**Лабораторная работа №1**

**Тема работы:** Создание простой JavaScript программы

**Цель работы:** изучить базовые принципы написания JavaScript программ.

**Теоретические сведения**

*Язык JavaScript предназначен для выполнения в браузере наряду с HTML и CSS. Но, если эти языки предназначены для верстки структуры сайта, то JavaScript позволяет 'оживлять' web-страницы - делать их реагирующими на действия пользователя или демонстрировать некоторую динамичность (к примеру, смена картинок в блоке или красивые плавно выпадающие менюшки).*

*Итак, давайте без лишних слов приступим к изучению этого, бесспорно, полезного языка.*

**Как запустить JavaScript**

Написать и запустить JavaScript можно двумя способами: первый заключается в том, что мы пишем код прямо на HTML странице внутри тега **<script>**:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Это заголовок тайтл</title>

<script>

var name **=** 'Дима';

alert('Привет, '**+**name);

</script>

</head>

<body>

Это основное содержимое страницы.

</body>

</html>

Тег **<**script**>** можно располагать в любом месте страницы - как в **<head>**, так и в **<body>**.

Второй вариант заключается в том, что JavaScript код хранится в отдельном файле (наподобие CSS) и подключается тоже с помощью тега **<script>** с атрибутом **src**, в котором указывается путь к файлу со скриптом:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Это заголовок тайтл</title>

<script src="путь к файлу со скриптом"></script>

</head>

<body>

Это основное содержимое страницы.

</body>

</html>

***Данные*** *- это объекты, которыми может оперировать JavaScript, к примеру, имя пользователя - это строка, а его возраст - это число. Данные могут быть разных* ***типов****.*

**Строки и числа**

Самыми простыми **типами данных** в JavaScript являются **строки** и **числа**.

**Числа** обозначают сами себя: 1, 12, 145, а вот **строки** требуется брать в кавычки (одинарные или двойные - без разницы):

'строка', "строка"; //это примеры строк

## Переменные

Одним из самых главных и распространенных объектов в программировании является переменная.

**Переменная** - это такой объект, который может хранить внутри себя различные данные, например, строки или числа.

**Имя переменной** должно состоять из английских букв: больших или маленьких, а также цифр и знака\_подчеркивания.

В JavaScript при **объявлении** переменной обязательно должно быть написано ключевое слово **var**:

var a; //тут мы объявили переменную

var a, a1, isVar, is\_var; //тут мы объявили группу переменных

**Операция присваивания**

Очень важным элементом программирования является **операция присваивания**. Пример присваивания:

var a **=** 4; //мы присвоили переменной a значение 4

**Комментарии**

В коде JavaScript, так же, как и в HTML и CSS, можно оставлять **комментарии**. Они могут быть многострочными и однострочными:

var a **=** 4; //это пример однострочного комментария.

/\*

Это пример

многострочного комментария.

\*/

var a **=** 4;

Комментарии игнорируются браузером при выполнении кода, в них можно оставлять какие-либо пометки или временно закрывать код от исполнения, чтобы потом его при необходимости вернуть (*откомментировать*).

**Функция alert**

В JavaScript существует специальная функция **alert**, которая позволяет вывести какой-либо текст в окно браузера в виде диалогового окошка.

Следующий код выводит на экран заданный текст:

alert('Привет, мир!'); //выведет на экран фразу 'Привет, мир!'

А в следующим коде переменной **text** присваивается фраза, а затем содержимое этой переменной выводится на экран:

var text **=** 'Привет, мир!';

alert(text); //выведет на экран фразу 'Привет, мир!'

