Лабораторная работа № 9

Тема: «Операторы JS»

Контрольные вопросы

Введение в JS

1. Опишите назначение языка программирования JavaScript.

JavaScript (JS) - это язык программирования, который в основном используется для придания интерактивности веб-страницам. Он позволяет создавать динамический контент, реагировать на действия пользователя, управлять мультимедиа, валидировать данные и многое другое прямо в браузере.

1. Какая актуальная последняя версия языка JS сейчас используется?

Сложно указать конкретную версию, так как JavaScript постоянно развивается. Современный JavaScript соответствует стандарту ECMAScript (ES). Наиболее широко поддерживаемой версией на данный момент является ES2021 (ES12), но браузеры уже активно поддерживают и более новые стандарты, такие как ES2022 (ES13) и ES2023 (ES14). Важно знать, что разработчики ориентируются не на конкретную версию, а на возможности, поддерживаемые большинством браузеров.

1. Для разработки каких элементов web-сайта используют скрипты на JS? Приведите примеры.

JS используется для разработки следующих элементов:

* Интерактивные элементы: Кнопки, всплывающие окна, модальные окна, выпадающие меню. Пример: Обработка нажатия кнопки для отправки формы.
* Динамическое изменение контента: Обновление частей страницы без перезагрузки, отображение скрытых блоков, изменение текста. Пример: Загрузка новых комментариев под статьей при нажатии кнопки «Показать больше».
* Анимация и эффекты: Создание плавных переходов, анимации элементов, параллакс-эффект. Пример: Появление элемента с анимацией при прокрутке страницы.
* Валидация форм: Проверка корректности введенных данных перед отправкой на сервер. Пример: Проверка формата email-адреса.
* Работа с Cookies и LocalStorage: Сохранение данных на стороне клиента. Пример: Запоминание предпочтений пользователя (тема оформления, язык).
* Взаимодействие с API: Получение данных с сервера и отображение их на странице. Пример: Отображение списка товаров, полученных из базы данных.

1. Есть ли сходство языка программирования JavaScript с языком Java? Ответ аргументируйте.

Название «JavaScript» вводит в заблуждение. Фактически, JavaScript и Java - это совершенно разные языки программирования. Их единственное сходство – в названии, которое было выбрано в маркетинговых целях в эпоху популярности Java. Они имеют разный синтаксис, разные области применения и разные принципы работы. Java - это компилируемый язык, часто используемый для серверной разработки, в то время как JavaScript - это интерпретируемый язык, в основном используемый для клиентской разработки.

1. Укажите название редактора кода для скриптов на JS в котором вы планируете в дальнейшем работать.

Я планирую работать с JS в редакторе кода Visual Studio Code.

1. Приведите пример синтаксиса скрипта на JS.

Пример синтаксиса JS:

// Переменная для хранения имени

let userName = "GeekBot";

// Функция, выводящая приветствие в консоль

function greet(name) {

console.log("Привет, " + name + "!");

}

// Вызов функции

greet(userName); // Вывод: Привет, GeekBot!

1. Приведите примеры на 4 разных способа внедрения JS в html код.

Способы внедрения JS:

* Встроенный тег <script>:

// Переменная для хранения имени

let userName = "GeekBot";

// Функция, выводящая приветствие в консоль

function greet(name) {

console.log("Привет, " + name + "!");

}

// Вызов функции

greet(userName); // Вывод: Привет, GeekBot!

* Внутренний <script> в <head> или <body>:

<!DOCTYPE html>

<html>

<head>

<title>Пример внутреннего скрипта</title>

<script>

function sayHello() {

alert("Привет из внутреннего скрипта!");

}

</script>

</head>

<body>

<button onclick="sayHello()">Нажми меня</button>

</body>

</html>

* Внешний файл .js:

<!DOCTYPE html>

<html>

<head>

<title>Пример внешнего скрипта</title>

<script src="script.js"></script>

</head>

<body>

<button onclick="sayHello()">Нажми меня</button>

</body>

</html>

script.js:

function sayHello() {

alert("Привет из внешнего скрипта!");

}

* Внедрение через атрибуты HTML:

<button onclick="alert('Этот способ не рекомендуется!')">Нажми меня</button>

СИНТАКСИС JS

1. Укажите с какими типами данных можно работать в языке программирования JavaScript. Приведите пример описания переменных целого, вещественного и строкового типов данных.

Number: Числовой тип (целые и с плавающей точкой):

let age = 30; // Целое число

let price = 99.99; // Число с плавающей точкой

String: Строковый тип (текст):

let name = "GeekBot"; // Строка

let message = 'Привет!';// Строка

Boolean: Логический тип (true или false):

let isAdult = true; // Булево значение

let isLoggedIn = false; // Булево значение

Undefined: Тип, присваиваемый переменной, которой не было присвоено значение:

let x; // x имеет значение undefined

console.log(x)

Null: Специальное значение, представляющее «отсутствие значения»;

let user = null; // user не содержит объекта

console.log(user)

Symbol (ES6): Уникальные идентификаторы:

const id = Symbol("id");

BigInt (ES2020): Целые числа произвольной длины:

const veryBigNumber = 12345678901234567890123456789012345n;

Object: Объекты (коллекции свойств):

let person = {

name: "GeekBot",

age: 30

};

1. Укажите используемые в языке JavaScript операции языка. Приведите примеры их использования.

