Лабораторная работа 9. Команды цикла

I. Общий вид цикла while и пример его использования:

Общий вид	Пример

`while (условие) команда;`	`var i = 1;`
	`var sum = 0;`
	`while(i <= 100)`
	`{`
	`sum += i;`
	`i ++;`
	`}`
	`alert("Сумма 1 + 2 +...+ 100 = " + sum);`

Полный текст программы с этим примером выглядит так:

```
<HTML>
   <HEAD>
     <TITLE>Опыты с командой while</TITLE>
   </HEAD>
   <BODY bgcolor=white text=black>
    <H2>Опыты с командой while</H2>
    <HR>
    <SCRIPT language=JavaScript>
    <!--
       var i = 1;
        var sum = 0;
        while(i <= 100)
         { sum += i; i ++;}
        alert("Сумма 1 + 2 +...+ 100 = " + sum);
    //-->
    </SCRIPT>
   </BODY>
</HTML>
```

Цикл работает так. Сначала проверяется условие. Если оно истинно, выполняется команда (тело цикла). И эти действия повторяются, т. е. снова проверяется условие, и если оно истинно, выполняется тело цикла, и т. д.

Цикл заканчивает работу, когда при очередной проверке условие оказывается ложным. Так как условие проверяется перед выполнением тела цикла, то команды, входящие в цикл, могут не выполниться ни разу.

Задания

1. По аналогии проверьте, какое сообщение будет выведено в окошко `alert` при выполнении следующих примеров. Если есть ошибки, устраните и поясните причину.

Пример 1	Пример 2

`var x = 5;`
`var s=0;`
`while(x) {s += x; x --;}`
`alert(s);`
	`var x = 5;`
	`var s = 0;`
	`while(-- x) s += x;`
	`alert(s);`

Пример 3	Пример 4

`var x = 5;`
`var s = 0;`
`while(x--) s += x;`
`alert(s);`
	`var x = 5;`
	`var s = 0;`
	`while(s) s += x;`
	`alert(s);`

Пример 5	Пример 6

`var x = 5;`
`var s = 0;`
`while(!x) s += x;`
`alert(s);`
	`var x = 5;`
	`var s = 0;`
	`while (--x) s += x; s ++;`
	`alert(s);`

Пример 7	Пример 8

`var x = 5;`
`var s = 0;`
`while(-- x && s < 10) s += x;`
`alert(s);`
	`var x = 5;`
	`var s = 0;`
	`while(-- x || s < 10) s += x;`
	`alert(s);`

Пример 9	Пример 10

`var x = 5;`
`var s = 0;`
`while(-- x || !s) s += x;`
`alert(s);`
	`var x = 5;`
	`var s = 0;`
	`while(-- x && s) s += x;`
	`alert(s);`

2. Определить пятый член последовательности и вывести его в окно `alert`:
   $a_1 = 2$; $a_n = a^2_{n-1} + 1$.
3. Определить сумму сл. последовательности и вывести её в окно `alert`:
   summa = 1+1/2+1/3+. . . + 1/10
4. Определить сумму сл. последовательности с точностью 0,0001 и вывести её в окно `alert`:
   summa = $1+1/2+1/4+\ldots+1/2^n+\ldots$
5. Определить суммы нечетных и четных чисел от 1 до 200 и вывести её в окно `alert`:

summa1 = 1+3+5+. . . + (2n-1) + . . . + 199; summa2 = 2+4+6+. . . + 2n + . . . + 200.

II. Общий вид цикла for и пример его использования:

Общий вид: `for(начало;условие;приращение)команда;`

Пример:

1 вариант	2 вариант

`var i;`
`var sum = 0;`
`for(i = 1; i <= 100; i ++)`
`  sum += i;`
`alert("Сумма 1 + 2 + ...`
`  + 100 = " + sum);`
	`var sum = 0;`
	`for(var i = 100; i; i --)`
	`  sum += i;`
	`alert("Сумма 1 + 2 + ...`
	`  + 100 = " + sum);`

Команда, помещаемая в начало, выполняется до циклического повторения (в примере это команда i = 1; или i =100;),а сам цикл образуете следующими действиями:

- проверка условия (в примере `i <= 100` или `i`);
- выполнение тела цикла (в примере `sum += i`);
- выполнение команды, записанной в разделе `приращение` (в примере `i ++` или `i --`).

Как и в команде while, тело цикла может не выполниться ни разу, если условие ложно с самого начала. Не выполнится при этом и команда из разделов `приращение`. А вот команда из раздела `начало` выполняется всегда, независимо от условия, и выполняется ровно один раз.

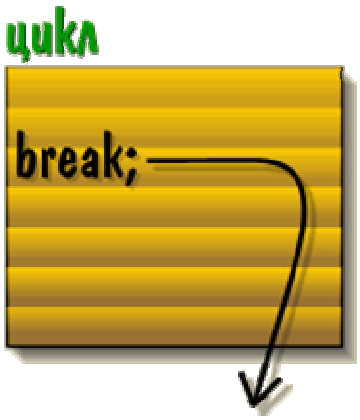
В заголовке цикла `for` любая из трех конструкций **начало, условие, приращение** может быть опущена, при этом соответствующую точку с запятой опускать нельзя. Когда опущено условие, считается, что оно имеет значение `true`. Таким образом, цикл превращается в бесконечный: `for (;;) команда`. Этот цикл не остановится, если только не будет содержать внутри себя команду `break`.

Команды *break* и *continue*

Эти команды используют в теле цикла для изменения последовательного хода выполнения команд (рис.).

Команда continue заставляет браузер пропустить выполнение всех команд после нее и до конца тела цикла. Но цикл продолжается.

Команда `break` заставляет браузер немедленно прекратить выполнение цикла.



Пример 1 (continue)

Найти сумму 5 четных чисел, случайным образом взятых из диапазона [1, 20].

```
<HTML>
   <HEAD> <TITLE>Сумма чисел</TITLE> </HEAD>
   <BODY bgcolor=white text=black>
       <H1>Сумма  5 четных чисел</H1>
       <HR>
       <SCRIPT language=JavaScript>
       <!--
        var len    =5; // Количество чисел.
        var a = 1; // Левая граница интервала,
        var b = 20; // Правая граница интервала.
        var sum    = 0; // Сумматор.
        var counter =0; // Счетчик чисел.
        var number; // Случайное число.
        var str = ""; // Строка для вывода.
        while(counter < len)
         {
          number = Math.floor(a + (b - a + 1) * Math.random());
          if(number%2) continue;
          sum += number;
          str += number;
          if(counter < len - 1) str += " + ";
          else  str += " = ";
          counter++;
         }
         str += sum;
         alert(str};
       //-->
       </SCRIPT>
    </BODY>
  </HTML>
```

Пример 2 (break). Целые числа случайным образом генерируются из диапазона [1, 20]. Требуется суммировать эти числа до тех пор, пока очередное случайное число не станет равным 10.

```
var a = 1; // Левая граница интервала.
var b = 20; // Правая граница интервала.
var special = 10; // Критическое значение случайного числа.
var sum = 0; // Сумматор.
var number; // Случайное число.
var str = ""; // Строка для вывода.
for(;;) // Бесконечный цикл.
{
 number = Math.floor(a + (b - a + 1) * Math.random());
 sum += number;
 str += number;
 if(number == special) break;
 str += " + ";
}
str += " = " + sum;
alert (str);
```

Результат работы примера может быть таким, как на рис., или числа могут быть гораздо больше.

Пример 3. Дана функция $y = x^2 - \sqrt{x} + 1.5$ Требуется найти значения **y** при изменениях аргумента **x** от 0 до 3 с шагом 0,5. Печатать значения **x** и **y** до тех пор, пока очередное значение **x** не станет равным 3.

```
var x0 = 1;
var xk = 3;
var dx = 0.5;
var x = x0;
var y;
while(x <= xk)
{y = x*x-Math.sqrt(x)+1.5;
 alert(" x= " + x + " y = " + y);
 x += dx
}
```

Пример 4. Дана функция $$y = \begin{cases} x^2 + x + 1, & \text{åñëè } \tilde{o} \le 0 \\ x - \sqrt{x+1}, & \text{åñëè } \tilde{o} > 0 \end{cases}$$

Требуется ввести значения x и найти значения y.

```
var x = prompt("Введите значение x:", "1");
y = (x <= 0) ? x*x+x+1: x-Math.sqrt(x+1);
alert ("x="+x+"  y=" + y);
```

Теперь попробуем изменить значения x от 0 до 3 сшагом 0,5 и найти значения y.

```
var  x0 = 0;     // Начальное значение x
var  xk = 3;     // Конечное значение x
var  dx = 0.5;   // шаг изменения x
var  x=x0;       // Присвоение x начального значения
var y;           // Объявление переменной  y
while (x =< xk)
{  y = x*x - Math.sqrt(x) +1.5;
   alert(" x= " + x + " y= " + y);
   x+=dx ;
}
```

Пример 5: Определить количество цифр в целом положительном числе.

```
// Количество цифр в целом положительном числе num.
// Вход: num (целое положительное число).
```

```
// Выход: количество цифр в num.
 function F(num)
 {
  var len = num ? 0 : 1;
  while(num)
  {   num = (num - num % 10) / 10;
     len ++;
  }
  return len;
 }
  var sum = 0;
  var num1, num2, num3;
  num1 = prompt("Введите первое число", "");
  sum += F(num1);
  num2 = prompt("Введите второе число", "");
  sum += F(num2) ;
  num3 = prompt("Введите третье число", "");
  sum += F(num3) ;
  alert("Общее число введенных цифр: " + sum);
```

Задания

1. По аналогии проверьте, какое сообщение будет выведено в окошко `alert` при выполнении следующих примеров. Если есть ошибки, устраните и поясните причину.

