

# «Массивы, циклы»



JavaScript  
Courses

[www.courses.dp.ua](http://www.courses.dp.ua)

## Если какие-либо действия нужно повторять, но заранее неизвестно сколько раз

```
1 <script>
2     var password = prompt("Введите пароль: ");
3
4     if((password == "12345")){
5         alert("Access approved.");
6     }else{
7         alert("Wrong password!");
8     }
9 </script>
```

*Если пароль не подходит, то нужно повторно запросить его у пользователя, и так повторять до тех пор пока не будет введён правильный пароль.*

*Т.е. нам нужен механизм который будет **повторять набор действий до тех пор пока будет верно условие** (например: пароль не равен «12345»)*

## Циклы – способ многократно повторить фрагмент кода

*Цикл `while / do..while` выполняет фрагмент кода пока условие заданное в нём верно (истинно, `true`).*

```
1  <script>
2
3  do{
4      var password = prompt("Введите пароль: ");
5
6      if((password == "12345")){
7          alert("Access approved.");
8      }else{
9          alert("Wrong password!");
10     }
11
12     }while(password != "12345");
13
14 </script>
```

*В данном примере помимо проверки данных оператором `if` еще следует проверка данных оператором `while`.*

# Цикл while/do..while – выполняет фрагмент кода, до тех пор пока верно условие в его заголовке

```
1 <script>
2   var temp_limit = 100;
3   var current_temp = 25;
4
5   while(temp_limit > current_temp){
6       console.log("Current/Limit: " + current_temp + "/" + temp_limit);
7       current_temp++;
8   }
9
10  console.log("The temperature limit is reached");
11 </script>
```



Current/Limit: 95/100	ex02.html:7
Current/Limit: 96/100	ex02.html:7
Current/Limit: 97/100	ex02.html:7
Current/Limit: 98/100	ex02.html:7
Current/Limit: 99/100	ex02.html:7
The temperature limit is reached	ex02.html:11

## Цикл while/do..while – выполняет фрагмент кода, до тех пор пока верно условие в его заголовке

```
1 <script>
2   var temp_limit = 100;
3   var current_temp = 25;
4
5   while(temp_limit > current_temp){
6       console.log("Current/Limit: " + current_temp + "/" + temp_limit);
7       current_temp++;
8   }
9
10  console.log("The temperature limit is reached");
11 </script>
```

*В условии используют переменные, с расчётом на то, что они будут менять своё значение и со временем условие станет ложным и цикл*

**!!!В теле цикла должны происходить какие-либо изменения тех переменных которые находятся в условии, иначе цикл будет выполняться вечно!!!**

# while, do/while, скобки

```
1 <script>
2
3   var a = 3;
4
5   var b = 7;
6
7   while(a < b) {
8
9       //Какие-то действия...
10      a++; //!!!
11
12  }
13
14  console.log(a);
15 </script>
```

```
1 <script>
2
3   var a = 3;
4
5   var b = 7;
6
7   do{
8       //Какие-то действия...
9       a++; //!!!
10
11  }while(a < b);
12
13  console.log(a);
14
15 </script>
```

**While** – проверяет условия перед входом в цикл, **do/while** после выполнения каждой итерации (шага) цикла. Т.е. в цикле **do/while** тело выполниться

минимум один раз.

В теле цикла должно происходить **что-то**, что повлияет на **условие цикла**, и рано или поздно заставит цикл прекратиться. Иначе цикл станет бесконечным.

