Experiment No._____ *Date___/___/2020*

TITLE OF EXPERIMENT: - A Program to Reverse string, Replace characters of a string, String is Palindrome operations on a given string.

DIVISION:_____ BRANCH: _____

BATCH:_____ ROLL NO.: _____

PERFORMED ON DATE: _____

SIGNATURE OF TEACHING STAFF:

EXPERIMENT NO. 3

Aim: Write a JavaScript program to following operations on a given string,

- Reverse string
- Replace characters of a string.
- String is Palindrome.

Prerequisites:

- For this **Javascript Lab**, it is assumed that you have a prior knowledge of HTML coding. It would help if you had some prior exposure to object-oriented programming concepts and a general idea on creating online applications.
- To understand this experiment, you should have the knowledge of the basic JavaScript,

✓ JavaScript String,
✓ JavaScript Function
✓ Function Expressions
✓ JavaScript String replace()

Editor:

1.	NotePad
2.	Visual studio code

Theory:

Strings:

Strings are used for storing text. A String variable contains a collection of characters surrounded by double quotes:

Example

Create a variable of type String and assign it a value:

```
String greeting = "Hello";
```

String Length

A String in Java is actually an object, which contain methods that can perform certain operations on strings. For example, the length of a string can be found with the length() method:

A. Reverse string :

Example 1: Reverse a String Using for Loop

```
// program to reverse a string

function reverseString(str) {

   // empty string
   let newString = "";
   for (let i = str.length - 1; i >= 0; i--)
{
      newString += str[i];
   }
   return newString;
}

// take input from the user
const string = prompt('Enter a string: ');

const result = reverseString(string);
console.log(result);
```

Output

```
Enter a string: hello world
dlrow olleh
```

In the above program, the user is prompted to enter a string. That string is passed to the reverseString() function.

Inside the reverseString() function,

- An empty newString variable is created.
- The for loop is used to iterate over the strings. During the first iteration, str.length - 1 gives the position of the last element. That element is added to the newString variable.

  This process continues for all the string elements.
- The value of i decreases in each iteration and continues until it becomes **0**.

Example 2: Reverse a String Using built-in Methods

For this, we will use three methods: the String.prototype.split() method, the Array.prototype.reverse() method and the Array.prototype.join() method.

- The split() method splits a String object into an array of string by separating the string into sub strings.
- The reverse() method reverses an array in place. The first array element becomes the last and the last becomes the first.
- The join() method joins all elements of an array into a string.

Steps:

Step 1. Use the split() method to return a new array
Step 2. Use the reverse() method to reverse the new created array
Step 3. Use the join() method to join all elements of the array into a string
Step 4. Return the reversed string

```
// program to reverse a string

function reverseString(str) {

    // Step 1. Use the split() method to return a new array of strings
    const arrayStrings = str.split(""); // var splitString = "hello".split("");
    // ["h", "e", "l", "l", "o"]


    // Step 2. Use the reverse() method to reverse the new created array elements
    const reverseArray = arrayStrings.reverse(); // var reverseArray = ["h", "e", "l", "l",
"o"].reverse();
    // ["o", "l", "l", "e", "h"]

    // Step 3. Use the join() method to join all elements of the array into a string
    const joinArray = reverseArray.join("");  // var joinArray = ["o", "l", "l", "e", "h"].join("");
    // "olleh"


    // Step 4.return the reversed string
    return joinArray;  // "olleh"
```

```
}

// take input from the user
const string = prompt('Enter a string: ');

const result = reverseString(string);
console.log(result);
```

Output

```
Enter a string: hello
olleh
```

In the above program, the built-in methods are used to reverse a string.

- First, the string is split into individual array elements using the split() method. str.split("") gives ["h", "e", "l", "l", "o"].
- The string elements are reversed using the reverse() method. arrayStrings.reverse() gives ["o", "l", "l", "e", "h"].
- The reversed string elements are joined into a single string using the join() method. reverseArray.join("") gives olleh.

