МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образование «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**Лабораторная работа №7**

по дисциплине: «Веб-технологии»

на тему: «Операторы управления, функции. Объекты ядра JavaScript.»

|  |  |
| --- | --- |
| Выполнил | Студент группы 22-ИТ-1  Короткевич П.А. |
|  |  |
| Проверил | Макарычева В. А. |

Полоцк 2024

**Цель:** Ознакомиться с функциями и операторами управления JavaScript, научиться применять их на практике.

**Задание 1**

Создайте объект calculator (калькулятор) с тремя методами:

1. read() (читать) запрашивает два значения и сохраняет их как свойства объекта.
2. sum() (суммировать) возвращает сумму сохранённых значений.
3. mul() (умножить) перемножает сохранённые значения и возвращает результат.

Листинг 1

function Calculator() {

let calculator = {

read() {

while (isNaN(this.a)) {

this.a = parseInt(prompt("Введите первое значение:", ""));

}

while (isNaN(this.b)) {

this.b = parseInt(prompt("Введите второе значение:", ""));

}

},

sum() {

return this.a + this.b;

},

mul() {

return this.a \* this.b;

}

};

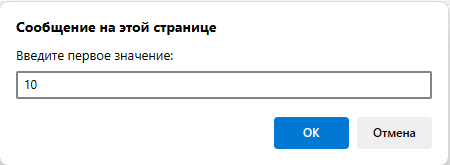
calculator.read(); // Запрашиваем значения

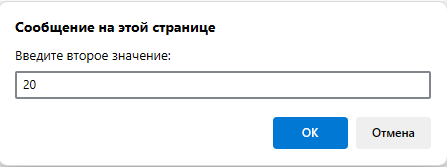
alert("Сумма: " + calculator.sum());

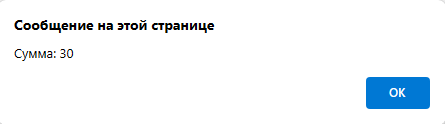
alert("Произведение: " + calculator.mul());

}

На следующем рисунке представлена реализация:







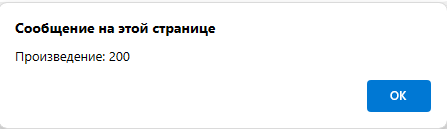


Рисунок 1 – Реализация первого задания

В данном коде определена функция Calculator, которая создает объект «calculator» с методами «read», «sum» и «mul». Метод «read» запрашивает у пользователя два числа с помощью prompt и сохраняет их в глобальные переменные a и b. Методы «sum» и «mul» выполняют операции сложения и умножения этих чисел соответственно, используя значения a и b, определенные в методе «read». Затем вызывается метод «read» для ввода пользователем значений, а затем выводятся результаты суммы и произведения этих чисел с помощью alert.

**Задание 2**

Напишите функцию-конструктор Accumulator(startingValue).

Объект, который она создаёт, должен уметь следующее:

* Хранить «текущее значение» в свойстве value. Начальное значение устанавливается в аргументе конструктора startingValue.
* Метод read() использует prompt для получения числа и прибавляет его к свойству value.

Таким образом, свойство value является текущей суммой всего, что ввёл пользователь при вызовах метода read(), с учётом начального значения startingValue.

Листинг 2

function Accumulatoring(){

function Accumulator(startingValue) {

while (isNaN(startingValue)) {

startingValue = parseFloat(prompt("Введите начальное значение (должно быть числом):", ""));

}

this.value = startingValue;

this.read = function() {

let input = parseFloat(prompt("Введите число (для завершения введите пустую строку):", ""));

while (!isNaN(input) && input !== null && input !== "") {

this.value += input;

input = parseFloat(prompt("Введите число (для завершения введите пустую строку):", ""));

}

};

}

let startingValue = parseFloat(prompt("Введите начальное значение:", ""));

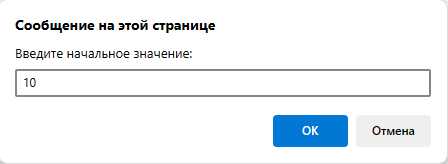
let accumulator = new Accumulator(startingValue);

