

Реализация основных алгоритмических структур на JavaScript.

Автор: Тюленев Евгений Анатольевич

Место работы: МБОУ "Боровихинская сош"

Должность: Учитель

Оглавление

1 Введение.....	3
1.1 Основные понятия.....	3
1.2 История JavaScript.....	3
1.3 Программа "Hello, World!!!".....	3
2 Вычисление значений выражений.....	5
3 Переменная. Операция присваивания. Организация ввода.....	8
3.1 Модельный ввод и вывод.....	9
4 Управляющие операторы.	11
4.1 Условный оператор.....	11
4.1.1 Регулярные выражения.....	12
4.2 Операторы цикла.....	13
5 Составные переменные.....	14
5.1 Массивы.....	14
5.1.1 Заполнение массивов.....	15
5.1.2 Обработка массива.....	16

1 Введение.

1.1 Основные понятия.

Исполнитель — это объект способный выполнять определенный набор команд.

Алгоритм — это последовательность команд, выполнение которой ведет к решению задачи.

Требования к алгоритму:

1. *Понятность*: инструкции алгоритма должны быть понятны исполнителю.
2. *Замкнутость*: в результате выполнения алгоритма должна быть решена именно данная задача.
3. *Дискретность*: выполнение очередной инструкции не начинается, пока не закончится выполнение текущей.
4. *Конечность*: выполнение инструкций алгоритма должно когда-то закончиться.

1.2 История JavaScript.

Первая версия языка JavaScript появилась в 1995 году. ведущий разработчик этого языка Брендан Аик, был нанят в компанию Netscape. Передним поставили задачу разработать язык программирования, который можно было внедрить в браузер.

Позже автор языка писал: *«JS был обязан «выглядеть как Java», только поменьше, быть эдаким младшим братом-тупицей для Java. Кроме того, он должен был быть написан за 10 дней, а иначе мы бы имели что-то похуже JS.»*

В 1996 году утвержден стандарт JavaScript.

1.3 Программа "Hello, World!!!".

*"Напечатать слова Hello, world
Вот первое препятствие, и чтобы его преодолеть,
вы должны суметь где-то создать текст
программы, успешно его скомпилировать,
загрузить, запустить на выполнение и
разобраться, куда будет отправлен результат. Как
только вы овладеете этим, все остальное
окажется относительно просто."
Б. Керниган, Д. Ритчи "Язык
программирования Си"*

Для работы понадобится текстовый редактор и браузер. В идеале нужен "продвинутой" текстовый редактор (с подсветкой синтаксиса, сворачиванием блоков кода, суфлером кода) и все популярные браузеры (IE, Firefox, Opera, Chromium, Netscape

N, и другие). Именно браузер является исполнителем скриптов на языке JavaScript. Лучше всего поддерживает стандарты Firefox.

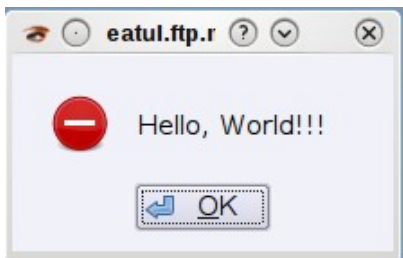
Можно использовать IDE для Web разработки, например Quanta plus.

Начнем, пожалуй!!!

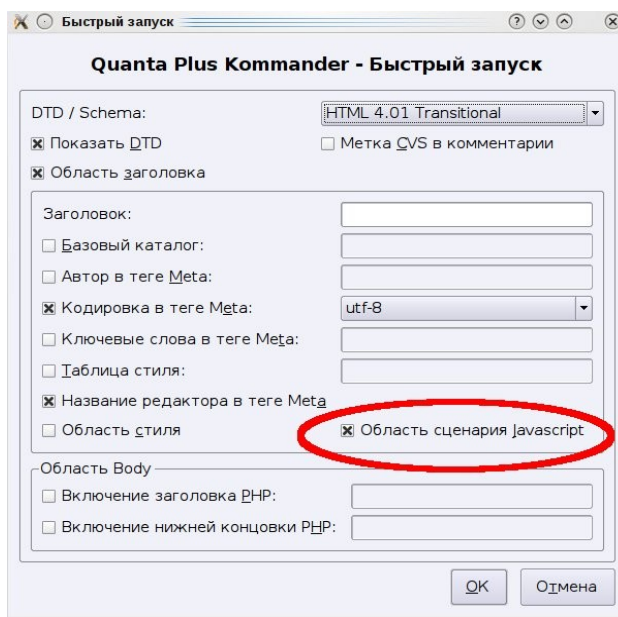
Запускаем текстовый редактор и набираем:

```
<!DOCTYPE HTML>
<html>
  <script type = "text/JavaScript">
    alert("Hello, World!!!");
  </script>
</html>
```

