

## String Methods

### 1) concat()

The concat method appends a string to the end of another string.

eg: ~~const~~ fname = "rondu"  
const lname = "vinod"

fname.concat(lname)

//output: rondivinod

### 2) includes()

The includes method return true if a string contains a specified string. otherwise it returns false. It is case sensitive.

eg: let text = "hello good morning"  
text.includes('good')

//output: true

### 3) indexOf()

The `indexOf()` method returns the position of the first occurrence of specified character in a string.

eg: let text = "hello good morning"  
text.indexOf('good')

"3" = text.indexOf('good')

//output : 6

### 4) lastIndexOf()

The `lastIndexOf()` method returns the position of the last occurrence of specified character in a string.

eg: let text = "hello planet earth, you are a great planet"

text.lastIndexOf('planet')

//output : 36

5) padEnd()

The `padEnd()` method pads a string at the end. It pads a string with another string (multiple times) until it reaches the given length.

eg: let text = "5"

text.padEnd(4, "x")

// output : 5xxx

6) padStart()

The `padStart()` method pads a string from the start. It pads a string with another string until it reaches a given length.

eg: let text = "5"

text.padStart(4, "x")

// output : xxx5



### 7) repeat ()

repeat () method returns a string with a number of copies of a string. It returns a new string. It does not change the original string.

eg: let text = "hello world"

let newText = text.repeat(2)

//output : hello worldhello world

### 8) replace ()

replace () method searches a string for a value or a regular expression. It returns a new string with the value(s) replaced. It does not change the original string.

eg: let text = "hello good morning"  
text.replace('morning', 'afternoon')

//output : hello good afternoon.

### 9) search()

search() method matches a string against a regular expression. It returns the index (position) of the first match. It returns -1 if no match is found. It is case sensitive.

eg: let text = "my name is nandini"  
text.search('my')

//output: 6

### 10) slice()

slice() method extracts a part of a string. It returns the extracted part as a new string. It does not change the original string.

eg: let text = "hello world"  
text.slice(0, 5)

//output = hello

## 11) split()

split method splits a string into an array of substrings. It returns the new array. It does not change the original string.

eg: let text = "hello world"  
text.split(" ")

// output: hello, world

## 12) startsWith()

startsWith method returns true if a string starts with a specified string. Otherwise it returns false. It is case sensitive.

eg: let text = "hello world"  
text.startsWith("hello")

// output: true



### 13) substr()

substr() method extracts a part of a string. It begins at a specified position and returns a specified number of characters. It does not change the original string.

eg: let text = 'hello world'  
text.substr(0, 5)

// output : hello

### 14) substring()

substring() method extracts characters between two indices (position), from a string and returns the substring. It extracts characters from start to end, but it does not change the original string.

eg: let text = 'hello world'

text.substring(0, 5)

// output: 'hello'

15) toLowerCase()

toLowerCase() method convert to lower case letter.

eg: let text = "HELLO WORLD"

text.toLowerCase()

// output: 'hello world'

16) toUpperCase()

toUpperCase() method convert a string to upper case letter.

eg: let text = "hello world"

text.toUpperCase()

// output: 'HELLO WORLD'



17) trim()

The trim() method removes ~~white~~ whitespace from both end of string.

eg: let text = " hello world "

text.trim()

//output : hello world

18) trimEnd()

trimEnd() method removes white from the end of a string. It does not change the original string. It works like trim() but remove whitespace only from the end of a string.

eg: let text = " hello world "

text.trimEnd()

//output : hello world.

19) trimStart()

trimStart method remove whitespace from the beginning of a string. It does not change the original string. It works like trim(), but remove whitespace only from the start of a string.

eg: let text = " hello world "

text.trimStart()

//output : hello world

20) charAt()

charAt() method returns the character at the specified index in a string.

eg: let text = "hello world"

text.charAt(0)

//output = h

## 21) charAt()

The `charAt()` method returns Unicode of the character at a specific index in a string.

eg: let `text = "hello world"`

`text.charAt(0)`

Output : `h`