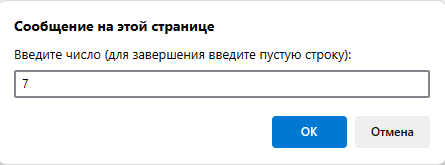
accumulator.read();

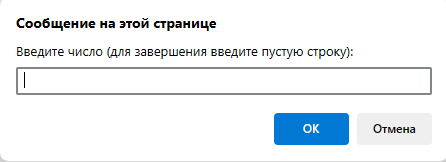
alert("Текущее значение: " + accumulator.value);

}

На следующем рисунке представлена реализация:







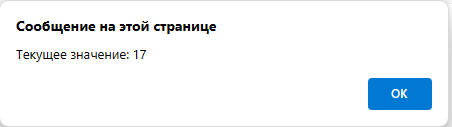


Рисунок 2 – Реализация второго задания

Эта функция Accumulatoring определяет внутри себя другую функцию Accumulator, которая является конструктором. В этом конструкторе осуществляется проверка на ввод числового начального значения. Если введенное значение не является числом, пользователю предлагается ввести его снова. Затем создается новый объект accumulator с помощью конструктора Accumulator, и вызывается его метод read(), который запрашивает у пользователя числа и прибавляет их к текущему значению value объекта accumulator. После завершения ввода метод read() выводит текущее значение value в виде сообщения с помощью alert.

**Задание 3**

Напишите функцию random(min, max), которая генерирует случайное число с плавающей точкой от min до max (но не включая max).

Листинг 3

function getRandomNumber() {

let min;

let max;

while(isNaN(min)){

min = parseFloat(prompt("Введите минимальное значение:"));

Продолжение листинга 3

}

while(isNaN(max)){

max = parseFloat(prompt("Введите максимальное значение:"));

}

function random(min, max) {

return Math.random() \* (max - min) + min;

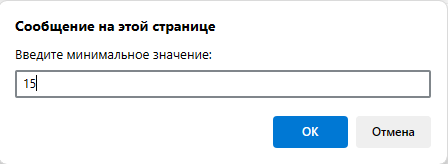
}

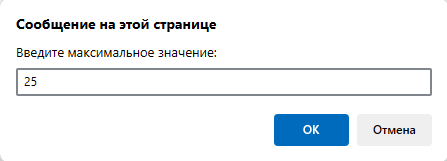
let randomNumber = random(min, max);

alert("Случайное число: " + randomNumber);

}

На следующем рисунке представлена реализация:





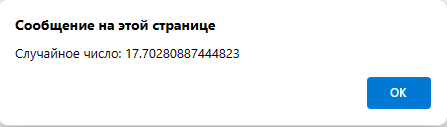


Рисунок 3 – Реализация третьего задания

Данная функция getRandomNumber() предназначена для генерации случайного числа в заданном пользователем диапазоне. Сначала она запрашивает у пользователя ввод минимального и максимального значений для диапазона. При этом выполняется проверка на то, что введенные значения являются числами. Если введенное значение не является числом, пользователю снова предлагается ввести значение до тех пор, пока не будет введено численное значение. После успешного ввода минимального и максимального значений функция вызывает вспомогательную функцию random(min, max), которая генерирует случайное число в заданном диапазоне, не включая верхнюю границу.

**Задание 4**

Есть стоимость в виде строки $120;. То есть сначала идёт знак валюты, а затем – число. Создайте функцию extractCurrencyValue(str), которая будет из такой строки выделять числовое значение и возвращать его.