Сохраните в файле например *hello.html* и откройте его в браузере. если все правильно, то должно появиться сообщение:



Если использовать IDE Quanta Plus, то при изучении основ JavaScript достаточно просто набрать указанный текст в окне редактора Quanta и воспользоваться встроенным браузером. Кроме того, можно автоматически создать скелет документа для этого выберите "теги+стандартная+быстрый запуск" в появившемся диалоге установите флажок «Область сценария JavaScript»



В результате получите:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title></title>
  <meta name="GENERATOR" content="Quanta Plus">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <script type="text/javascript">
    // <!--
    // именно здесь пишем код!!!
    // -->
  </script>
</head>
<body>

</body>
</html>
```

Контрольные вопросы

1. Назовите исполнителей для которых пишут на JavaScript.
2. В каком теге размещается код на JavaScript?
3. Какая функция позволяет выводить информацию в модальный диалог?
4. Какое программное обеспечение необходимо для разработки приложений на JavaScript?

Упражнения.

1. Измените программу примера так, чтобы выводились фразы:
 - а) "JavaScript"
 - б) "Программируем для браузера!!!"

проверьте работоспособность примеров в доступных браузерах.

2 Вычисление значений выражений.

Для того, чтобы вычислить значение выражения $3(12-6)$ нужно действовать по алгоритму:

1. из 12 вычесть 6;
2. результат первого действия умножить на 3;
3. сообщить результат второго действия.

Составление выражений ведущих к решению задачи тоже программирование.

Требования к записи выражений:

1. расставлять знаки всех операций.
2. дробная черта заменяется делением.
3. у десятичной дроби целая от дробной части отделяется точкой.
4. если необходимо, то числитель и знаменатель заключается в скобки.

Обозначение арифметических операций:

+	сложение
-	вычитание
*	умножение
/	деление (для целых чисел - нацело)
%	нахождение остатка
++	увеличение на 1
--	уменьшение на 1

Например:
$$\frac{5,6 - 11(0,12 + 4)}{6,11 + 3,2^2}$$

на языке программирования записывается так: $(5.6 - 11 * (0.12 + 4)) / (6.11 + 3.2 * 3.2)$.

Для того, чтобы выводить значения выражений можно использовать уже известную функцию `alert`. Например:

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<script language = "JavaScript" type="text/javascript">
    alert(( 5.6 - 11 * ( 0.12 + 4 ) ) / ( 6.11 + 3.2 * 3.2 ));
</script>
</head>
</html>
```

Контрольные вопросы

1. Каковы требования к записи выражений на языке программирования?
2. Как обозначают основные арифметические операции?

Упражнения.

2. Напишите программы вычисления значений выражений.

а) $56,5(11-5,8)$

б) $\frac{23-1,123(3,5+6)}{5^3-6}$

в) $(4,6-1)^4-87,3$

г) $1,1+\frac{89^2}{5,6}+7$

3. Запишите на языке алгебраических формул:

а) $567.5 * 4 - 5 * 5;$

б) $33.7 * (23 - 4.5 * 4.5);$

в) $(4 - 6.78)/(23*5);$

г) $5.45 - 24 / 67 + 4.5$

4. Вычислите.

а) $23/5-8*3;$

б) $6-78/(4+2);$

в) $37\%5 -2+6;$

г) $(56-23\%7)+76/4;$

5. Напишите программы решения задач:

а) Найти площадь и периметр прямоугольника если его длина — 5м, а ширина 20,45см.

б) Автомобиль движется со скоростью 5.4 м/с. Сколько времени ему понадобится, чтобы проехать 55 км.

в) Сторона куба 3,4см. Сколько метров составляет объем куба.

г) С полуночи прошло 5 ч 34 м 20 с. Сколько секунд осталось до полудня.

