**Вопросы 💎**

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

Строку начинают и оканчивают одинаковыми одинарными или двойными кавычками. Можно использовать кавычки внутри строки,отлтчные от наружных.

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

Плюс одно преимущество обратных кавычек — они могут занимать более одной строки.

Для одинарных или двойных кавычек такое не работает.

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

- Метод indexOf(): Этот метод возвращает индекс первого вхождения подстроки в строку. Если подстрока не найдена, метод возвращает -1. (в JavaScript индексация начинается с нуля)

/\* indexOf чувствителен к регистру \*/

'ITGirls'.indexOf('T'); // 1

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

'ITGirls ITGirls'.lastIndexOf('s'); // 14

- Похожим образом работает метод search() — он тоже ищет строки по заданному значению и возвращает позицию совпадения:

const str = "Я живу в Москве";

console.log(str.search("Москве")); // 9

- Метод includes(): Этот метод позволяет проверить, содержится ли подстрока в строке. Он возвращает булево значение true, если подстрока найдена, и false в противном случае.

'хлеб да соль'.indexOf('соль') !== -1; // true

'хлеб да соль'.indexOf('сахар') !== -1; // false

- JavaScript также можно использовать регулярные выражения для поиска подстроки в строке. Например, вы можете использовать метод match() в сочетании с регулярным выражением для поиска всех вхождений подстроки.

const str = "Hello, world!";

const regex = /o/g;

const matches = str.match(regex);

console.log(matches); // Выводит: ["o", "o"]

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

интерпретируется как символ новой строки

Напишите код, который делает первый символ заглавным. Например, "настя" ⇒ "Настя"

str = `настя`;

console.log(str.charAt(0));// н

const firstLetter= str.charAt(0).toUpperCase() + str.slice(1);

console.log(firstLetter);

Какой из вариантов округления делает это по математическим правилам?

// округляет число вниз до ближайшего целого, поэтому результат будет 9

Math.floor(9.99)

// округляет число вверх до ближайшего целого, поэтому результат будет 10

Math.ceil(9.01)

// округляет число до ближайшего целого, поэтому результат будет 10

Math.round(9.51)

// возвращает наибольшее из переданных чисел, поэтому результат будет 5

Math.max(1, 2, 3, 4, 5)

// возвращает наименьшее из переданных чисел, поэтому результат будет 1

Math.min(1, 2, 3, 4, 5)

// возвращает случайное число в диапазоне от 0 (включительно) до 1 (не включительно)

Math.random(); // 0.31764219954126016

Как сгенерировать случайное число от 1 до 100?

const randomNumber = Math.floor(Math.random() \* 100) + 1;

console.log(randomNumber);

// генерирует случайное число в диапазоне от 0 (включительно) до 1 (исключая 1). Затем это число умножается на 100 для получения случайного числа от 0 (включительно) до 100 (исключая 100). Затем используется Math.floor() для округления результата вниз до ближайшего целого числа.

Наконец, добавляется 1 к округленному числу для получения случайного числа в диапазоне от 1 до 100. Это нужно, чтобы исключить 0 из диапазона и сделать верхнюю границу включительной.

1. Что выведет в консоль? **The string length is 6**

const x="Солнце";

console.log("The string length is " + x.length)

1. Каким будет результат в представленном ниже фрагменте кода? **"ВЕЛОСИПЕД"**

const upperText="ВелоСипед" ;

document.write(upperText.toUpperCase())

1. Каким будет результат в приведенном ниже фрагменте кода? **“красивые такие розы”**

let str = "розы такие розы";

str.replace("розы","красивые")

1. Что выведет в консоль? **(2,3): cde**

const str = "abcdefghij";

console.log("(2,3): " + str.substr(2,3));