String In JavaScript

In JavaScript, there are two primary ways to create strings: using string literals and using the `String` object.

String Literals

String is created by enclosing text within single quotes (`'`), double quotes (`"`), or backticks (`` ` ``).

```
<!-- ? example: -->

<!--
let str1 = 'this is string1';
let str2 = "this is string2";
let str3 = `this is string3`; -->
```

Characteristics of String Literals:

- They are simple to use and are the preferred way to create strings for most cases.
- Strings created using literals are primitive values.
- They have all the methods and properties of the `String` prototype.

String Objects

String objects are created using the `String` constructor function.

```
<!-- ? example: -->

<!-- let strObj = new String('this is string object'); -->
```

Characteristics of String Objects:

- They are objects, not primitive values.
- Because they are objects, they have additional properties and methods inherited from the `Object` prototype.
- String objects are rarely used in practice because they can introduce unnecessary complexity and performance overhead.

Key Differences

1. **Type Checking:**
   - **String Literal:** `typeof str1` returns `"string"`.
   - **String Object:** `typeof strObj` returns `"object"`.

2. **Performance:**
   - **String Literal:** Generally faster and more efficient.
   - **String Object:** Slower due to additional overhead of creating an object.

3. **Equality:**
   - **String Literal:** Compared by value.
   - **String Object:** Compared by reference, which can lead to unexpected results.

```
let str1 = 'Hello';
let str2 = 'Hello';
let strObj1 = new String('Hello');
let strObj2 = new String('Hello');

console.log(str1 === str2);       // true
console.log(strObj1 === strObj2); // false
console.log(str1 === strObj1);    // false
```

<!-- !   String Methods   -->

1. `toLowerCase()`

Converts a string to lowercase letters and returns one new string.

2 . `toUpperCase()`

Converts a string to uppercase letters and returns one new string.

3. `includes()`

Checks if a string contains specified characters and returns true/false.

4. `startsWith()`

Checks if a string starts with specified characters and returns true/false.

5. `endsWith()`

Checks if a string ends with specified characters and returns true/false.

6. `charAt()`

Returns the character at a specified index.

7. `indexOf()`

Returns the index of the first occurrence of a specified value.

8. `lastIndexOf()`

Returns the index of the last occurrence of a specified value.

9. `concat()`

Joins two or more strings and return new string.

10. `replace()`

Searches a string for a specified value, or a regular expression, and returns a new string where the specified values are replaced.

11. `replaceAll()`

this method is used to return a new string with all matches of a pattern replaced by a replacement

12. `split()`

Splits a string into an array of substrings.

13. `trim()`

Removes whitespace from both ends of a string and returns new string.

14. `trimStart()`

Removes whitespace from the beginning of a string.

15. `trimEnd()`

Removes whitespace from the end of a string.

16. `valueOf()`

Returns the primitive value of a String object.

17. `substr()`

The `substr()` method extracts a part of a string, beginning at a specified position, and returns a specified number of characters.

Syntax string.substr(start, length)

- `start`: The position where to start the extraction. If negative, it starts counting from the end of the string.
- `length`: The number of characters to extract. If omitted, it extracts to the end of the string.

18. `slice()`

The `slice()` method extracts a part of a string and returns it as a new string, without modifying the original string.

Syntax string.slice(startIndex, endIndex)

- `start`: The position where to start the extraction. If negative, it starts counting from the end of the string.
- `end`: The position where to end the extraction. If omitted, it extracts to the end of the string. The character at this index will not be included.

19. `substring()`

The `substring()` method extracts characters from a string, between two specified indices, and returns the new substring. The `substring()` method swaps the two arguments if `start` is greater than `end`.

Syntax string.substring(start, end)

Key Differences

1. **Handling of Negative Indices**:
   - `substr()`: If `start` is negative, it counts from the end of the string.
   - `slice()`: If `start` or `end` is negative, it counts from the end of the string.
   - `substring()`: Treats negative values as 0.

2. **Parameter Swapping**:
   - `substring(start, end)`: Swaps the two arguments if `start` is greater than `end`.
   - `slice(start, end)` and `substr(start, length)`: Do not swap the arguments.

20. `padStart()`

Pads the current string with another string (multiple times, if needed) until the resulting string reaches the given length. The padding is applied from the start (left) of the current string.