3 Переменная. Операция присваивания. Организация ввода.

Для хранения информации используют переменные.

Переменная — это именованная область памяти.

Прежде чем использовать переменную, ее желательно определить.

Общий вид определения переменной:

```
var имя1={начальное значение};
```

В одном блоке var можно определить несколько переменных через запятую.

Примеры:

```
var a=0,b=0;//две переменных для чисел.
```

Основной способ записи значения в переменную — *операция присваивания*.

Синтаксис операции присваивания:

```
имя = выражение;
```

Например:

```
result = 2 * ( 56.4 — 4 );//в result будет записано 104.8
```

Найти площадь круга S и длину окружности C , если радиус 4,67см.

Переменные S и C должны быть целого типа. Расчетные формулы $S = \pi \cdot r^2$; $C = 2 \cdot \pi \cdot r$, таким образом:

```
<script language="JavaScript">
var s=0,c=0;
s = 3.14 * 4.67 * 4.67;
p = 2 * 3.14 * 4.67;
alert("s="+s+"p="+p);
</script>
```

Для того, чтобы организовать запрос значения переменной в процессе выполнения, можно использовать функцию prompt:

```
имя = prompt(заголовок окна, значение по умолчанию)
```

Например, пусть браузер спросит наше имя, и потом поприветствует нас.

```
<script language="JavaScript">
    var name=prompt("Как Вас зовут?","");
    alert("Привет, "+name);
```



```
</script>
```

Обратите внимание: *несколько выражений сцепляются в одну операцию +*.

Упражнения.

1. Что будет записано в переменные:

а) $tt = 6 * 7 - 11$;

б) $result = 5 \% 2 + 7$;

в) $y = 5,67 - 8/7$;

г) $r = 34 / 2 + 5$;

2. Напишите программы решения задач:

а) Найти скорость **v**, если расстояние 2345 км, пройдено за **t** ч. (движение равномерное и прямолинейное);

б) Определить сколько минут **mm** содержится в **hh** часах **ss** секундах;

3.1 Модельный ввод и вывод.

Использование функций *alert* и *prompt* позволяет организовать модальный ввод и вывод. То есть для каждой из них определен диалог, внешний вид которого изменить невозможно.

На практике чаще используют другой — модельный. Для этого в самом документе создают форму в которую включают необходимые элементы интерфейса: поля ввода, флажки, переключатели и т. п.

Вся форма заключается в тег `<form>`, а элементы интерфейса определяются тегами `<input ...>`

Например необходимо предоставить возможность пользователю ввести свои анкетные данные: фамилию, имя, отчество, год рождения.

```
<form name="anketa">
  Фамилия:<input type="text" name="surname"><br>
  Имя:<input type="text" name="myname"><br>
  Отчество:<input type="text" name="secondname"><br>
  Год рождения:<input type="text" name="year"><br>
  <input type="button" value="Передать">
</form>
```

Если все правильно, то в окне просмотра Вы увидите форму:

Фамилия:

Имя:

Отчество:

Год рождения:

Вы можете ввести данные, однако при нажатии на кнопку передать ни каких действий происходить не будет. Для того чтобы организовать обработку информации, нужно описать алгоритм обработки данных формы. Для это лучше создать отдельную функцию.

Синтаксис определения функции:

```
function имя_функции(аргументы)
{
    //операторы
}
```

Задача. Пусть в результате обработки данных формы anketa, браузер сообщит возраст пользователя.

В теге <script> пишем:

```
function hello_friend()
{
    var myform=document.anketa;
    var d=new Date(),l=0;
    var fio=myform.surname.value+" "+myform.myname.value+" "
    +myform.secondname.value;
    alert("Здравствуйте, "+fio);
    l=1900+d.getYear()-(+myform.year.value);
    alert("Вам "+l+" лет");
}
```

Затем в объекте input типа button определяем отклик на событие onclick, вызывая определенную нами функцию hello_friend():

```
<input type="button" value="Передать" onclick="hello_friend();">
```

Запускаем просмотр, и, если все правильно то радуемся результату.

Упражнения.

1. Напишите программу, вычисления значения переменной, организовав модельный диалог:

а) $y=5x^2-1$

б) $y=2a-3b$

в) $y=(a+b+c)^3$

г) $y=x^3-\sin x+\frac{2}{(x^2+1)}$

2. Напишите программу расчета возраста

а) В полных годах.

