатия кнопки «*Cancel*»

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.2

Для выполнения этого задания необходимо было изменить пример из лекции так, чтобы на экран выводились цифры от одного до пяти. Функция выглядит следующим образом:

*function task312() {*

*var x = 0*

*do {*

*++x*

*alert(x)*

*}*

*while (x < 5)*

*}*

В результате появляется первое диалоговое окно с цифрой один, продемонстрированное на рисунке 6, и при нажатии на кнопку «*ОК*» оно сменяется следующей цифрой, и так до пяти, после чего диалоговое окно закрывается.

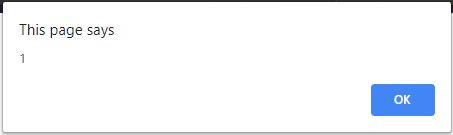


Рисунок 6 – Первое диалоговое окно задания 3.1.2

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.3

В данном задании необходимо было изменить пример из лекции так, чтобы часть *do* при *while* равном *false* выполнялось не более трёх раз. Код функции:

*function task313() {*

*var x = -1;*

*do {*

*if (x > 3) break*

*x++;*

*sometext = prompt("Введите текст", "Впечатайте любой текст");*

*}*

*while ((sometext == "Впечатайте любой текст" ||sometext == "" ||sometext == null) && x < 2)*

*console.log(x)*

*if (x > 3) { alert("Если это задание 3.5.3, то вы забыли выполнить его условие."); }*

*alert(sometext);*

*}*

Результат работы представлен на рисунках 7-8.

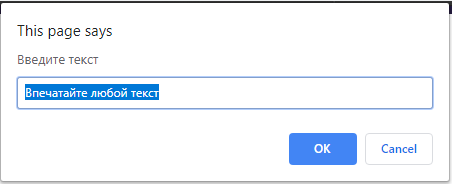


Рисунок 7 – Окно для ввода текста

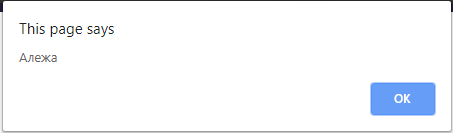


Рисунок 8 – Окно для вывода текста

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.4

В данном задании необходимо изменить пример Задания 2.1.2, чтобы на экран выводились цифры от 1 до 5-ти, но с использованием оператора *while*. Код функции:

*function task314() {*

*var x = 0*

*while (x < 5) {*

*++x*

*alert(x)*

*}*

*}*

Результат аналогичен с выполнением задания 3.1.1.

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.5

В данном задании необходимо изменить пример из лекции таким образом, чтобы создавался массив с индексами и значениями в порядке убывания (от девяти до нуля). Код функции:

*function task315() {*

*var myarray = []*

*for (i = 10; i > 0; i--) {*

*myarray[10 - i] = i - 1*

*}*

*alert(myarray)*

*if (i != 0) { alert("Если это задание 3.5.5, то вы забыли выполнить его условие") }*

*}*

Результат работы продемонстрирован на рисунке 9.

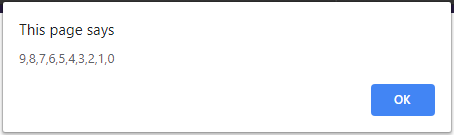


Рисунок 9 – Вывод массива в окно по убыванию

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.6

В данном задании необходимо изменить пример из лекции так, чтобы массив содержал до пяти действующих ссылок на ваши картинки и (или) фотографии (только лишь на существующие) в известном вам *JPEG*-формате (с расширениям *jpg*). Сейчас эти ссылки всего лишь выводятся на экран. Код функции:

*function task316() {*

*var myarray = new Array("", "", "", "", "", "", "", "", "", "");*

*var y = -1, ys = "", extent1 = ".gif", extent2 = ".jpg"; //вставьте в extent2 расширение для формата JPEG*

*for (x in myarray) {*

*y++;*

*ys = y;*

*myarray[x] = "\n<a href=\"../image1/" + x + ".jpg\"" + ">рис." + ys + "</a>";*

*}*

*alert("Ссылки на мои картинки gif-формата:" + myarray);*

*console.log(myarray)*

*if (extent2 != ".jpg") { alert("Если это задание 356, то Вы, забыли про фото в формате jpeg"); }*

*}*

Результат работы представлен на рисунке 10.

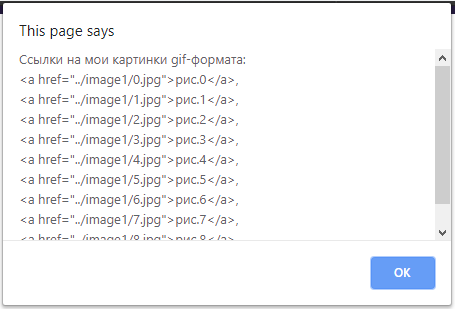


Рисунок 10 – Результат работы задания 3.1.6

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.7

В данном задании необходимо изменить пример из лекции так, чтобы на экран выводились одни чётные числа. Код функции:

*function task317() {*

*var x = 0;*

*while (x < 10) {*

*x++;*

*if (x % 2 !== 0) {*

*continue;*

*}*

*alert(x);*

*}*

Выводимые диалоговые окна аналогичны тем, что выводились в задании 2.1.2, только показывают чётные числа от двух до десяти.

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.8

В данном задании необходимо ввести число для прерывания цикла. Также присутствует проверка на ввод некорректного значения. Код функции:

*function task318() {*

*let breakat = prompt("Введите число для прерывания цикла от 1 до 10-ти ", "");*

*if (breakat>10 || breakat<1 || !Number.isInteger(breakat)) {*

*alert('Вы не ввели число в заданном промежутке')*

*return*

*}*

*for (let i = 0; i < 10; i++) {*

*if (i == breakat) {*

*break;*

*}*

*alert(i);*

*}*

*}*

Результат работы проиллюстрирован на рисунках 11-12. При вводе корректного значения запускается цикл, как в задании 2.1.2, который останавливается при достижении введенного числа.

