

## Оглавление

Задание 1. Каким будет результат? .....	2
Задание 2. Проверка на спам. ....	3
Задание 3. Случайное число от min до max.....	4
Задание 4. Палиндром. ....	5
Задание 5. FizzBuzz.....	6
Задание 6. Поиск гласных .....	7
Полезные правила.....	8

### Задание 1. Каким будет результат?

Вывести результат и объяснить его в комментариях к каждой строке.



```
1 "" + 1 + 0
2 "" - 1 + 0
3 true + false
4 6 / "3"
5 "2" * "3"
6 4 + 5 + "px"
7 "$" + 4 + 5
8 "4" - 2
9 "4px" - 2
10 "  -9  " + 5
11 "  -9  " - 5
12 null + 1
13 undefined + 1
14 " \t \n" - 2
```



```
1 5 > 4
2 "ананас" > "яблоко"
3 "2" > "12"
4 undefined == null
5 undefined === null
6 null == "\n0\n"
7 null === +"\n0\n"
```

## Задание 2. Проверка на спам.

Напишите функцию `checkSpam(str)`, возвращающую `true`, если `str` содержит `'viagra'` или `'XXX'`, а иначе `false`. Функция должна быть нечувствительна к регистру:

Тест кейсы. (Примеры использования и как сломать)



```
1 checkSpam('buy ViAgRA now') == true
2 checkSpam('free xxxxx') == true
3 checkSpam("innocent rabbit") == false
```

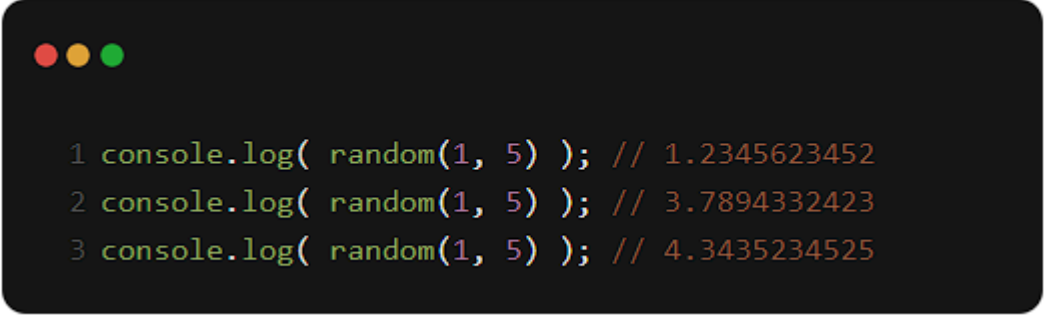
### Задание 3. Случайное число от `min` до `max`.

Встроенный метод `Math.random()` возвращает случайное число от 0 (включительно) до 1 (но не включая 1)

Напишите функцию `random(min, max)`, которая генерирует случайное число с плавающей точкой от `min` до `max` (но не включая `max`).

Тест кейсы. (Примеры использования и как сломать)

Пример работы функции:



```
1 console.log( random(1, 5) ); // 1.2345623452
2 console.log( random(1, 5) ); // 3.7894332423
3 console.log( random(1, 5) ); // 4.3435234525
```

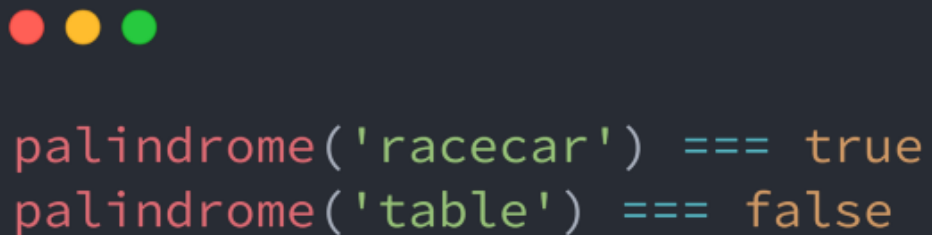
## Задание 4. Палиндром.