# Что происходит с переменными которые «участвовали» и изменялись в цикле

```
1 <script>
2   var temp_limit = 100;
3   var current_temp = 25;
4
5   while(temp_limit > current_temp){
6     console.log("Current/Limit: " + current_temp + "/" + temp_limit);
7     current_temp++;
8   }
9
10  console.log("The temperature limit is reached");
11
12  console.log("Last current temp: " + current_temp);
13 </script>
```



Current/Limit: 97/100	ex02.html:6
Current/Limit: 98/100	ex02.html:6
Current/Limit: 99/100	ex02.html:6
The temperature limit is reached	ex02.html:10
Last current temp: 100	ex02.html:12

*Без сюрпризов. Все изменения которые вносились в переменные по ходу работы цикла сохраняются и после завершения его работы.*

# Массивы

```
var a = [456, "lalala", 12.78, true];
```

↑            ↑            ↑            ↑  
0            1            2            3

**Массив** – это нумерованный набор



*Оператор [] – основной признак массива, он позволяет обратиться по номеру к конкретному элементу массива.*

[http://www.w3schools.com/js/js\\_arrays.asp](http://www.w3schools.com/js/js_arrays.asp)

[http://www.w3schools.com/js/js\\_array\\_methods.asp](http://www.w3schools.com/js/js_array_methods.asp)

# В JavaScript массивы представляют собой гибрид классических массивов, стека, очереди и ассоциативных массивов.

```
1 <script>
2   var arr = [1, 3, "Elena", true];
3
4   console.log(arr);
5   console.log(arr.length);
6   //Свойство указывающее длину массива.
7   console.log(typeof(arr));
8
9   console.log(arr[0], typeof(arr[0]));
10  console.log(arr[2], typeof(arr[2]));
11  console.log(arr[3], typeof(arr[3]));
12
13  arr[0] = "Ivan";
14  console.log(arr[0], typeof(arr[0]));
15
16  arr.push(777);
17  //Добавление элемента в конец массива
18  console.log(arr, arr.length);
19
20  var x = arr.pop();
21  //Удаление последнего элемента из массива
22  console.log(arr, arr.length);
23  console.log(x);
24
25  arr.unshift("Julia");
26  //Добавление элемента в начало массива
27  console.log(arr, arr.length);
28
29  arr.shift();
30  //Удаление первого элемента массива
31  console.log(arr, arr.length);
32
33 </script>
```



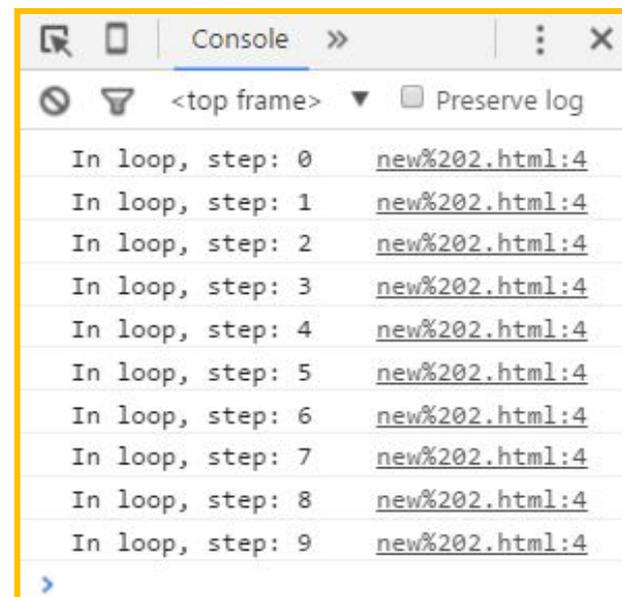
[1, 3, "Elena", true]	example.html:4
4	example.html:5
object	example.html:7
1 "number"	example.html:9
Elena string	example.html:10
true "boolean"	example.html:11
Ivan string	example.html:14
["Ivan", 3, "Elena", true, 777] 5	example.html:18
["Ivan", 3, "Elena", true] 4	example.html:22
777	example.html:23
["Julia", "Ivan", 3, "Elena", true] 5	example.html:27
["Ivan", 3, "Elena", true] 4	example.html:31

*В JavaScript массивы не типизированы, т.е. могут одновременно хранить элементы*

# Цикл for – когда известно сколько раз нужно повторить действия

*Нужно повторят последовательность действий заранее известное количество раз. В таком случае, как правило, применяется цикл **for**.*

```
1 <script>
2
3   for(var i = 0; i < 10; i++){
4     console.log("In loop, step: " + i);
5   }
6
7 </script>
```