Операции в JS:

* Арифметические операторы: + (сложение), - (вычитание), \* (умножение), / (деление), % (остаток от деления), \*\* (возведение в степень).

let sum = 5 + 3; // 8

let difference = 10 - 4; // 6

let product = 6 \* 7; // 42

let quotient = 20 / 5; // 4

let remainder = 10 % 3; // 1

let power = 2 \*\* 3; // 8

* Операторы присваивания: =, +=, -=, \*=, /=, %=, \*\*=.

let x = 10;

x += 5; // x = x + 5 (x становится 15)

x -= 2; // x = x - 2 (x становится 13)

* Операторы сравнения: == (равно), != (не равно), === (строго равно), !== (строго не равно), > (больше), < (меньше), >= (больше или равно), <= (меньше или равно).

let a = 5;

let b = "5";

console.log(a == b); // true (сравнение по значению)

console.log(a === b); // false (сравнение по значению и типу)

* Логические операторы: && (логическое И), || (логическое ИЛИ), ! (логическое НЕ).

let sunny = true;

let warm = true;

if (sunny && warm) {

console.log("Отличная погода!");

}

if (sunny || warm) {

console.log("Хорошая погода");

}

* Строковые операторы: + (конкатенация).

let firstName = "Geek";

let lastName = "Bot";

let fullName = firstName + " " + lastName; // "Geek Bot"

* Унарные операторы: ++ (инкремент), -- (декремент), typeof (определение типа).

let count = 0;

count++; // count становится 1

console.log(typeof count); // "number"

1. Приведите пример использования методов alert(…), confirm(…) и promt() объекта window для вывода окон сообщений.

1) alert() - Выводит простое диалоговое окно с сообщением.

alert("Привет! Это сообщение от alert().");

2) confirm() - Выводит диалоговое окно с вопросом и кнопками "OK" и «Отмена».

let result = confirm("Вы уверены?");

if (result) {

alert("Вы нажали OK!");

} else {

alert("Вы нажали Отмена!");

}

3) prompt() - Выводит диалоговое окно с полем ввода для получения данных от пользователя.

let name = prompt("Пожалуйста, введите ваше имя:", "GeekBot"); // Второй аргумент - значение по умолчанию

if (name) {

alert("Привет, " + name + "!");

} else {

alert("Вы не ввели имя.");

}

ОПЕРАТОРЫ JS Опишите синтаксис следующих операторов языка программирования JavaScript. Приведите примеры их использования.

1. Условный оператор (if/else):

let age = 20;

if (age >= 18) {

console.log("Вы совершеннолетний.");

} else {

console.log("Вы несовершеннолетний.");

}

// Сокращенная запись (тернарный оператор)

let status = (age >= 18) ? "Совершеннолетний" : "Несовершеннолетний";

console.log(status);

1. Оператор множественного выбора (switсh):

let day = 3;

switch (day) {

case 1:

console.log("Понедельник");

break;

case 2:

console.log("Вторник");

break;

case 3:

console.log("Среда");

break;

default:

console.log("Неизвестный день");

}

1. Операторы организации циклов (for, while, do…while):

for:

for (let i = 0; i < 5; i++) {

console.log(i); // Выводит 0, 1, 2, 3, 4

}

while:

let i = 0;

while (i < 5) {

console.log(i);

i++;

}

do…while:

let i = 0;

do {

console.log(i);

i++;

} while (i < 5); // Выполнится хотя бы один раз

1. Оператор досрочного выхода из циклов (break):

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

if (i === 5) {

break; // Выход из цикла, когда i равно 5

}

console.log(i); // Выводит 0, 1, 2, 3, 4

}

1. Оператор продолжения (Continue):

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

if (i % 2 === 0) {

continue; // Пропуск итерации, если i четное

}

console.log(i); // Выводит только нечетные числа: 1, 3, 5, 7, 9

}

1. Оператор возврата Return:

function add(a, b) {

return a + b; // Возвращает сумму a и b

}

let result = add(5, 3);

console.log(result); // 8

ПОДПРОГРАММЫ В JS (Функции)

1. Опишите правила создания и вызова функций в JavaScript. Приведите примеры. Опишите назначение оператора Return и приведите пример его использования при работе с функциями в JavaScript. В чем состоит отличие между локальными и глобальными переменными в JavaScript.
   * Правила создания и вызова функции:

Создание функции:

// 1. function declaration

function functionName(parameter1, parameter2) {

// Тело функции

return result; // необязательно

}

// 2. function expression

const myFunction = function(parameter1, parameter2) {

// Тело функции

return result; // необязательно

}

// 3. arrow function (ES6)

const arrowFunction = (parameter1, parameter2) => {

// Тело функции

return result; // необязательно

};

const shortArrowFunction = (parameter) => result; // Сокращенная запись для однострочных функций

Вызов функции:

functionName(argument1, argument2); // для function declaration и function expression

arrowFunction(argument1, argument2); // для arrow function

* + Назначение return: Оператор return завершает выполнение функции и возвращает указанное значение. Если return отсутствует, функция возвращает undefined.
  + Пример использования return:

function multiply(x, y) {

return x \* y; // Возвращает произведение x и y

}

let product = multiply(4, 6); // product получает значение 24

* + Локальные переменные объявлены внутри функции. Видны только внутри этой функции:

function myFunc() {

let localVar = "Локальная переменная";

console.log(localVar); // "Локальная переменная"

}

myFunc();

//console.log(localVar); // Ошибка: localVar не определена вне функции

* + Глобальные переменные объявлены вне функций или объявлены внутри функции без использования let, const или var:

let globalVar = "Глобальная переменная"; // let - глобальная переменная, но в пределах модуля (или скрипта)

function myFunc() {

console.log(globalVar);

}

myFunc(); // Выводит "Глобальная переменная"

console.log(globalVar); // Выводит "Глобальная переменная"