Палиндром — слово, предложение или последовательность символов, которая абсолютно одинаково читается как в привычном направлении, так и в обратном. К примеру, “Anna” — это палиндром, а “table” и “John” — нет.

### *Постановка*

Дана строка; нужно написать функцию, которая позволяет вернуть значение true, если строка является палиндромом, и false — если нет. При этом нужно учитывать пробелы и знаки препинания.

Так же нужно написать тест кейсы. ( Разные варианты использования функции – с пробелами, запятыми и т.д )



```
palindrome('racecar') === true
palindrome('table') === false
```

### *Разбираем задание*

Основная идея здесь — перевернуть строку в обратном направлении. Если «реверсная» строка полностью идентична исходной, значит, мы получили палиндром и функция должна вернуть значение true. Если же нет — false.

## Задание 5. FizzBuzz


Одна из самых популярных на собеседованиях задач.

### Постановка

Требуется написать функцию, выводящую в консоль числа от 1 до  $n$ , где  $n$  — это целое число, которая функция принимает в качестве параметра, с такими условиями:

- вывод `fizz` вместо чисел, кратных 3;
- вывод `buzz` вместо чисел, кратных 5;
- вывод `fizzbuzz` вместо чисел, кратных как 3, так и 5.

Так же нужно написать тест кейсы.



```
Fizzbuzz(5)
// Результат:
// 1
// 2
// fizz
// 4
// buzz
```

### Разбираем задание

Главное здесь — способ поиска кратных чисел с использованием JavaScript. Его можно реализовать при помощи оператора модуля или же остатка — `%`, который позволяет показать остаток при делении двух чисел. Если остаток 0, это означает, что первое число кратно второму.

`12 % 5 // 2 -> 12 is not a multiple of 5`  
`12 % 3 // 0 -> 12 is multiple of 3`

Так, если разделить 12 на 5, получаем 2 с остатком 2. Если же разделить 12 на 3, то получаем 4 с остатком 0. В первом случае 12 не кратно 5, во втором — 12 кратно 3.

## Задание 6. Поиск гласных

Достаточно простая задача, которая часто попадает на собеседованиях.

### Постановка

Нужно написать функцию, принимающую строку в качестве аргумента и возвращающую количество гласных, которые содержатся в строке. Гласными являются «а», «е», «i», «о», «и».

Так же нужно написать тест кейсы.

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It displays two lines of code with their corresponding outputs: 

```
findVowels('hello') // --> 2  
findVowels('why') // --> 0
```


```
findVowels('hello') // --> 2  
findVowels('why') // --> 0
```

### Разбираем задание

Важно обратить внимание на использование метода `.includes()`. Он доступен и для строк, и для массивов. Его стоит применять для того, чтобы выявить, содержит ли массив определенное значение. Этот метод возвращает `true`, если массив содержит указанное значение, и `false`, если нет.

Либо задействуем метод `.match()`, который позволяет реализовать эффективный поиск. Если регулярное выражение как аргумент метода обнаружено внутри указанной строки, то возвращаемым значением становится массив совпадающих символов. Ну а если совпадений нет, то `.match()` возвращает `null`.

## Полезные правила.

- 
- 1) НЕТ волшебных чисел
  - 2) НЕТ имен со скрытым значением и двумя смыслами
  - 3) НЕТ имен с малозаметным различием
  - 4) НЕТ не информативных имен
  - 5) НЕТ типов в названии имен
  - 6) НЕТ неудобно произносимых имен
  - 7) НЕТ префиксов членов классов
  - 8) НЕТ закодированного интерфейса в имени
  - 9) СУЩЕСТВИТЕЛЬНЫЕ в именах классов и объектов
  - 10) ГЛАГОЛЫ в именах методов
  - 11) ОДНО слово для каждой концепции
  - 12) ИМЕНА из пространства решений и задачи
- ИМЕНА с содержательным контекстом