

«Массивы, циклы»



JavaScript
Courses

vk.com/js.courses

js.courses.dp.ua/files

Если какое-либо действие (или блок действий) нужно повторить многократно (здесь и сейчас, без перерывов на другие действия) то циклы в помощь

```
1 <script>
2
3   var secret_key = "123";
4   var user_key;
5
6   do{
7
8       user_key = prompt("Введите пароль: ");
9
10      if(secret_key != user_key) alert("Пароль не верный!");
11
12
13      }while(secret_key != user_key);
14
15      alert("Пароль правильный!");
16
17 </script>
```

*Классический цикл do/while, выполняется пока **условие** истинно (true)*

while, do/while

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

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

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  console.log(a);
14
15 </script>
```

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

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

Игра «Угадай число»

```
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 = Number(prompt("Осталось попыток: " + attempts + "/10. Введите ваш
10    вариант: "));
11
12    if(version == answer){
13      break;
14    }else{
15      alert("Ответ не верный. Моё число " + ((answer > version) ? "больше" : "меньше")
16        + " чем " + version);
17    }
18
19    attempts--;
20
21  }while(attempts > 0);
22
23  if(version == answer){
24    alert("Поздравляю, вы угадали!!!");
25  }else{
26    alert("Вы проиграли, загаданное число: " + answer);
27  }
28 </script>
```

Циклы можно прерывать в любое время в любом месте, делает это оператор **break**. Но злоупотребление этим оператором усложняет чтение кода.

Игра «Угадай число»

```
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 = Number(prompt("Осталось попыток: " + attempts + "/10. Введите ваш
10    вариант: "));
11
12     if(version == answer){
13       break;
14     }else{
15       alert("Ответ не верный. Моё число " + ((answer > version) ? "больше" : "меньше")
16         + " чем " + version);
17     }
18
19     attempts--;
20
21   }while(attempts > 0);
22
23   if(version == answer){
24     alert("Поздравляю, вы угадали!!!");
25   }else{
26     alert("Вы проиграли, загаданное число: " + answer);
27   }
28 </script>
```

Домашнее задание

Проанализировать код, придумать (или найти в интернете) способ гарантированно выигрывать в игру «угадай число» во всех случаях, независимо от загаданного числа.

Массивы, когда переменных не хватает...

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

Массивы – упорядоченный, сгруппированный набор элементов.

```
a = [ "a", "b", "c", "d", "e" ]
```

↑ ↑ ↑ ↑ ↑
0 1 2 3 4



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 и массивы

```
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>
```

0	5	exmaple.html:6
1	34	exmaple.html:6
2	"computer"	exmaple.html:6
3	true	exmaple.html:6
4	"phone"	exmaple.html:6
5	77.355	exmaple.html:6

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.

Цикл for и массивы

```
1 <script>
2 //Создаём массив, заполняем числами, находим среднее арифметическое.
3 var mas_len = Number(prompt("Сколько будет элементов?"));
4
5 var mas = [];
6
7 for(var i = 0; i < mas_len; i++){
8     var new_element = Number(prompt("Введите элемент №" + i));
9     mas.push(new_element);
10 }
11
12 var sum = 0;
13
14 for(var i = 0; i < mas_len; i++){
15     sum = sum + mas[i];
16 }
17
18 var avr = sum / mas.length;
19
20 alert("Среднее арифметическое в массиве: " + avr);
21
22 </script>
```

*Например: часто цикл **for** применяют для обхода массива и обработка его элементов.*

Все циклы взаимозаменяемы