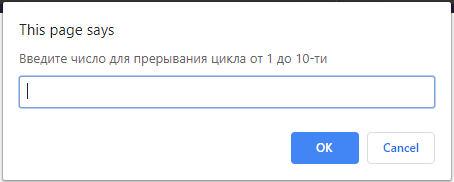


Рисунок 11 – Окно для ввода значения

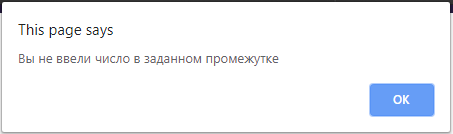


Рисунок 12 – Введено некорректное значение

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.9

В данном задании необходимо воспользоваться оператором switch для написания сценария, в котором будут выполняться различные операторы в зависимости от того, какая буква введена в окне запроса: a, b, c, d или e. Иное значение также обрабатывается. Код функции:

*function task319() {*

*var yourchoice;*

*yourchoice = prompt("Введите a/b/c/d или что-то другое", "a, b, c, d или e")*

*switch (yourchoice) {*

*case "a":*

*alert("Вы ввели букву а");*

*break;*

*case "b":*

*alert("Вы ввели букву b");*

*break;*

*case "c":*

*alert("Вы ввели букву c");*

*break;*

*case "d":*

*alert("Вы ввели букву d");*

*break;*

*case "e":*

*alert("Вы ввели букву e");*

*break;*

*default:*

*alert("Вы ввели что-то непонятное");*

*break;*

*}*

*}*

Результат работы продемонстрирован на рисунках 13-15. На рисунке 14 показано окно, которое появляется при вводе предложенных букв, на рисунке 15 – при вводе иного значения.

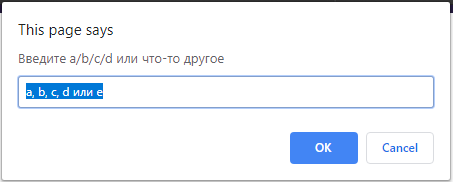


Рисунок 13 – Окно ввода



Рисунок 14 – Результат ввода предлагаемого значения

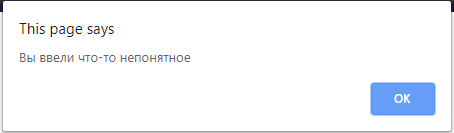


Рисунок 15 – Результат ввода неподходящего значения

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.10

В данном задании необходимо с использованием любого оператора цикла изменить пример 3.5.10 из лекции так, чтобы форма с требованием ввода пароля выводилась три раза и лишь после этого загружалась дополнительная страница. Код функции:

*for (let i = 0; i < 3; i++) {*

*pass = prompt('Введите пароль, пожалуйста:');*

*if (pass == '1111') { alert('Поздравляю! Вы ввели верный пароль!') }*

*else {*

*alert('Сожалею, но вы ввели неправильный пароль. Повторите, пожалуйста!'), top.location.href = "refuse.html"*

*}*

*}*

Результат работы продемонстрирован на рисунках 16-20.

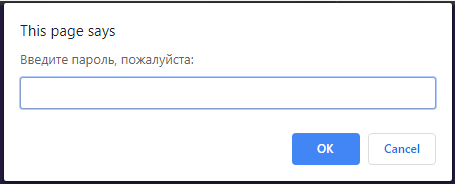


Рисунок 16 – Окно ввода пароля

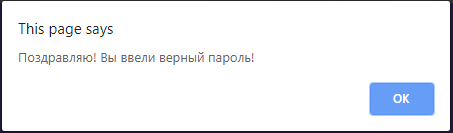


Рисунок 17 – Окно подтверждения после трехкратного ввода правильного пароля

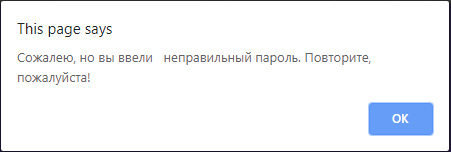


Рисунок 18 – Окно отказа после ввода неправильного пароля

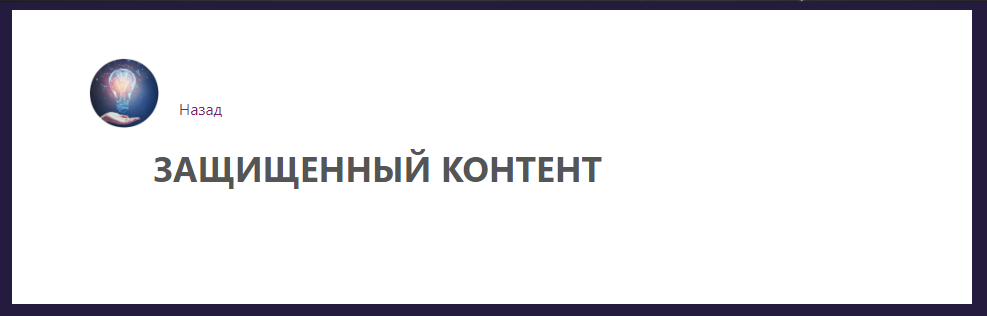


Рисунок 19 – Страница с информацией, «защищенной» окном ввода

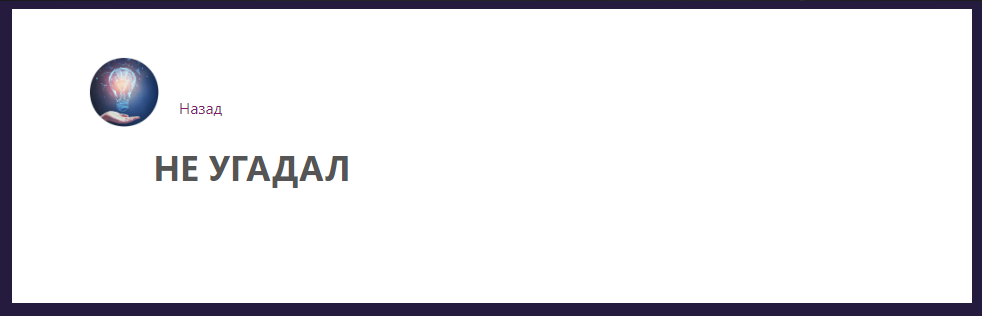


Рисунок 20 – Страница, выводящаяся при вводе неправильного пароля

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.11

В данном задании необходимо создать страницу с функцией, которая будет выполняться после того, как пользователь уберет курсор с гиперссылки. При этом должна загружаться страница со списком примеров и заданий.

Функция имеет следующий вид:

*function mouseout() {*

*alert('Вы убрали курсор с гиперссылки')*

*top.location.href = "lab3.html"*

*}*

Обработчик перемещения курсора со ссылки находится в ее атрибуте *onmouseout*:

*<a onmouseout='mouseout()' href='lab3.html'>mouseout event ref</a>*

Результат работы продемонстрирован на рисунках 21-22. При нажатии на кнопку «*ОК*» диалогового окна происходит перенаправление на страницу со списком заданий.

