

String Method

```
// 1. string
// total number of length of string
const abcd = "abcdefghijklmnopqrstuvwxyz";
console.log(abcd.length);

// 2. toLowerCase()
const var1 = "THIS IS SOME TEXT"
// the lowercase method convert a string to lowercase letters.
console.log(var1.toLowerCase());

// 3. toUpperCase()
// the uppercase method converts a string to upper case letters.
const var2 = "this is a some text"
console.log(var2.toUpperCase());

// 4. includes()
// the include method return true if a string contains a specified string.
// otherwise it return false.
// the include method is a case sensitive.
const var3 = "this is a search engine"
console.log(var3.includes("rch"));

// 5. startWith()
// the startWith method return a true if a string start with a specified
string.
// otherwise return false
// the startwith method is a case sensitive.

const var4 = "javascript is a greate language"
console.log(var4.startsWith("javascript")); // output : true
console.log(var4.startsWith("java")); //output : true
console.log(var4.startsWith("Java")); //output : false

// 6. endWith
// the endwith method return a true if a string ends with a specific string.
// otherwise it reuturn false.
// the endsWith method is case sensitive.
const var5 = "javascript is a greate language"
console.log(var5.endsWith("language")); // output : true
console.log(var5.endsWith("uage")); // output : true
console.log(var5.endsWith("Language")); // output : false

// 7. search
// this method matches a string against regular expression.
// this method return the index position of the first match.
```

```
// this method return -1 if no match is found.
// this method is a case sensitive.
let var6 = "Mr. Blue has a blue house";
console.log(var6.search("has")); // output : 9
console.log(var6.search("Has")); // output : -1
console.log(var6.search("use")); // output : 22
```

```
// 8. match()
// match method matches a string against a regular expression **
// this method return an array with the matches.
// the match method return null if no match is found.
// g stands for globally and i stands for case-insensitive
let var7 = "The rain in SPAIN stays mainly in the plain";
console.log(var7.match(/in/g)); // output : [ 'in', 'in', 'in', 'in', 'in' ]
console.log(var7.match(/in/gi)); // output : [ 'in', 'in', 'IN', 'in', 'in',
'in' ]
console.log(var7.match(/ipt/g)); // output : null
```

```
// 9. indexOf()
// this method return the position of the occurrence of a value in a string.
// means jo usko pahela keyword milega uski position return karega.
// this method return -1 if the value is not found.
// method is a case-sensitive.
let var8 = "Hello worlde, welcome to the universe.";
console.log(var8.indexOf("wel")); // output : 13
console.log(var8.indexOf("e",5)); // output : 11
console.log(var8.indexOf("e")); // output : 1
let var9 = "Hello world, welcome to the universe.";
console.log(var9.indexOf("e",5)); // output : 14
let var10 = "Hello world, wflcome to the universe.";
console.log(var10.indexOf("e",5)); // output : 19
```

```
// 10. lastIndexOf()
// this method search the string from the end to the beginning.
// this method return the index from the beginning (position 0)
// this method return -1 if the value is not found.
// this is case-sensitive
let var11 = "Hello planet earth, you are a great planet.";
console.log(var11.lastIndexOf("great")); // output : 30
console.log(var11.lastIndexOf("Great")); // output : -1
console.log(var11.lastIndexOf("th")); // output : 16
```

```
// 11. replace()
// this method searches a string for a value.
// method return a new string with the values replaced.
// method does not changes the original string.
let var12 = "Visit Microsoft!";
console.log(var12.replace("Microsoft","google")); // output : Visit google!
```

```
let var13 = "Mr Blue has a blue house and a blue car";
console.log(var13.replace(/blue/g,"red")); // output : Mr Blue has a red
house and a red car
```

```
// 12. replaceAll()
// method return a new string with all values replaced.
let var14 = "I love cats. Cats are very easy to love. Cats are very
popular."
console.log(var14.replaceAll("Cats","Dogs")); //output : I love cats. Dogs
are very easy to love. Dogs are very popular.
console.log(var14.replaceAll("cats","dogs")); //output : I love dogs. Cats
are very easy to love. Cats are very popular.
```

```
// 13. trim()
// the trim method removes whitespace from both side of a string.
// trim method does not change the original string.
const var15 = " Hello World "
console.log(var15); // output : before : Hello World
console.log(var15.trim()); // output : after : Hello World
```

```
// trimEnd()
// this method remove whitespace from the end of the String.
// they does not change the original string.
```

```
// trimStart()
// this method removes whiespaces from the beginning of a string.
// they does nor change the original string.
```

```
//remove spaces with replace() using a regular expression.
const var16 = " Hello World ";
console.log(var16);
console.log(var16.replace(/^\s+|\s+$/gm,'')); // same output above example
```

```
//14. charAt()
// this method return the character at a specified index (position) in a
string.
let var17 = "Visit Microsoft!";
console.log(var17.charAt(0)); //output : V
console.log(var17.charAt(6)); //output : M
```

```
// 15. charCodeAt()
// this method returns the unicode of the character at a specified
index(position) in a string.
// if a index(position) not found in string then return NaN.
let var18 = "HELLO WORLD";
console.log(var18.charCodeAt(0)); // output : 72
```

```
console.log(var18.charCodeAt(1)); // output : 69
console.log(var18.charCodeAt(2)); // output : 76
console.log(var18.charCodeAt(24)); // output : NaN
```

```
// 16. fromCharCode()
// the String.fromCharCode() method convert unicode values to characters.
// the String.fromCharCode() is a static method of the string object.
console.log(String.fromCharCode(65)); //output : A
console.log(String.fromCharCode(66)); //output : B
console.log(String.fromCharCode(70)); //output : F
```

```
// 17. the concat method joins two or more string.
// this method does not change the existing string.
// this method return a new string.
```

```
const var19 = "Hello"
const var20 = "World"
const var21 = "welcome";
console.log(var19.concat(" ",var21," ",var20)) // output : Hello Welcome
World
```

```
// 18. split()
// this method split a string and return an array.
//method return the new array.
// they does not change the original string.
let var22 = "How are you doing today?";
console.log(var22.split(" ")); //output : [ 'How', 'are', 'you', 'doing',
'today?' ]
const var23 = "new/string/here/for/example"
console.log(var23.split("/")); //output : [ 'new', 'string', 'here', 'for',
'example' ]
```

```
//19. repeat()
// this method return a string with a number of copies of a string.
// repeat method return a new string.
// they not change original string.
```

```
const var24 = "Hello World / "
console.log(var24.repeat(2)); //output : Hello World Hello World
console.log(var24.repeat(4)); //output : Hello World / Hello World / Hello
World / Hello World /
```

```
//20. slice()
// method extracts a part of a string.
const var25 = "javascript is a Great language"
console.log(var25.slice(0)); //output : javascript is a Great language
```

```
console.log(var25.slice(5)); //output : cript is a Great language
console.log(var25.slice(-1)); //output : e
console.log(var25.slice(-2)); //output : e
console.log(var25.slice(5,12)); //output : cript i
```

```
// 21. substr()
// this method extracts a part of a string.
// this begins at a specified position and return a specified number of
character.
const var26 = "javascript is a Great language"
console.log(var26.substring(0)); //output : javascript is a Great language
console.log(var26.substring(5)); //output : cript is a Great language
console.log(var26.substring(0,5)); //output : javas
```