String Length

let text = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
let length = text.length;

String slice()

let text = "Apple, Banana, Kiwi";  
let part = text.slice(7, 13);

If a parameter is negative, the position is counted from the end of the string:

let text = "Apple, Banana, Kiwi";  
let part = text.slice(-12);

## String substring()

let str = "Apple, Banana, Kiwi";  
let part = str.substring(7, 13);

## String substr()

let str = "Apple, Banana, Kiwi";  
let part = str.substr(7, 6);

## Replacing String

let text = "Please visit Microsoft!";  
let newText = text.replace("Microsoft", "W3Schools");

replace case insensitive, use a **regular expression** with an /i flag (insensitive):

let text = "Please visit Microsoft!";  
let newText = text.replace(/MICROSOFT/i, "W3Schools");

To replace all matches, use a **regular expression** with a /g flag (global match):

let text = "Please visit Microsoft and Microsoft!";  
let newText = text.replace(/Microsoft/g, "W3Schools");

## String ReplaceAll()

text = text.replaceAll("Cats","Dogs");  
text = text.replaceAll("cats","dogs");

## String toUpperCase()

## let text1 = "Hello World!"; let text2 = text1.toUpperCase();

## String toLowerCase()

## let text1 = "Hello World!";       // String let text2 = text1.toLowerCase();  // text2 is text1 converted to lower

## String concat()

concat() joins two or more strings:

let text1 = "Hello";  
let text2 = "World";  
let text3 = text1.concat(" ", text2);

## String trim()

The trim() method removes whitespace from both sides of a string:

## let text1 = "      Hello World!      "; let text2 = text1.trim();

## String trimStart()

The trimStart() method works like trim(), but removes whitespace only from the start of a string.

let text1 = "     Hello World!     ";  
let text2 = text1.trimStart();

## String trimEnd()

let text1 = "     Hello World!     ";  
let text2 = text1.trimEnd();

## padStart()

The padStart() method pads a string from the start.

It pads a string with another string (multiple times) until it reaches a given length.Pad a string with "0" until it reaches the length 4:

let text = "5";  
let padded = text.padStart(4,"0");

## String padEnd()

The padEnd() method pads a string from the end.

It pads a string with another string (multiple times) until it reaches a given length.

let text = "5";  
let padded = text.padEnd(4,"0")

The padEnd() method is a string method.

To pad a number, convert the number to a string first.

See the example below.

let numb = 5;  
let text = numb.toString();  
let padded = text.padEnd(4,"0");

## String charAt()

The charAt() method returns the character at a specified index (position) in a string:

let text = "HELLO WORLD";  
let char = text.charAt(0);

## String charCodeAt()

The charCodeAt() method returns the unicode of the character at a specified index in a string:The method returns a UTF-16 code (an integer between 0 and 65535).

let text = "HELLO WORLD";  
let char = text.charCodeAt(0);

## Property Access

ECMAScript 5 (2009) allows property access [ ] on strings:

let text = "HELLO WORLD";  
let char = text[0];

## String split()

A string can be converted to an array with the split() method:

text.split(",")    // Split on commas  
text.split(" ")    // Split on spaces  
text.split("|")    // Split on pipe