Листинг 4

function getCost(){

function extractCurrencyValue(str) {

return parseFloat(str.slice(1));

}

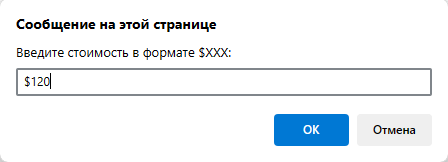
let userInput = prompt("Введите стоимость в формате $XXX:");

let numericValue = extractCurrencyValue(userInput);

alert("Конечный результат:" + numericValue);

}

На следующем рисунке представлена реализация:



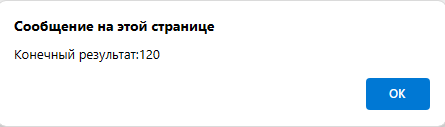


Рисунок 4 – Реализация четвертого задания

Эта функция getCost() предназначена для запроса у пользователя стоимости в формате $XXX и вывода числового значения этой стоимости без знака валюты. Сначала функция запрашивает у пользователя ввод стоимости с помощью prompt. Затем внутренняя функция extractCurrencyValue(str) извлекает числовое значение из строки, удаляя символ валюты $ с помощью метода slice(1) и преобразуя оставшуюся часть строки в число с плавающей точкой с помощью parseFloat().

Задание 5

Напишите функцию sumInput(), которая:

* Просит пользователя ввести значения, используя prompt и сохраняет их в массив.
* Заканчивает запрашивать значения, когда пользователь введёт не числовое значение, пустую строку или нажмёт «Отмена».
* Подсчитывает и возвращает сумму элементов массива.

Ноль 0 считается числом, не останавливайте ввод значений при вводе «0».

Листинг 5

function task5(){

function sumInput(){

let numbers=[];

while(true){

let input = prompt("Введите число:", "");

if (input === null || input === "") {

break;

}

if (!isNaN(input)) {

numbers.push(+input);

} else {

alert("Введите число, а не букву или другой символ.");

Продолжение листинга 5

}

}

let sum =0;

for(let number of numbers){

sum+=number;

}

return sum;

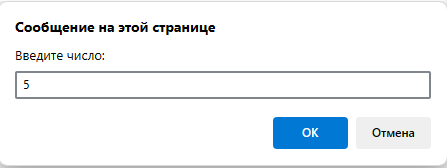
}

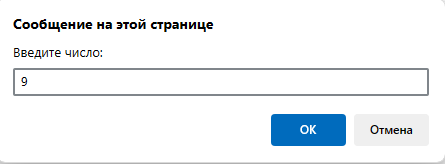
let result = sumInput();

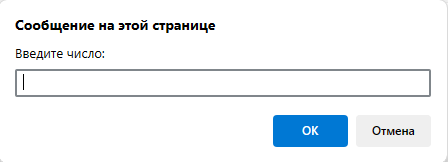
alert("Сумма введенных чисел:" + result);

}

На следующем рисунке представлена реализация:







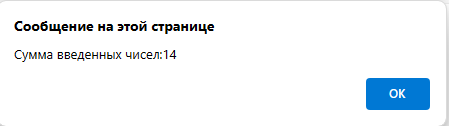


Рисунок 5 – Реализация пятого задания

В этой функции task5 определена внутренняя функция sumInput(), которая запрашивает у пользователя ввод чисел с помощью prompt, сохраняет их в массиве numbers и затем возвращает их сумму. Цикл while(true) выполняется до тех пор, пока пользователь не введет пустую строку или нажмет отмену. В каждой итерации цикла запрашивается ввод числа с помощью prompt. Если пользователь вводит число, то оно добавляется в массив numbers после преобразования в число с помощью унарного плюса +. Если же ввод содержит нечисловые символы, то выводится сообщение об ошибке с помощью alert, и пользователь снова приглашается ввести число.

**Задание 6**

Есть объект salaries с произвольным количеством свойств, содержащих заработные платы.

Напишите функцию sumSalaries(salaries), которая возвращает сумму всех зарплат с помощью метода Object.values и цикла for..of.

Если объект salaries пуст, то результат должен быть 0.