**Математические операции**

В JavaScript между числами можно совершать различные **математические операции**:

alert(2 **+** 3); //выведет 5

alert(5 **-** 1); //выведет 4

alert(2 **\*** 3); //выведет 6

alert(6 **/** 2); //выведет 3

**Получение определенного символа строки**

В JavaScript можно получить доступ к **определенному символу строки** по его номеру таким образом: **a[n]** – n-ный символ строки (учтите, что *нумерация идет с нуля*):

var a, b; //объявим наши переменные

a **=** 'abcde'; //в переменной а будет хранится значение 'abcde'

b **=** a[0]; //в переменной b будет 'a'

b **=** a[1]; //в переменной b будет 'b'

b **=** a[4]; //в переменной b будет 'e'

**Сложности с операцией присваивания**

Очень часто новички не понимают, что **присваивание отличается от обычного равенства**. Посмотрите следующий пример:

var a **=** 1;

a **=** a **+** 2;

С точки зрения математики запись **a = a + 2** абсурдна, но не с точки зрения программирования.

В данном случае переменная **a** имела значение **1**, а затем мы переменной **a** присвоили новое значение - старое значение переменной **a** плюс **2**.

**Операции инкремента и декремента**

Операция **a++** или **++a** – увеличивает переменную **a** на единицу. Эта операция называется **инкремент**.

Операция **a--** или **--a** – уменьшает переменную **a** на единицу. Эта операция называется **декремент**.

Примеры:

var a **=** 1;

a**++**; //увеличит a на 1, что соответствует коду a = a + 1;

alert(a); //выведет 2

var a **=** 1;

a**--**; //уменьшит a на 1, что соответствует коду a = a - 1;

alert(a); //выведет 0

Давайте посмотрим, в каких случаях проявляется разница между **++a** и **a++**.

Пусть у нас есть код **alert(++a)** и код **alert(a++)**.

В первом случае переменная сначала увеличится на единицу, а потом выведется, а во втором случае - сначала выведется, а потом увеличится.

**Операции +=, -=, \*=, /=**

Мы уже рассматривали код, который демонстрирует сложности с операцией присваивания:

var a **=** 2;

a **=** a **+** 3;

В данном случае мы присваиваем переменной **a** ее текущее значение, увеличенное на 2. Однако JavaScript позволяет записать этот код еще короче с помощью оператора **+=**:

var a **=** 1;

a **+=** 3; //этот код полностью эквивалентен коду a = a + 3;

Кроме того, существуют операторы -=, \*=, /=, которые эквивалентны следующему коду:

var a **=** 2;

a **-=** 3; //этот код полностью эквивалентен коду a = a - 3;

var a **=** 2;

a **\*=** 3; //этот код полностью эквивалентен коду a = a \* 3;

var a **=** 2;

a **/=** 3; //этот код полностью эквивалентен коду a = a / 3;

**Специальные значения**

В JavaScript, как и в других языках программирования, существуют ключевые слова для некоторых специальных значений. Вот они: undefined, null, true, false, NaN, Infinity, -Infinity.

**Значения undefined и null**

Значение **undefined** обозначает неопределенность. К примеру, если мы попробуем обратиться к переменной, которой мы еще не задали значение - то ее значение и будет **undefined**.

var a;

alert(a); //выведет undefined

Значение **null** обозначает 'ничего'. К примеру, мы можем присвоить переменной значение **null** в знак того, что там ничего не лежит.

Это значение очень похоже на **undefined**, отличие в том, что **undefined** - это не определенное значение, а **null** - определенное - ничего.

### **Значения true и false**

Значения **true** и **false** обозначают истину и ложь соответственно. Они используются для таких вещей, которые предполагают два варианта ответа - да или нет.

К примеру, на вопрос 'вам уже есть 18 лет?' в можете ответить да, то есть **true**, или нет, то есть **false**.

**Значение NaN**

Значение **NaN** (*Not-A-Number*) обозначает не число. Оно может получится, к примеру, в таком случае - когда вы умножаете строку с буквами на число:

alert('abc'**\***3); //выведет NaN

### **Значения Infinity и -Infinity**

Значения **Infinity** и **-Infinity** обозначают соответственно бесконечность и минус бесконечность. Они получаются если какое-то число поделить на ноль - в этом случает JavaScript не выдает ошибку, как в других языках программирования, а возвращает эти значения.