B. Replace characters of a string.

Example: Replace First Occurrence of a Character in a String

```
// program to replace a character of a string

const string = 'Mr Red has a red house and a red car';

// replace the characters
const newText = string.replace('red', 'blue');

// display the result
console.log(newText);
```

Output

```
Mr Red has a blue house and a red car
```

In the above program, the replace() method is used to replace the specified string with another string.

When a string is passed in the replace() method, it replaces only the first instance of the string. So if there is a second match in the string, it won't be replaced.

You could also pass a **regular expression (regex)** inside the replace() method to replace the string.

Example 2: Replace Character of a String Using RegEx

```
// program to replace a character of a string

const string = 'Mr Red has a red house and a red car';

// regex expression
const regex = /red/g;

// replace the characters
const newText = string.replace(regex, 'blue');

// display the result
console.log(newText);
```

Output

```
Mr Red has a blue house and a blue car
```

In the above program, a **regex** expression is used as the first parameter inside the replace() method.

/g refers to **global**. It means that all the matching characters in the string are replaced.

Since JavaScript is case-sensitive, **R** and **r** are treated as different values.

You could also use the regex to perform case-insensitive replacement using **/gi,** where i represents case-insensitive.

C. String is Palindrome

A string is a palindrome if it is read the same from forward or backward.

For example, **dad** reads the same either from forward or backward. So the word **dad** is a palindrome. Similarly, **madam** is also a palindrome.

Example 1: Check Palindrome Using for Loop

```
// program to check if the string is palindrome or not

function checkPalindrome(string) {

    // find the length of a string
    const len = string.length;

    // loop through half of the string
    for (let i = 0; i < len / 2; i++) {

        // check if first and last string are same
        if (string[i] !== string[len - 1 - i]) {
            return 'It is not a palindrome';
        }
    }
    return 'It is a palindrome';
}

// take input
const string = prompt('Enter a string: ');

// call the function
const value = checkPalindrome(string);

console.log(value);
```

Output

```
Enter a string: madam
It is a palindrome
```

In the above program, the **checkPalindrome()** function takes input from the user.

- The length of the string is calculated using the length property.
- The for loop is used to iterate up to the half of the string. The if condition is used to check if the first and the corresponding last characters are the same. This loop continues till the half of the string.
- During the iteration, if any character of the string, when compared with its corresponding last string is not equal, the string is not considered a palindrome.

Example 2: Check Palindrome using built-in Functions

```
// program to check if the string is palindrome or not

function checkPalindrome(string) {

    // convert string to an array
    const arrayValues = string.split('');

    // reverse the array values
    const reverseArrayValues = arrayValues.reverse();

    // convert array to string
    const reverseString = reverseArrayValues.join('');

    if(string == reverseString) {
        console.log('It is a palindrome');
    }
    else {
        console.log('It is not a palindrome');
    }
}

//take input
const string = prompt('Enter a string: ');

checkPalindrome(string);
```

Output

```
Enter a string: hello
It is not a palindrome
```

In the above program, the palindrome is checked using the built-in methods available in JavaScript.

- The split('') method converts the string into individual array characters.

```
const arrayValues = string.split(''); // ["h", "e", "l", "l", "o"]
```

- The reverse() method reverses the position in an array.

```
// ["o", "l", "l", "e", "h"]
const reverseArrayValues = arrayValues.reverse();
```

- The join('') method joins all the elements of an array into a string.

```
const reverseString = reverseArrayValues.join(''); // "olleh"
```

- Then the if...else statement is used to check if the string and the reversed string are equal. If they are equal, the string is a palindrome.

  Note: The multiple lines of code can be reduced and written in one line:

```
const reverseString = string.split('').reverse().join('');
```

Program:

A. Reverse string :

```
<html>
<head>
<title> program to reverse a string </title>
</head>
<body>
<script type="text/javascript">
function reverseString(str) {

   // return a new array of strings
   const arrayStrings = str.split("");
```

```
    // reverse the new created array elements
    const reverseArray = arrayStrings.reverse();

    // join all elements of the array into a string
    const joinArray = reverseArray.join("");
        // return the reversed string
    return joinArray;
}
 // take input from the user
const string = prompt('Enter a string: ');

const result = reverseString(string);
console.log(result);
 </script>
</body>
</html>
```

