1. Write a simple script that displays “Hello, World!” on the web page using an alert box.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Task1</title>

</head>

<body>

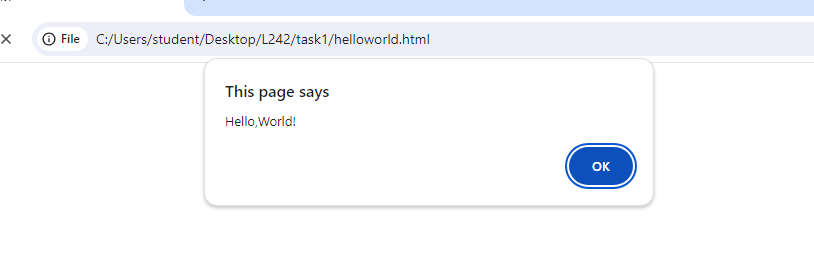
    <script>

        alert("Hello,World!");

    </script>

</body>

</html>



1. Experiment with different data types in JavaScript (e.g., string, number, boolean) by declaring and logging them in the console.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Datatype</title>

</head>

<body>

<script>

    let name="Rahini G";

    console.log(name);

    var num=1234;

    console.log(num);

    var age=20;

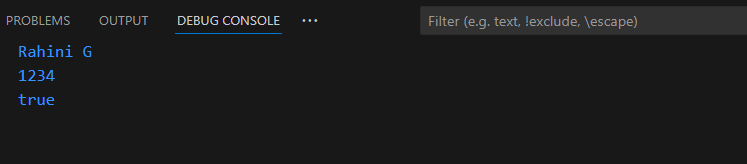
    var bool=Boolean(age>=18);

    console.log(bool);

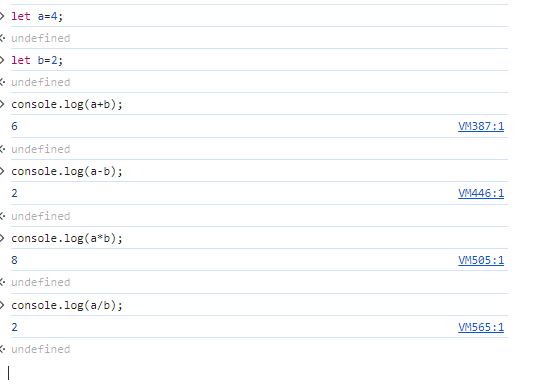
</script>

</body>

</html>



1. Use the console to perform basic math operations like addition, subtraction, multiplication, and division.



1. Declare two strings and concatenate them using the + operator.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <script>

    var s1="Hello";

    var s2="Everyone";

    document.write(s1+s2);

</script>

</body>

</html>



1. Use the typeof operator to check the data type of various variables.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <script>

    let data={

        name:"Rahini",

        dept:"ECE",

        rollno:242,

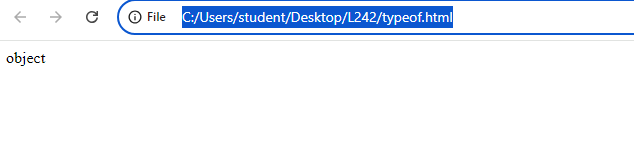
    };

    document.write(typeof(data));

</script>

</body>

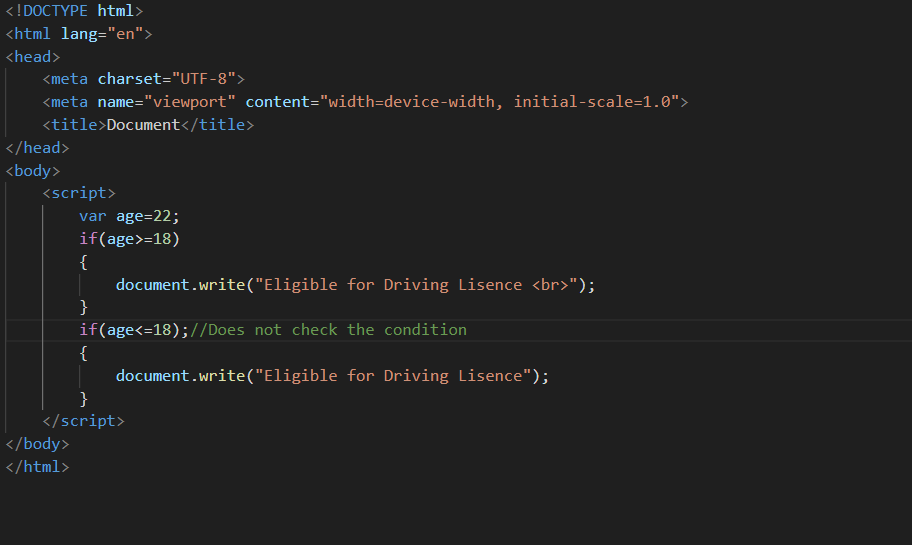
</html>

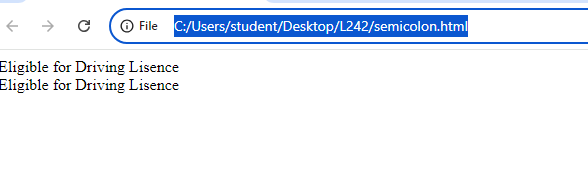


1. Write a multi-line JavaScript comment and a single-line comment. Explain the difference.



1. Create a script with both semicolon-separated and not separated lines. Note any differences in behavior.





1. Use proper indentation to format a nested loop.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <script>

        for(let i=0;i<5;i++)

        {

            for(let j=0;j<=i;j++)

            {

                document.write("\*");

            }

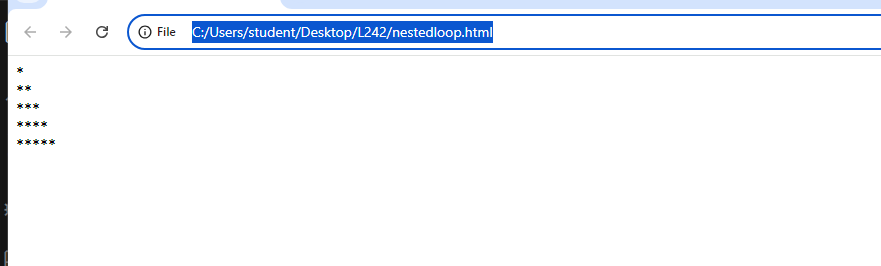
            document.write("<br>");

        }

    </script>

</body>

</html>



1. Declare multiple variables in a single line.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

<script>

    var a="Rahini",b=" ECE",c= 242;

    document.write(a);

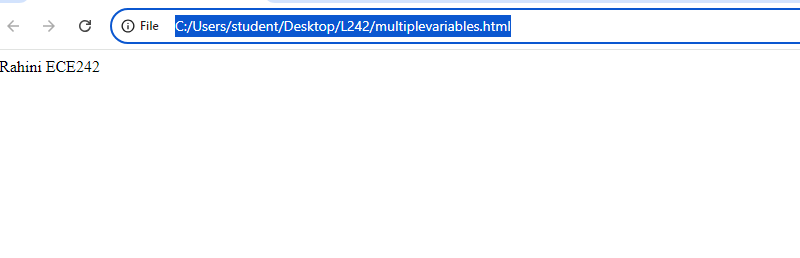
    document.write(b);

    document.write(c);

</script>

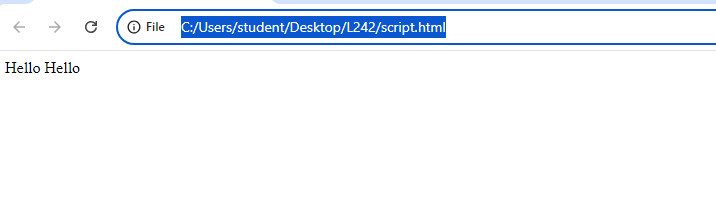
</body>

</html>



1. Place a script tag at the top and bottom of an HTML document. Note any differences in behavior.





1. Write a script without using “use strict” and try to assign a value to an undeclared variable. Note the result.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <script>

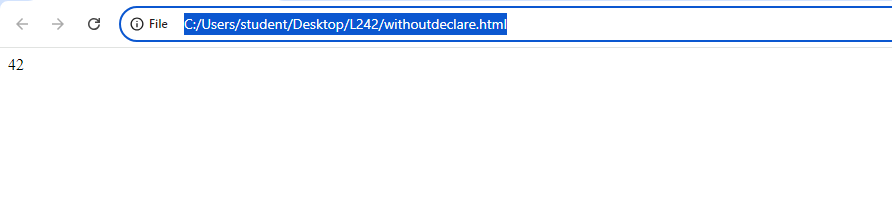
    a=42;

    document.write(a);

</script>

</body>

</html>



1. Enable “use strict” mode and repeat the above action, noting the difference.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <script>

    "use strict"

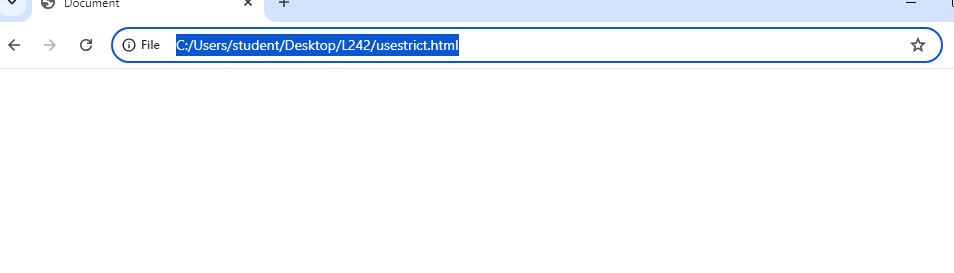
    a=42;

    document.write(a);

