

# Задачи на циклы

## while, for в JavaScript

### Задача

**Задача.** Выведите столбец чисел от 1 до 50.

**Решение:** воспользуемся циклом while (отделим числа тегом `br` друг от друга, чтобы получить столбец, а не строку):

```
var i = 1;
while (i <= 50) {
    document.write(i + '<br>');
    i++;
}
```

Можно также воспользоваться и циклом for:

```
for (var i = 1; i <= 50; i++) {
    document.write(i + '<br>');
}
```

### Задача

**Задача.** Дан массив с элементами [1, 2, 3, 4, 5]. С помощью цикла `for` выведите все эти элементы на экран.

**Решение:** будем повторять цикл `for` от 0 до номера последнего элемента массива. Этот номер на единицу меньше количества элементов в массиве, которое можно найти с помощью [свойства length](#) таким образом: `arr.length`.

Чтобы цикл прокрутился на единицу меньше длины массива, в условие окончания мы поставим `<`, а не `<=`.

К элементам массива будем обращаться так: `arr[i]`. При этом переменная `i` - это счетчик цикла, который будет меняться от нуля до `arr.length` (не включительно). Таким образом мы последовательно выведем все элементы массива на экран (отделив их [тегом br](#) друг от друга):

```
var arr = [1, 2, 3, 4, 5];
for (var i = 0; i < arr.length; i++) {
    document.write(arr[i] + '<br>');
}
```

### Задача

**Задача.** Дан массив с элементами [2, 3, 4, 5]. С помощью цикла `for` найдите произведение элементов этого массива.

**Решение:** для таких задач существует **стандартное решение**, которое заключается в том, что циклом `for` перебираются элементы массива и их произведение

последовательно записывается в переменную **result** (в ней постепенно накапливается искомое произведение):

```
var result = 1;
var arr = [2, 3, 4, 5];
for (var i = 0; i < arr.length; i++) {
    result = result * arr[i];
}
alert(result);
```

Как это работает: изначально переменная **result** имеет значение 1, затем при первом проходе цикла в нее записывается ее текущее содержимое (это 1), умноженное на первый элемент массива (это 2). Получится, что в **result** запишется  $1 \cdot 2$  и теперь там будет лежать число 2.

При следующем проходе цикла в **result** запишется текущее значение **result**, умноженное на второй элемент массива (то есть  $2 \cdot 3 = 6$ ). И так далее **пока массив не закончится**.

Можно переписать строчку **result = result \* arr[i]** через **\*=** для краткости:

```
var result = 1;
var arr = [2, 3, 4, 5];
for (var i = 0; i < arr.length; i++) {
    result *= arr[i];
}
alert(result);
```

## Задача . Цикл for-in

**Задача.** Дан объект **obj** с ключами 'Минск', 'Москва', 'Киев' с элементами 'Беларусь', 'Россия', 'Украина'. С помощью цикла **for-in** выведите на экран строки такого формата: 'Минск - это Беларусь'.

**Решение:** задача не представляет сложности если уметь работать с [циклом for-in](#). Давайте решать задачу поэтапно. Для начала выведем на экран все ключи объекта (это названия городов):

```
var obj = {
    'Минск': 'Беларусь',
    'Москва': 'Россия',
    'Киев': 'Украина'
};

for (var key in obj) {
    alert(key);
}
```

А теперь выведем все значения объекта (это страны):

```
var obj = {
```

```
        'Минск': 'Беларусь',  
        'Москва': 'Россия',  
        'Киев': 'Украина'  
    };
```

```
for (var key in obj) {  
    alert(obj[key]);  
}
```

Ну, а теперь сформируем строки нужного нам формата:

```
var obj = {  
    'Минск': 'Беларусь',  
    'Москва': 'Россия',  
    'Киев': 'Украина'  
};
```

```
for (var key in obj) {  
    alert(key + ' - это ' + obj[key] + '.');  
}
```

## Задачи для решения

### Циклы while и for

*Решите эти задачи сначала через цикл while, а затем через цикл for.*

Выведите столбец чисел от 1 до 100.

Выведите столбец чисел от 11 до 33.

Выведите столбец четных чисел в промежутке от 0 до 100.

С помощью цикла найдите сумму чисел от 1 до 100.

### Работа с for для массивов

Дан массив с элементами [1, 2, 3, 4, 5]. С помощью цикла **for** выведите все эти элементы на экран.

Дан массив с элементами [1, 2, 3, 4, 5]. С помощью цикла **for** найдите сумму элементов этого массива. Запишите ее в переменную **result**.