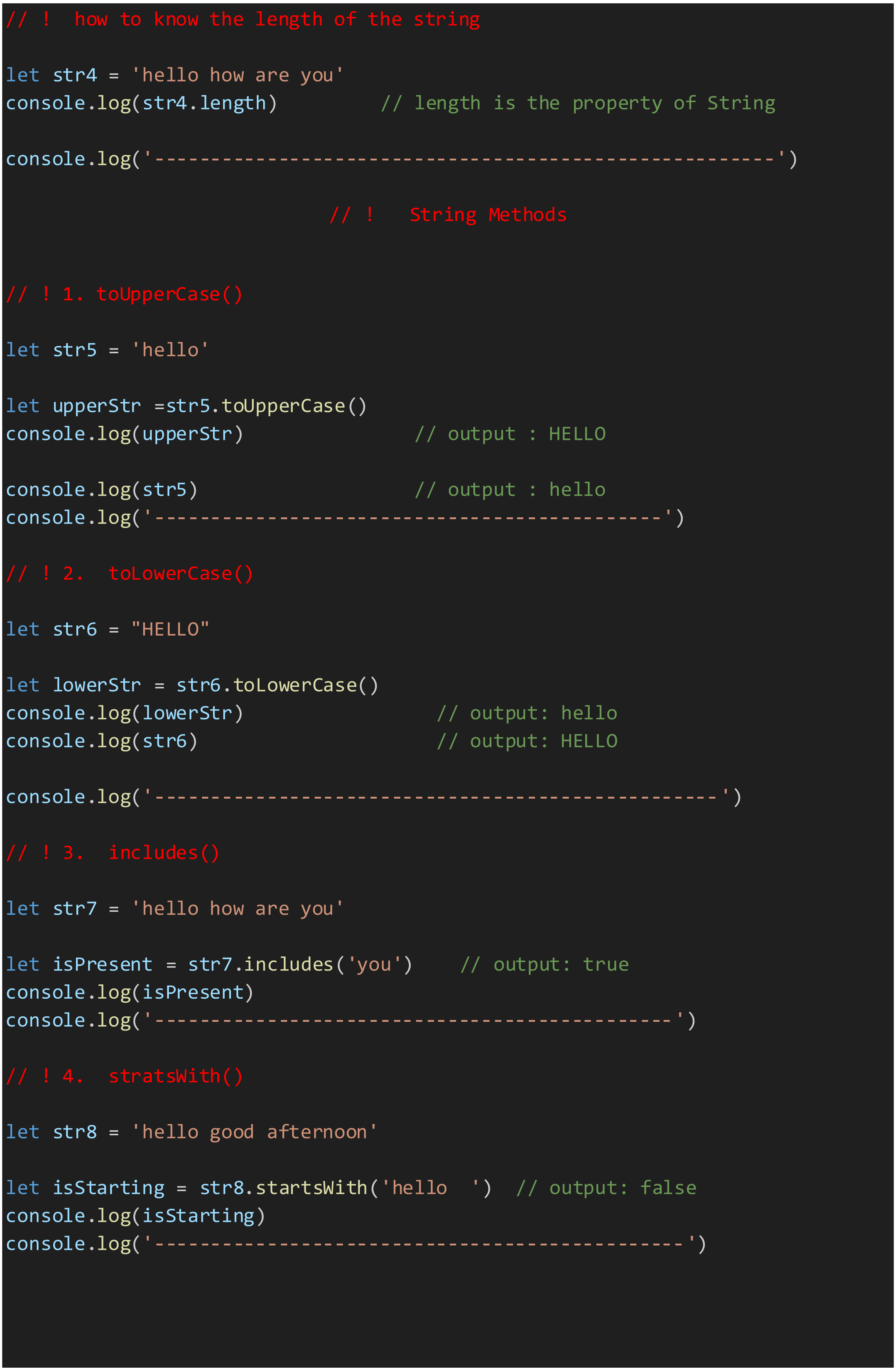
21. `padEnd()`

Pads the current string with another string (multiple times, if needed) until the resulting string reaches the given length. The padding is applied from the end (right) of the current string.