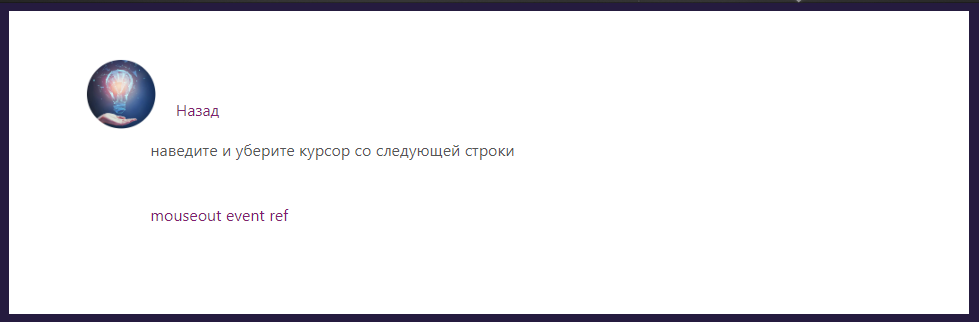


Рисунок 21 – Главная страница задания

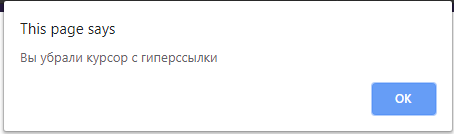


Рисунок 22 – Окно, возникающее после перемещения курсора со ссылки

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.12

В данном задании необходимо создать страницу, при закрытии которой пользователь получит прощальное сообщение. Для этого можно воспользоваться функцией *onbeforeunload()*:

*function onunload() {*

*alert('Прощайте')*

*}*

*window.onbeforeunload(onunload())*

Таким образом, перед загрузкой страницы появляется окно, которое можно видеть на рисунке 23, и только после закрытия окна загружается страница, показанная на рисунке 24.



Рисунок 23 – Окно с прощальным сообщением



Рисунок 24 – Загружаемая страница

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.13

В данном задании необходимо на основе примера 3.6.9 создать сценарий, дополнив сценарий из примера еще четырьмя  функциями и  соответствующими кнопками переводов: Цельсий-Реомюр, Фаренгейт-Реомюр, Реомюр-Цельсий, Реомюр-Фаренгейт, при этом кнопки должны активизировать добавленные функции после двойного щелчка мышью с помощью соответствующего события из приведенной выше таблицы.

Обработчики двойных нажатий на кнопки имеют следующий вид:

*function inputCels() {*

*var cels = prompt("Введите значение температуры в градусах по цельсию", "Введите температуру");*

*ansFah = doFahCalc(cels);*

*alert(cels + " градусов по Цельсию равно " + ansFah + " градусов по Фаренгейту");*

*}*

*function inputFah() {*

*var fah = prompt("Введите значение температуры в градусах по Фаренгейту", "Введите температуру");*

*ansCel = doCelCalc(fah);*

*alert(fah + " градусов по Фаренгейту равно " + ansCel + " градусов по Цельсию");*

*}*

*function inputCr() {*

*var fah = prompt("Введите значение температуры в градусах по Цельсию", "Введите температуру");*

*ansCel = cr(fah);*

*alert(fah + " градусов по Цельсию равно " + ansCel + " градусов по Реомюру");*

*}*

*function inputFr() {*

*var fah = prompt("Введите значение температуры в градусах по Фаренгейту", "Введите температуру");*

*ansCel = fr(fah);*

*alert(fah + " градусов по Фаренгейту равно " + ansCel + " градусов по Реомюру");*

*}*

*function inputRc() {*

*var fah = prompt("Введите значение температуры в градусах по Фаренгейту", "Введите температуру");*

*ansCel = rc(fah);*

*alert(fah + " градусов по Реомюру равно " + ansCel + " градусов по Цельсию");*

*}*

*function inputRf() {*

*var fah = prompt("Введите значение температуры в градусах по Фаренгейту", "Введите температуру");*

*ansCel = rf(fah);*

*alert(fah + " градусов по Реомюру равно " + ansCel + " градусов по Фаренгейту");*

*}*

*function doCelCalc(fah) {*

*var ans = ((Number(fah) - 32) / 1.8);*

*return (ans);*

*}*

*function doFahCalc(cels) {*

*var ans = ((1.8 \* Number(cels)) + 32);*

*return (ans);*

*}*

*function cr(deg) {*

*return deg\*0.8*

*}*

*function fr(deg) {*

*return (deg-32)\*0.444444*

*}*

*function rc(deg) {*

*return (deg/0.8)*

*}*

*function rf(deg) {*

*return deg\*2.25+32*

*}*

Главная страница задания представлена на рисунке 25.

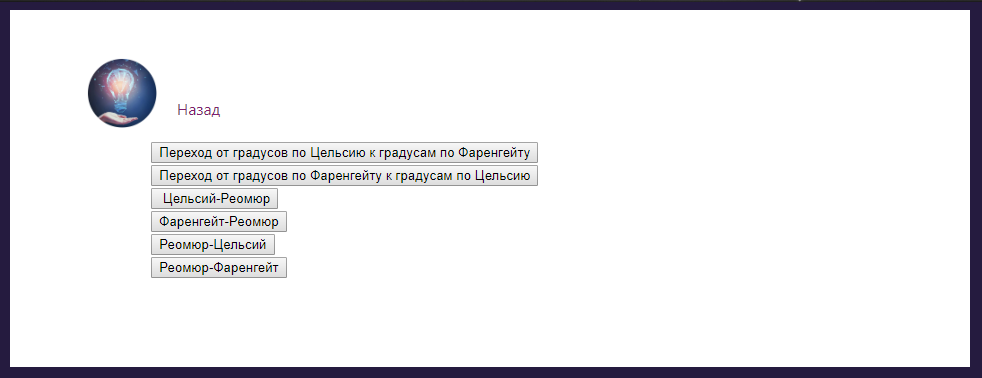


Рисунок 25 – Главная страница задания

Результат перевода градусов Цельсия в Фаренгейт представлен на рисунках 26-27.

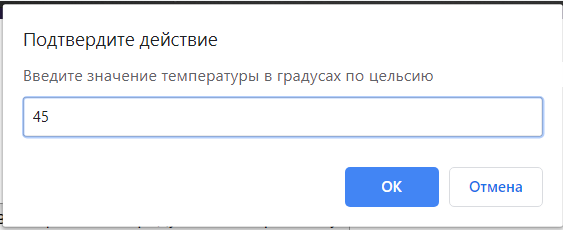


Рисунок 26 – Окно ввода температуры по Цельсию

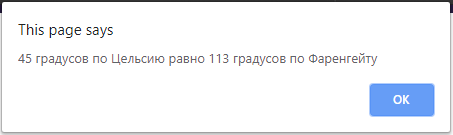


Рисунок 27 – Окно перевода температуры из Цельсия в Фаренгейт

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.14

В задании необходимо дополнить пример 3.7.1. из лекции №16 вычислением псевдослучайного числа в промежутке от 0 до 100, которое затем, предварительно округлив до целого, используйте в качестве радиуса для вычисления площади круга.

