«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

```
<script>

    var a = 3;

    var b = 7;

    while(a < b){

        //Какие-то действия...
        a++; //!!!
    }

    console.log(a);

</script>
```

```
<script>

    var a = 3;

    var b = 7;

    do{
        //Какие-то действия...
        a++; //!!!

    }while(a < b);

    console.log(a);

</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
11     }while(a < b);
12
13     console.log(a);
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»

```
<script>
    var answer      = Math.floor(Math.random() * 1000);
    var attempts    = 10;
    var version;

    alert("Я загадал число от 1 до 1000, у вас 10 попыток угадать его!");

    do{
        version = Number(prompt("Осталось попыток: " + attempts + "/10. Введите ваш
        вариант: "));

        if(version == answer){
            break;
        }else{
            alert("Ответ не верный. Моё число " + ((answer > version) ? "больше" : "меньше")
              + " чем " + version);
        }

        attempts--;

    }while(attempts > 0);

    if(version == answer){
        alert("Поздравляю, вы угадали!!!");
    }else{
        alert("Вы проиграли, загаданное число: " + answer);
    }
</script>
```

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

Игра «Угадай число»

```
<script>
    var answer      = Math.floor(Math.random() * 1000);
    var attempts    = 10;
    var version;

    alert("Я загадал число от 1 до 1000, у вас 10 попыток угадать его!");

    do{
        version = Number(prompt("Осталось попыток: " + attempts + "/10. Введите ваш
        вариант: "));

        if(version == answer){
            break;
        }else{
            alert("Ответ не верный. Моё число " + ((answer > version) ? "больше" : "меньше")
              + " чем " + version);
        }

        attempts--;

    }while(attempts > 0);

    if(version == answer){
        alert("Поздравляю, вы угадали!!!");
    }else{
        alert("Вы проиграли, загаданное число: " + answer);
    }
</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" ]
```

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

```
<script>
    var arr = [1, 3, "Elena", true];

    console.log(arr);
    console.log(arr.length);
    //Свойство указывающее длинну массива.
    console.log(typeof(arr));

    console.log(arr[0], typeof(arr[0]));
    console.log(arr[2], typeof(arr[2]));
    console.log(arr[3], typeof(arr[3]));

    arr[0] = "Ivan";
    console.log(arr[0], typeof(arr[0]));

    arr.push(777);
    //Добавление элемента в конец массива
    console.log(arr, arr.length);

    var x = arr.pop();
    //Удаление последнего элемента из массива
    console.log(arr, arr.length);
    console.log(x);

    arr.unshift("Julia");
    //Добавление элемента в начало массива
    console.log(arr, arr.length);

    arr.shift();
    //Удаление первого элемента массива
    console.log(arr, arr.length);

</script>
```

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

```
<script>
    //Создаём массив, заполняем числами, находим среднее арифметическое.
    var mas_len = Number(prompt("Сколько будет элементов?"));

    var mas = [];

    for(var i = 0; i < mas_len; i++){
        var new_element = Number(prompt("Введите элемент №" + i));
        mas.push(new_element);
    }

    var sum = 0;

    for(var i = 0; i < mas_len; i++){
        sum = sum + mas[i];
    }

    var avr = sum / mas.length;

    alert("Среднее арифметическое в массиве: " + avr);

</script>
```

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

Все циклы взаимозаменяемы

```
<script>

    var mas = [5,34,"computer", true, "phone", 77.355];

    for(var i = 0; i < mas.lenght; i++){
        console.log(i, mas[i]);
    }

</script>
```

```
<script>

    var mas = [5,34,"computer", true, "phone", 77.355];

    var i = 0;

    while(i < mas.length){

        console.log(i, mas[i]);

        i++
    }

</script>
```

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

Контрольный вопрос #1

```
<script>

    var mas = [5,34,"computer", true, "phone", 77.355];

    for(var i = 0; i < mas.length; i++){
        console.log(i, mas[i]);
    }

    console.log(i);

</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?

