// JavaScript Strings:JavaScript strings are for storing and manipulating text.

// A JavaScript string is zero or more characters written inside quotes.

//Example

// let text = "John Doe";

// console.log(text);

// //Example

// let carName1 = "Volvo XC60"; // Double quotes

// let carName2 = "Volvo XC60"; // Single quotes

// let text1 = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

// let length = text1.length;

// console.log(length);

//JavaScript String slice()

//slice() extracts a part of a string and returns the extracted part in a new string.

//The method takes 2 parameters: start position, and end position (end not included).

//Example1

//Slice out a portion of a string from position 7 to position 13:

// let text = "Apple, Banana, Kiwi";

// let part = text.slice(7, 13);

// console.log(part)

//Example2

// let text = "Apple, Banana, Kiwi";

// let part = text.slice(7);

// console.log(part)

// Example3

//end is start from -1

// let text = "Apple, Banana, Kiwi";

// let part = text.slice(-12);

// console.log(part)

//Example4

// let text = "Apple, Banana, Kiwi";

// let l = text.length;

// let part = text.slice(-12. - 6);

// console.log(part);

//JavaScript String substring()

//substring() is similar to slice().

//The difference is that start and end values less than 0 are treated as 0 in substring().

//Example

// let str = "Apple, Banana, Kiwi";

// let part = str.substring(7, 13);

// console.log(part)

// let str = "Apple, Banana, Kiwi";

// let part = str.substring(-12);

// console.log(part)

// let str = "Apple, Banana, Kiwi";

// let part = str.substr(7, 6);

// console.log(part);

// let str = "Apple, Banana, Kiwi";

// let part = str.substr(7);

// console.log(part);

// let str = "Apple, Banana, Kiwi";

// let part = str.substr(-4);

// console.log(part);

//Replacing String Content

// The replace() method replaces a specified value with another value in a string:

// let text = "Please visit Microsoft!";

// let newText = text.replace("Microsoft", "W3Schools");

// console.log(newText);

// JavaScript String ReplaceAll()

// In 2021, JavaScript introduced the string method replaceAll():

// let text = "I love Cats. Cats are very easy to love. Cats are very popular";

// let text1 = text.replaceAll("Cats", "Dogs");

// console.log(text1);

// Converting to Upper and Lower Case

// A string is converted to upper case with toUpperCase():

// A string is converted to lower case with toLowerCase():

// JavaScript String toUpperCase()

// let text1 = "Hello World!";

// let text2 = text1.toUpperCase();

// console.log(text2);

// JavaScript String toLowerCase()

// let text1 = "Hello World!";       // String

// let text2 = text1.toLowerCase();  // text2 is text1 converted to lower

// console.log(text2);

//JavaScript String concat()

// concat() joins two or more strings:

// let text1 = "Hello";

// let text2 = "World";

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

// console.log(text3);

// JavaScript String charAt()

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

// let text = "HELLO WORLD";

// let char = text.charAt(3);

// console.log(char);

//Indexof:

// let text = "Pleae locate where 'locate' occurs!" ;

// let index = text.indexOf('locate');

// console.log(index);

//Last Indexof:

// let text = "Please locate where 'locate' occurs!" ;

// let index = text.lastIndexOf('locate');

// console.log(index);

//includes:

// let text = "Hello world, welcome to the universe." ;

// let text2 = text.includes('world');

// console.log(text2);

// let text = "Hello world, welcome to the universe." ;

// let text2 = text.includes('worlds');

// console.log(text2);

// Back-Tics Syntax

// Template Literals use back-ticks (``) rather than the quotes ("") to define a string:

// let text = `Hello World!`;

// console.log(text);

// Quotes Inside Strings

// With template literals, you can use both single and double quotes inside a string:

// let text = `He's often called "Johnny"`;

// console.log(text);

// Multiline Strings

// Template literals allows multiline strings:

// let text =

// `The quick

// brown fox

// jumps over

// the lazy dog`;

// console.log(text);

// Interpolation

// Template literals provide an easy way to interpolate variables and expressions into strings.

// The method is called string interpolation.

// The syntax is: ${...}

// Variable Substitutions

// Template literals allow variables in strings:

*let* firstName = "John";

*let* lastName = "Doe";

*let* text = `Welcome ${firstName}, ${lastName}!`;

console.log(text);

Topic: JavaScript Strings And String Function:String Length,slice(),substring(),substr(),Replacing String Content,ReplaceAll(),charAt()concat()Converting to Upper and Lower Case,indexOf(),lastIndexOf(),search(),includes(),startsWith(),endsWith(),Back-Tics Syntax,Quotes Inside Strings,Multiline Strings,Interpolation,Variable Substitutions