B. Replace characters of a string.

```
<html>
<head>
<title> program to reverse a string </title>
</head>
<body>
<script type="text/javascript">
// program to replace a character of a string

const string = 'Mr Red has a red house and a red car';

// regex expression
const regex = /red/g;
```

```
// replace the characters
const newText = string.replace(regex, 'blue');

// display the result
console.log(newText);
</script>
</body>
</html>
```

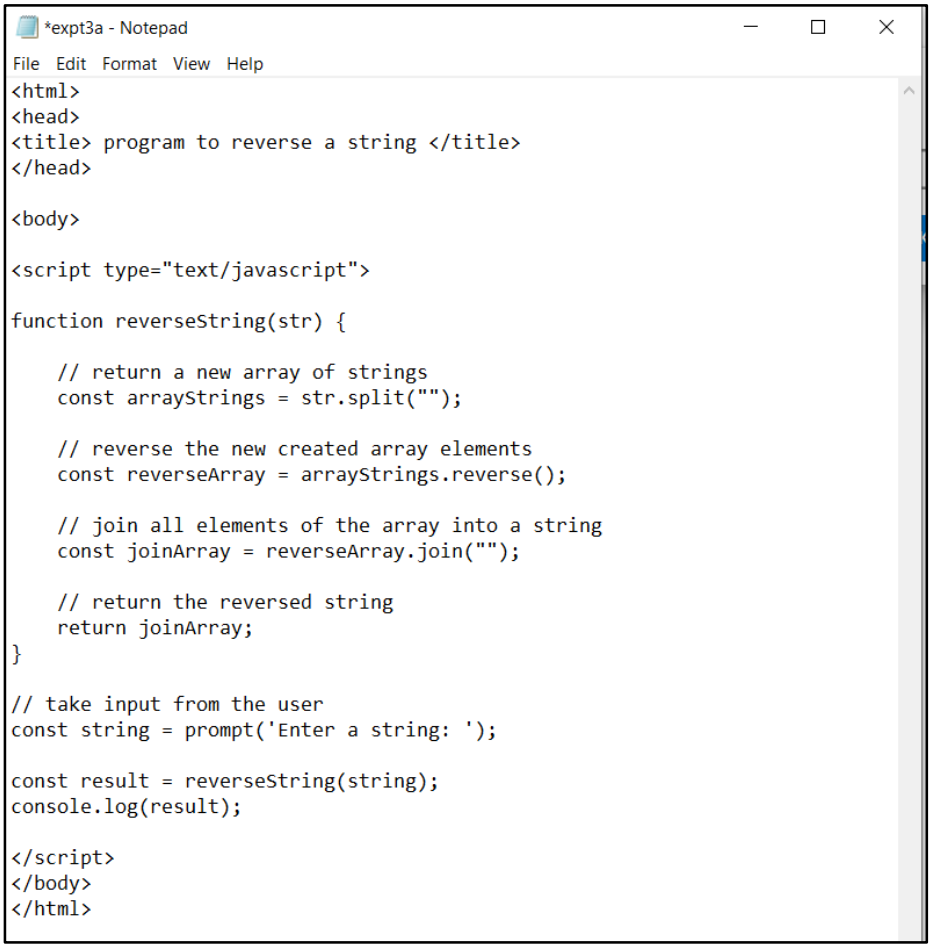
C. String is Palindrome

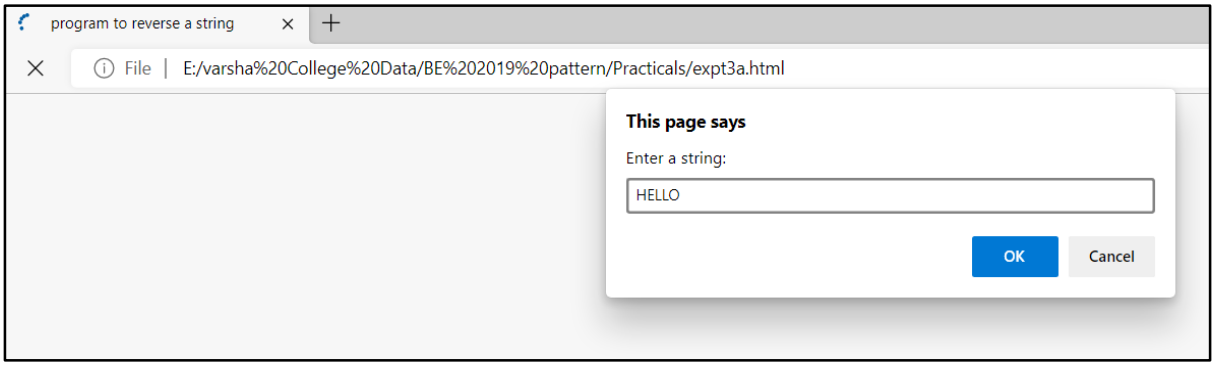
```
</head>
<body>
<script type="text/javascript">
// program to check if the string is palindrome or not