Функция имеет следующий вид:

*function task331() {*

*rndNum = Math.random(); //1-й вариант случайного числа*

*alert("1-й радиус = " + rndNum);*

*alert("Площадь круга с 1-м радиусом = " + Math.PI \* ((rndNum\*rndNum)));*

*rndNum = rndNum \* 10; //2-й вариант случайного числа*

*alert(" 2-й радиус = " + Math.round(rndNum));*

*alert("Площадь круга со 2-м радиусом = " + Math.PI \* ((Math.round(rndNum)\*Math.round(rndNum))));*

*var min=0;*

*var max=100;*

*var rad = Math.round(Math.random() \* (+max - +min)) + +min; //3-й вариант случайного числа*

*alert("3-й радиус = " + Math.round(rad));*

*alert('Площадь круга c 3-м радиусом = '+Math.PI\*rad\*rad)*

*}*

Результат выполнения задания представлен на рисунках 28-33.

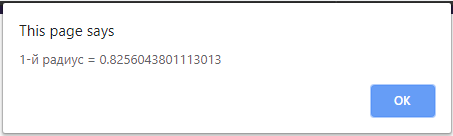


Рисунок 28 – Генерация первого рандомного радиуса от 0 до 1

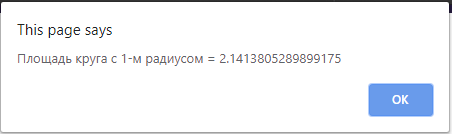


Рисунок 29 – Вычисление площади с первым радиусом



Рисунок 30 – Генерация второго рандомного радиуса от 0 до 10

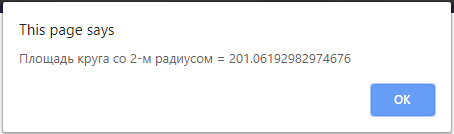


Рисунок 31 – Вычисление площади со вторым радиусом



Рисунок 32 – Генерация третьего рандомного радиуса от 0 до 100

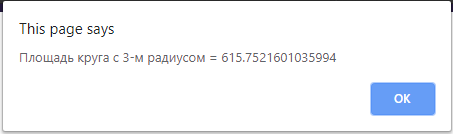


Рисунок 33 – Вычисление площади с третьим радиусом

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.15

В задании необходимо выполнить расчет и вывод на экран даты, которая наступит через неделю аналогично тому, как это сделано в примере 3.7.4 из лекции №16, но без учета времени (оперировать необходимо датами без времени).

После изменений примера из лекции функция примет следующий вид:

*function task332() {*

*dateVar = new Date();*

*day\_per\_ms = 1000 \* 60 \* 60 \* 24;*

*ten\_day\_per\_ms = 7 \* day\_per\_ms;*

*today\_ms = dateVar.getTime();*

*today\_ms += ten\_day\_per\_ms;*

*dateVar.setTime(today\_ms);*

*ten\_day = dateVar.toLocaleString();*

*let str = ten\_day.split(',')[0]*

*alert("Дата через " + ten\_day\_per\_ms / day\_per\_ms +*

*" дней от текущей даты :-) : " + str);*

*}*

Результат выполнения скрипта представлен на рисунке 34.

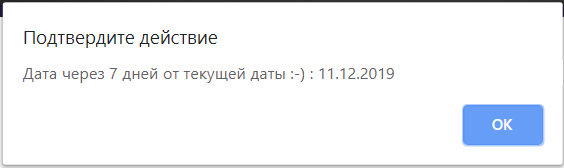


Рисунок 34 – Результат выполнения задания

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.16

В данном задании требуется, используя пример 3.7.10 из лекции №16, написать сценарий для определения среднемесячного дохода за шестимесячный период. Для обращения к значениям, передаваемым из формы нужно использовать не имена (как это было сделано в примере 3.7.10 лекции №16), а свойство elements частной коллекции forms.

Функция, обрабатывающая данные из формы имеет следующий вид:

*function income(ob\_frm) {*

*let sum = 0*

*for (i=0;i<6;i++) {*

*sum+= Number(ob\_frm.elements[i].value)*

*}*

*ob\_frm.result.value = (sum/6).toFixed(2);*

*}*

Форма продемонстрирована на рисунке 35. Результат вычислений средней зарплаты показан на рисунке 36.

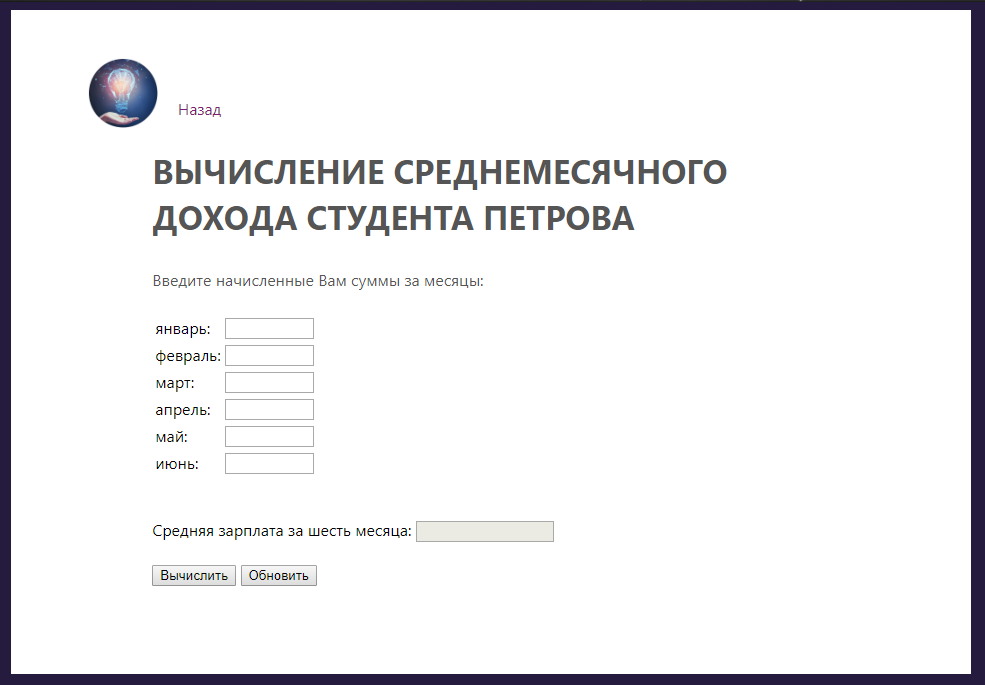


Рисунок 35 – Форма для вычисления заработной платы за полгода

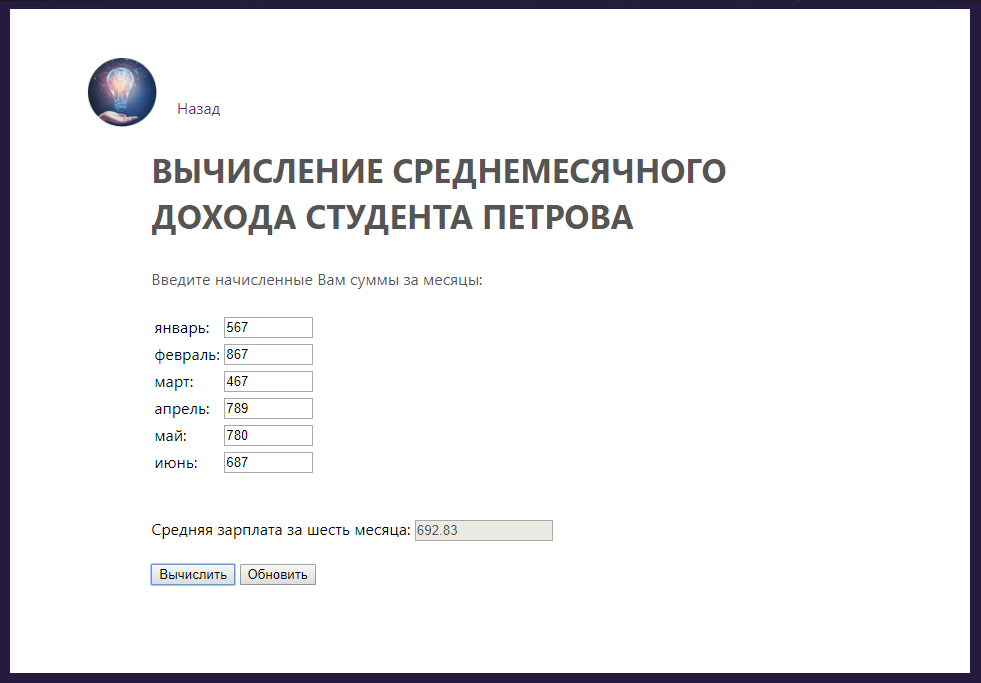


Рисунок 36 – Результат вычислений заработной платы за полгода

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.17

В данном задании необходимо на основе примера из лекции №17 усложните программу вычисления площадей фигур введя еще одну фигуру – равносторонний треугольник, а также возможность расчета площадей в различных единицах: миллиметрах, сантиметрах, метрах, дюймах. Единица для входного параметра – метр.

Функция вычисления площади примет следующий вид:

*function testsel() {*

*var d = document;*

*var a = d.forms["form1"].num.value;*

*var s;*

*var r = " ";*

*if (((d.forms["form1"].elements[0])[0]).selected) {*

*r = r + ((d.forms["form1"].elements[0])[0]).text;*

*s = a \* a \* 3.14;*

*}*

*if (((d.forms["form1"].elements[0])[1]).selected) {*

*r = r + ((d.forms["form1"].elements[0])[1]).text;*

*s = a \* a;*

*}*

*if (((d.forms["form1"].elements[0])[2]).selected) {*

*r = r + ((d.forms["form1"].elements[0])[2]).text;*

*s = a \* a / 2;*

*}*

*if (((d.forms["form1"].elements[0])[3]).selected) {*

*r = r + ((d.forms["form1"].elements[0])[3]).text;*

*s = Math.sqrt(3)/4 \* a\*a ;*

*}*

*if (((d.forms["form1"].elements[1])[0]).selected) {*

*s = s*

*}*

*if (((d.forms["form1"].elements[1])[1]).selected) {*

*s = s \* 10000*

*}*

*if (((d.forms["form1"].elements[1])[2]).selected) {*

*s = s \* 1000000*

*}*

*if (((d.forms["form1"].elements[1])[3]).selected) {*

*s = s \* 39.37 \* 39.37*

*}*

*d.forms["form1"]["res"].value = s.toFixed(3);*

*d.forms["form1"]["fig"].value = r;*

*}*

Форма для вычисления площадей фигур представлена на рисунке 37. Результат вычисления площади равностороннего треугольника продемонстрирован на рисунке 38.

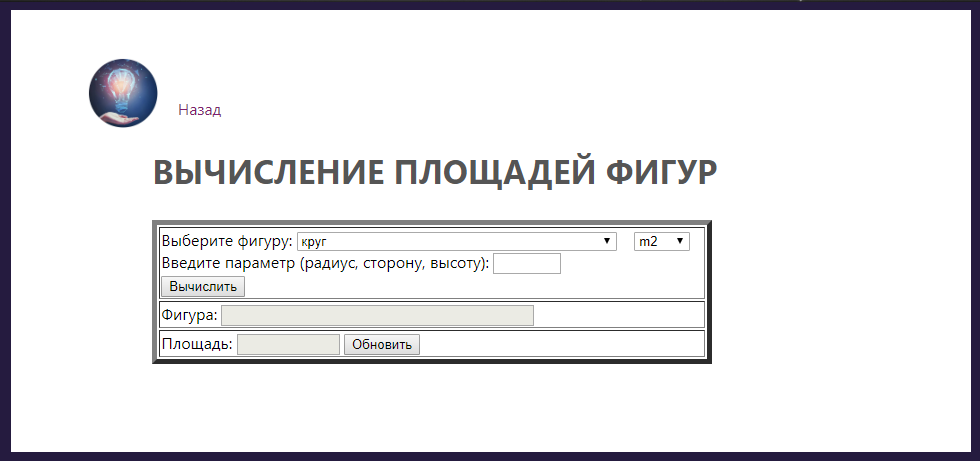


Рисунок 37 – Форма для вычисления площадей фигур

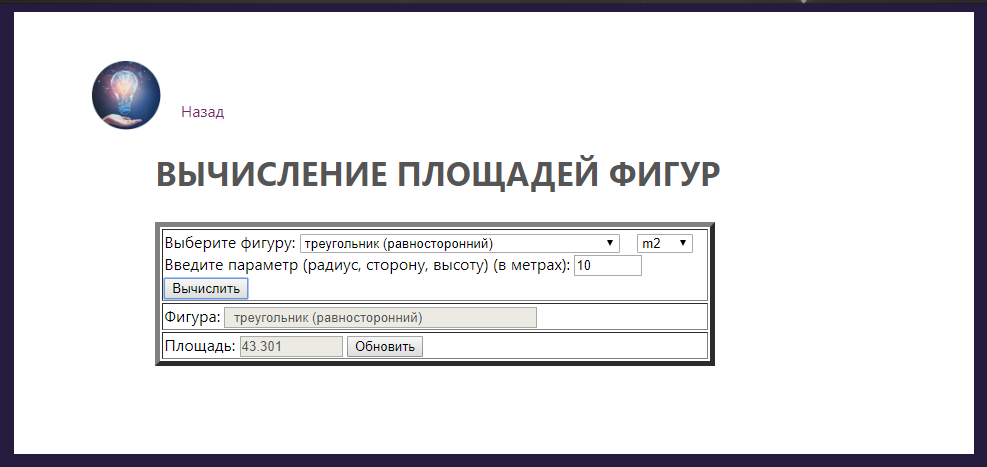


Рисунок 38 – Результат вычисления площади равностороннего треугольника

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.18

В данном задании требуется изменить задание 2.1.16, добавив вычисление и вывод максимального и минимального месячных доходов и соответствующих им названий месяцев с использованием данных за 12 месяцев.

Функция, обрабатывающая данные из формы после изменений примет следующий вид:

*function income(ob\_frm) {*

*let sum = 0*

*let arr = []*

*for (i = 0; i < 12; i++) {*

*sum += Number(ob\_frm.elements[i].value)*

*arr.push(Number(ob\_frm.elements[i].value))*

*}*

*ob\_frm.result.value = (sum/12).toFixed(2);*

*ob\_frm.min.value = Math.min(...arr)*

*ob\_frm.max.value = Math.max(...arr)*

*}*

Начальная форма и результат вычислений представлены на рисунках 39 и 40 соответственно.

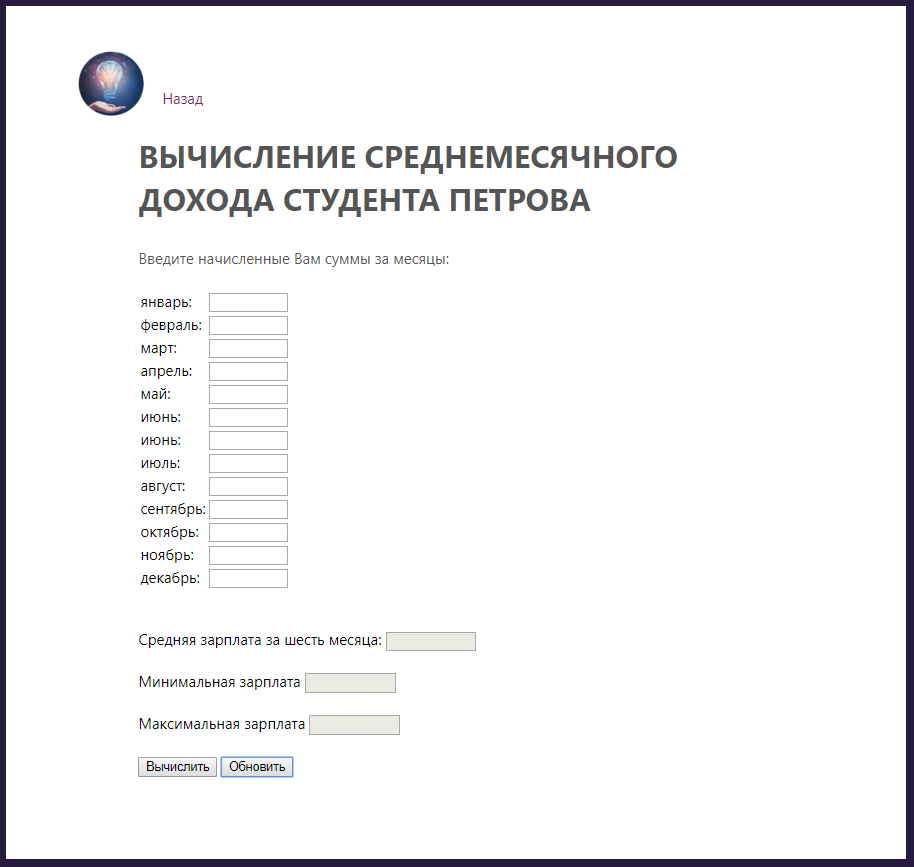


Рисунок 39 – Начальная форма

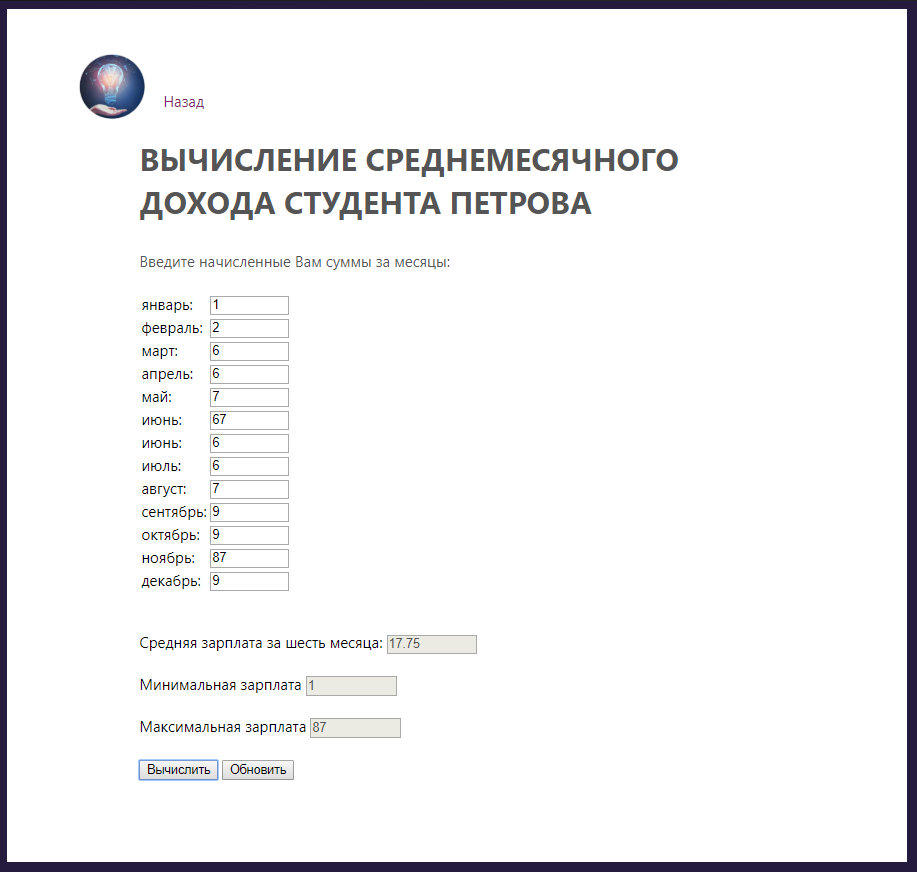


Рисунок 40 – Результат вычислений

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.19

В данном задании необходимо создать анимационный фрагмент, в котором вокруг перемещающегося справа налево изображения-центра вращается по окружности другое изображение-спутник.