### Работа с for-in

Дан объект **obj**. С помощью цикла **for-in** выведите на экран ключи и элементы этого объекта.

```
var obj = {green: 'зеленый', red: 'красный', blue: 'голубой'}
```

Дан объект **obj** с ключами **Коля**, **Вася**, **Петя** с элементами **'200'**, **'300'**, **'400'**. С помощью цикла **for-in** выведите на экран строки такого формата: **'Коля - зарплата 200 долларов.'**

## Задачи

Дан массив с элементами **2, 5, 9, 15, 0, 4**. С помощью цикла **for** и оператора **if** выведите на экран столбец тех элементов массива, которые больше **3-х**, но меньше **10**.

Дан массив с числами. Числа могут быть положительными и отрицательными. Найдите сумму положительных элементов массива.

Дан массив с элементами **1, 2, 5, 9, 4, 13, 4, 10**. С помощью цикла **for** и оператора **if** проверьте есть ли в массиве элемент со значением, равным 4. Если есть - выведите на экран **'Есть!'** и выйдите из цикла. Если нет - ничего делать не надо.

Дан массив числами, например: **[10, 20, 30, 50, 235, 3000]**. Выведите на экран только те числа из массива, которые начинаются на цифру **1, 2** или **5**.

Дан массив с элементами **1, 2, 3, 4, 5, 6, 7, 8, 9**. С помощью цикла **for** создайте строку **'1-2-3-4-5-6-7-8-9'**.

Составьте массив дней недели. С помощью цикла **for** выведите все дни недели, а выходные дни выведите жирным.

Составьте массив **дней недели**. С помощью цикла **for** выведите все дни недели, а **текущий** день выведите *курсивом*. Текущий день должен храниться в переменной **day**.

Дано число **n=1000**. Делите его на **2** столько раз, пока результат деления не станет меньше **50**. Какое число получится? Посчитайте количество итераций, необходимых для этого (*итерация* - это проход цикла), и запишите его в переменную **num**.

# Задачи на математические функции JavaScript

## Работа с %

Даны переменные **a = 10** и **b = 3**. Найдите остаток от деления **a** на **b**. \_

Даны переменные **a** и **b**. Проверьте, что **a** делится без остатка на **b**. Если это так - выведите **'Делится'** и результат деления, иначе выведите **'Делится с остатком'** и остаток от деления. \_

## Работа со степенью и корнем

Для решения задач данного блока вам понадобятся следующие методы: [Math.pow](#), [Math.sqrt](#).

Возведите **2** в **10** степень. Результат запишите в переменную **st**.

Найдите квадратный корень из **245**.

Дан массив с элементами **4, 2, 5, 19, 13, 0, 10**. Найдите **квадратный корень** из **суммы кубов** его элементов. Для решения воспользуйтесь циклом **for**.

## Работа с функциями округления

Для решения задач данного блока вам понадобятся следующие функции: [Math.round](#), [Math.ceil](#), [Math.floor](#), [toFixed](#), [toPrecision](#).

Найдите квадратный корень из **379**. Результат округлите до целых, до десятых, до сотых.

Найдите квадратный корень из **587**. Округлите результат в **большую** и **меньшую** стороны, запишите результаты округления в **объект** с ключами **'floor'** и **'ceil'**.

## Нахождение максимального и минимального числа

Для решения задач данного блока вам понадобятся следующие методы: [Math.max](#), [Math.min](#).

Даны числа **4, -2, 5, 19, -130, 0, 10**. Найдите **минимальное** и **максимальное** число.

## Работа с randомом

Для решения задач данного блока вам понадобятся следующие методы: [Math.random](#).

Выведите на экран случайное целое число от **1** до **100**.

Заполните массив **10-ю** **случайными** целыми числами. (Подсказка: нужно воспользоваться циклами **for** или **while**). \_

## Работа с модулем

Для решения задач данного блока вам понадобятся следующие методы: [Math.abs](#).

Даны переменные **a** и **b**. Найдите модуль разности **a** и **b**. Проверьте работу скрипта самостоятельно для различных **a** и **b**.

Даны переменные **a** и **b**. Отнимите от **a** переменную **b** и результат присвойте переменной **c**. Сделайте так, чтобы в любом случае в переменную **c** записалось **положительное** значение. Проверьте работу скрипта при **a** и **b**, равных соответственно 3 и 5, 6 и 1.

## Задачи

Дан массив **arr**. Найдите среднее арифметическое его элементов. Проверьте задачу на массиве с элементами **12, 15, 20, 25, 59, 79**.

Напишите скрипт, который будет находить **факториал** числа. **Факториал** (обозначается **!**) - это произведение (умножение) всех целых чисел, меньше данного, и его самого. Например, **4! = 1\*2\*3\*4**.