Пример 11

```
var s = 0;
for(var i = 1; i < 10; i ++)
     s += i;
alert(s);
```

Пример 12

```
var s = 0;
for(var i = 1; i < 10; i += 2)
     s += i;
alert(s);
```

Пример 13

```
var s = 0;
for(var i = 10; i; i --) s += i;
alert(s);
```

Пример 14

```
var s = 0;
for(var i = 10; -- i;) s += i;
alert(s);
```

Пример 15

```
var s = 0;
for(var i = 10; i --) s += i;

alert(s);
```

Пример 16

```
var s = 0;
for(var i = 10;;)
(if(-- i) break; s +- i;}
alert(s);
```

Пример 17

```
var s = 0;
var i = 10;
for(;;)
{if(!(-- i) break; s += i;}
alert(s);
```

Пример 18

```
var s = 0;
var i = 10;
for(;;)
{if(s > 6) break; s += -- i; }
alert(s);
```

Пример 19

```
var s = 0;
for(var i = 10;-- i; s += i);

alert(s);
```

Пример 20

```
var s = 0;
for(var i = 10; -- i;)
{if(i > 5) continue; s += i;}
alert(s);
```

Пример 21

```
var s = 0;
var i =10;
while(-- i)
{if(i > 5) continue; s += i;}
alert(s);
```

Пример 22

```
var s = 0;
var i = 10;
while(-- i)
{if(i < 5) break; s += i;}
alert(s);
```

Во всех следующих заданиях под числом понимается натуральное число.

2. Напишите скрипты с циклом `for` затем и `while` для решения следующих задач:

а) найти 10! = 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10;

б) найти сумму цифр числа;

в) найти произведение цифр числа;

г) найти наибольшую цифру числа;

д) найти наименьшую цифру числа;

е) найти количество цифр «3» в числе;

ж) найти количество цифр «3» и «5» в числе;

з) вывести цифры числа в обратном порядке;

и) каждую цифру числа заменить дополнением до 9;

к) между цифрами числа записать знак «+»;

л) удвоить каждую цифру числа.

3. Напишите скрипты с циклом `for` для решения следующих задач:

а) найти 10! = 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10;

б) найти сумму 10 случайных чисел из диапазона [1, 100];

в) найти сумму 10 случайных четных чисел из диапазона [1, 100];

г) найти пятый член последовательности

$a_1 = 2$; $a_n = a^2_{n-1} + 1$;

д) перевести двоичное число из 8 единиц в десятичную систему;

е) перевести шестнадцатеричное число из 8 единиц в десятичную систему;

ж) перевести троичное число из 8 двоек в десятичную систему;

з) построить 10-значное число, в котором цифра на i-м месте определяется как последняя цифра числа i^2;

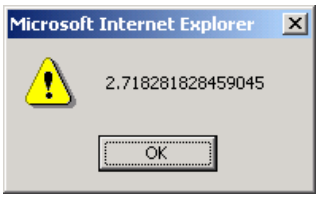
и) построить 10-значное число, в котором цифра на i-м месте определяется как последняя цифра члена последовательности $a_j = 11 * i + 13$;

к) проверить, выдает ли функция math.random() числа 0 и 1.

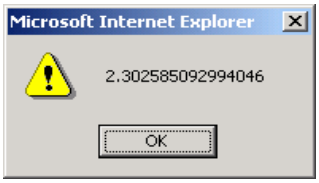
л) подсчитать число чисел, которые попадут в интервале (0,4, 0,6) за 1000 обращений к функции math.random().

7.1. Стандартные константы

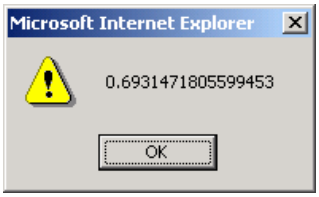
1. Основание натурального логарифма - E.
Пример:Alert(Math.E)



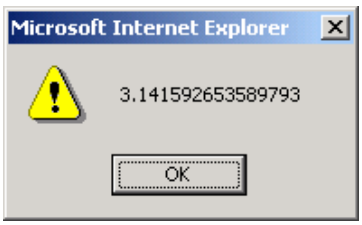
2. Натуральный логарифм числа 10 - LN10.
Пример:Alert(Math.LN10)



