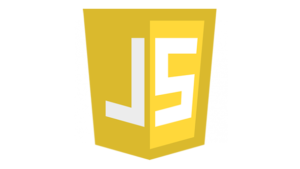
**JAVA SCRIPT STRINGS**

[Javascript](https://www.edureka.co/javascript-jquery-training) is a high-level programming language which is having curly bracket syntax, dynamic typing & prototype based on object-orientation. In [Java programming](https://www.edureka.co/blog/java-tutorial/), the strings are treated as articles. Java stage gives the String class to create and control strings. This **JavaScript String Functions** will list out some of the most commonly used string functions.

Some of the important javascript string functions include:

* [char](https://www.edureka.co/blog/javascript-string-functions/#charAt(x)) At(x)
* [charCodeAt(x)](https://www.edureka.co/blog/javascript-string-functions/#charCodeAt(x))
* [concat(v1,v2..)](https://www.edureka.co/blog/javascript-string-functions/#concat(v1,v2..))
* [fromCharcode(c1,c2)](https://www.edureka.co/blog/javascript-string-functions/#fromCharcode(c1,c2))
* [indexOf(substr, [start])](https://www.edureka.co/blog/javascript-string-functions/#indexOf(substr,%5Bstart%5D))
* [lastIndexOf(substr, [start])](https://www.edureka.co/blog/javascript-string-functions/#lastIndexOf(substr,%5Bstart%5D))
* [match(regexp)](https://www.edureka.co/blog/javascript-string-functions/#match(regexp))
* [replace(regexp/substr, replacetext)](https://www.edureka.co/blog/javascript-string-functions/#replace(regexp/substr,replacetext))
* [search(regexp)](https://www.edureka.co/blog/javascript-string-functions/#search(regexp))
* [slice(start, [end])](https://www.edureka.co/blog/javascript-string-functions/#slice(start,%5Bend%5D))
* [split(delimiter, [limit])](https://www.edureka.co/blog/javascript-string-functions/#split(delimiter,%5Blimit%5D))
* [substr(start, [length])](https://www.edureka.co/blog/javascript-string-functions/#substr(start,%20%5Blength%5D))
* [substring(from, [to])](https://www.edureka.co/blog/javascript-string-functions/#substring(from,%20%5Bto%5D))
* [toLowerCase()](https://www.edureka.co/blog/javascript-string-functions/#toLowerCase())
* [toUpperCase()](https://www.edureka.co/blog/javascript-string-functions/#toUpperCase())
* [includes()](https://www.edureka.co/blog/javascript-string-functions/#includes)
* [endsWith()](https://www.edureka.co/blog/javascript-string-functions/#endswith)
* [repeat()](https://www.edureka.co/blog/javascript-string-functions/#repeat)
* [valueOf()](https://www.edureka.co/blog/javascript-string-functions/#linkValueOf)
* [trim()](https://www.edureka.co/blog/javascript-string-functions/#trim)

Now let’s move ahead and have a look the string functions.

**1.charAt(x)**

This function will return the character at the x position within the string.

|  |  |
| --- | --- |
| 1  2  3  4 | //charAt(x)  var myString = 'jQuery FTW!!!';  console.log(myString.charAt(7));  //output: F |

**2.charCodeAt(x)**

This function will return the unicode value of the character at position ‘x’ within the string.

|  |  |
| --- | --- |
| 1  2  3  4 | //charAt(position)  var message="jquery4u"  //alerts "q"  alert(message.charAt(1) |

**3.concat(v1,v2..)**

This function combines one or more strings(argv1,v2 etc) into existing one.

|  |  |
| --- | --- |
| 1  2  3  4  5 | //concat(v1, v2,..)  var message="Sam"  var final=message.concat(" is a"," hopeless romantic.")  //alerts "Sam is a hopeless romantic."  alert(final) |

**4.fromCharcode(c1,c2)**

Function will return a string created by using specified sequence of unicode values(argc1,c2).

|  |  |
| --- | --- |
| 1  2  3  4  5 | //fromCharCode(c1, c2,...)  console.log(String.fromCharCode(97,98,99,120,121,122))  //output: abcxyz  console.log(String.fromCharCode(72,69,76,76,79))  //output: HELLO |

**5.indexOf(substr, [start])**

Searches and (if found) returns the index number of the searched character or [substring within the string](https://www.edureka.co/blog/javascript-substring/). If not found, -1 is returned. “Start” is an optional argument specifying the position within string to begin the search. Default is 0.

|  |  |
| --- | --- |
| 1  2  3  4 | //indexOf(char/substring)  var sentence="Hi, my name is Sam!"  if (sentence.indexOf("Sam")!=-1)  alert("Sam is in there!") |

Moving on with this article on Javascript String functions

**6.lastIndexOf(substr, [start])**

Searches and (if found) returns the index number of the searched character or substring within the string. Searches the string from end to the beginning. If not found, -1 is returned. “Start” is an optional argument specifying the position within string to begin the search. Default is string.length-1.