Если мы делим на ноль положительное число, то получаем **Infinity**, а если отрицательное - то **-Infinity**.

**Функция prompt**

Кроме функции alert, которая выдает диалоговое окошко, существует функция **prompt**, которая не только выдает окошко с текстом, но и позволяет получить от пользователя какой-либо текст.

Этот текст можно записать в переменную и затем выполнить над ним какие-либо операции.

В следующем примере мы спросим имя пользователя, запишем его в переменную **name** и с помощью функции **alert** выведем на экран:

var name **=** prompt('Ваше имя?');

alert('Ваше имя: '**+**name);

**Функция confirm**

Если вам нужно просто спросить у пользователя 'Да' или 'Нет', не давая ему возможность ввести иной текст - используйте функцию **confirm**.

Эта функция вызывает окошко с вопросом, на который нужно ответить пользователю, и двумя кнопками для ответа: с кнопкой 'ОК' и с кнопкой 'Отмена'.

Если пользователь нажмет 'ОК' - то функция вернет **true**, а если 'Отмена' - то вернет **false**.

В следующем примере функция **confirm** выведет диалоговое окно с вопросом '*Вам уже есть 18 лет?*'.

Если вы нажмете 'Ок', то в переменную ok запишется **true** и выведется на экран функцией alert, а если нажмете 'Отмена' - то **false**:

var ok **=** confirm('Вам уже есть 18 лет?');

alert(ok);

**Типизация переменных**

Что будет, если попробовать перемножить, к примеру, число и строку, вот так: **3 \* '3'**? В результате вы получите число **9**. Это значит, что JavaScript автоматически осуществляет преобразование типов при необходимости, вам не нужно за это переживать.

Однако, есть нюанс: если мы попытаемся **сложить** строку и число, то JavaScript сложит их как строки, а не как числа, вот так: **'3' + 3** получится строка **'33'**, а не число 6.

В случае, например, с умножением JavaScript понимал, что нельзя перемножить строки, поэтому строки переводил в числа и перемножал их. А случай со сложением можно трактовать двояко: складывать как строки или как числа (плюс-то используется как для сложения строк, так и чисел).

Бороться с этим можно следующем способом: нужно сделать недопустимую для строк операцию, например, так: **+'3' + 3** - поставим плюс перед строкой и она преобразуется к числу.

Второй вариант такой: можно сказать яваскрипту, что мы хотим явно преобразовать строку к числу. Это делается с помощью функции Number, вот так: **Number('3') + 3**. В результате получится 6, а не '33'.

К числам могут преобразовываться не только строки, но и любые другие типы данных, например true тоже можно преобразовать к числу таким образом: **Number(true)**.

Можно преобразовывать и к другим типам с помощью функций **Boolean**, **String** и других подобных.

**Порядок выполнения работы**

**Примеры решения задач**

**Задача №1**

**Задача.** Создайте переменную **str** и присвойте ей значение 'abcde'. Обращаясь к отдельным символам этой строки выведите на экран символ 'a', символ 'b', символ 'e'.

**Решение:** будем обращаться к отдельным символам этой строки, к примеру, буква 'a' имеет номер 0 и ее можно вывести так - **str[0]**, буква 'b' имеет номер 1 и так далее:

var str **=** 'abcde';

alert(str[0]); //выведем букву 'a'

alert(str[1]); //выведем букву 'b'

alert(str[4]); //выведем букву 'e'

**Задача №2**

**Задача.** Напишите скрипт, который считает **количество секунд** в часе.

**Решение:** так как в минуте **60** секунд, а в часе - **60** минут, то умножив **60** на **60** мы получим **количество секунд в часе**:

alert(60 **\*** 60);

Если нам нужно получить количество секунд в **сутках**, то умножим еще и на **24** (так как в сутках **24** часа):

alert(60 **\*** 60 **\*** 24);