function checkPalindrome(string) {

    // convert string to an array
    const arrayValues = string.split('');

    // reverse the array values
    const reverseArrayValues = arrayValues.reverse();

    // convert array to string
    const reverseString = reverseArrayValues.join('');

    if(string == reverseString) {
        console.log('It is a palindrome');
    }
    else {
        console.log('It is not a palindrome');
```

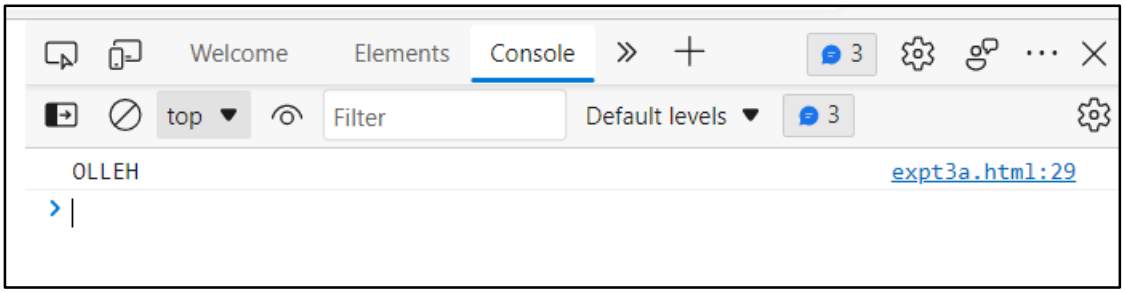
```
   }
}


//take input
const string = prompt('Enter a string: ');
checkPalindrome(string);
</script>
</body>
</html>
```

Screenshot's of Output:

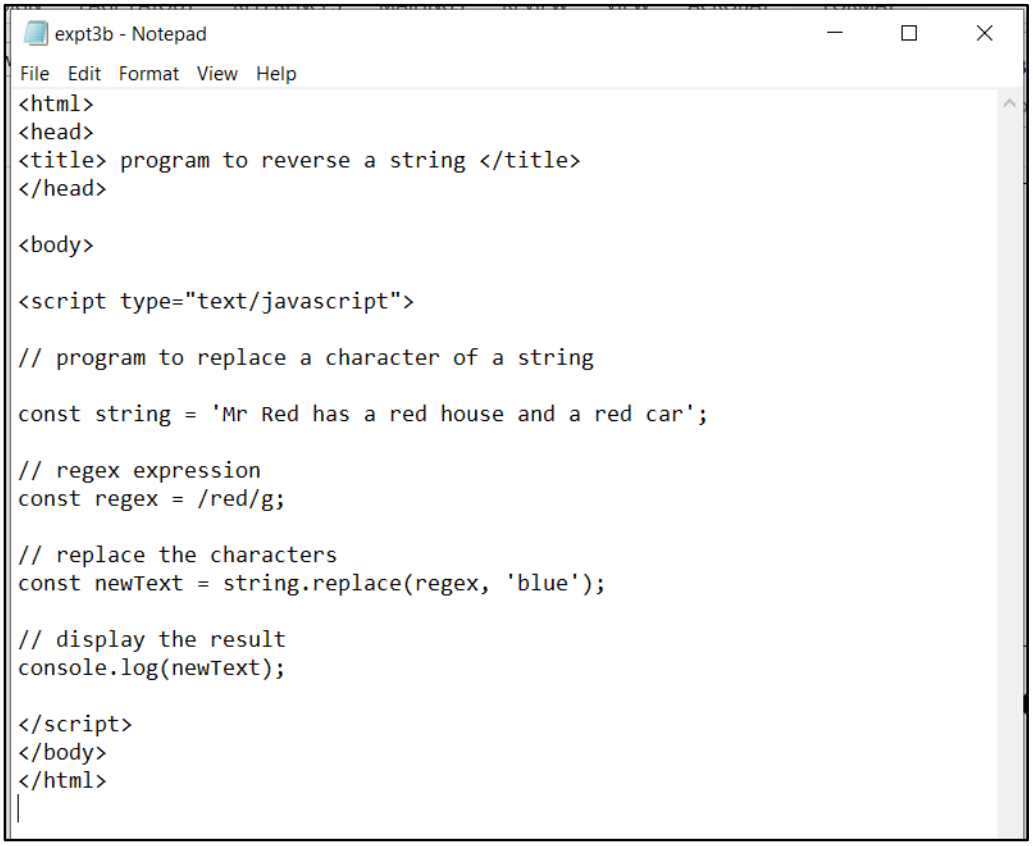
a. Reverse string :

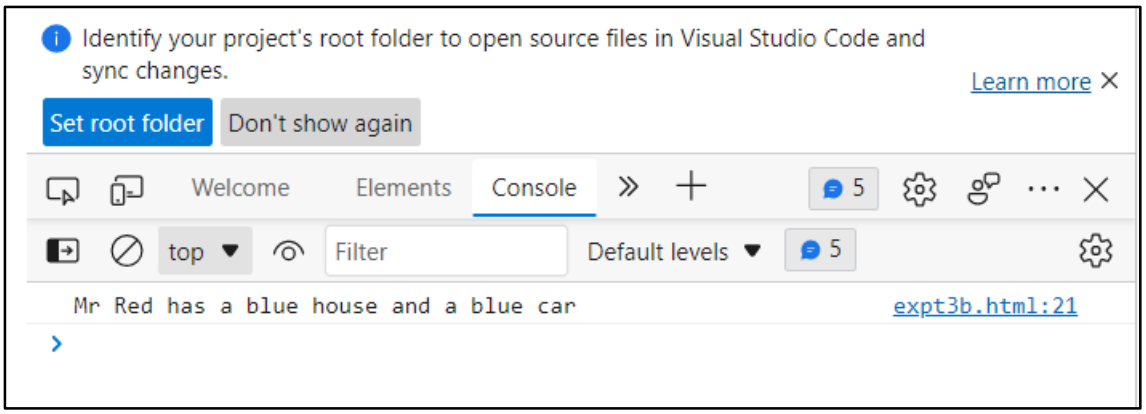






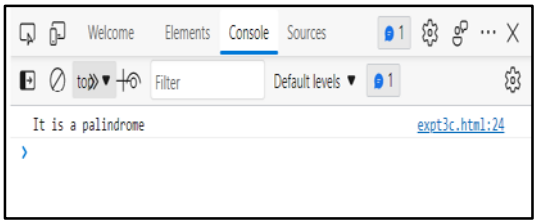
b. Replace characters of a string.





c. String is Palindrome

```
File  Edit  Format  View  Help
</head>

<body>

<script type="text/javascript">

// program to check if the string is palindrome or not

function checkPalindrome(string) {

    // convert string to an array
    const arrayValues = string.split('');

    // reverse the array values
    const reverseArrayValues = arrayValues.reverse();

    // convert array to string
    const reverseString = reverseArrayValues.join('');

    if(string == reverseString) {
        console.log('It is a palindrome');
    }
    else {
        console.log('It is not a palindrome');
    }
}

//take input
const string = prompt('Enter a string: ');

checkPalindrome(string);

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



Conclusion: