Internet Programming I

Chapter 4 JavaScript: String and the use of new looping statement

Group: 9
3rd Year Sec B Students,
Dept. of Software Eng.

Jan 2022, AASTU

Objectives

After success completion of the session you will be able to:

- Understand JS strings
- Apply operations, methods
- Work with string using various looping techniques

Outline

String

- Creating String
- Escape sequences
- String Operations
- String methods

- Looping Techniques

1. String

- String is a type used to holding data that can be represented in text form.
- Values of string is *not object* they are ***primitives***.
- Strings are **immutable** (i.e. all operations doesn't modify the original string but return new string) like all primitives(int, Boolean, number)
- But they have built-in properties
- String is primitive ***wrapper object*** for string primitive values.

Creating String

- Strings can be created as
  1. primitives, from string literals, or
  2. as objects, using the String() constructor:

```
// 1. using string literal
const str1 = "A string primitive";
const str2 = 'Also a string primitive';
const str3 = `Yet another string primitive`;
// 2. using String() constructor
const str4 = new String("A String object");
```

Escape Sequences

- Some character can't be represented directly in the string. We need to escape them using escape character \

Escape sequence	Unicode code point
\0	null character
\'	single quote
\"	double quote
\\	Backslash
\n	Newline
\r	carriage return
\v	vertical tab

String Operations

- Some of the most-used operations on strings are
  - to check length, e.g 'hello'.length
  - to build and **concatenate** them using the + and += string operators
  - checking for the existence or location of substrings with the indexOf() method
  - extracting substrings with the substring() method.
  - Comparing strings by using **comparison operators**(e.g. >, <, >=, <=, == …etc)
  - Accessing character using charAt() method or **index**(introduced ES6)

String Methods

Method	Description
charAt, charCodeAt, codePointAt	Return the character or character code at the specified position in string.
indexOf, lastIndexOf	Return the position of specified substring in the string or last position of specified substring, respectively.
startsWith, endsWith, includes	Returns whether or not the string starts, ends or contains a specified string.
concat	Combines the text of two strings and returns a new string.
fromCharCode, fromCodePoint	Constructs a string from the specified sequence of Unicode values. This is a method of the String class, not a String instance.
split	Splits a String object into an array of strings by separating the string into substrings.
slice	Extracts a section of a string and returns a new string.

String Methods

Method	Description
substring, substr	Return the specified subset of the string, either by specifying the start and end indexes or the start index and a length.
match, matchAll, replace, replaceAll, search	Work with regular expressions.
toLowerCase, toUpperCase	Return the string in all lowercase or all uppercase, respectively.
normalize	Returns the Unicode Normalization Form of the calling string value.
repeat	Returns a string consisting of the elements of the object repeated the given times.
trim	Trims whitespace from the beginning and end of the string.
substring, substr	Return the specified subset of the string, either by specifying the start and end indexes or the start index and a length.

String Methods Example

```
const str = "Internet programming I and Networking"
console.log(str.length)
console.log(str.charAt(1))
console.log(str[1])
console.log(str + ": string")
console.log(str.concat(": string"))
console.log(str.indexOf('t'))
console.log(str.match(/net/gi))
console.log(str.search(/net/gi))
console.log(str.replace(' ', '_'))
console.log(str.replaceAll(' ', '_'))
console.log(str.slice(-10, str.length))
console.log(str.substring(-1, 8))
console.log(str.toUpperCase())
console.log(str.toLowerCase())
```

OUTPUT

```
n
n
Internet programming I and Networking: string
Internet programming I and Networking: string
2
[ 'net', 'Net' ]
5
Internet_programming I and Networking
Internet_programming_I_and_Networking
Networking
net
INTERNET PROGRAMMING I AND NETWORKING
internet programming i and networking
```

2. Looping Techniques

- Loops offer a quick and easy way to do something ***repeatedly***
- There are different loops provided by JavaScript. These are:
  a) for statement
  b) do..while statement
  c) while statement
  d) for..in statement
  e) for..of statement
  f) using Array.map() or array.forEach()

Their syntax and usage are the ***same*** in JavaScript like, C, C++ and Java

2. Looping Techniques

A) *for ... in :* iterates a specified variable over all the ***enumerable*** properties of an object.

```
for (variable in object)
    statement;
```

B) *for ... of* statement creates a loop Iterating over ***iterable objects*** (e.g. Array, Map, Set, String ..etc)

```
for (variable of object)
    statement;
```

