|  |  |  |
| --- | --- | --- |
| Код | Выполнение | Объяснение |
| var a = "DOTA";  a.toLocaleLowerCase(); | dota | метод **toLocaleLowerCase()** преобразует строку в строчные буквы (нижний регистр) с учетом текущего языкового стандарта. Для того, чтобы преобразовать строку в заглавные буквы (верхний регистр) с учетом текущего языкового стандарта вы можете воспользоваться методом [**toLocaleUpperCase()**](https://basicweb.ru/javascript/js_string_tolocaleuppercase.php)**.** |
| var a= "Game".concat("", "Dota!"); | Game Dota! | метод **concat()** используется для объединения двух, или более массивов в один, при этом он не изменяет существующие массивы, а возвращает новый массив. |
| var sim = "d,o,t,a";  sim.split(","); | ["d", "o", "t", "a"] | метод **split()** позволяет разбить строки на массив подстрок, используя заданную строку разделитель для определения места разбиения. |
| a = "Go DOTA"; var b = str.replace("DOTA", "RUST"); | Go RUST | Метод Replace () заменяет указанное значение другим значением в строке: |
| **var** a = "**sun**"  *a*.toLocaleUpperCase(); | **"SUN"** | ToLocaleUpper Case  Делает тоже самое, что и ToUpperCase, но с учетом текущего языкового стандарта |
| **var** a = "**somebody**";  *a*.lastIndexOf( "**body**" );  Просмотр демо в редакторе | **4** | метод **lastIndexOf()** позволяет возвратить **последний** индекс искомого элемента в массиве при первом совпадении, или -1 если элемент не найден. Для того, чтобы произвести поиск необходимого элемента с **начала** массива, Вы можете воспользоваться методом [**indexOf()**](https://basicweb.ru/javascript/js_array_indexof.php)**.** |
| var a = "work on Russia"; var b = str.search("Russia");  alert(b) | -1 | Метод Search () ищет строку для заданного значения и возвращает позицию совпадения: |
| var a = "Doter"; a.charCodeAt(2); | **Возвращает 116** | charCodeAt()  Возвращает числовое значение символа по указанному индексу в стандарте кодирования символов Unicode |
| var a = " are you doter?";  a.substring(4); | **you doter?** | Substring  аналогична slice(), но substring() не может принимать отрицательные индексы. |
| String.fromCharCode( 115, 116, 117 ); | stu | String.fromChar Code  Статический метод String.fromCharCode() возвращает строку, созданную из указанной последовательности значений никода |
| **var** a = "**gamer**",  **console**.log(a.valueOf() ); | **gamer** | метод **valueOf()** возвращает примитивное значение строкового объекта в виде строкового типа данных. |
| var a = "study in KIPU";  var b = a.substr(6);  console.log(b); | in KIPU | substr () похож на slice (). Разница в том, что второй параметр указывает длину извлеченной детали.  Если опустить второй параметр, substr () будет разрезать оставшуюся часть строки. |
|  |  |  |