|  |  |  |  |
| --- | --- | --- | --- |
|  | **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 | var arr = str.split([separator][, limit]); |  | **1)separator**  регулярное выражение или строка, по которой делить str  **limit**  максимальное количество кусков, на которые может быть разбита строка  2)Метод split возвращает новый массив.  Строка бьется по separator, при разбивании separator пропадает:   |  | | --- | | arr = "a,b,c".split(',')  // массив ["a", "b", "c"] | |
| 3 | var codepoint = string.charCodeAt(index) |  | **1)index**  целое число от 0 до длины строки-1  2) Юникодное значение от 0 до 1,114,111. Первые 128 значений Unicode совпадают с кодировкой ASCII.  charCodeAt() возвращает NaN, если указанный индекс меньше нуля или больше/равен длине строки. |
| 4 | String.fromCharCode(num1, ..., numN) |  | **1)num1, ..., numN**  последовательность чисел, представляющих собой коды Unicode-символов  2) Этот метод возвращает элементарную строку, а не объект типа String. |
| 5 | str.charAt(index) |  | **1)index**  Целое число от 0 до количества символов в строке минус 1  2) Символы идут слева направо. Первый символ имеет позицию 0, а последний string.length-1.  Если указанный index лежит за пределами этого диапазона, яваскрипт вернет пустую строку. |
| 6 | result = string1.concat(string2, string3[, ..., stringN]) |  | **1)string2...stringN**  сроки, которые будут прибавлены к string1  2) Выполняет действие, аналогичное оператору + для строк. |
| 7 | str.lastIndexOf(searchValue[, fromIndex]) |  | **1)searchValue**  искомая подстрока  **fromIndex**  индекс, от которого начинать поиск по направлению к началу строки. От 0 до str.length-1. Значение по умолчанию: str.length  2) Символы в строке идут слева направо. Позиция первого символа 0, последнего - длина строки минус один.  Метод lastIndexOf чувствителен к регистру символов. |
| 8 | idx = str.search(regexp) |  | **1)regexp**  объект типа [RegExp](http://javascript.ru/regexp) или строка  2) Этот метод удобен, когда нужно проверить, есть ли совпадения с регулярным выражением (аналогично [RegExp#test](http://javascript.ru/RegExp/test)).  Чтобы получить больше информации, используйте более медленный метод [match](http://javascript.ru/string/match) (или родственный ему метод [RegExp#exec](http://javascript.ru/RegExp/exec)). |
| 9 | str.match(regexp) |  | **1)regexp**  регулярное выражение в виде объекта [RegExp](http://javascript.ru/regexp) или строка  2) Если регулярное выражение без флага "g", то возвращает такое же результат, как [regexp.exec(str)](http://javascript.ru/RegExp/exec).  Если же для регулярного выражения указан флаг "g", то метод возвращает массив, содержащий все совпадения. Если совпадений нет, то возвращает null. |
| 10 | str.toLowerCase() |  | Перевод в нижний регистр. |
| 11 | str.toUpperCase() |  | Перевод в верхний регистр |
| 12 | str.toLocaleLowerCase() |  | Обычно работает в точности как [toLowerCase](http://javascript.ru/string/toLowerCase). Различия есть только для некоторых языков, например, для турецкого, правила которых противоречат соглашениям Unicode.  Для русского результаты совпадают. |
| 13 | str.toLocaleUpperCase() |  | Обычно работает в точности как [toUpperCase](http://javascript.ru/String/toUpperCase). Различия есть только для некоторых языков, например, для турецкого, правила которых противоречат соглашениям Unicode.  Для русского результаты совпадают. |
| 14 | str.toString() |  | Перекрывает соответствующий метод [Object#toString](http://javascript.ru/object/tostring), возвращает элементарную строку. |
| 15 | str.valueOf() |  | Метод valueOf возвращает элементарное значение для объекта [String](http://javascript.ru/string). Это значение эквивалентно [String#toString](http://javascript.ru/String/tostring).  Обычно этот метод вызывается неявно, внутри интерпретатора javascript, а не в коде. |
| 16 | str.substring(indexA, [indexB]) |  | **1)indexA**  целое число от 0 до длины строки-1  **indexB**  целое число от 0 до длины строки-1  2) Метод substring возвращает подстроку, начиная с позиции indexA до, но не включая indexB.  В частности:   * Если indexA = indexB, возвращается пустая строка * Если indexB не указан, substring возвращает символы до конца строки * Если какой-то из аргументов меньше 0 или является NaN - он считается равным 0 * Если какой-то из аргументов больше, чем длина строки - он считается равным длине строки   Если indexA > indexB, тогда substring ведет себя, как будто аргументы поменялись местами. |
| 17 | var sub = string.slice(beginSlice[, endSlice]) |  | **1)beginSlice**  позиция, начиная от 0, с которой начинается участок  **endSlice**  позиция, на которой заканчивается участок  2) Метод slice возвращает часть строки от beginSlice до endSlice, но не включая символ под номером endSlice, не меняя вызывающую строку.  Если параметр endSlice меньше нуля, то он обозначает сдвиг относительно конца строки. |
| 18 | str.indexOf(searchValue[, fromIndex]) |  | **1)searchValue**  искомая подстрока  **fromIndex**  позиция, с которой начинать поиск: от 0 до длины строки-1  2) Поиск подстроки осуществляется слева направо. Метод чувствителен к регистру символов. |
| 19 | var sub = string.substr(start[, length]) |  | **1)start**  начальная позиция, начиная от 0  **length**  сколько символов брать с позиции start  2) Этот метод описан только в ненормативной части стандарта ECMA, но реализован во всех браузерах.  Метод substr начинает собирать строку с позиции start и заканчивает, когда соберет length или дойдет до конца строки. |
| 20 | str.replace(regexp, newSubStr|function) |  | **1)regexp**  объект типа [RegExp](http://javascript.ru/RegExp) или строка  **newSubStr|function**  новая подстрока или функция, которая ее возвращает  2) Этот метод не меняет вызывающую строку, а возвращает новую, после замен.  Чтобы произвести глобальный поиск и замену, используйте regexp c флагом "g". |