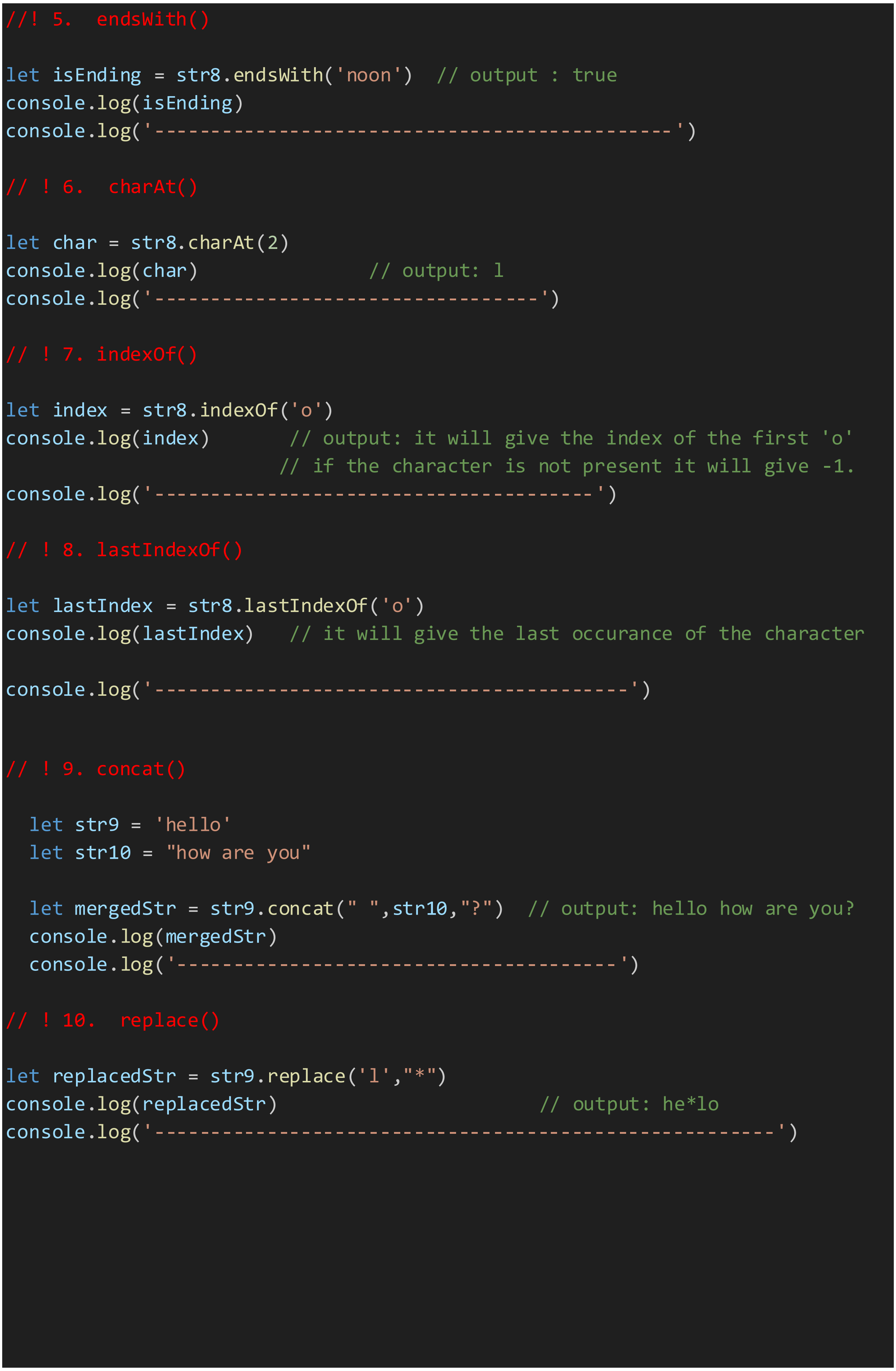
22. `search()`

Searches a string for a specified value, or a regular expression, and returns the position of the match.