</script>

</body>

</html>



1. In “use strict” mode, try to delete a variable, function, or function parameter.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

<script>

    "use strict"

    var emp={name:'Rahini',

        age:22,deptartment:'ECE'};

    document.write("Age="+emp.age);

    document.write("<br>");

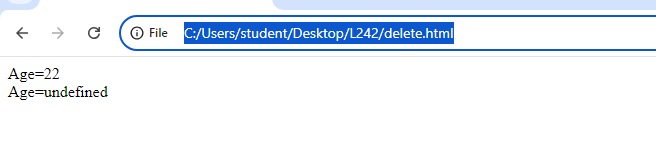
    delete(emp.age);

    document.write("Age="+emp.age);

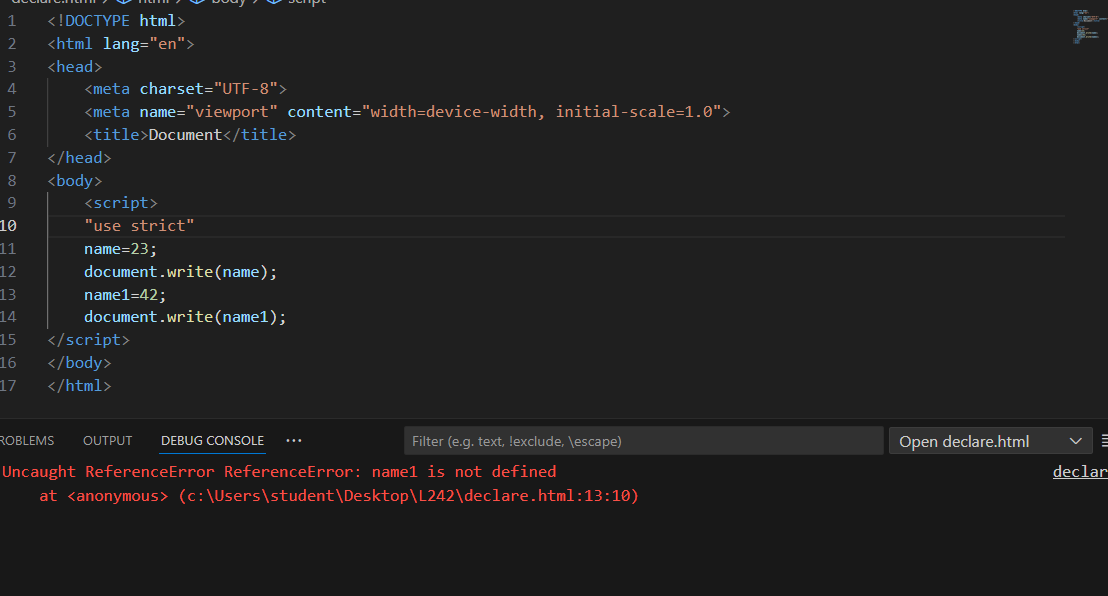
</script>

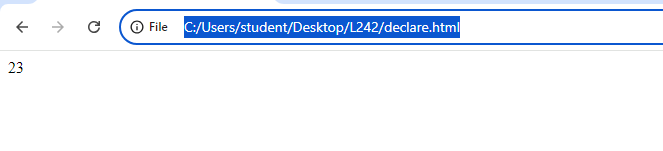
</body>

</html>



1. Assign a value to an undeclared variable without “use strict” and then with “use strict”.





1. Declare a variable with a reserved keyword in “use strict” mode.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

<script>

    "use strict"

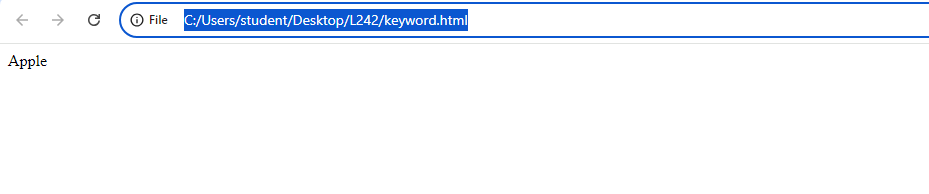
    const a='Apple';

    document.write(a);

