ANECO ACADEMY

JS-STRING_METHODS_LET_CONST

STRING METHODS

length: Returns the number of characters in a string

toUpperCase(): A string is converted to upper case

toLowerCase(): A string is converted to lower case

concat(): joins two or more strings, can be used instead of the plus operator

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

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

indexOf(): method returns the position of the first occurrence of a specified value in a string, method returns -1 if the value is not found

Slice(): extracts a part of a string and returns the extracted part in a new string, method takes 2 parameters: the start position, and the end position

split(): method divides a String into an ordered list of substrings and returns them as an array

STRING METHODS

Example

```
let txt = “JavaScript";
let length = txt.length;

let text = “I am learning JavaScript";
let newText = text.replace(“learning", “working");

let text = "Please visit Madras, I love Madras!";
let newText = text.replace(/Madras/g, “Chennai");

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

let text1 = "Hello World!";       // String
let text2 = text1.toLowerCase();

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

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

Example

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

const message = "JavaScript is fun";

// check the first occurrence of 'i' in message
let result = message.indexOf("i");
console.log(result);

str.indexOf(searchValue, fromIndex)

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

let str = "Apple, Banana, Kiwi";
let part = str.substring(7, 13);
let part1 = str.substring(7);

let message = "JavaScript is fun";
let result = message.split(" ");
```

HISTORY OF JAVSACRIPT

- **1996:** Changed from LiveScript to JavaScript to attract Java developers. **JavaScript has almost nothing to do with Java** ☝️
- **1997:** ES1 (ECMAScript 1) became the first version of the JavaScript language standard:
  - ECMAScript: The language standard;
  - JavaScript: The language in practice.
- **2009:** ES5 (ECMAScript 5) was released with lots of new features.
- **2015:** ES6/ES2015 (ECMAScript 2015) was released: **the biggest update to the language ever!**
- **2015:** Changed to an **annual release cycle** 🙏
- **2016/2017/2018/2019/...:** Release of ES2016/ES2017/ES2018/ES2019/...

ES6 INTRODUCTION

ES5

- Fully supported in all browsers;
- Ready to be used today 👍

http://kangax.github.io/compat-table

ES6/ES2015

ES7/ES2016

ES8/ES2017

- Well supported in all **modern** browsers
- No support in older browsers;
- Can use **most** features in production with transpiling and polyfilling (converting to ES5) 😃

ES9/ES2018

ES10/ES2019

- Future versions, together called ESNext;
- Some features supported in modern browsers;
- Can already use **some** features in production with transpiling and polyfilling 😁

let & const

Read more about `let` : https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/let

Read more about `const` : https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/const

`let` and `const` basically replace `var` . You use `let` instead of `var` and `const` instead of `var` if you plan on never re-assigning this "variable" (effectively turning it into a constant therefore).

VAR & LET & CONST

```
/*
// ES5
var name5 = 'Jane Smith';
var age5 = 23;
name5 = 'Jane Miller';
console.log(name5);

// ES6
const name6 = 'Jane Smith';
let age6 = 23;
name6 = 'Jane Miller';
console.log(name6);
```

VAR & LET & CONST

```
// ES5
function driversLicence5(passedTest) {

    if (passedTest) {
        console.log(firstName);
        var firstName = 'John';
        var yearOfBirth = 1990;
    }


    console.log(firstName + ', born in ' + yearOfBirth + ', is now officially allowed to drive a car.');
}

driversLicence5(true);


// ES6
function driversLicence6(passedTest) {

    //console.log(firstName);
    let firstName;
    const yearOfBirth = 1990;

    if (passedTest) {
        firstName = 'John';
    }

    console.log(firstName + ', born in ' + yearOfBirth + ', is now officially allowed to drive a car.');
}

driversLicence6(true);
```

VAR & LET & CONST

```
var i = 23;

for (var i = 0; i < 5; i++) {
    console.log(i);
}

console.log(i);
*/
```

VARIABLE SCOPE

JavaScript has 3 types of scope

- **Block scope** - Variables declared inside a { } block **cannot be accessed** from outside the block
- **Function scope** - Variables declared within a JavaScript function, become **LOCAL** to the function
- **Global scope** - A variable declared outside a function, becomes **GLOBAL**

```
{
 let x = 2;
}
// x can NOT be used here
```

Variables declared with the var keyword can NOT have block scope.

```
{
 var x = 2;
}
// x CAN be used here
```

```
// code here can NOT use myName

function myFunction() {
 let myName = "Raja";
 // code here CAN use myName
}

// code here can NOT use myName
```

```
let myName = "Ram";
// code here can use myName

function myFunction() {
// code here can also use myName
}
```

VARIABLE SETUP

Declare Variable

Three ways to declare a variable in javaScript

- Using var

```
var x = 5;
var y = 6;
var z = x + y;
```

- Using let

```
let x = "John Doe";
```

- Using const

```
const PI = 3.141592653589793;
```