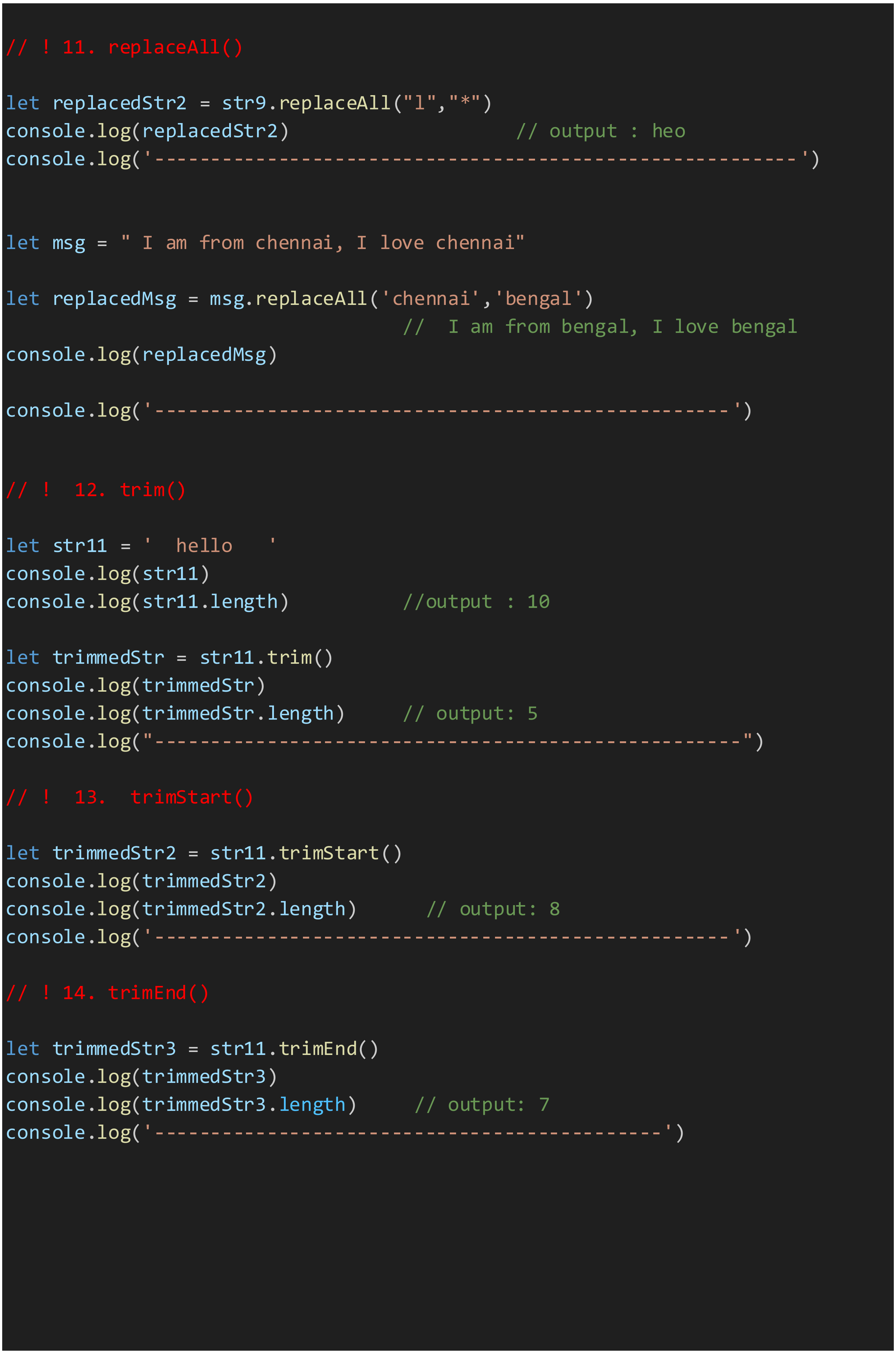
```
// !   How to declare String


                              // !  by literals

let str1 = 'this is string1'
console.log(str1)
console.log(typeof str1)

let str2 = "this is string2"
console.log(str2)
console.log(typeof str2)

let str3 = `this is string3`
console.log(str3)
console.log(typeof str3)
```