Код программы:

*var p1;*

*var sp1;*

*var Circle = false;*

*var Line = false;*

*function initLine() {*

*Line = false;*

*p1 = 24;*

*sp1 = 1;*

*l1.style.left = 24;*

*}*

*var timerLine = 10;*

*function succLine() {*

*sp1 = 1;*

*p1 += sp1;*

*if (p1 > 700) p1 = 24;*

*l1.style.left = p1;*

*timerLine = setTimeout(succLine, 200);*

*if (Line == true) stoppedLine();*

*}*

*var timerCircle = 10;*

*function initCircle() {*

*timerCircle = setInterval(moveImage, 10);*

*}*

*var angle = 0;*

*var radius = 50;*

*var xbegin = 80;*

*var ybegin = 80;*

*var d = document;*

*function moveImage() {*

*var rad = angle \* Math.PI / 180;*

*d.myimage.style.pixelLeft = xbegin + radius \* Math.sin(rad);*

*d.myimage.style.pixelTop = ybegin + radius \* Math.cos(rad);*

*angle += 1;*

*if (angle >= 360) angle = 0;*

*if (Circle == true) stoppedCircle();*

*}*

*function stoppedCircle() {*

*clearInterval(timerCircle);*

*}*

*function stoppedLine() {*

*clearTimeout(timerLine);*

*}*

Внешний вид фрагмента представлен на рисунке 41.

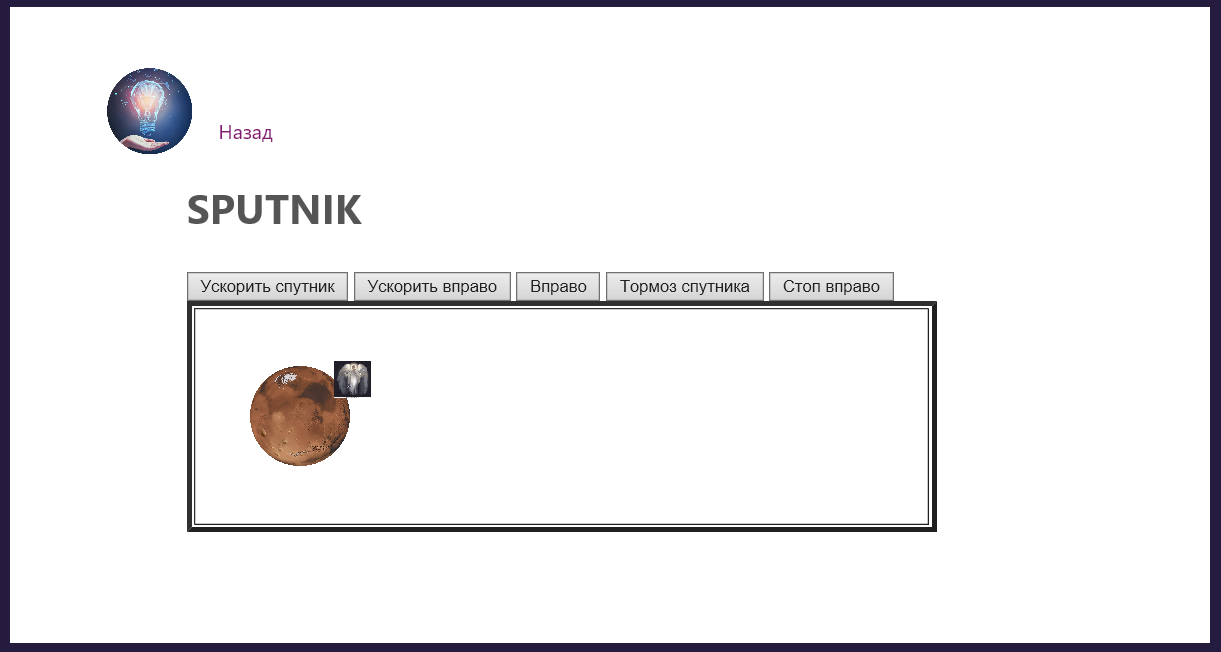
**

Рисунок 41 – Страница с анимационным фрагментом

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.20

Создадим страницу, добавим на неё несколько тегов и получим информацию с помощью функции *document.body.getElementsByTagName()*. Результат выполнения программы приведен на рисунке 42.

Листинг кода:

*<HTML id="n\_1">*

*<HEAD id="n\_2">*

*<TITLE name="n\_3">Простая страница</TITLE>*

*</HEAD>*

*<BODY id="n\_4" bgcolor="F8F8FF">*

*<H2 id="n\_5">Пример для работы с коллекциями элементов</H2>*

*<P id="n\_6">Это 6-ой элемент коллекции (порядковый номер - 5)</P>*

*<P id="n\_7">Это 7-ой элемент коллекции (порядковый номер - 6)</P>*

*<SCRIPT id="n\_8" language="JavaScript">*

*for (var i = 0; i < document.all.length; i++) {*

*alert("1-й способ. " + "Тэг с индексом " + i + ": " +*

*document.all.item(i).tagName);*

*alert("2-й способ. " + "Тэг с индексом " + i + ": " +*

*document.all[i].tagName);*

*}*

*alert("Встретилось элементов в документе:" +*

*document.all.length); // выдаст 8*

*alert("6-й элемент: " +*

*document.all.item("n\_6").tagName);*

*alert("7-й элемент: " +*

*document.all["n\_7"].tagName);*

*alert("8-й элемент (последний): " +*

*document.all.n\_8.tagName);*

*alert("8-й элемент (первый в коллекции SCRIPTS): " +*

*document.scripts[0].tagName);*

*</SCRIPT>*

*<SCRIPT id="n\_9" language="JavaScript">*

*alert("9-й элемент (второй в коллекции SCRIPTS): " +*

*document.scripts[1].tagName);*

*alert("Всего элементов в документе: " + document.all.length); // выдаст 9*

*</SCRIPT>*

*</BODY>*

*</HTML>*

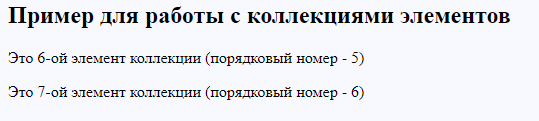


Рисунок 42 – Результат выполнения задания

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.21

В данном задании необходимо создать страницу и добавить область для отображения картинок. Для этого необходимо добавить обработку двух событий *onmouseover* – для обработки наведения курсора на картинку*, onmouseout* – для обработки покидания курсором области картинки. Результат выполнения программы представлен на рисунках 43 и 44.

