1. Какие кавычки можно использовать для создания строк в JS и в чем разница между ними?

Строкой может быть любой текст внутри кавычек. Можно использовать как одинарные, так и двойные кавычки, можно использовать кавычки внутри строки, пока они не будут соответствовать кавычкам вокруг строки:

И также можно использовать третий вид кавычек - обратные. Если использовать обратные кавычки, то в такую строку мы сможем вставлять произвольные выражения, обернув их в `${…}`

1. Какими методами можно найти подстроку в строке? Приведите пример.

Метод `indexOf()` возвращает индекс (позицию) в первое вхождение указанного текста в строку:

var str = "Я живу в Москве";

var pos = str.indexOf("Москве");

Метод `lastIndexOf()` возвращает индекс последнего вхождения указанного текста в строку:

Оба метода `indexof()` и `lastIndexOf()` возвращают значение `-1`, если текст не найден.

1. Самостоятельно разберитесь, зачем нужен специальный символ [\\n](file:///\\n)?

В строке одна обратная косая черта-это так называемый символ 'escape'. Это используется для включения специальных символов,  таких как tab (\t) или новая строка (\n).

[\\n](file:///\\n) – символ обозначает новую строку.

1. Напишите код, который делает первый символ заглавным. Например, "настя" ⇒ "Настя"
2. Как создать дату 24 января 2021 года, 22 часа 51 минута? Временная зона – местная.

let date = new Date (2021,0,24,22,51,);

alert(date);

1. Как посчитать, сколько секунд осталось до завтра?

Создайте функцию getSecondsToTomorrow(), возвращающую количество секунд до завтрашней даты.

Например, если сейчас 23:00, то:

getSecondsToTomorrow() == 3600

Функция должна работать в любой день, т.е. в ней не должно быть конкретного значения сегодняшней даты.

1. Как выделить из строки с денежной суммой (например, *120р.* или *99€*) только цифры, т.е. *120 и 99*?

Функция parseInt() принимает строку в качестве аргумента и возвращает целое число в соответствии с указанным основанием системы счисления.

parseInt(‘*120р*’) // вернет число 120

1. Зачем нужны функции join и split?

Метод [str.split(delim)](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/String/split) именно это и делает. Он разбивает строку на массив по заданному разделителю delim.

let names = 'Вася, Петя, Маша';

let arr = names.split(', ');

for (let name of arr) {

alert( `Сообщение получат: ${name}.` ); // Сообщение получат: Вася (и другие имена)

}

Метод [str.split(delim)](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/String/split) именно это и делает. Он разбивает строку на массив по заданному разделителю delim.

let arr = ['Вася', 'Петя', 'Маша'];

let str = arr.join(';'); // объединить массив в строку через ;

alert( str ); // Вася;Петя;Маша

1. Какой из вариантов округления делает это по математическим правилам?

Math.round(параметр) – математическое округление, т.е. когда дробная часть числа больше или равна 0.5, то оно округляется в большую сторону, а иначе в меньшую сторону;

1. Как сгенерировать случайное число от 1 до 100?

let numberRandom = Math.round((Math.random()\*100));

1. Зачем нужна функция str.trim() ?

Метод **trim()** удаляет пробельные символы с начала и конца строки. Пробельными символами в этом контексте считаются все собственно пробельные символы (пробел, табуляция, неразрывный пробел и прочие) и все символы конца строки (LF, CR и прочие).

Метод trim() возвращает строку с вырезанными пробельными символами с её концов. Метод trim() не изменяет значение самой строки.

1. Что такое флаг?

Регулярные выражения могут иметь флаги, которые влияют на поиск.

В JavaScript их всего шесть

Регулярное выражение (оно же «регэксп», «регулярка» или просто «рег»), состоит из *шаблона* (также говорят «паттерн») и необязательных *флагов*.

Существует два синтаксиса для создания регулярного выражения.

«Длинный» синтаксис:

regexp = new RegExp("шаблон", "флаги");

…И короткий синтаксис, использующий слеши "/":

regexp = /шаблон/; // без флагов

regexp = /шаблон/gmi; // с флагами gmi (будут описаны далее)