Листинг 6

function task6(){

function sumSalaries() {

let salaries = {};

let name = prompt("Введите имя сотрудника:", "");

while (name) {

let salary = parseFloat(prompt("Введите зарплату для " + name + ":", ""));

if (!isNaN(salary)) {

salaries[name] = salary;

} else {

alert("Введите числовое значение для зарплаты.");

}

name = prompt("Введите имя следующего сотрудника (или нажмите Отмена для завершения):", "");

}

let sum = 0;

for (let salary of Object.values(salaries)) {

sum += salary;

}

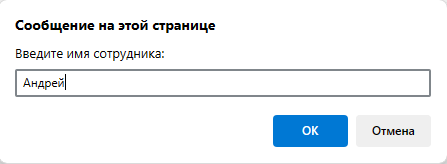
return sum;

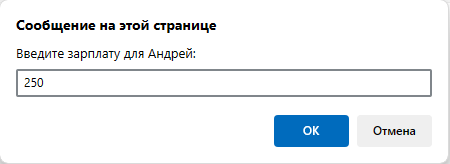
}

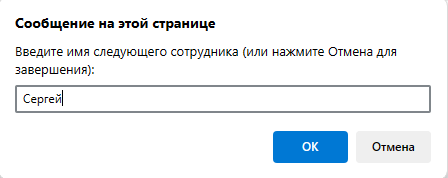
alert("Сумма зарплат: " + sumSalaries());

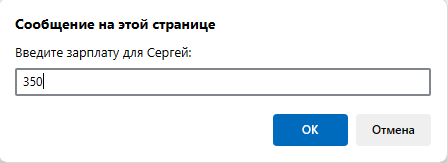
}

На следующем рисунке представлена реализация:









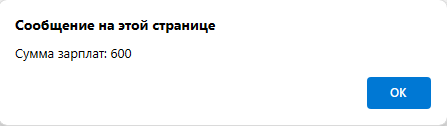


Рисунок 6 – Реализация шестого задания

Эта функция запрашивает у пользователя ввод имени сотрудника и их зарплаты до тех пор, пока пользователь не нажмет кнопку "Отмена" или не введет пустую строку. Для каждого сотрудника вводится их зарплата, которая добавляется в объект salaries. Если вводится некорректное значение для зарплаты (не числовое), выводится сообщение об ошибке. После завершения ввода данных программа вычисляет сумму всех зарплат, используя метод Object.values() для получения массива всех зарплат, а затем суммирует их с помощью цикла for...of.

**Задание 7**

Создайте функцию topSalary(salaries), которая возвращает имя самого высокооплачиваемого сотрудника.

* Если объект salaries пустой, то нужно вернуть null.
* Если несколько высокооплачиваемых сотрудников, можно вернуть любого из них.

P.S. Используйте Object.entries и деструктурирование, чтобы перебрать пары ключ/значение.

Листинг 7

function task7(){

function topSalary() {

let salaries = {};

let name = prompt("Введите имя сотрудника:", "");

while (name !== "") {

let salary = parseFloat(prompt("Введите зарплату для " + name + ":", ""));

if (!isNaN(salary)) {

salaries[name] = salary;

} else {

alert("Введите числовое значение для зарплаты.");

}

name = prompt("Введите имя следующего сотрудника (или нажмите Отмена для завершения):", "");

}

let maxSalary = 0;

let topPaidEmployee = null;

for (let [name, salary] of Object.entries(salaries)) {

if (salary > maxSalary) {

maxSalary = salary;

topPaidEmployee = name;

}

}

return topPaidEmployee;

}

let topPaidEmployee = topSalary();

if (topPaidEmployee) {

alert("Самая высокая зарплата у сотрудника: " + topPaidEmployee);

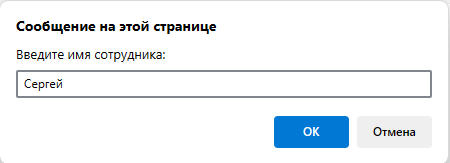
} else {

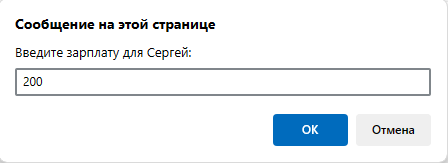
alert("Нет данных о зарплатах сотрудников.");