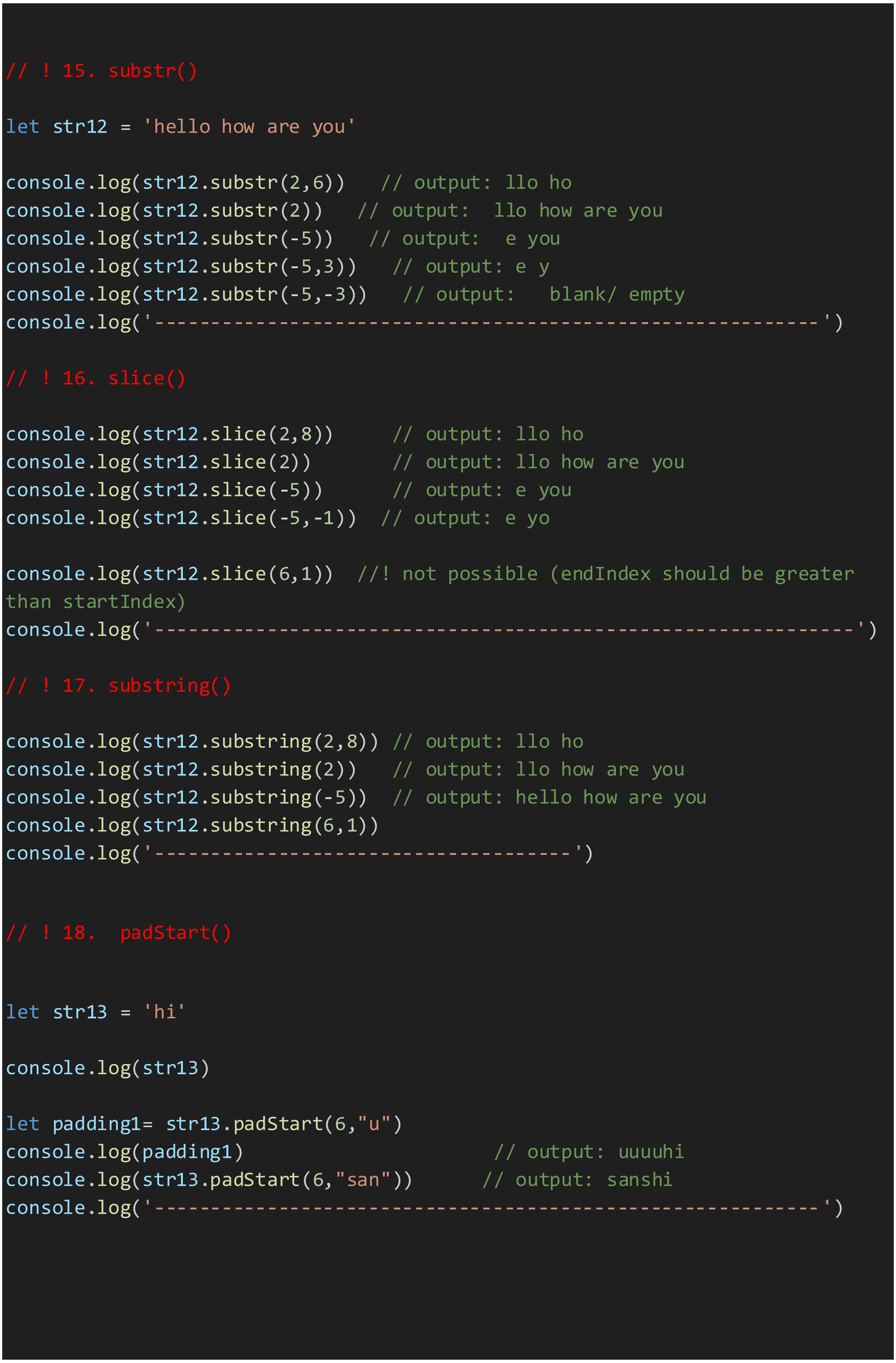
```
// !  how to know the length of the string

let str4 = 'hello how are you'
console.log(str4.length)         // length is the property of String

console.log('-------------------------------------------------------')

                             // !   String Methods


// ! 1. toUpperCase()

let str5 = 'hello'

let upperStr =str5.toUpperCase()
console.log(upperStr)               // output : HELLO

console.log(str5)                   // output : hello
console.log('---------------------------------------------')

// ! 2.  toLowerCase()

let str6 = "HELLO"

let lowerStr = str6.toLowerCase()
console.log(lowerStr)                 // output: hello
console.log(str6)                     // output: HELLO

console.log('--------------------------------------------------')

// ! 3.  includes()

let str7 = 'hello how are you'

let isPresent = str7.includes('you')    // output: true
console.log(isPresent)
console.log('----------------------------------------------')

// ! 4.  stratsWith()

let str8 = 'hello good afternoon'

let isStarting = str8.startsWith('hello  ')  // output: false
console.log(isStarting)
console.log('-----------------------------------------------')
```



