**Challenge1**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>ES6 Template Literals</title>

</head>

<body>

    <script src="challenge1.js"></script>

</body>

</html>

**Challenge1.js**

/\* Create a function "templateLiteral" with one parameter "num".

Function is expected to return template literal. Sample function calls are below.

\*/

const templateLiteral= num =>{

     return `Number is ${num}.\nThis number is ${num>10 ? `greater` : `less`} than 10.\nSquare root of this number is ${Math.sqrt(num)}.`;

}

// TEST 1

const myNumber = 9;

console.log(templateLiteral(myNumber));

/\*

Number is 9.

This number is less than 10.

Square root of this number is 3.

\*/

// TEST 2

const myAnotherNumber = 25;

console.log(templateLiteral(myAnotherNumber));

/\*

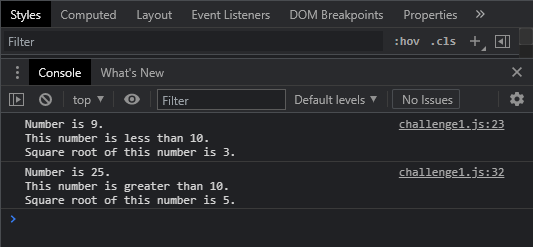
Number is 25.

This number is greater than 10.

Square root of this number is 5.

\*/

**Result**

****

**Challenge2**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>ES6 Template Literals</title>

</head>

<body>

    <script src="challenge2.js"></script>

</body>

</html>

**Challenge2.js**

function taggedTemplate(arrayOfStrings){

    const vals=Array.from(arguments).slice(1);

    // console.log(vals);

    return arrayOfStrings.reduce((concatStr,element,index)=>{

        return concatStr+element+ (typeof vals[index] !== "undefined" ? vals[index]: "");

    },"");

  }

  const a = 10;

  const b = 5;

  const sum = taggedTemplate`Sum of the two variables a(${a}) and b(${b}) is ${a +

    b}`;

  console.log(sum);

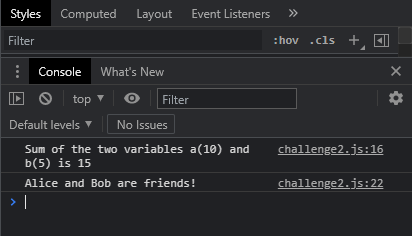
  const girl = "Alice";

  const boy = "Bob";

  const friendsInfo = taggedTemplate`${girl} and ${boy} are friends!`;

  console.log(friendsInfo);

**Result**

****