б) В месяцах.

в) В днях

г) В минутах.

3. Напишите программу вычисления:

а) Площади квадрата по его диагонали.

б) Объема прямоугольного параллелепипеда по его измерениям.

в) Площади трапеции.

г) Площади параллелограмма.

4 Управляющие операторы.

К управляющим относятся:

1. Условный.
2. Оператор выбора.
3. Операторы цикла.

4.1 Условный оператор.

Очень часто встречаются задачи в которых, прежде чем выполнять действие, нужно проверить условие. Например, если будет введен неверный пароль вход должен

быть закрыт.

Для этого служит условный оператор.

Синтаксис оператора.

```
if(условие){  
    //серия 1;  
}  
else {  
    //серия 2;  
}
```

выполнение.

1. *проверяется условие.*
2. *Если условие верно выполняется серия 1, иначе выполняется серия 2.*

вот программа проверки пароля (паролем является слово "Java"):

```
<script language = "JavaScript" type="text/javascript">  
    var pass=prompt("Пароль?","");  
    if(pass == "Java"){  
        alert("Вход разрешен!");  
    }  
    else {  
        alert("неверный пароль!!!");  
    }  
</script>
```

Для записи неравенств используются операции отношения. К ним относятся

== - равно.

<= - меньше или равно.

>= - больше или равно.

!= - не равно.

4.1.1 Регулярные выражения.

Регулярные выражения — это шаблоны соответствия строки определенным требованиям.

Например:

.* - любое количество любых символов.

[abcsv] одна из четырех букв a, b, c или d.

Использование регулярных выражений позволяет быстро определить соответствие строки определенному шаблону или поиск таких соответствий в тексте:

Например:

/[xyz]*/ -любое количество одной из букв x y или z

Упражнения.

1. Напишите программы проверки

а) Правильности ввода четного числа.

б) Правильности ввода ip-адреса.

в) Правильности ввода доменного имени.

г) Правильности ввода пароля (латинские буквы, цифры, без пробелов, не менее 6-ти символов)

4.2 Операторы цикла.

Цикл — это повторение действия, пока верно условие.

Условие можно проверять перед выполнением действия (цикл с предусловием) или после него (цикл с постусловием).

Для организации цикла существует несколько операторов:

1. оператор while (пока)
2. оператор for (для)
3. оператор do/while (делать/пока)

Первые два служат для организации цикла с предусловием, а последний — с постусловием.

Оператор while.

Синтаксис оператора:

```
while(условие)
{
    //операторы
}
```

Выполнение:

1. *проверяется условие.*

2. Если условие верно, то выполняется оператор, иначе переход к пункту 4.
3. Переход к пункту 1.
4. Следующий оператор.

Упражнения.

1. Добиться ввода задуманного числа.
2. Трехзначное число — пароль. Добиться ввода пароля не более чем с 4 попыток.

5 Составные переменные.

В большинстве реальных задач требуется специальным образом организовать хранение информации. В этом случае используют составные переменные. К ним относятся массивы, файлы, структуры(записи), объекты. Здесь будут рассмотрены массивы и файлы.

5.1 Массивы.

Для хранения многих однотипных данных удобно использовать составную переменную — массив.

Массив — это совокупность пронумерованных элементов, как правило, одного типа.

Вся последовательность имеет имя, а конкретный элемент имеет номер.

На языке Javascript нумерация элементов массива ведется с 0.

Пример:

массив **a[11]** из 11 целых чисел

0	1	2	3	4	5	6	7	8	9	10
345	12	56	98	11	5	98	4	11	8	6

a[7]=4

Определение массива

```
var имя = []; //пустой массив
var имя = new Array(количество элементов); //массив с заданным количеством
элементов.
var имя = [элемент1, элемент2, элемент3, ....] //массив заданный пречислением
элементов.
```

Например:

```
var a=[]; //пустой массив
var x = new Array(10); //массив из 10-ти элементов
var s=["red","green","blue"]; //массив из 3-х слов.
```

Виды задач на массивы:

1. Заполнение массива;
2. Обработка массива.

5.1.1 Заполнение массивов.

По формуле n-го элемента.