**[pt and JQue](https://www.edureka.co/javascript-jquery-training" \t "_blank)**

[m](https://www.edureka.co/javascript-jquery-training" \t "_blank)

|  |  |
| --- | --- |
| 1  2  3  4 | //lastIndexOf(substr, [start])  var myString = 'javascript rox';  console.log(myString.lastIndexOf('r'));  //output: 11 |

**7.match(regexp)**

Executes a search for a match within a string based on a regular expression. It returns an [array](https://www.edureka.co/blog/javascript-array/) of information or null if no match is found.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | //match(regexp) //select integers only  var intRegex = /[0-9 -()+]+$/;    var myNumber = '999';  var myInt = myNumber.match(intRegex);  console.log(isInt);  //output: 999    var myString = '999 JS Coders';  var myInt = myString.match(intRegex);  console.log(isInt);  //output: null |

**8.replace(regexp/substr, replacetext)**

Searches and replaces the regular expression (or sub string) portion (match) with the replaced text instead.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | //replace(substr, replacetext)  var myString = '999 JavaScript Coders';  console.log(myString.replace(/JavaScript/i, "jQuery"));  //output: 999 jQuery Coders    //replace(regexp, replacetext)  var myString = '999 JavaScript Coders';  console.log(myString.replace(new RegExp( "999", "gi" ), "The"));  //output: The JavaScript Coders |

**9.search(regexp)**

Tests for a match in a string. It returns the index of the match, or -1 if not found.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | //search(regexp)  var intRegex = /[0-9 -()+]+$/;    var myNumber = '999';  var isInt = myNumber.search(intRegex);  console.log(isInt);  //output: 0 |

**10.slice(start, [end])**

This function returns a substring of the string based on the “start” and “end” index arguments, NOT including the “end” index itself. “End” is optional, and if none is specified, the slice includes all characters from “start” to end of the string.

|  |  |
| --- | --- |
| 1  2  3  4 | //slice(start, end)  var text="excellent"  text.slice(0,4) //returns "exce"  text.slice(2,4) //returns "ce" |

Moving on with this article on Javascript String functions

**11.split(delimiter, [limit])**

This will split a string into many according to the specified delimiter, and returns an array containing each element. The optional “limit” is an integer that lets you specify the maximum number of elements to return.

|  |  |
| --- | --- |
| 1  2  3  4  5 | //split(delimiter)  var message="Welcome to jQuery4u"  //word[0] contains "We"  //word[1] contains "lcome to jQuery4u"  var word=message.split("l") |

**12.substr(start, [length])**

This function returns the characters in a string beginning at “start” and through the specified number of characters, “length”. “Length” is optional, and if omitted, up to the end of the string is assumed.

|  |  |
| --- | --- |
| 1  2  3  4 | //substring(from, to)  var text="excellent"  text.substring(0,4) //returns "exce"  text.substring(2,4) //returns "ce" |

**Front End Web Development Training**

**[WEB DEVELOPMENT CERTIFICATION TRAINING](https://www.edureka.co/complete-web-developer" \o "Web Development Certification Training " \t "_blank)**

**[Web Development Certification Training](https://www.edureka.co/complete-web-developer" \o "Web Development Certification Training " \t "_blank)**

*[Reviews](https://www.edureka.co/complete-web-developer" \o "Web Development Certification Training " \t "_blank)*

**[5](https://www.edureka.co/complete-web-developer" \o "Web Development Certification Training " \t "_blank)**[(6447)](https://www.edureka.co/complete-web-developer" \o "Web Development Certification Training " \t "_blank)

**[ANGULAR CERTIFICATION TRAINING](https://www.edureka.co/angular-training" \o "Angular Certification Training" \t "_blank)**

**[Angular Certification Training](https://www.edureka.co/angular-training" \o "Angular Certification Training" \t "_blank)**

*[Reviews](https://www.edureka.co/angular-training" \o "Angular Certification Training" \t "_blank)*

**[5](https://www.edureka.co/angular-training" \o "Angular Certification Training" \t "_blank)**[(16002)](https://www.edureka.co/angular-training" \o "Angular Certification Training" \t "_blank)

**[REACT WITH REDUX CERTIFICATION TRAINING](https://www.edureka.co/reactjs-redux-certification-training" \o "React with Redux Certification Training " \t "_blank)**

