Лабораторная работа 1.

**Введение.**

JS – язык программирования, применяющийся в основном для создания скриптов и программ, использующихся в веб страницах. В данный момент практически все браузеры поддерживают JS.

Js язык с динамической, слабой, не явной типизацией.

Ecmascript – стандартизированная расширенная версия языка.

Однострочный комментарий //

Многострочный комментарий /\* \*/

Разделение инструкций ; (не обязательно)

Зарезервированные слова: break, default, function, return, var, case, delete, if, switch, void, catch, do, in, this, while, const, else, instanceof, throw, with, continue, finally, let, try, debugger, for, new, typeof, class, enum, export, extends, import, super, Implements, interface, package, private, protected, public, static

**ПО.**

Текстовый редактор (vsCode + open-in-browser + html snipets)

Браузер (chrome)

**Подготовка.**

Внедрение JS в html.

Внедрить сценарий js в html можно несколькими способами:

* Его можно разместить в теге <script></script>. Например,

<script>alert('Hello world');</script>

* В html сценарий можно подключить js файл, содержащий код. Например,

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

*Во многих случаях удобнее второй способ*

* JS код можно встроить в обработчики событий. Например,

<h1 onclick="alert('hello world');">Нажми меня</h1>

* При помощи атрибута href и псевдопротокола JavaScript. Например,

<a href="JavaScript: alert('Hello world');">Ссылка</a>

Создадим папку lr1. Откроем ее в vsCode (файл – открыть папку…). Это будет проектом. Создадим в нем файлы Index.html и script.js.

Index.html

<html>

<head>

<meta charset="UTF-8">

    <script src="script.js"></script>

<title>Лабораторная работа 1</title>

</head>

<body>

</body>

</html>

**Выполнение.**

1. Переменные и типы данных
   1. Для объявления переменных используются ключевые слова var, let и const.

Отличия var и let:

* Переменная, объявленная var, видна везде в функции, объявленная через let, видна только в рамках блока {...}, в котором объявлена.
* Переменная, объявленная var, видна до объявления, объявленная через let только после.

Объявление const задаёт константу, то есть переменную, которую нельзя менять.

* 1. В JS существует 2 категории типов данных: простые типы и объекты.

Простые типы это: числа, строки, логические данные, null и undefined.

Чтобы получить тип данных переменной можно воспользоваться функцией typeof().

NULL TYPE: переменные этого типа хранят в себе ничего.

UNDEFINED TYPE: переменным этого типа не присвоено значение, еще этот тип возвращается функциями, которые не возвращают значение.

NUMBER TYPE: все числовые значения (-2^53 - 1 до 2^53 - 1 64 бита на число ). Может содержать значения NaN (не число), infinity и -infinity. Для проверки на эти значения используются функции isNaN() и isFinite(). Приведение к числовому типу данных Number() (или поставить+).

let str = +'10' + Number('20');

STRING TYPE: все строковые значения, содержаться в “ ”, ` ` или ‘ ’. Символ экранирования \. Приведение к строковому типу данных String(). Возможен доступ к символу по индексу.

BOOLEAN TYPE: true/false. Приведение к логическому типу данных Boolean()

Пример:

let a = 10;

let str = 'str';

let test;

console.log(typeof(a));

console.log(typeof(str));

console.log(typeof(test));

console.log(str[str.length-1]);

1. Операторы

Арифметические операторы:

+, -, \*, /, %, ++, --

Логические операторы:

&&, ||, !

Операторы присваивания:

=, +=, -=, \*=, /=

Строковые операторы:

+, +=

Операторы сравнения:

==, !=, ===, !==, >, >=, <, <=

*=== - идентичность.*

1. Условные операторы
   1. If else

if(условие) {

действия

} else if(условие) {

действия

} else {

действия

}

Пример:

let a = 10;

let a1 = 20;

let str = 'Большее из чисел равно ';

let b = true;

if (b) {

    console.log('b равно ' + b);

}

if (a > a1) {

    console.log(str + a);

}

else if (a < a1) {

    console.log(str + a1);

} else {

    console.log('Они равны');

}

* 1. Тернарный оператор условного присваивания

(условие) ? истина: ложь

Пример:

let a = 0;

function isEven(num) {

    return (num % 2) ? 'не четное' : 'четное';

}

console.log(isEven(a));

* 1. Конструкция switch (для чисел и строк, сравнение по идентичности)

switch (переменная) {

case значение переменной:

действие

break;

default:

действия

break;

}

Пример:

let v = '';

switch (v) {

    case '':

    case null:

        console.log('пусто');

        break;

    case 'str':

        console.log('str');

        break;

    default:

        console.log('default');

        break;

}

1. Циклы
   1. while (с предусловием)

while(условие) {

тело цикла

}

Пример:

let i = 10;

while (i >= 0) {

    console.log(i--);

}

* 1. do while (с пост условием)

do {

тело цикла

} while (условие)

Пример:

let i = 10;

do {

    console.log(i);

}

while (--i >= 0)

* 1. Цикл for

for(инициализация счетчика;проверка;инкремент) {

тело цикла

}

Пример:

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

    console.log(i);

}

* 1. continue – переход к следующему шагу цикла

let i = 10;

while (i >= 0) {

    if (i === 5) {

        continue;

    }

    console.log(i--);

}

* 1. break – выход из цикла

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

    if (i === 5) {

        break;

    }

    console.log(i);

}

1. Методы ввода/вывода объекта window
   1. alert() - сообщение

let str = 'Hello world';

alert(str);

* 1. confirm() – сообщение с подтверждением

let str = 'Да или отмена?';

console.log(confirm(str));

* 1. prompt() – сообщением с вводом строки

console.log(prompt('Введите слово', 'слово'));

* 1. eval() – разбор строковых данных

let str = '2+4';

console.log(eval(str));