Lesson 04 Demo 02

Using Various String Built-in Methods

Objective: To demonstrate the usage of various built-in string methods in JavaScript for string manipulation and analysis

Tools Required: Visual Studio Code and Node.js

Prerequisites: None

Steps to be followed:

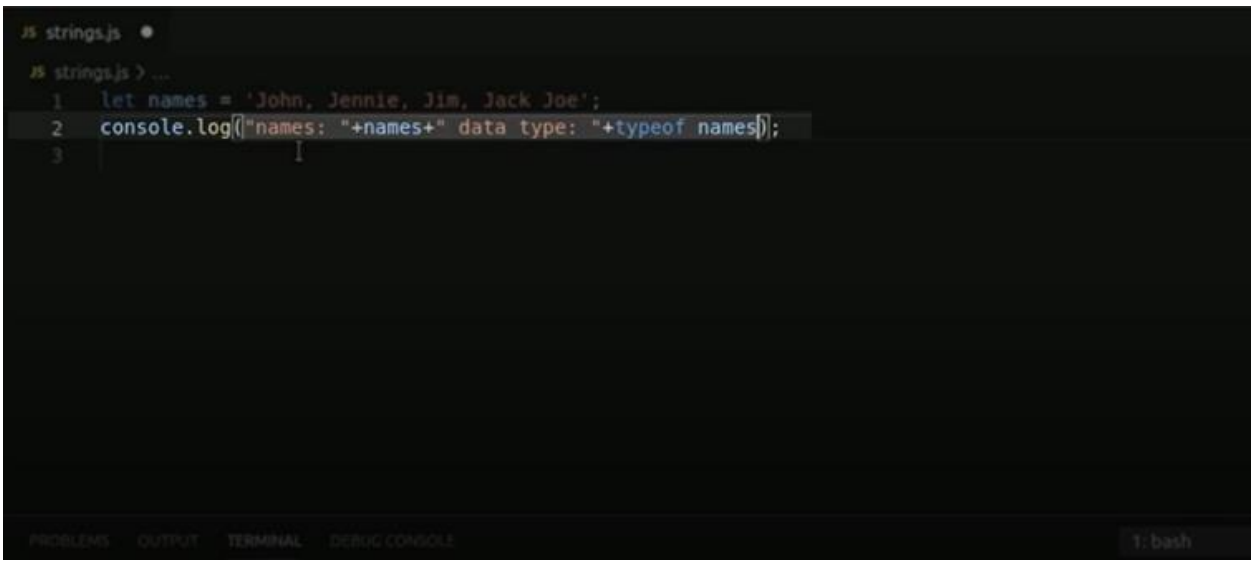
1. Explore built-in string methods

Step 1: Explore built-in string methods

1.1 Open Visual Studio Code and create a new file named **strings.js**

1.2 Declare a string variable called **names** using single or double quotes. Assign multiple names to the **names** variable, separated by commas

1.3 Use **console.log()** to print the **names** variable. Use the **typeof** operator to verify the data type of the **names** variable



1.4 Run the program

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                    1: bash
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node strings.js
names: John, Jennie, Jim, Jack Joe
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node strings.js
names: John, Jennie, Jim, Jack Joe data type: string
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$
```

You can see the names printed on the console with the data type.

1.5 Utilize the length property on the **names** variable to retrieve the length of the string

```
strings.js
strings.js > ...
1   let names = 'John, Jennie, Jim, Jack Joe';
2   console.log("names: "+names+" data type: "+typeof names);
3   console.log(names.length);
4
5
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                    1: bash
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node strings.js
names: John, Jennie, Jim, Jack Joe
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node strings.js
names: John, Jennie, Jim, Jack Joe data type: string
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node strings.js
names: John, Jennie, Jim, Jack Joe data type: string
27
```

1.6 Apply the **toUpperCase()** method to the **names** variable to convert it to uppercase. Print the converted string

1.7 Utilize the **replace()** method on the **names** variable to replace a specific character or substring. Print the manipulated string

1.8 Rerun the program and observe the output

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                    1: bash
27
Names in Uppercase: John, Jennie, Jim, Jack Joe
Result is: JOHN, JENNIE, JIM, JACK JOE
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node strings.js
names: John, Jennie, Jim, Jack, Joe data type: string
28
Names in Uppercase: John, Jennie, Jim, Jack, Joe
Result is: JOHN, JENNIE, JIM, JACK, JOE
Result is: Kohn, Jennie, Jim, Jack, Joe
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$
```

1.9 Create an array of contact names

```
JS strings.js ●
JS strings.js > [@] contactNames
3   console.log(names.length);
4
5   let result = names.toUpperCase()
6   console.log("Names in Uppercase: "+names);
7   console.log("Result is: "+result);
8
9   result = names.replace('J', 'K');
10  console.log("Result is: "+result);
11
12  let contactNames = [
13      "John",
14      "George",
15      "Kia",
16      "Ana",
17      "Shawn"
18  ];
```

1.10 Iterate over the array using a **for loop** with a specific condition and print the filtered names

```
JS strings.js ×
JS strings.js > ...
17      "Shawn",
18      "Sia"
19  ];
20
21  let searchKeyword = 'ia';
22
23  for(let idx=0;idx<contactNames.length;idx++){
24      if(contactNames[idx].endsWith(searchKeyword)){
25          console.log(contactNames[idx]);
26
27      }
28  }
```