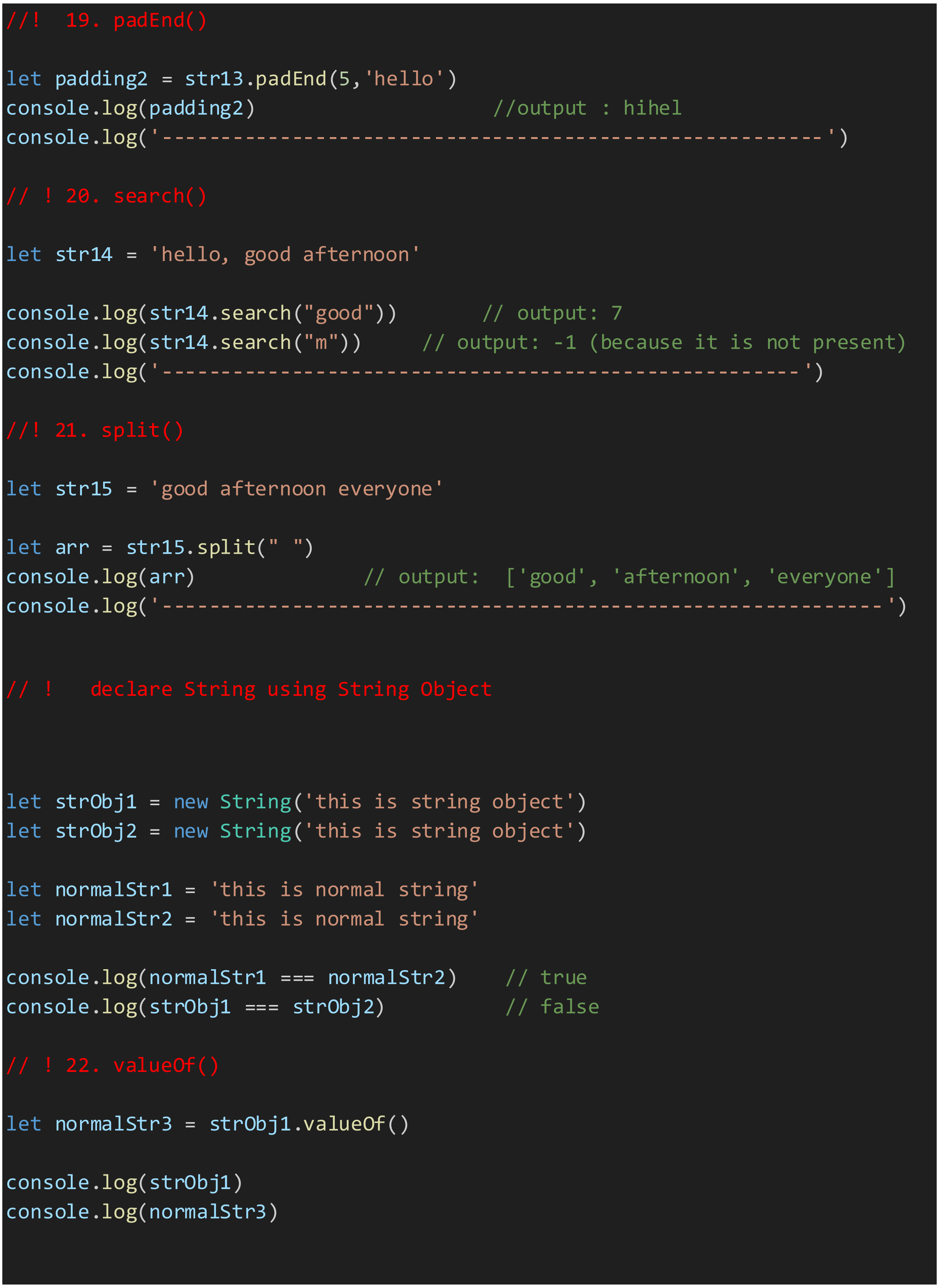
```
//! 5.  endsWith()

let isEnding = str8.endsWith('noon')  // output : true
console.log(isEnding)
console.log('---------------------------------------------')

// ! 6.  charAt()

let char = str8.charAt(2)
console.log(char)               // output: l
console.log('---------------------------------')

// ! 7. indexOf()

let index = str8.indexOf('o')
console.log(index)       // output: it will give the index of the first 'o'
                        // if the character is not present it will give -1.
console.log('--------------------------------------')

// ! 8. lastIndexOf()

let lastIndex = str8.lastIndexOf('o')
console.log(lastIndex)   // it will give the last occurance of the character

console.log('-----------------------------------------')


// ! 9. concat()

  let str9 = 'hello'
  let str10 = "how are you"

  let mergedStr = str9.concat(" ",str10,"?")  // output: hello how are you?
  console.log(mergedStr)
  console.log('--------------------------------------')

// ! 10.  replace()

let replacedStr = str9.replace('l',"*")
console.log(replacedStr)                       // output: he*lo
console.log('------------------------------------------------------')
```



```
// ! 11. replaceAll()

let replacedStr2 = str9.replaceAll("l","*")
console.log(replacedStr2)                    // output : heo
console.log('--------------------------------------------------------')


let msg = " I am from chennai, I love chennai"

let replacedMsg = msg.replaceAll('chennai','bengal')
                                   //  I am from bengal, I love bengal
console.log(replacedMsg)

console.log('---------------------------------------------------')


// !  12. trim()

let str11 = '  hello   '
console.log(str11)
console.log(str11.length)          //output : 10

let trimmedStr = str11.trim()
console.log(trimmedStr)
console.log(trimmedStr.length)     // output: 5
console.log("----------------------------------------------------")

// !  13.  trimStart()

let trimmedStr2 = str11.trimStart()
console.log(trimmedStr2)
console.log(trimmedStr2.length)      // output: 8
console.log('---------------------------------------------------')

// ! 14. trimEnd()

let trimmedStr3 = str11.trimEnd()
console.log(trimmedStr3)
console.log(trimmedStr3.length)     // output: 7
console.log('---------------------------------------------')
```



```
// ! 15. substr()

let str12 = 'hello how are you'