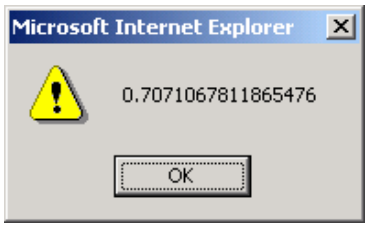
3. Натуральный логарифм числа 2 - LN2.
Пример:Alert(Math.LN2)



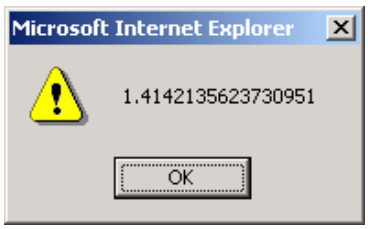
4. Число ПИ – PI
Пример:Alert(Math.PI)



5. Квадратный корень из 1/2 – SQRT1_2
Пример:Alert(Math.SQRT1_2)

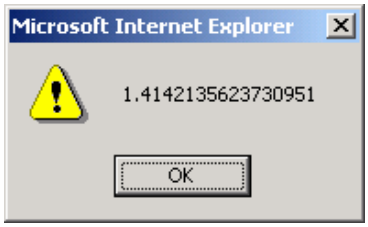


6. Квадратный корень из 2 – SQRT2
Пример:Alert(Math.SQRT2)

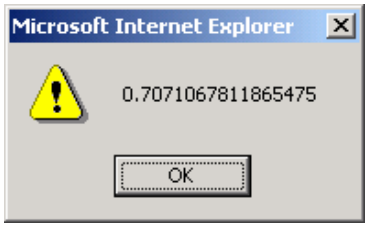


7.2. Стандартные функции

1.Абсолютное значение аргумента – abs(arg)
Пример:
```
var x = -25;
var y
y = Math.abs(x);
alert(y);
```

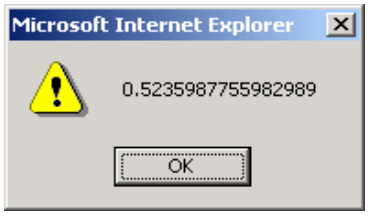


2.Тригонометрические функции: sin(arg), cos(arg), tan(arg)
Пример:
```
var x = Math.PI/4;
var y
y = Math.sin(x);
alert(y);
```



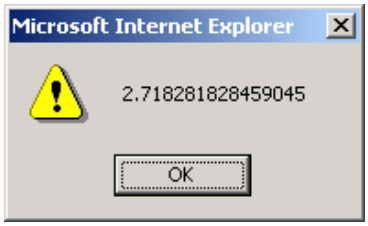
3.Обратные тригонометрические функции:
asin(arg),acos(arg),atan(arg)

```
var x = 0.5;
var y
y = Math.asin(x);
alert(y);
```



3. Экспонента и натуральный логарифм:
exp(arg), log(arg)

```
var x = 1;
var y
y = Math.exp(x);
alert(y);
```



5. Ближайшее целое число, большее или равное аргументу: ceil(arg)

```
var x = -2.69;
var y
y = Math.ceil(x);
alert(y);
```

6. Ближайшее целое число, меньшее или равное аргументу: `floor(arg)`

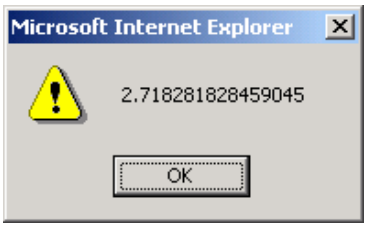
```
var x = -2.69;
var y
y = Math.floor(x);
alert(y);
```

7. Округляет аргумент до ближайшего целого: `round(x)`

```
var x = -2.69;
var y = Math.round(x);
alert(y);
```

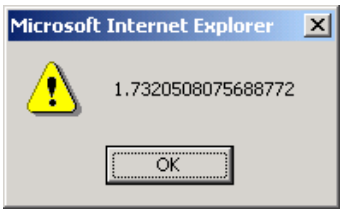
7. Экспонента и натуральный логарифм: `exp(arg)`, `log(arg)`

```
var x = 1;
var y
y = Math.exp(x);
alert(y);
```



9. Корень квадратный из аргумента: `sqrt(arg)`

```
var x = 3;
var y
y = Math.sqrt(x);
alert(y);
```



10. Максимальное или минимальное значение из двух числовых аргументов: `max(arg1, arg2)`, `min(arg1, arg2)`

```
var x = 3;
var y = Math.min(x,10);
alert(y);
```

11. Число в степени : `pow(arg1, arg2)`

Вычисляет arg1 в степени `arg2var x = 2;`

```
var y = Math.pow(x,10);
alert(y);
```

12. Выдает случайное число из диапазона [0,1]: `random()`

```
alert(Math.random());
```