*Цикл **for** хорош тем, что в нём есть переменная-счётчик, которая последовательно принимает значения.*

[http://www.w3schools.com/js/js\\_loop\\_for.asp](http://www.w3schools.com/js/js_loop_for.asp)

# Цикл for и массивы

```
1 <script>
2   var mas = [5, 76, "mouse", true, "phone", 77.884];
3
4   for(var i = 0; i < mas.length; i++){
5       console.log(i, mas[i]);
6       mas[i] = i*i*i;
7   }
8
9   console.log(mas);
10 </script>
```

0 5	ex02.html:5
1 76	ex02.html:5
2 "mouse"	ex02.html:5
3 true	ex02.html:5
4 "phone"	ex02.html:5
5 77.884	ex02.html:5
[0, 1, 8, 27, 64, 125]	ex02.html:9

Свойство `mas.length` –  
свойство массива  
содержащее его длину.

Цикл `for` удобен для тех случаев, когда заранее известно (или можно просчитать на основе уже имеющихся данных), сколько раз нужно будет повторить то или иное действие. Например: **обработка массивов**.

# Базовые алгоритмы работы с данными

```
1 <script>
2   var n = 10;
3   var arr = [];
4
5   for(var i = 0; i < n; i++){ arr.push( Math.floor(Math.random() * 100) ); };
6
7   console.log(arr);
8
9   //Код можно писать после этой строки//
10
11
12 </script>
```

[18, 93, 72, 82, 59, 15, 25, 50, 89, 95] arr\_base.html:7

[http://files.courses.dp.ua/js/04/arr\\_base.html](http://files.courses.dp.ua/js/04/arr_base.html)

1. Поиск максимального (минимального) элемента и среднеарифметического значения;
2. Создание нового набора данных на основе имеющегося;
3. Изменение позиций элементов (сортировка).

# Базовые алгоритмы работы с данными

Поиск максимального (минимального)  
элемента и среднеарифметического  
значения

[62, 64, 81, 16, 74, 11, 56, 47, 63, 99]



# Базовые алгоритмы работы с

Создание ~~нового~~ **данными** набора данных на основе имеющегося набора

**Например:** в первом массиве задана температура в градусах по Цельсию, необходимо получить температуру в градусы по Фаренгейту.

[4, 99, 65, 40, 66, 65, 92, 24, 4, 26]



[59, 35, 12, 17, 6, 20, 6, 29, 18, 77]

# Сортировка

Частный случай, когда необходимо внести изменения в существующий набор данных.

```
9 //Код можно писать после этой строки//
10
11 for(var i = 0; i < arr.length - 1; i++){
12     for(var j = 0; j < arr.length - i; j++){
13         if(arr[j] > arr[j+1]){
14             var temp = arr[j];
15             arr[j] = arr[j+1];
16             arr[j+1] = temp;
17         }
18     }
19 }
20
21 console.log(arr);
```

На слайде представлен классический алгоритм «пузырьковой» сортировки.

# Все циклы взаимозаменяемы

```
1 <script>
2
3   var mas = [5,34,"computer", true, "phone", 77.355];
4
5   for(var i = 0; i < mas.lenght; i++){
6       console.log(i, mas[i]);
7   }
8
9 </script>
```



```
1 <script>
2
3   var mas = [5,34,"computer", true, "phone", 77.355];
4
5   var i = 0;
6
7   while(i < mas.length){
8
9       console.log(i, mas[i]);
10
11       i++;
12   }
13
14 </script>
```

*Все циклы полностью взаимозаменяемые, цикла while хватает на все случаи, но специализированные версии циклов (for, for/in, for/of и др.) уменьшают объем кода.*

# Контрольный вопрос #1

```
1 <script>
2
3   var mas = [5,34,"computer", true, "phone", 77.355];
4
5   for(var i = 0; i < mas.length; i++){
6       console.log(i, mas[i]);
7   }
8
9   console.log(i);
10
11 </script>
```

