@CODE.CLASH

JavaScript : String Methods

NEXT →

1

@CODE.CLASH

Hey Everyone 👋

Strings often need to be manipulated to display a desired result. It is a must-have skill for the frontend.

In this post, I will share with you some String methods that help you work with strings.

Do Like, save and Share This Post If You Found This Helpful.

NEXT →

2

@CODE.CLASH

String Length

1. length

- The length property returns the length of a string:

```
let myName = 'Imtiyaz'
const length = myName.length;
console.log(length);
// Expected Output 7
```

NEXT →

3

@CODE.CLASH

Finding SubString

1. indexof()

- If substring find return substring first character index else return -1. You can pass a second parameter as a search starting point.

2. lastIndexOf()

- this method traverse from end to start.

3. search()

- you cannot pass a second parameter.

```
let myBio = "I am a React Developer";

console.log(myBio.indexOf('React'));
// Expected Output 7

console.log(myBio.lastIndexOf('ac',14));
// Expected Output 9

console.log(myBio.search('name'));
// Expected Output -1
```

NEXT →

4

@CODE.CLASH

Extracting SubString

1. slice()

- return the substring as a new string.
  two parameter --> start position, End position.

2. substring()

- similar to slice(). but negative index value is ingnore by JS.

3. substr()

- second perameter specifies the length of extrated string.

```
let frStr = "Apple, banana, orange, mango";

console.log(frStr.slice(7,13));
// Expected Output :-  banana

console.log(frStr.substring(15,-2));
// Expected Output :-  Apple, banana

console.log(frStr.substr(7,13));
// Expected Output :-  banana, orang
```

NEXT →

5

@CODE.CLASH

Extracting Character

1. charAt()

- return the charachter at a specified index in string, if it's not available then return nothoing.

2. charCodeAt()

- return the Unicode value of character. if it's not available then return NAN.

3. popertyAccess[]

- ES5 allow property access[] on string it's return a charachter.

```
let Str = "Hello i am a Student";

console.log(Str.charAt(13));
// Expected outPut :- S

console.log(Str.charCodeAt(58));
// Expected outPut :- NaN

console.log(Str[8]);
// Expected outPut :- a
```

NEXT →

6

@CODE.CLASH

Other Useful Methods

1. toUpperCase()

- Converts all string characters to uppercase.

2. toLowerCase()

- Converts all string characters to lowercase.

3. concat()

- Used for concatenation of two strings.

```
let myStr = "Hello i am a Student";

console.log(myStr.toUpperCase());
// Expected outPut :- HELLO I AM A STUDENT

console.log(myStr.toLowerCase());
// Expected outPut :- hello i am a student

console.log(myStr.concat(", From India"));
// Expected outPut :- Hello i am a Student, From India
```

NEXT →

4. replace()

- It only replaces the first match. and return new replaced string.

5. trim()

- Used to remove whitespace from either side of a string.

6. split()

- Used to convert a string to an array.

```
let myBio = 'I am a React Developer';

let repData = myBio.replace('React', 'Javascript');
console.log(repData);
// Expected Output : I am a Javascript Developer

let Str = '        Hii    How Are You?          ';
console.log(Str.trim());
// Expected Output : Hii    How Are You?

let text = 'a,b,c,d,e,f';
console.log(text.split(','));
// Expected Output : ['a', 'b', 'c', 'd', 'e', 'f']
```

NEXT →

Best Of Luck :)

Next →

@CODE.CLASH

THANKS FOR YOUR ATTENTION

CODE.CLASH

IMTIYAZ NANDASANIYA

LIKE AND SAVE IT FOR LATER