**Упражнение 4.** Используя учебные ресурсы <http://javascript.ru/string>и <https://learn.javascript.ru/string> , привести описание не рассмотренных выше методов для работы со строками. Результаты поместить в таблицу:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Код** | **Выполнение** | **Объяснение** |
| 1 | var str = "Учим Ява Скрипт"  str.search("Ява") | 5 | search() ищет совпадения между выражением и строкой текста и возвращает позицию найденного совпадения (если оно имеется).  Данный метод возвращает -1 если совпадение не было найдено. |
| 2 | str.split([separator][, limit]); |  | **separator**  регулярное выражение или строка, по которой делить str  **limit**  максимальное количество кусков, на которые может быть разбита строка  Метод split возвращает новый массив.  Строка бьется по separator, при разбивании separator пропадает:   |  | | --- | | arr = "a,b,c".split(',')  // массив ["a", "b", "c"] | | Иными словами, разбитие строки в массив | |
| 3 | Str.fromCharCode(num1, ..., numN) |  | **num1, ..., numN**  последовательность чисел, представляющих собой коды Unicode-символов  String.fromCharCode(65,66,67)  // вернет строку "ABC".  Создание строки по таблице символов Unicode |
| 4 | str.charCodeAt(index) |  | **index**  целое число от 0 до длины строки-1  Юникодное значение от 0 до 1,114,111. Первые 128 значений Unicode совпадают с кодировкой ASCII.  Заметим, что charCodeAt() всегда возвращает значение, меньшее 65536.  charCodeAt() возвращает NaN, если указанный индекс меньше нуля или больше/равен длине строки   |  | | --- | | // возвратит 65, unicode-код для А |  |  | | --- | | "ABC".charCodeAt(0) // 65 | |
| 5 | str.concat(string1, string2[, ..., stringN]) |  | **String1...stringN**  сроки, которые будут прибавлены к str   |  | | --- | | a = "строка" |  |  |  | | --- | --- | | 2 | b = " очень" |  |  |  | | --- | --- | | 3 | с = " длинная" |  |  |  | | --- | --- | | 4 |  |  |  |  | | --- | --- | | 5 | alert(a.concat(b,c))  // строка очень длинная | |
| 6 | str.valueOf() |  | Метод valueOf возвращает элементарное значение для объекта String.   |  | | --- | | x = new String("Hello world"); |  |  | | --- | | alert(x.valueOf()) |    // покажет"Hello world" |
| 7 | string.substr(start[, length]) |  | **start**  начальная позиция, начиная от 0  **length**  сколько символов брать с позиции start  Метод substr начинает собирать строку с позиции start и заканчивает, когда соберет length или дойдет до конца строки.   |  | | --- | | var str = "abcdefghij"; |  |  |  | | --- | --- | | 2 | alert("(1,2): "    + str.substr(1,2));  //выводит (1,2): bc | |
| 8 | str.replace(regexp, newSubStr|function) |  | **regexp**  объект типа [RegExp](http://javascript.ru/RegExp) или строка  **newSubStr|function**  новая подстрока или функция, которая ее возвращает  Этот метод не меняет вызывающую строку, а возвращает новую, после замен.   |  | | --- | | str = "тест еще тест" |  |  | | --- | | str.replace("тест","прошел")  // = "прошел еще тест" | |

**Упражнение 5.** В скороговорке «Шла cаша по шоссе» сделать имя с большой буквы.

var g="Шла саша по шоссе";

var netw=g.slice(0,4)+ g[4].toUpperCase() + g.slice(5,g.length)

console.log(netw)

**Упражнение 6.** Исправить ошибки: «Не хочу учится а хочу женится».

var word = "Не хочу учится а хочу женится"

console.log(word)

word = word.replace("учится", "учиться,")

word = word.replace("женится", "жениться")