*Чему рано  $i$  после выполнения цикла?*

*А если **let** вместо **var**???*

# Контрольный вопрос #2

```
1 <script>
2
3     var i = 5;
4
5     while(i) {
6         console.log(i--);
7     }
8
9     console.log("После цикла: " + i);
10
11 </script>
```

*Что мы увидим в консоле?*

*А если **let** вместо **var**???*

# Не всё так просто...

```
1 <script>
2
3   var arr = [6,77,23];
4
5   console.log("Len: " + arr.length);
6   for(var i = 0; i < arr.length; i++){
7       console.log(i + " >> " + arr[i]);
8   }
9
10  arr[7] = 89;
11
12  console.log("Len: " + arr.length);
13  for(var i = 0; i < arr.length; i++){
14      console.log(i + " >> " + arr[i]);
15  }
16
17 </script>
```

Len: 3	<a href="#">exmaple.html:5</a>
0 >> 6	<a href="#">exmaple.html:7</a>
1 >> 77	<a href="#">exmaple.html:7</a>
2 >> 23	<a href="#">exmaple.html:7</a>

?!?

*Что мы увидим в консоли?*

# Многомерные

```
1 <script>
2   var arr = [{"iPhone", "399$"}, {"Samsung", "459$"}, {"HTC",
3     "150$"}, {"LG", "799$"}, {"Lenovo", "199$"}, {"Sony", "430$"},
4     {"Nokia", "39$"}];
5
6   console.dir(arr);
7 </script>
```

*Многомерные массивы – массивы элементы которого сами являются массивами.*

```
▼ Array[7] ⓘ
  ▼ 0: Array[2]
    0: "iPhone"
    1: "399$"
    length: 2
    ▶ __proto__: Array[0]
  ▼ 1: Array[2]
    0: "Samsung"
    1: "459$"
    length: 2
    ▶ __proto__: Array[0]
  ▶ 2: Array[2]
  ▶ 3: Array[2]
  ▶ 4: Array[2]
  ▶ 5: Array[2]
  ▶ 6: Array[2]
  length: 7
  ▶ __proto__: Array[0]
```

# Многомерные

```
1 <script>
2   var arr = [{"iPhone", "399$"}, {"Samsung", "459$"}, {"HTC",
3     "150$"}, {"LG", "799$"}, {"Lenovo", "199$"}, {"Sony", "430$"},
4     {"Nokia", "39$"}];
5
6   console.log(arr[3]);
7
8   console.log(arr[3][0]);
9
10  console.log(arr[3][1]);
11 </script>
```

All	Errors	Warnings	Info	Logs	Debug	Handled
				["LG", "799\$"]		ex01.html:4
				LG		ex01.html:6
				799\$		ex01.html:8
				>		

*Многомерные массивы – массивы элементы которого сами являются массивами. Обращение к элементам осуществляется использованием нескольких пар скобок **[][]***

Немного практики

# Игра «Угадай число»

Необходимо написать скрипт который загадает число, в диапазон от 1 до 1000 включительно. И даст пользователю 10 попыток на угадывание. Если пользователь во время попытки не угадал число, ему даётся подсказка в виде «число которое я загадал больше чем ваш вариант» или «...меньше...».



# Домашнее задание

# Домашнее задание №1

1. Узнать зачем в циклах оператор *continue*;
2. Узнать зачем в циклах оператор *break*.

# Домашнее задание №2

```
1 <script>
2
3   var phones = ["Apple", "Nokia", "Samsung", "LG", "HTC", "Lenovo", "Sony", "Fly",
4     "Microsoft", "Huawei"].map(function(element) {
5       return [element, (Math.random() * 1000).toFixed(0) + "$" ];
6     });
7
8   phones.forEach(function(element) { console.log(element); });
9
10  //phones содержит массив со списком телефонов, каждый элемент которого тоже массив
11  //состоящий их названия и цены телефона. Вам необходимо написать код который
12  //отсортирует массив phones по возрастанию цены. Цены генерируются автоматически
13  //(каждый раз разные). Ваш код можно писать только в отведённом месте, править
14  //остальной код нельзя. Вид массива [{"Apple", "233$"}, {"Nokia", "315$"}... и т.д.]
15  //Код можно писать после этой строки//
16
17  //Код можно писать до этой строки//
18
19  console.log("Вывод после сортировки: ");
20  phones.forEach(function(element) { console.log(element); });
21
22 </script>
```