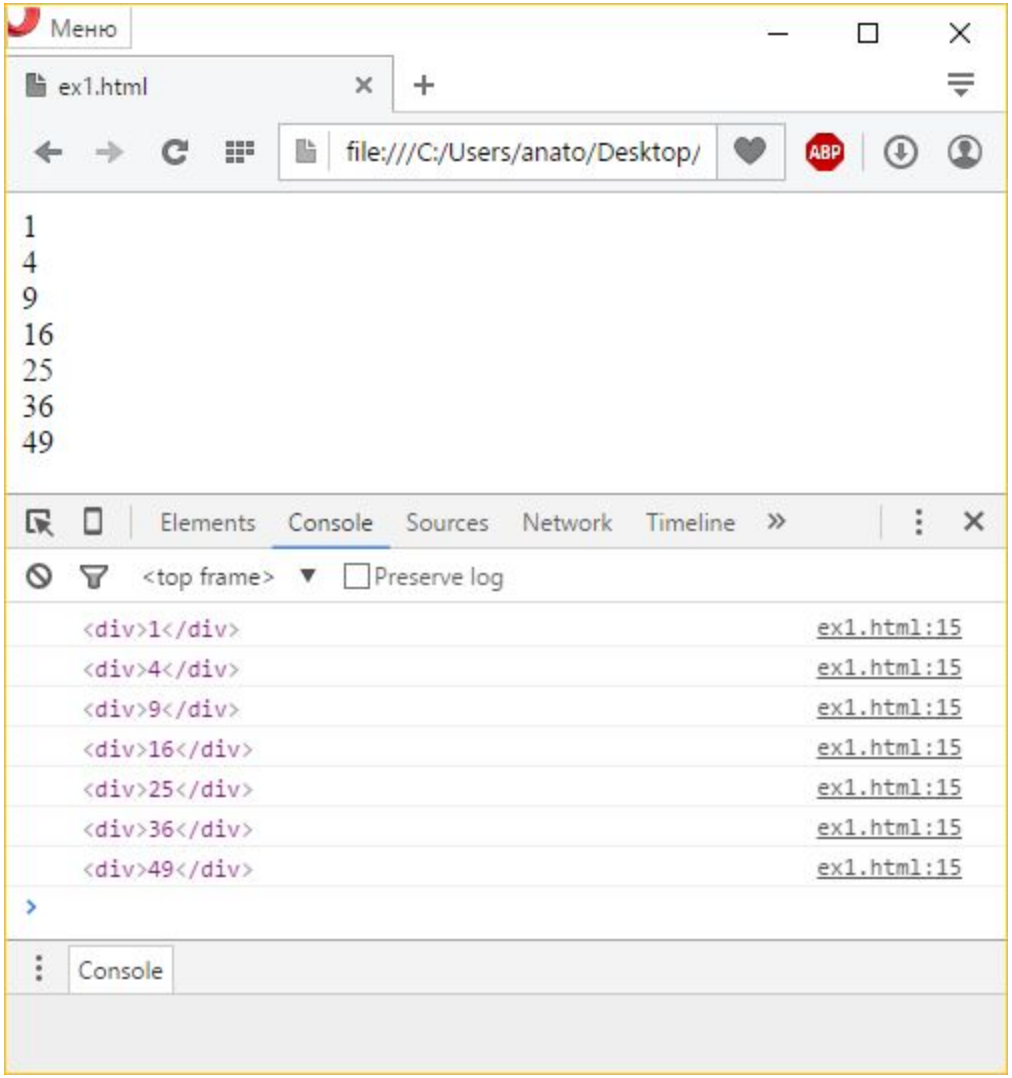
Зачем всё это надо?

```
<!DOCTYPE html>
<html>
<body>
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
    <div>5</div>
    <div>6</div>
    <div>7</div>
<script>
    var divs = document.getElementsByTagName("div");

    for(var i = 0; i < divs.length; i++){
        console.log(divs[i]);
    }

    for(var i = 0; i < divs.length; i++){
        divs[i].innerHTML = divs[i].innerHTML * divs[i].innerHTML;
    }
</script>
</body>
</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.

```
<script>
    var mas_length = Math.floor(Math.random() * (15) + 5);
    var mas = [];

    for(var i = 0; i < mas_length; i++){
        mas[i] = Math.floor(Math.random() * (100) + 1);
    }

    console.log(mas);
    /*
        Напишите здесь код, который отсортирует массив по возрастанию.
        Править другой код нельзя. Использовать встроенную функцию
        JavaScript для сортировки тоже нельзя. Подсказка: "Пузырьковая
        сортировка".
    */
    console.log(mas);
</script>
```

JavaScript как язык программирования

его

концепции

Переменные / Типы / Операции

Ветвления (условные операторы)

Циклы / Массивы (структуры данных)

Функции

И

Объекты

По просьбам трудящихся (с)

Получение данных из элементов ввода

```
<!DOCTYPE html>
<html>
<head>
<script>
    window.onload = function(){
        document.forms.user_form.onsubmit = function(event){

            var a = document.forms.user_form.user_name.value;
            var element = document.getElementById("user_name_id");
            var b = element.value;

            console.log(a);
            console.log(b);
            console.log(typeof(b));

            element.value = "this is text";

            console.log(document.forms.user_form.user_description.value);

            event.preventDefault();
        }
    }
</script>
</head>
<body>
    <form name="user_form">
        <input type="text" name="user_name" id="user_name_id">
        <textarea name="user_description" id="user_description_id"></textarea>
        <input type="submit" name="submit_button" id="submit_button_id">
    </form>
</body>
</html>
```

Получение данных из элементов ввода