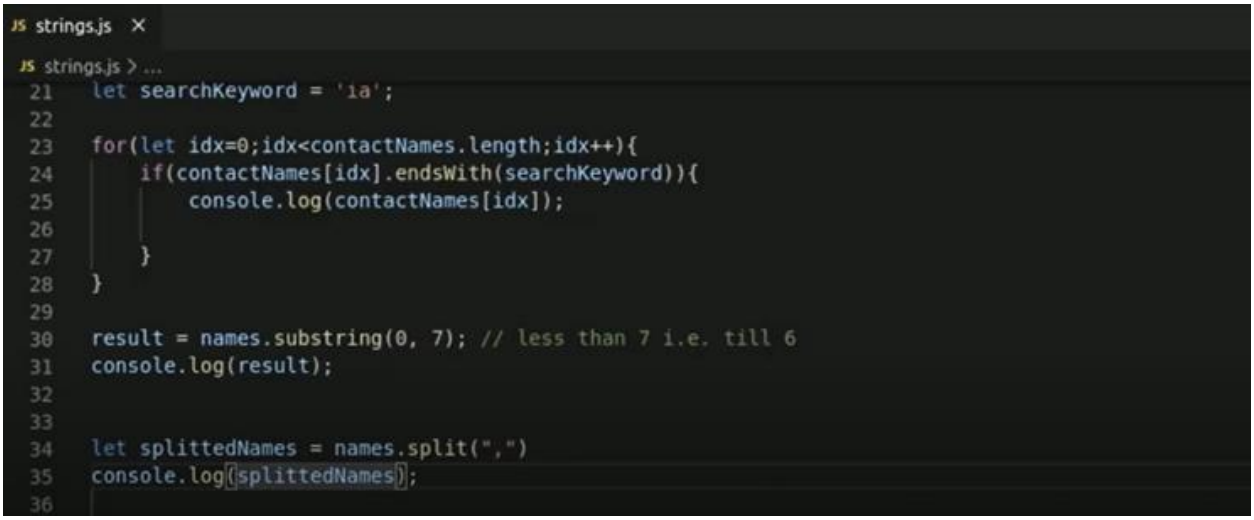
```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                    1: bash
Result is: Kohn, Jennie, Jim, Jack, Joe
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node strings.js
names: John, Jennie, Jim, Jack, Joe data type: string
28
Names in Uppercase: John, Jennie, Jim, Jack, Joe
Result is: JOHN, JENNIE, JIM, JACK, JOE
Result is: Kohn, Jennie, Jim, Jack, Joe
Kia
Sia
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$
```

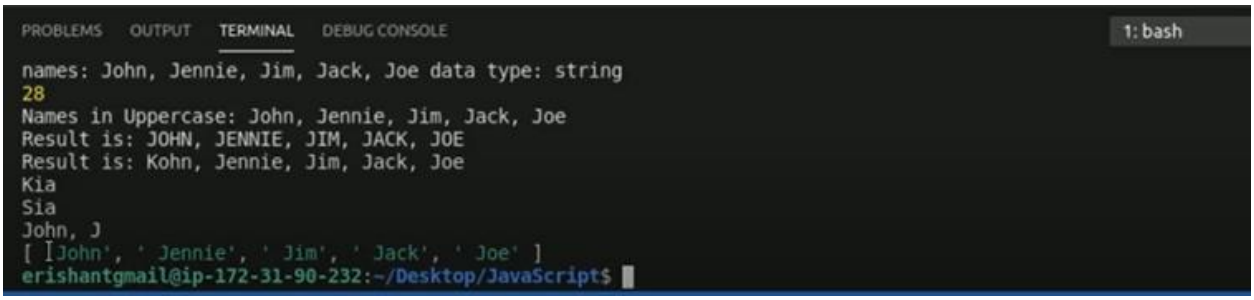
1.11 Utilize the **substring()** method on the **names** variable to extract a portion of the string. Print the extracted substring

```
JS strings.js
JS strings.js > ...
17      "Shawn",
18      "Sia"
19  ];
20
21  let searchKeyword = 'ia';
22
23  for(let idx=0;idx<contactNames.length;idx++){
24      if(contactNames[idx].endsWith(searchKeyword)){
25          console.log(contactNames[idx]);
26
27      }
28  }
29
30  result = names.substring(0, 7);
31  console.log(result);
32
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                    1: bash
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$ node strings.js
names: John, Jennie, Jim, Jack, Joe data type: string
28
Names in Uppercase: John, Jennie, Jim, Jack, Joe
Result is: JOHN, JENNIE, JIM, JACK, JOE
Result is: Kohn, Jennie, Jim, Jack, Joe
Kia
Sia
John, J
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$
```

1.12 Utilize the **split()** method on the **names** variable to split the string based on a specified delimiter. Print the resulting array





By following these steps, you have successfully demonstrated the usage of various built-in string methods in JavaScript for effective string manipulation and analysis, enhancing your ability to manage and process text within your applications.