[http://files.courses.dp.ua/js/04/homework\\_1.html](http://files.courses.dp.ua/js/04/homework_1.html)

*Задача, отсортировать массив со списком телефонов по возрастанию цены.*

## Домашнее задание №3

Разработать скрипт, проверяющий знания (умение) таблицы умножения двузначных чисел. Скрипт должен задать пользователю 12 задач на умножение двузначных чисел (используйте **prompt**). По результатам проверки, пользователю выставляется оценка (используйте **alert**), а также выводиться два списка: верных ответов, и ошибочных ответов, указанием какой ответ был правильный (используйте **console.log**).

# Домашнее задание №4

## Игра «Угадай число»

*Придумать (или найти в интернете, намёк: «**prometheus**») способ гарантированно выигрывать в игру «угадай число» во всех случаях, независимо от загаданного числа.*

Решения

# Базовые алгоритмы работы с данными

## Поиск максимального (минимального) элемента и среднееарифметического

Решение

```
1 <script>
2   var n = 10;
3   var arr = [];
4
5   for(var i = 0; i < n; i++){ arr.push( Math.floor(Math.random() * 100) ); };
6
7   console.log(arr);
8
9   var max = arr[0];
10  for(var i = 0; i < arr.length; i++){
11    if(arr[i] > max) max = arr[i];
12  }
13  console.log("Max: ", max);
14
15  var min = arr[0];
16  for(var i = 0; i < arr.length; i++){
17    if(arr[i] < min) min = arr[i];
18  }
19  console.log("Min: ", min);
20
21  var avr = 0;
22  for(var i = 0; i < arr.length; i++){
23    avr += arr[i];
24  }
25  avr = avr / arr.length;
26  console.log("Avr: ", avr);
27
28 </script>
```

[36, 49, 76, 61, 4, 35, 25, 85, 34, 7] [ex05 base h.html:7](#)

Max: 85 [ex05 base h.html:13](#)

Min: 4 [ex05 base h.html:19](#)

Avr: 41.2 [ex05 base h.html:26](#)

# Базовые алгоритмы работы с

данными  
Создание нового набора данных на основе имеющегося набора

**Например:** в первом массиве задана температура воздуха по Цельсию, необходимо получить температуру по Фаренгейту.

Решение

```
1 <script>
2   var n = 10;
3   var arr = [];
4
5   for(var i = 0; i < n; i++){ arr.push( Math.floor(Math.random() * 100) ); };
6
7   console.log(arr);
8
9   var faren_temps = [];
10
11  for(var i = 0; i < arr.length; i++){
12      var f_temp = arr[i] * 1.8 + 32;
13      faren_temps.push(f_temp);
14  }
15
16  console.log(faren_temps);
17 </script>
```

[20, 95, 93, 2, 85, 76, 9, 28, 1, 54]

[68, 203, 199.4, 35.6, 185, 168.8, 48.2, 82.4, 33.8, 129.2]

# Игра «Угадай число»

Решение

```
1 <script>
2   var answer      = Math.floor(Math.random() * 1000);
3   var attempts   = 10;
4   var version;
5
6   alert("Я загадал число от 1 до 1000, у вас 10 попыток угадать его!");
7
8   do{
9     version = parseInt(prompt("Осталось попыток: " + attempts + "/10. Введите вариант: "));
10
11     if(version == answer || isNaN(version)){
12       break;
13     }else{
14       alert("Ответ не верный. Моё число " + ((answer > version) ? "больше" : "меньше") + " чем " + version);
15     }
16
17     attempts--;
18
19   }while(attempts > 0);
20
21   if(version == answer && !isNaN(version)){
22     alert("Поздравляю, вы угадали!");
23   }else{
24     alert("Вы проиграли, загаданное число " + answer);
25   }
26 </script>
```