Заполнить массив из 12 целых чисел первыми четными.

```
var a= new Array(12),i;
for(i=0;i<12;i++) a[i]=2*i;
for(i=0;i<12;i++) document.write(a[i]+" ");
```

По рекуррентному соотношению

Заполнить массив из 8 действительных чисел, если $a[0]=1,1$ $a[i]=2*a[i-1]+1$

```
var a = new Array(8),i;
a[0]=1.1;
for(i=1;i<8;i++) a[i]=2*a[i-1]+1;
for(i=0;i<8;i++) document.write(a[i]+" ");
```

Из потока ввода.

Пример

Заполнить массив $a[5]$ со стандартного потока ввода (клавиатура).

```
var a = new Array(5), i;
for(i=0;i<5;i++) a[i]=prompt("Введите a["+i+"]:","0");
for(i=0;i<5;i++) document.write(a[i]+" ");
```

Случайными числами.

Случайное число получается в результате функции *random()* из класса *Math*. Это число принадлежит отрезку $[0;1]$. Чтобы взять только целую часть нужно использовать метод *floor* из этого же класса.

Чтобы получить случайное целое число из отрезка $[a;b]$ можно использовать

формулу $\text{Math.floor}(\text{Math.random()}*(b-a))+a$;

Пример.

Заполнить массив `int a[10]` случайными числами из отрезка `[10;99]`

```
var a = new Array(10), i;  
for(i=0;i<10;i++) a[i] = Math.floor(Math.random(99-10))+10;  
for(i=0;i<10;i++) document.write(a[i]+" ");
```

Упражнения.

1. Заполнить массив:

а) `a[17]` числами кратными 5;

б) `b[9]` первыми элементами последовательности Фибоначчи.

в) `x[7]` с клавиатуры.

г) `m[10]` случайными трехзначными числами.

2. Заполнить массив `a[20]` элементами последовательности
$$a_i = \begin{cases} i+1 & i < 3 \\ 2*a_{i-1} & i \geq 3 \end{cases}$$

3. Заполнить массив `a[14]` элементами последовательности:

а) 1 0 1 0 1 0 ...

б) 1 2 3 1 2 3 1 2 3 ...

в) 1 2 3 2 3 4 3 4 5 ...

г) * 1 0 1 1 0 1 1 1 ...

5.1.2 Обработка массива

Поиск элементов.

Пример

Найти минимальный элемент в массиве двузначных чисел `int a[12]`

```
var a= new Array(10),i,min;  
for(i=0;i<10;i++) a[i]=Math.floor(Math.random()*99)+10;  
min = a[0];  
for(i=1;i<10;i++)  
    if(a[i]<min) a[i]=min;  
for(i=0;i<10;i++) document.write(a[i]+" ");
```



```
document.write("<br");  
document.write("min="+min);
```

Поиск и замена.

Пример.

В массиве a[19] трехзначных чисел заменить все четные числа нулями.

```
var a= new Array(19),i;  
for(i=0;i<10;i++) a[i]=Math.floor(Math.random()*999)+100;  
for(i=0;i<10;i++)  
    if(a[i]%2==0) a[i]=0;  
for(i=0;i<10;i++) document.write(a[i]+" ");
```

Изменение порядка элементов.

Пример.

Упорядочить массив целых неотрицательных чисел меньших 50 int a[10] в порядке возрастания.

```
var a=new Array(19),i,j,b;  
for(i=0;i<10;i++) {a[i]=Math.floor(Math.random()*50); document.write(a[i]+" ");}  
for(i=0;i<10;i++)  
    for(j=9;j>i;j--)  
        if ( a[j]<a[j-1] ) {b=a[j];a[j]=a[j-1];a[j-1]=b;}  
  
document.write("<br>");  
for(i=0;i<10;i++) document.write(a[i]+" ");
```

1. Дан массив двузначных чисел $a[10]$. Найти

а) Номера четных элементов;

б) Элементы большие 50;

в) Элементы, оканчивающиеся на 5.

г) Элементы, сумма цифр которых больше 10.

2. Дан массив трехзначных чисел $a[15]$. Определить какие из них не могут входить в ip адрес и заменить их нулями.

3. В массиве целых неотрицательных чисел меньших 20 $a[30]$ найти:

а) минимальный.

б) максимальный

в) сумму минимального и максимального.

г) * первый элемент больший минимального.