Looping Technique Example

```
const str = "Abc";
for (let i = 0; i < str.length; i++) {
    console.log(str[i]) // or str.charAt(i)
}
for (const c of str) {
    console.log(c)
}
// not recommended for array and string
for (const i in str) {
    console.log(str[i])
}
[...str].forEach(i => {
    console.log(i)
});
```

OUTPUT

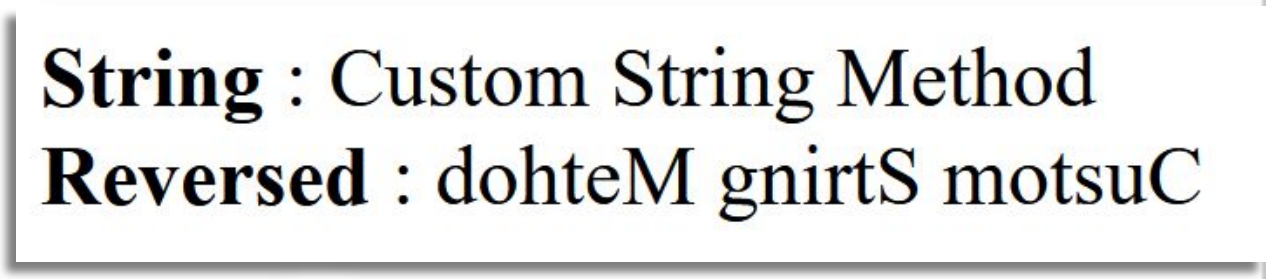
```
A
b
c
A
b
c
A
b
c
A
b
c
```

Extending The functionality of String

- We can add new methods and functionalities to builitin objects due to dynamic nature of JavaScript. E.g. let us add new string method

```
<!-- in the body of HTML document -->
<div id="output"></div>
<script>
    String.prototype.reverse = function () {
        return this.split('').reverse().join('')
    }
    const str = "Custom String Method"
    const output = document.getElementById("output")
    output.innerHTML = `
            <b>String</b> : ${str}<br>
            <b>Reversed</b> : ${str.reverse()}
        `
</script>
```

OUTPUT



String Methods Demonstration (part I)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Demo: String Exercise</title>
</head>
<body>
    <div id="output"></div>
    <script>
        // custom method for string
        String.prototype.reverse = function () {
            return this.split('').reverse().join('')
        }
        const str = "Internet programming I and Networking"
        const output = document.getElementById("output")
```

String Methods Demonstration(Part II)

```
output.innerHTML = `
        String : ${str }<br> Length : ${str.length } <br>
        Char at 3: ${str.charAt(3)} <br>  Str[3]: ${str[3]} <br>
        Concat 'test': ${str.concat('test', '!')}<br>
        Concat 'test' using +: ${str + 'test' + '!'}<br>
        Index of 't': ${str.indexOf('t')} <br>
        Match 'net': ${str.match(/net/gi)} <br>
        Search 'net': ${str.search(/net/i)} <br>
        Replace space with _ : ${str.replace(' ', '_')} <br>
        Replace All space with _ : ${str.replaceAll(' ', '_')} <br>
        Slice last 10 chars: ${str.slice(-10, str.length)} <br>
        Substring between 5 and 8: ${str.substring(5, 8)} <br>
        Uppercase: ${str.toUpperCase()} <br>
        Lowercase: ${str.toLowerCase()} <br>
        <mark>Reversed: ${str.reverse()}</mark>
`

</script> </body> </html>
```

String Methods Demonstration Output

```
String : Internet programming I and Networking
Length : 37
Char at 3: e
Str[3]: e
Concat 'test': Internet programming I and Networkingtest!
Concat 'test' using +: Internet programming I and Networkingtest!
Index of 't': 2
Match 'net': net,Net
Search 'net': 5
Replace space with _ : Internet_programming I and Networking
Replace All space with _ : Internet_programming_I_and_Networking
Slice last 10 chars: Networking
Substring between 5 and 8: net
Uppercase: INTERNET PROGRAMMING I AND NETWORKING
Lowercase: internet programming i and networking
Reversed: gnikrowteN dna I gnimmargorp tenretnI
```

Demonstration Cont..

Check out Github for Palindrome Checker & Exercise

(click the following icon or here to get code)

Reading Resources/Materials

Links

- ✔ mdn documentations
- ✔ tutorialsTeacher

Thank You
For Your Attention!!