# Задачи на функции работы со строками в JavaScript

## Задача . Поиск и замена

**Задача.** Дана строка 'aaa@bbb@ccc'. Замените все @ на '!' с помощью **глобального** поиска и замены.

**Решение:** в данном случае необходимо воспользоваться [методом replace](#), который выполняет поиск и замену. Однако, при простом варианте использования, этот метод найдет и заменит **только первое совпадение**:

```
var str = 'aaa@bbb@ccc';  
alert(str.replace('@', '!')); //получим 'aaa!bbb@ccc'
```

Чтобы заменить все совпадения, воспользуемся **глобальным** поиском с помощью **регулярного выражения**:

```
var str = 'aaa@bbb@ccc';  
alert(str.replace(/@/g, '!')); //получим 'aaa!bbb!ccc'
```

## Задача . Методы substr, substring, slice

**Задача.** Дана строка 'aaa bbb ccc'. Вырежьте из нее слово 'bbb' тремя разными способами (через [substr](#), [substring](#), [slice](#)).

**Решение:** слово 'bbb' начинается с символа номер 4 (нумерация с нуля), а заканчивается символом номер 6. Воспользуемся указанными методами:

```
var str = 'aaa bbb ccc';  
alert(str.substr(4, 3)); //substr(откуда отрезать, сколько отрезать)  
alert(str.substring(4, 7)); //substring(откуда отрезать, докуда отрезать)  
alert(str.slice(4, 7)); //slice(откуда отрезать, докуда отрезать)
```

Обратите внимание на то, что в методах **substring** и **slice** второй параметр должен быть на 1 больше того символа, который мы хотим забрать (то есть, если указать число 7 - то отрезание произойдет до 6 символа включительно).

## Задача . Преобразование формата даты

**Задача.** В переменной **date** лежит дата в формате '2025-12-31'. Преобразуйте эту дату в формат '31/12/2025'.

**Решение:** с помощью [метода split](#) разобьем нашу строку '2025-12-31' в массив по разделителю '-', при этом в нулевом элементе окажется год, в первом - месяц, во втором - день:

```
var str = '2025-12-31';  
var arr = split('-');  
alert(arr); //получим массив ['2025', '12', '31']
```

Теперь, обращаясь к разным элементам массива по их ключам, сформируем нужную нам строку:

```
var str = '2025-12-31';
var arr = split('-');
var newStr = arr[2] + '/' + arr[1] + '/' + arr[0];
alert(newStr); //получим строку '31/12/2025'
```

## Задачи для решения

### Работа с регистром символов

Для решения задач данного блока вам понадобятся следующие методы: [toUpperCase](#), [toLowerCase](#).

Дана строка 'js'. Сделайте из нее строку 'JS'.

Дана строка 'JS'. Сделайте из нее строку 'js'.

### Работа с length, substr, substring, slice. Работа с indexOf

Для решения задач данного блока вам понадобятся следующие методы: [length](#), [substr](#), [substring](#), [slice](#), [indexOf](#).

Дана строка 'я учу javascript!'. Найдите количество символов в этой строке.

Дана строка 'я учу javascript!'. Вырежьте из нее слово 'учу' и слово 'javascript' тремя разными способами (через [substr](#), [substring](#), [slice](#)).

Дана строка 'я учу javascript!'. Найдите позицию подстроки 'учу'.

Дана переменная **str**, в которой хранится какой-либо текст. Реализуйте обрезание длинного текста по следующему принципу: если количество символов этого текста больше заданного в переменной **n**, то в переменную **result** запишем первые **n** символов строки **str** и добавим в конец троеточие '...'. В противном случае в переменную **result** запишем содержимое переменной **str**.

### Работа с replace

Для решения задач данного блока вам понадобятся следующие методы: [replace](#).

Дана строка 'Я-учу-javascript!'. Замените все дефисы на '!' с помощью глобального поиска и замены.

### Работа с split

Для решения задач данного блока вам понадобятся следующие методы: [split](#).

Дана строка 'я учу javascript!'. С помощью метода **split** запишите каждое слово этой строки в отдельный элемент массива.

Дана строка 'я учу javascript!'. С помощью метода **split** запишите каждый символ этой строки в отдельный элемент массива.

В переменной **date** лежит дата в формате '2025-12-31'. Преобразуйте эту дату в формат '31.12.2025'.



## Работа с join

*Для решения задач данного блока вам понадобятся следующие методы: [join](#).*

Дан массив ['я', 'учу', 'javascript', '!']. С помощью метода **join** преобразуйте массив в строку 'я+учу+javascript+!'.

## Задачи

Преобразуйте **первую букву** строки в верхний регистр.

Преобразуйте первую букву **каждого слова** строки в верхний регистр.

Преобразуйте строку 'var\_test\_text' в 'varTestText'. Скрипт, конечно же, должен работать с любыми аналогичными строками.