Листинг кода:

*<html><head><title>Простая страница</title></head>*

*<body bgcolor="F8F8FF">*

*<h2>Проведите над изображением указателем мыши.</h2>*

*<A HREF=""*

*onmouseover = "document.myImage.src = './image1/second.jpg';"*

*onmouseout = "document.myImage.src = './image1/first.png';">*

*<IMG BORDER=0 name="myImage" src = './image1/second.jpg'>*

*</A></body>*

*</html>*



Рисунок 43 – Страница до перемещения курсора по изображению



Рисунок 44 – Страница после перемещения курсора по изображению

## Описание хода выполнения индивидуального задания, приведенного в пункте 2.1.22

В данном задании требуется создать страницу и добавить область для отображения изображения. Для этого необходимо создать три тега *ADLIST*, которые будут содержать в себе ссылки на картинки и периодичность смены изображений. Результат выполнения программы приведен на рисунке 45.

Листинг кода:

*<HTML>*

*<HEAD>*

*<TITLE>Периодически сменяющиеся объявления-изображения</TITLE>*

*<LINK REL="stylesheet" HREF="pr424.css" TYPE="text/css">*

*<!-- Можно добавлять объявления в этот список.*

*Для этого придуман нераспознаваемы элемент. -->*

*<ADLIST src="./image1/0.jpg" duration=3000>*

*<ADLIST src="./image1/1.jpg" duration=2000>*

*<ADLIST src="./image1/2.jpg" duration=3000>*

*<ADLIST src="./image1/3.jpg">*

*<ADLIST src="./image1/4.jpg" duration=3000>*

*<ADLIST src="./image1/img5.jpg">*

*<ADLIST src="./image1/angel.gif" duration=3000>*

*<ADLIST src="./image1/clever.png">*

*<ADLIST src="./image1/first.png" duration=3000>*

*<ADLIST src="./image1/img5.gif">*

*<ADLIST src="./image1/second.jpg" duration=4000>*

*<SCRIPT LANGUAGE="JavaScript">*

*var adSet = document.all.tags("ADLIST");*

*adSet.current = 0;*

*var nextImage = document.createElement("IMG");*

*function preLoad() {*

*nextImage.onerror = preLoad;*

*nextImage.src =*

*adSet[adSet.current].getAttribute("src");*

*nextImage.duration =*

*adSet[adSet.current].getAttribute("duration");*

*if (null == nextImage.duration)*

*nextImage.duration = 2000;*

*if (++adSet.current == adSet.length)*

*adSet.current = 0; //Стартуем снова.*

*}*

*function skipImage() {*

*if (nextImage.complete) {*

*document.all.ad.src = nextImage.src;*

*var duration = nextImage.duration;*

*preLoad();*

*window.tm = setTimeout('skipImage()', duration);*

*}*

*else*

*window.tm = setTimeout('skipImage()', 10);*

*}*

*preLoad();*

*</SCRIPT>*

*</HEAD>*

*<BODY ONLOAD="window.tm = setTimeout('skipImage()', 1);"*

*ONUNLOAD="clearTimeout(window.tm);">*

*<H2>Периодически сменяющиеся изображения.</H2>*

*<IMG ID="ad" SRC="./image1/0.jpg">*

*</BODY>*

*</HTML>*

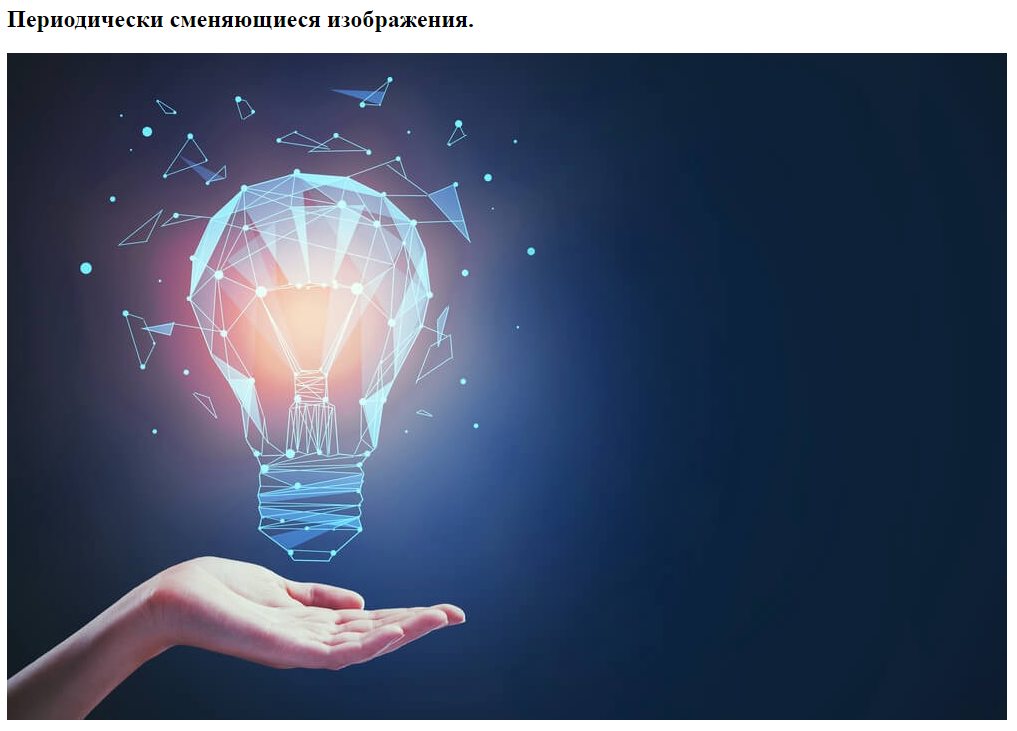


Рисунок 45 – Результат выполнения задания

**ЗАКЛЮЧЕНИЕ**

*JavaScript* — полноценный динамический язык программирования, который применяется к HTML документу, и может обеспечить динамическую интерактивность на веб-сайтах.

С помощью *JavaScript*-анимаций можно осуществить движение по сложному пути с временной функцией, отличной от кривой Безье, или *canvas*-анимации.