console.log(str12.substr(2,6))   // output: llo ho
console.log(str12.substr(2))   // output:  llo how are you
console.log(str12.substr(-5))   // output:  e you
console.log(str12.substr(-5,3))   // output: e y
console.log(str12.substr(-5,-3))   // output:   blank/ empty
console.log('-----------------------------------------------------------')

// ! 16. slice()

console.log(str12.slice(2,8))     // output: llo ho
console.log(str12.slice(2))       // output: llo how are you
console.log(str12.slice(-5))      // output: e you
console.log(str12.slice(-5,-1))  // output: e yo

console.log(str12.slice(6,1))  //! not possible (endIndex should be greater than startIndex)
console.log('--------------------------------------------------------------')

// ! 17. substring()

console.log(str12.substring(2,8)) // output: llo ho
console.log(str12.substring(2))   // output: llo how are you
console.log(str12.substring(-5))  // output: hello how are you
console.log(str12.substring(6,1))
console.log('-------------------------------------')


// ! 18.  padStart()


let str13 = 'hi'

console.log(str13)

let padding1= str13.padStart(6,"u")
console.log(padding1)                      // output: uuuuhi
console.log(str13.padStart(6,"san"))      // output: sanshi
console.log('-----------------------------------------------------------')
```



```
//!  19. padEnd()

let padding2 = str13.padEnd(5,'hello')
console.log(padding2)                    //output : hihel
console.log('--------------------------------------------------------')

// ! 20. search()

let str14 = 'hello, good afternoon'

console.log(str14.search("good"))       // output: 7
console.log(str14.search("m"))     // output: -1 (because it is not present)
console.log('-----------------------------------------------------')

//! 21. split()

let str15 = 'good afternoon everyone'

let arr = str15.split(" ")
console.log(arr)              // output:  ['good', 'afternoon', 'everyone']
console.log('------------------------------------------------------------')


// !   declare String using String Object



let strObj1 = new String('this is string object')
let strObj2 = new String('this is string object')

let normalStr1 = 'this is normal string'
let normalStr2 = 'this is normal string'

console.log(normalStr1 === normalStr2)    // true
console.log(strObj1 === strObj2)          // false

// ! 22. valueOf()

let normalStr3 = strObj1.valueOf()

console.log(strObj1)
console.log(normalStr3)
```