```
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 </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       console.log(i, mas[i]);
9
10      i++;
11  }
12
13
14 </script>
```

Все циклы полностью взаимозаменяемые, цикла while хватает на все случаи, но специализированные версии циклов (for, for/in) 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?

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?

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	exmaple.html:5
0 >> 6	exmaple.html:7
1 >> 77	exmaple.html:7
2 >> 23	exmaple.html:7



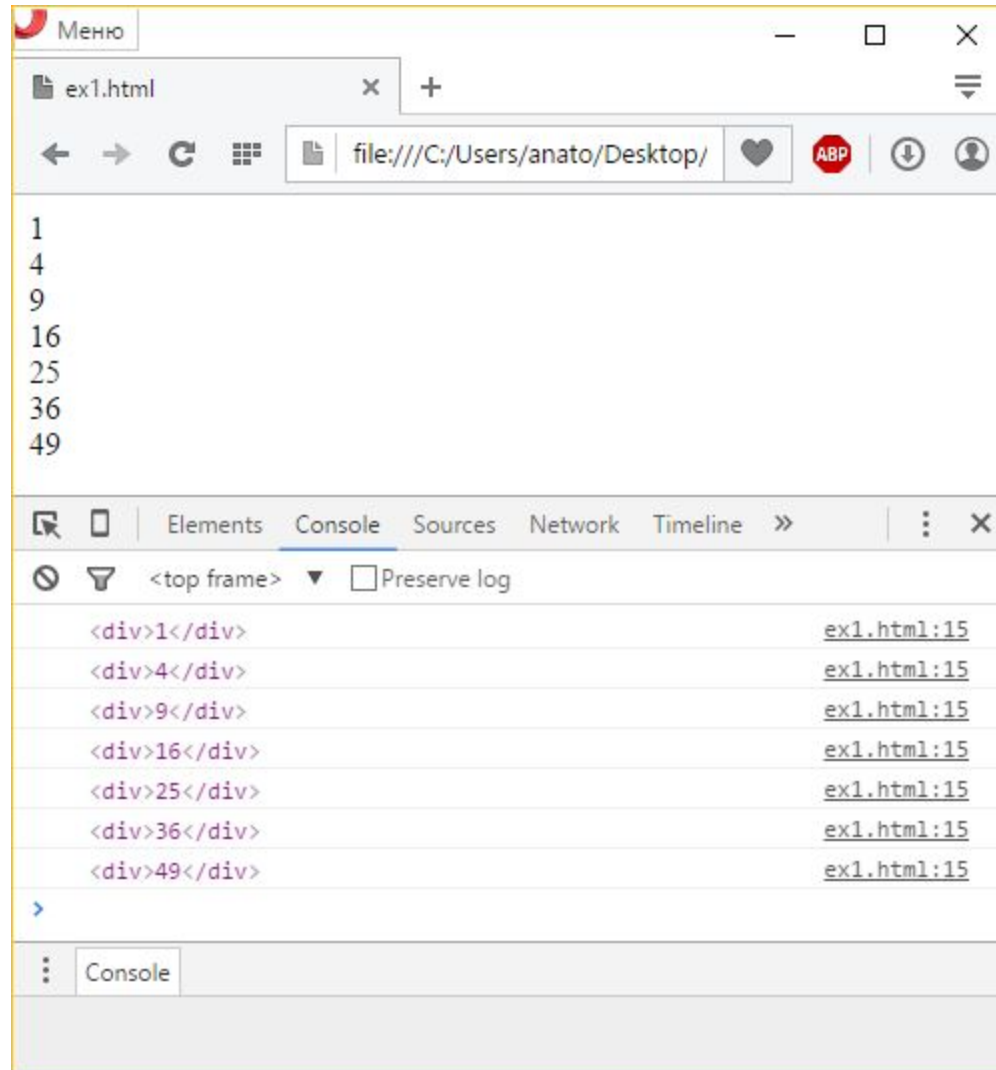
?!?

Что мы увидим в консоле?

Зачем всё это надо?

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4     <div>1</div>
5     <div>2</div>
6     <div>3</div>
7     <div>4</div>
8     <div>5</div>
9     <div>6</div>
10    <div>7</div>
11 <script>
12     var divs = document.getElementsByTagName("div");
13
14     for(var i = 0; i < divs.length; i++){
15         console.log(divs[i]);
16     }
17
18     for(var i = 0; i < divs.length; i++){
19         divs[i].innerHTML = divs[i].innerHTML * divs[i].innerHTML;
20     }
21 </script>
22 </body>
23 </html>
```

Зачем всё это надо?



Домашнее задание

1. На слайде № 6.
2. Узнать зачем в циклах оператор *continue*.
3. Узнать, что такое **многомерные массивы**.

4. Задача по сортировке

```
1 <script>
2   var mas_length = Math.floor(Math.random() * (15) + 5);
3   var mas = [];
4
5   for(var i = 0; i < mas_length; i++){
6       mas[i] = Math.floor(Math.random() * (100) + 1);
7   }
8
9   console.log(mas);
10  /*
11     Напишите здесь код, который отсортирует массив по возрастанию.
12     Править другой код нельзя. Использовать встроенную функцию
13     JavaScript для сортировки тоже нельзя. Подсказка: "Пузырьковая
14     сортировка".
15  */
16  console.log(mas);
17 </script>
```