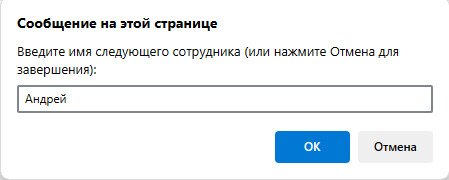
}

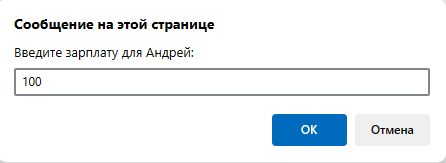
}

На следующем рисунке представлена реализация:









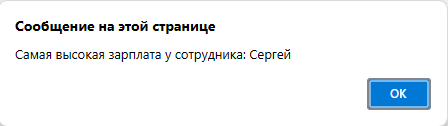
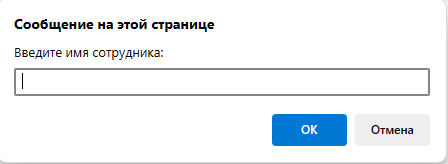


Рисунок 7 – Реализация седьмого задания



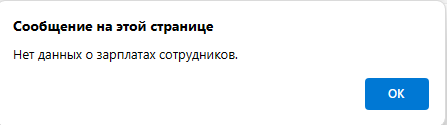


Рисунок 8 – Отсутствие данных

Эта функция task7 запрашивает у пользователя ввод имени и зарплаты сотрудников до тех пор, пока пользователь не введет пустую строку в качестве имени сотрудника. Затем она определяет сотрудника с наивысшей зарплатой, перебирая объект salaries и сравнивая значения зарплат. После этого функция возвращает имя сотрудника с наивысшей зарплатой. В конце task7 проверяет, есть ли вообще данные о зарплатах сотрудников. Если есть, то выводится сообщение с именем сотрудника, у которого наивысшая зарплата. Если данных о зарплатах нет, то выводится сообщение о том, что нет информации о зарплатах.

**Задание 8**

Напишите функцию sum, которая работает таким образом: sum(a)(b) =a+b.

Да, именно таким образом, используя двойные круглые скобки (не опечатка).

Листинг 8

function task8(){

function sum(a){

let b = parseFloat(prompt("Введите число b:"));

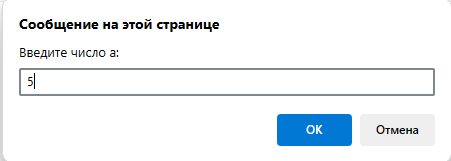
return a+b;

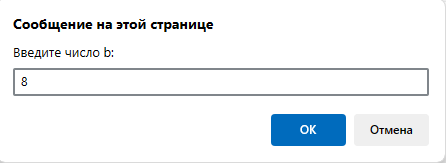
}

let a = parseFloat(prompt("Введите число a:"));

alert("Сумма чисел:" + sum(a));}

На следующем рисунке представлена реализация:





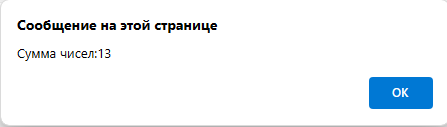


Рисунок 9 – Реализация девятого задания

Функция sum(a) запрашивает у пользователя ввод числа b с помощью prompt. Если введенное значение b является числом, то оно добавляется к числу a, и результат возвращается. В противном случае выводится сообщение об ошибке, и возвращается NaN. Затем осуществляется запрос на ввод числа a с помощью prompt. Если введенное значение a является числом, то вызывается функция sum(a), и результат суммирования выводится с помощью alert. Если введенное значение a не является числом, выводится сообщение об ошибке.

**Вывод:** В процессе освоения функций и операторов управления JavaScript, открывается возможность создавать более эффективный и гибкий код. Функции помогают организовать логику программы в отдельные блоки, повторно используемые в различных частях приложения. Операторы управления, такие как условные операторы и циклы, предоставляют возможность контролировать ход выполнения программы в зависимости от различных условий.