

Лабораторная работа № 2

Основные положения

Программа (сценарий) на языке JavaScript представляет собой последовательность операторов с "точкой с запятой" (;) между ними. Если каждый оператор размещается на одной строке, то разделитель можно не писать. Один оператор может располагаться на нескольких строках.

В программах на JavaScript можно использовать комментарии. Для того чтобы задать комментарий, располагающийся на одной строке, достаточно перед его текстом поставить две косые черты (//). Если же поясняющий текст занимает несколько строк, то его следует заключать между символами /* и */. В JavaScript строчные и прописные буквы алфавита считаются разными символами. Любой язык программирования оперирует с постоянными и переменными величинами. В JavaScript это литералы и переменные.

Опр.: Простейшие данные, с которыми может оперировать программа, называются литералами.

Литералы не могут изменяться. Литералы целого типа могут быть заданы в десятичном (по основанию 10), шестнадцатеричном (по основанию 16) или восьмеричном (по основанию 8) представлении. Шестнадцатеричные числа включают цифры 0-9 и буквы a, b, c, d, e, f. Шестнадцатеричные числа записываются с символами 0x перед числом, например, 0x25, 0xa1, 0xff. Запись вещественного литерала отличается от записи вещественного числа в математике тем, что вместо запятой, отделяющей целую часть от дробной, указывается точка, например, 123.34, -22.56. Кроме того, для записи вещественных чисел можно использовать так называемую экспоненциальную форму.

Кроме целых и вещественных значений в языке JavaScript могут встречаться так называемые логические значения. Существуют только два логических значения: истина и ложь. Первое представляется литералом true, второе - false. В некоторых реализациях JavaScript может быть использована единица в качестве true, и ноль в качестве false.

Строковый литерал представляется последовательностью символов, заключенной в одинарные или двойные кавычки. Примером строкового литерала может быть строка "результат" или 'результат'.

Опр.: Элемент, используемый для хранения данных, называется переменной.

Тип переменной зависит от хранимых в ней данных, при изменении типа данных меняется тип переменной. Определить переменную можно с помощью оператора var, например: var test1.

В данном случае тип переменной test1 не определен и станет известен только после присвоения переменной некоторого значения. Оператор var можно использовать и для инициализации переменной, например, конструкцией var test2=276 определяется переменная test2 и ей присваивается значение 276.

Значение переменной изменяется в результате выполнения оператора присваивания. Оператор присваивания может быть использован в любом месте программы и способен изменить не только значение, но и тип переменной. Оператор присваивания выглядит так: $a=b$, где a - переменная, которой мы хотим задать некоторое значение; b - выражение, определяющее новое значение переменной.

```
Пусть в сценарии описаны следующие переменные
var n=3725
var x=2.75
var p=true
var s="Выполнение завершено"
```

n и x имеют тип `number`, тип переменной p - логический, переменная s имеет тип `string`. В JavaScript определен тип `function` для всех стандартных функций и функций, определяемых пользователем. Объекты JavaScript имеют тип `object`. Переменные типа `object` часто называют просто объектами, они могут хранить объекты.

Опр.: Выражения строятся из литералов, переменных, знаков операций, скобок. В зависимости от типа вычисленного значения выражения можно разделить на арифметические, логические и строковые. Арифметические выражения получаются при выполнении операций, перечисленных в табл.1.

Операции отношения применимы к операндам любого типа. Результат операции - логическое значение `true`, если сравнение верно, и `false` - в противном случае.

Приоритет операций определяет порядок, в котором выполняются операции в выражении. В табл.2 перечислены рассмотренные операции в порядке убывания приоритетов.

Сценарии, написанные на языке JavaScript, могут располагаться непосредственно в HTML-документе между тегами `<script>` и `</script>`.

Одним из параметров тега `<script>` является `language`, который определяет используемый язык сценариев. Для языка JavaScript значение параметра равно "JavaScript". Если применяется язык сценариев VBScript, то значение параметра должно быть равным "VBScript". В случае использования языка JavaScript параметр `language` можно опускать, т. к. этот язык выбирается браузером по умолчанию.

Обычно браузеры, не поддерживающие какие-либо теги HTML, эти теги просто игнорируют. Попытка браузера проанализировать содержимое не поддерживаемых тегов может привести к неверному отображению страницы. Чтобы избежать такой ситуации, рекомендуется помещать операторы языка JavaScript в теги комментария `<!-- ... -->`. Для правильной работы интерпретатора перед закрывающим тегом комментария `-->` следует поставить символы `//`.

Итак, для размещения сценария в HTML-документе следует написать следующее:

```
<script language="JavaScript">
```

```
</script>
```

Документ может содержать несколько тегов `<script>`. Все они последовательно обрабатываются интерпретатором JavaScript. В следующем примере в раздел `<body>` (в тело) HTML-документа вставлены операторы языка JavaScript.

Пример 1. Вычисление площади треугольника

Необходимо написать сценарий, определяющий площадь прямоугольного треугольника по заданным катетам. Сценарий разместим в разделе `<body>` HTML-документа (листинг 1).

Листинг 1. Первый сценарий в документе :

```
<HTML>
<HEAD>
<title>Первый сценарий в документе</title>
</HEAD>
<BODY>
<P>Страница, содержащая сценарий.</P>
<script>
<!--
var a=8; h=10 /*Инициализируются две переменные*/
document.write ("Площадь прямоугольного треугольника равна ", a*h/2, ".")
/*Для формирования вывода используется метод write объекта document*/
//-->
</script>
<P>Конец формирования страницы, содержащей сценарий</P>
</BODY>
</HTML>
```

Задания

1. Проверить пример из лабораторной работы.
2. Составить сценарий, в котором вычисляется площадь круга по заданному радиусу.
3. Составить сценарий, вычисляющий гипотенузу по заданным катетам.