# Задачи на функции работы с массивами в JavaScript

## Примеры решения задач

### Задача

**Задача.** Даны два массива: ['a', 'b', 'c'] и [1, 2, 3]. Объедините их вместе.

**Решение:** объединим эти массивы с помощью [метода concat](#):

```
var a = ['a', 'b', 'c'];
```

```
var b = [1, 2, 3];
```

```
//Вариант 1:
```

```
var c = a.concat(b);
```

```
alert(c); //получится массив ['a', 'b', 'c', 1, 2, 3]
```

```
//Вариант 2:
```

```
var c = b.concat(a);
```

```
alert(c); //получится массив [1, 2, 3, 'a', 'b', 'c']
```

### Задача

**Задача.** Дан массив ['a', 'b', 'c']. Добавьте ему в конец элементы 1, 2, 3.

**Решение:** воспользуемся [методом push](#):

```
var arr = ['a', 'b', 'c'];
```

```
arr.push(1, 2, 3);
```

```
alert(arr); //получится массив ['a', 'b', 'c', 1, 2, 3]
```

## Задачи для решения

### Работа с concat

Для решения задач данного блока вам понадобятся следующие методы: [concat](#).

Даны два массива: [1, 2, 3] и [4, 5, 6]. Объедините их вместе.

### Работа с reverse

Для решения задач данного блока вам понадобятся следующие методы: [reverse](#).

Дан массив [1, 2, 3]. Сделайте из него массив [3, 2, 1].

## Работа с push, unshift

Для решения задач данного блока вам понадобятся следующие методы: [push](#), [unshift](#).

Дан массив [1, 2, 3]. Добавьте ему в конец элементы 4, 5, 6.

Дан массив [1, 2, 3]. Добавьте ему в начало элементы 4, 5, 6.

## Работа с shift, pop

Для решения задач данного блока вам понадобятся следующие методы: [shift](#), [pop](#).

Дан массив ['js', 'css', 'jq']. Выведите на экран **первый** элемент.

Дан массив ['js', 'css', 'jq']. Выведите на экран **последний** элемент.

## Работа с slice

Для решения задач данного блока вам понадобятся следующие методы: [slice](#).

Дан массив [1, 2, 3, 4, 5]. С помощью метода **slice** запишите в новый массив элементы [1, 2, 3].

Дан массив [1, 2, 3, 4, 5]. С помощью метода **slice** запишите в новый массив элементы [4, 5].

## Работа с splice

Для решения задач данного блока вам понадобятся следующие методы: [splice](#).

Дан массив [1, 2, 3, 4, 5]. С помощью метода **splice** преобразуйте массив в [1, 4, 5].

Дан массив [1, 2, 3, 4, 5]. С помощью метода **splice** запишите в новый массив элементы [2, 3, 4].

Дан массив [1, 2, 3, 4, 5]. С помощью метода **splice** сделайте из него массив [1, 2, 3, 'a', 'b', 'c', 4, 5].

Дан массив [1, 2, 3, 4, 5]. С помощью метода **splice** сделайте из него массив [1, 'a', 'b', 2, 3, 4, 'c', 5, 'e'].

## Работа с sort

Для решения задач данного блока вам понадобятся следующие методы: [sort](#).

Дан массив [3, 4, 1, 2, 7]. Отсортируйте его.

## Работа с Object.keys

Для решения задач данного блока вам понадобятся следующие методы: [Object.keys](#).

Дан объект {js: 'test', jq: 'hello', css: 'world'}. Получите массив его **ключей**.

# Практика на комбинации стандартных функций JavaScript

## Задачи для решения

### Отработка стандартных функций JavaScript

*В следующих задачах вы должны применить комбинацию стандартных функций JavaScript, чтобы решить задачу. Все методы вам известны, если вы делали задачи предыдущих уроков и проходили тесты.*

Дана строка. Сделайте заглавным первый символ этой строки не используя цикл. Найдите два решения. \_

Дана строка, например, '123456'. Переверните эту строку (сделайте из нее '654321') не используя цикл. \_

Проверьте, что строка начинается на **http://**. \_

Проверьте, что строка заканчивается на **.html**. \_

# Задачи на основы работы с пользовательскими функциями в JavaScript

## Примеры решения задач

### Задача

**Задача.** Сделайте функцию, которая возвращает куб числа. Число передается параметром.

**Решение:**

```
function cube($num) {  
    return $num * $num * $num  
}
```

## Задачи для решения

### Простые функции

Сделайте функцию, которая возвращает квадрат числа. Число передается параметром.

Сделайте функцию, которая возвращает сумму двух чисел.

Сделайте функцию, которая отнимает от первого числа второе и делит на третье.

Сделайте функцию, которая принимает параметром число от 1 до 7, а возвращает день недели на русском языке.