

# JavaScript

Основы программирования,  
часть 2



Web.Dev.  
Courses

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

# JavaScript - язык программирования

*его*

*концепции*

**Переменные / Типы /**

**Операции**

**Ветвления (условные**

**Циклы / Массивы (структуры**

**данных)  
Функции**

**Объект**

**ы**

# Циклы

## 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>
```

*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!!!**

# 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

# 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

# Массивы

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

          ↑          ↑          ↑          ↑  
          0          1          2          3

**Массив** – это нумерованный набор



[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)

# Массивы

```
1 <script>
2
3   var mas = ["axe", 22, "Ivan", true, 55.77, "test", "b", 865];
4
5   console.log(mas);
6   console.log("Element 0:", mas[0]);
7   console.log("Element 2:", mas[2]);
8   console.log("Element 3:", mas[3]);
9
10  mas[2] = 0;
11  mas[3] = "Lalalala";
12
13  console.log(mas);
14  console.log("Element 2:", mas[2]);
15  console.log("Element 3:", mas[3]);
16
17  mas.push("Irina!");
18  mas.unshift(777.77);
19
20  console.log(mas);
21
22 </script>
```

`["axe", 22, "Ivan", true, 55.77, "test", "b", 865]` [ex02.html:5](#)

Element 0: axe [ex02.html:6](#)

Element 2: Ivan [ex02.html:7](#)

Element 3: true [ex02.html:8](#)

`["axe", 22, 0, "Lalalala", 55.77, "test", "b", 865]` [ex02.html:13](#)

Element 2: 0 [ex02.html:14](#)

Element 3: Lalalala [ex02.html:15](#)

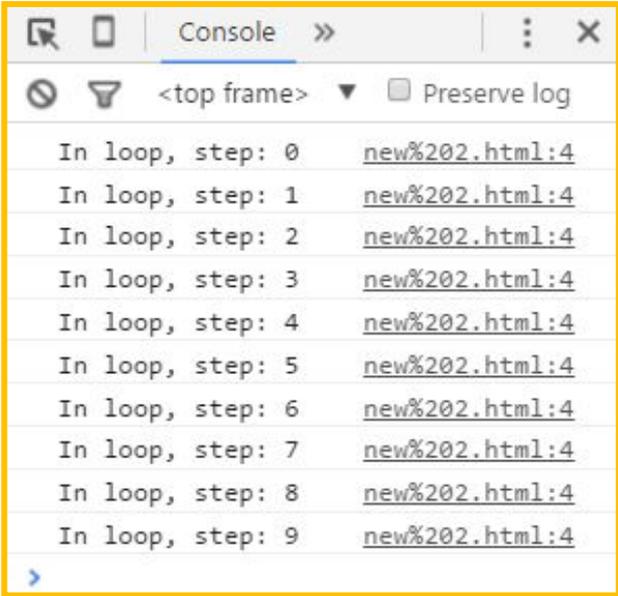
`[777.77, "axe", 22, 0, "Lalalala", 55.77, "test", "b", 865, "Irina!"]` [ex02.html:20](#)

>

# Цикл for – когда известно сколько раз нужно повторить действия

*Нужно повторят последовательность действий заранее известное количество раз? В таком случае удобно применять цикл **for**.*

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



```
Console >>
<top frame>
Preserve log

In loop, step: 0 new%202.html:4
In loop, step: 1 new%202.html:4
In loop, step: 2 new%202.html:4
In loop, step: 3 new%202.html:4
In loop, step: 4 new%202.html:4
In loop, step: 5 new%202.html:4
In loop, step: 6 new%202.html:4
In loop, step: 7 new%202.html:4
In loop, step: 8 new%202.html:4
In loop, step: 9 new%202.html:4
```

# Цикл for – когда известно сколько раз нужно повторить действия

```
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**.

# Алгоритмы обработки набора данных

# Базовые алгоритмы работы с данными

1. Поиск *максимального* (минимального) *элемента* и *среднеарифметического значения*;
2. *Создание нового набора данных* (массива) *на основе имеющегося*;
3. *Анализ данных*;

# Немного практики: базовые алгоритмы работы с

```
1 <script>
2   //Массив хранит курс доллара по дням за месяц.
3   var usd = [];
4
5   for(var i = 0; i < 30; i++){
6     usd.push(Math.floor(Math.random() * (30 - 25) + 25)*100)/100);
7   };
8
9   console.log("На входе курсы валют за месяц (30 дней): " , usd);
10
11   //Дальше можно писать код
12 </script>
```

top Preserve log

На входе курсы валют за месяц (30 дней): ex04.html:9

```
[25.51, 27.39, 27.46, 29.72, 28.71, 27.36, 27.27, 25.94, 26.02, 26.78, 29.3, 25.14, 26, 29, 29.59, 27.35, 29.39, 27.57, 26.52, 29.73, 27.5, 29.07, 26.77, 26.14, 29.77, 27.18, 25.78, 26.01, 29.38, 28.12]
```

Заготовка по

адресу:

<http://files.courses.dp.ua/js/arrays/ex02.html>



# 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 набора данных на основе имеющегося набора

**Например:** зная сколько гривен стоит один доллар узнаем сколько долларов дадут за одну гривну.

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



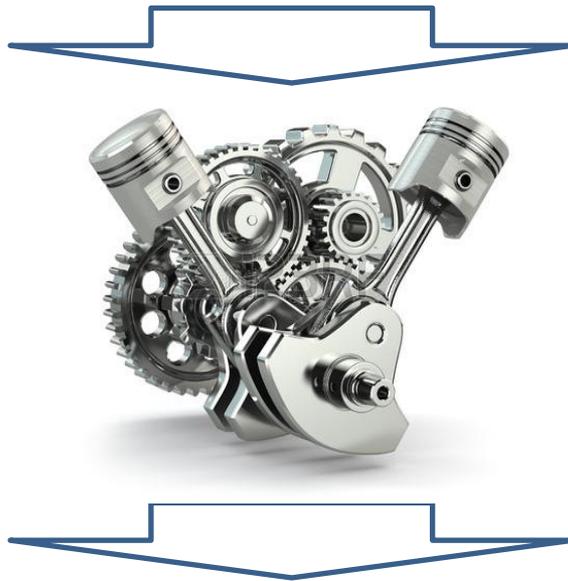
[59, 35, 12, 17, 6, 20, 6, 29, 18, 77]

# Базовые алгоритмы работы с данными

## Анализ данных

*Например: определить какая тенденция преобладала в течении месяца: рост, падение или разнонаправленно?*

[41, 26, 53, 37, 80, 6, 59, 82, 62, 23]



Да, нет,  
возможно?!?

# Примеры алгоритмов обработки набора данных

# Базовые алгоритмы работы с данными

## Поиск максимального (минимального) элемента и среднеарифметического значения

Решение

```
6 //Дальше можно писать код
7
8 var max = usd[0];
9 var max_day = 0;
10
11 for(var i = 1; i < usd.length; i++){
12     if(usd[i] > max){
13         max = usd[i];
14         max_day = i;
15     }
16 }
17 max_day = max_day + 1;
18 console.log("Max: " + max + "; day: " + max_day);
19
20 //=====//
21
22 var min = usd[0];
23 var min_day = 0;
24
25 for(var i = 1; i < usd.length; i++){
26     if(usd[i] < min){
27         min = usd[i];
28         min_day = i;
29     }
30 }
31 min_day = min_day + 1;
32 console.log("Min: " + min + "; day: " + min_day);
33
34 //=====//
35
36 var sum = 0;
37
38 for(var i = 1; i < usd.length; i++){
39     sum = sum + usd[i];
40 }
41 var average = sum / usd.length;
42 console.log("Average: " + average);
43
44
45 </script>
```

Max: 26.98; day: 28

[ex03\\_1.html:18](#)

Min: 24.06; day: 4

[ex03\\_1.html:32](#)

Average: 24.560000000000002

[ex03\\_1.html:42](#)

> |

# Базовые алгоритмы работы с

данными  
Создание нового набора данных на основе имеющегося набора

**Например:** зная сколько гривен стоит один доллар узнаем сколько долларов дадут за одну гривну.

Решение

```
6 //Дальше можно писать код
7
8 var uah = [];
9
10 for(var i = 0; i < usd.length; i++){
11     var rate = 1 / usd[i];
12     rate = rate.toFixed(4);
13
14     uah.push(rate);
15 }
16
17 console.log(uah);
18
19
20
21 </script>
```

ex03\_2.html:19

```
▶ ["0.0379", "0.0407", "0.0383", "0.0416", "0.0396", "0.0396", "0.0406", "0.0373", "0.0415", "0.0388",  
"0.0392", "0.0407", "0.0375", "0.0374", "0.0388", "0.0412", "0.0383", "0.0376", "0.0412", "0.0411",  
"0.0400", "0.0392", "0.0396", "0.0380", "0.0375", "0.0410", "0.0396", "0.0371", "0.0392", "0.0408"]
```

# Базовые алгоритмы работы с данными

## Анализ данных

**Например:** определить какая тенденция преобладала в течении месяца: рост, падение или разнонаправленно?

```
5
6 //Дальше можно писать код
7 var up = 0;
8 var down = 0;
9
10 for(var i = 0; i < usd.length; i++){
11     if(usd[i+1] > usd[i]){
12         up = up + 1;
13     }else if(usd[i+1] < usd[i]){
14         down = down + 1;
15     };
16 }
17
18 if(up > down){
19     console.log("Up trend.");
20 }else if(up < down){
21     console.log("Down trend.");
22 }else{
23     console.log("In different directions.");
24 }
25
26
27
28
29
30
31
32
33 </script>
```

Решение

Up trend.

ex03\_2.html:25

# Домашнее задание

# Домашнее задание №2

```
1 <script>
2   var winter_temps = [];
3
4   for(var i = 0; i < 90; i++){
5       winter_temps.push( Math.floor(Math.random() * 100 - 50) );
6   };
7
8   console.log(winter_temps);
9
10  //После этой строки вы можете писать свой код//
11
12 </script>
```

<http://files.courses.dp.ua/web/09/ex02.html>

```
[-24, -26, -47, -28, 35, -2, -31, -15, -1, -19, 22, -9, -47, -34, -23, 11, -7, 3, -41, 14, 18,
-19, 42, 30, 10, 33, 31, -30, -46, -25, 44, -13, -33, -34, -45, -7, 43, -23, -45, -26, -50, 21,
48, 11, -7, 29, 22, -39, -41, -35, -33, 20, 40, -32, -10, 11, 16, 39, 21, -37, 26, -39, 21, 22,
46, -45, -38, -39, 38, -44, 0, -50, 46, 27, 7, 8, -17, -22, 43, 23, -1, -4, 17, 20, 10, -19,
24, 30, 49, -15]
```

**Задача:** в массиве содержатся данные о ежедневной температуре в течении зимних месяцев. Нужно подсчитать сколько было циклом замерзания в этот период (замерзания это когда сегодня температура плюсовая, а завтра минусовая).