**[React with Redux Certification Training](https://www.edureka.co/reactjs-redux-certification-training" \o "React with Redux Certification Training " \t "_blank)**

*[Reviews](https://www.edureka.co/reactjs-redux-certification-training" \o "React with Redux Certification Training " \t "_blank)*

**[5](https://www.edureka.co/reactjs-redux-certification-training" \o "React with Redux Certification Training " \t "_blank)**[(5246)](https://www.edureka.co/reactjs-redux-certification-training" \o "React with Redux Certification Training " \t "_blank)

**[JAVASCRIPT AND JQUERY ESSENTIALS TRAINING AND CERTIFICATION](https://www.edureka.co/javascript-jquery-training" \o "JavaScript and JQuery Essentials Training and Certification" \t "_blank)**

**[JavaScript and JQuery Essentials Training and Certification](https://www.edureka.co/javascript-jquery-training" \o "JavaScript and JQuery Essentials Training and Certification" \t "_blank)**

*[Reviews](https://www.edureka.co/javascript-jquery-training" \o "JavaScript and JQuery Essentials Training and Certification" \t "_blank)*

**[5](https://www.edureka.co/javascript-jquery-training" \o "JavaScript and JQuery Essentials Training and Certification" \t "_blank)**[(2508)](https://www.edureka.co/javascript-jquery-training" \o "JavaScript and JQuery Essentials Training and Certification" \t "_blank)

**[JQUERY UI DEVELOPMENT](https://www.edureka.co/jquery-ui-development" \o "jQuery UI Development" \t "_blank)**

**[jQuery UI Development](https://www.edureka.co/jquery-ui-development" \o "jQuery UI Development" \t "_blank)**

*[Reviews](https://www.edureka.co/jquery-ui-development" \o "jQuery UI Development" \t "_blank)*

**[5](https://www.edureka.co/jquery-ui-development" \o "jQuery UI Development" \t "_blank)**[(996)](https://www.edureka.co/jquery-ui-development" \o "jQuery UI Development" \t "_blank)

Next

**13.substring(from, [to])**

It returns the characters in a string between “from” and “to” indexes, NOT including “to” itself. “To” is optional, and if omitted, up to the end of the string is assumed.

|  |  |
| --- | --- |
| 1  2  3  4  5 | //substring(from, [to])  var myString = 'javascript rox';  myString = myString.substring(0,10);  console.log(myString)  //output: javascript |

**14.toLowerCase()**

This will return the string with all of its characters converted to lowercase.

|  |  |
| --- | --- |
| 1  2  3  4  5 | //toLowerCase()  var myString = 'JAVASCRIPT ROX';  myString = myString.toLowerCase();  console.log(myString)  //output: javascript rox |

**15.toUpperCase()**

This will return the string with all of its characters converted to uppercase.

|  |  |
| --- | --- |
| 1  2  3  4  5 | //toUpperCase()  var myString = 'javascript rox';  myString = myString.toUpperCase();  console.log(myString)  //output: JAVASCRIPT ROX |

**16. includes()**

It is used to check whether a string contains the specified string or characters.

|  |  |
| --- | --- |
| 1  2  3  4 | //includes()  var mystring = "Hello, welcome to MIT";  var n = mystring.includes("MIT");  //output: True |

**17. endsWith()**

This function checks whether a string ends with specified string or characters.

|  |  |
| --- | --- |
| 1  2  3  4 | //endsWith()  var mystr = "List of javascript functions";  var n = mystr.endsWith("functions");  //output: True |

**18. repeat()**

This returns a new string with a specified number of copies of an existing string.

|  |  |
| --- | --- |
| 1  2  3  4 | //repeat()  var string = "Welcome to MIT";  string.repeat(2);  //output: Welcome to MIT Welcome to MIT |

**19. valueOf()**

It is used to return the primitive value of a String object.

|  |  |
| --- | --- |
| 1  2  3  4 | //valueOf()  var mystr = "Hello World!";  var res = mystr.valueOf();  //output: Hello World! |

[[](https://www.edureka.co/javascript-jquery-training)](https://www.edureka.co/javascript-jquery-training" \t "_blank)

**[JavaScript and JQuery Essentials Training and Certification](https://www.edureka.co/javascript-jquery-training" \t "_blank)**

[Weekday / Weekend BatchesSee Batch Details](https://www.edureka.co/javascript-jquery-training" \t "_blank)

**20. trim()**

This function removes whitespace from both ends of a string.

|  |  |
| --- | --- |
| 1  2  3 | //trim()  var str = "     Hello MIT!     ";  alert(str.trim()); |

These were some of the most commonly used JavaScript String functions. With this, we have come to the end of our article.