JavaScript как язык программирования

его

концепции

**Переменные / Типы /
Операции
Ветвления (условные**

**операторы)
Циклы / Массивы (структуры**

**данных)
Функции**

Объект

ы

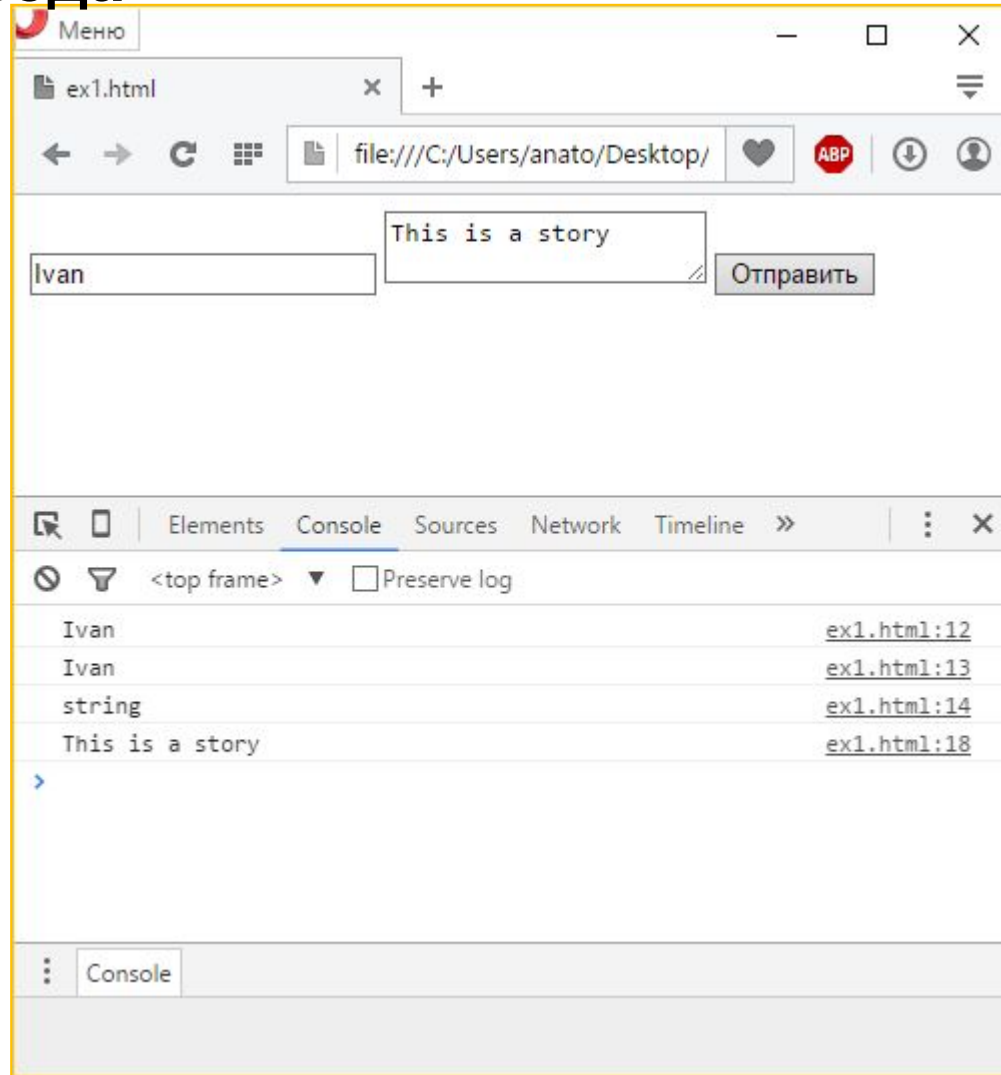
**По просьбам
трудящихся (с)**

Получение данных из элементов

Ввода

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <script>
5      window.onload = function() {
6          document.forms.user_form.onsubmit = function(event) {
7
8              var a = document.forms.user_form.user_name.value;
9              var element = document.getElementById("user_name_id");
10             var b = element.value;
11
12             console.log(a);
13             console.log(b);
14             console.log(typeof(b));
15
16             element.value = "this is text";
17
18             console.log(document.forms.user_form.user_description.value);
19
20             event.preventDefault();
21         }
22     }
23 </script>
24 </head>
25 <body>
26     <form name="user_form">
27         <input type="text" name="user_name" id="user_name_id">
28         <textarea name="user_description" id="user_description_id"></textarea>
29         <input type="submit" name="submit_button" id="submit_button_id">
30     </form>
31 </body>
32 </html>
```

Получение данных из элементов ввода



The image shows a web browser window with a single tab titled "ex1.html". The address bar shows the file path: `file:///C:/Users/anato/Desktop/`. The browser's developer tools are open, with the "Console" tab selected. The console displays the following log entries:

Value	Location
Ivan	ex1.html:12
Ivan	ex1.html:13
string	ex1.html:14
This is a story	ex1.html:18

The browser window itself displays a simple form with two input fields and a button. The first input field contains the text "Ivan", and the second input field contains the text "This is a story". The button is labeled "Отправить".