**Задача №3**

**Задача.** Переделайте приведенный код так, чтобы в нем использовались операции **+=, -=, \*=, /=, ++, --**. Количество строк кода при этом не должно измениться. Код для переделки:

var num **=** 1;

num **=** num **+** 12;

num **=** num **-** 14;

num **=** num **\*** 5;

num **=** num **/** 7;

num **=** num **+** 1;

num **=** num **-** 1;

alert(num);

**Решение:** заменим все подходящие места на сокращенную форму записи. К примеру, вместо **num = num + 12** можно написать **num += 12**, а вместо **num = num + 1** будет **num++**. Результат переделки будет выглядеть так:

var num **=** 1;

num **+=** 12;

num **-=** 14;

num **\*=** 5;

num **/=** 7;

num**++**;

num**--**;

alert(num);

**Практическая часть**

**Работа с переменными**

1. Создайте переменную **num** и присвойте ей значение **3**. Выведите значение этой переменной на экран с помощью метода **alert**.
2. Создайте переменные **a=10** и **b=2**. Выведите на экран их сумму, разность, произведение и частное (результат деления).
3. Создайте переменные **c=15** и **d=2**. Просуммируйте их, а результат присвойте переменной **result**. Выведите на экран значение переменной **result**.
4. Создайте переменные **a=10**, **b=2** и **c=5**. Выведите на экран их **сумму**.
5. Создайте переменные **a=17** и **b=10**. Отнимите от **a** переменную **b** и результат присвойте переменной **c**. Затем создайте переменную **d**, присвойте ей значение **7**. Сложите переменные **c** и **d**, а результат запишите в переменную **result**. Выведите на экран значение переменной **result**.

**Работа со строками**

1. Создайте переменную **str** и присвойте ей значение **'Привет, Мир!'**. Выведите значение этой переменной на экран.
2. Создайте переменные **str1='Привет, '** и **str2='Мир!'**. С помощью этих переменных и операции сложения строк выведите на экран фразу **'Привет, Мир!'**.
3. Создайте переменную **name** и присвойте ей ваше имя. Выведите на экран фразу **'Привет, %Имя%!'**.
4. Создайте переменную **age** и присвойте ей ваш возраст. Выведите на экран **'Мне %Возраст% лет!'**.

**Функция prompt**

1. Спросите имя пользователя с помощью методы **prompt**. Выведите с помощью **alert** сообщение **'Ваше имя %имя%'**.
2. Спросите у пользователя число. Выведите с помощью **alert** квадрат этого числа.

**Обращение к символам строки**

1. Создайте переменную **str** и присвойте ей значение 'abcde'. Обращаясь к отдельным символам этой строки выведите на экран символ 'a', символ 'c', символ 'e'.
2. Создайте переменную **num** и присвойте ей значение '12345'. Найдите произведение (умножение) цифр этого числа.

**Практика**

1. Напишите скрипт, который считает количество секунд в **часе**, в **сутках**, в **месяце**.
2. Создайте три переменные - **час**, **минута**, **секунда**. С их помощью выведите текущее время в формате **'час:минута:секунда'**.
3. Создайте переменную, присвойте ей число. Возведите это число в квадрат. Выведите его на экран.

**Работа с присваиванием и декрементами**

1. Переделайте этот код так, чтобы в нем использовались операции **+=, -=, \*=, /=**. Количество строк кода при этом не должно измениться.

var num **=** 47;

num **=** num **+** 7;

num **=** num **-** 18;

num **=** num **\*** 10;

num **=** num **/** 15;

alert(num);

1. Переделайте этот код так, чтобы в нем использовались операции **++** и **--**. Количество строк кода при этом не должно измениться.

var num **=** 10;

num **=** num **+** 1;

num **=** num **+** 1;

num **=** num **-** 1;

alert(num);

**Контрольные вопросы**

1. Какими способами можно запустить JavaScript?
2. Что такое переменная?
3. Через какое ключевое слово объявляется переменная?
4. Каким образом можно получить определенный символ строки?
5. В чем разниуа между **null** и **undefined**?
6. В каких случаях можно получить значение **NaN**?