</script>

</body>

</html>



1. Declare variables using let, const, and var. Discuss when each should be used.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <script>

    let a=42;//we cannot redeclare in the same variable

    var b=245;//we can redeclare and reassign

    const c=224;//we cannot redeclare or reassign

    document.write(a +"<br>");

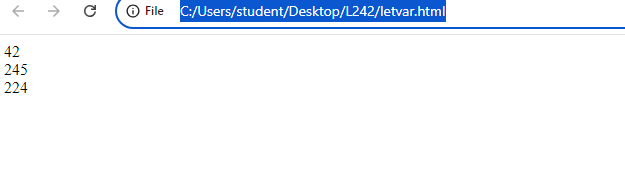
    document.write(b +"<br>");

    document.write(c +"<br>");

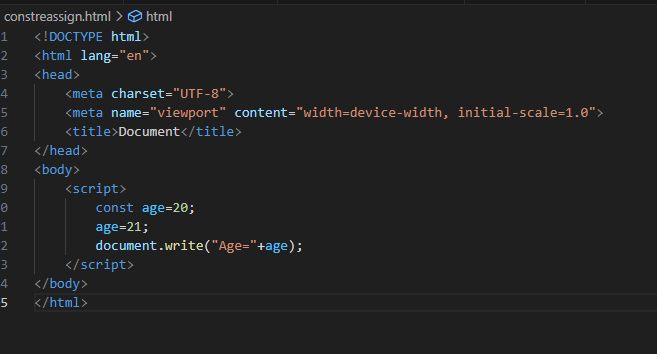
</script>

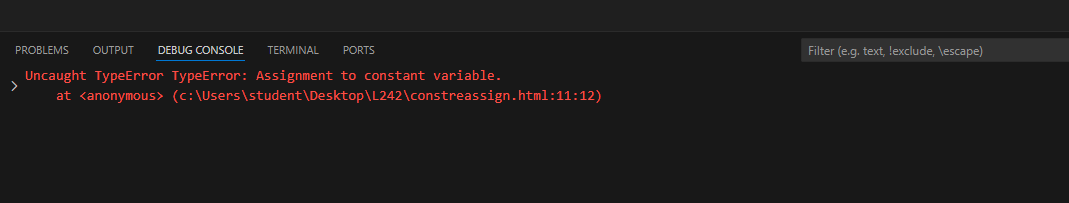
</body>

</html>



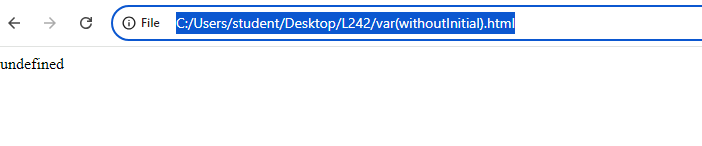
1. Attempt to reassign a const variable and observe the result.





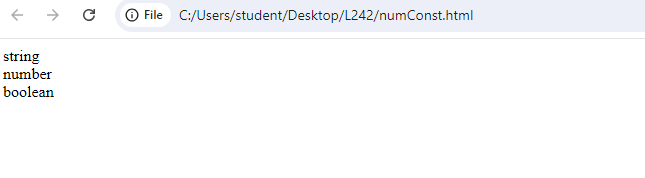
1. Declare a variable without initializing it and print its value.





1. Assign a number, string, and boolean value to a variable and print its type using typeof.





1. Rename a variable and observe the outcome.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <script>

        let age=78;

        document.write(age+"<br>");

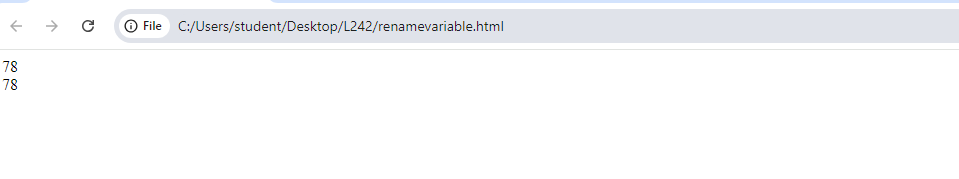
        let a=78;

        document.write(a);

    </script>

</body>

</html>



1. Create variables of different data types (e.g., string, number, boolean, null, undefined, object).

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <script>

        let num="twenty";

        let age=20;

        let a=Boolean(age==20);

        let data={

            name:"Rahini",

            dept:"ECE",

            rollno:242,

        };

        let b=null;

        let c;

        document.write(data.name+"<br>");

        document.write(num+"<br>");

        document.write(age+"<br>");

        document.write(b+"<br>");

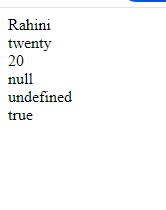
        document.write(c+"<br>");

        document.write(a);

    </script>

</body>

</html>



1. Use the typeof operator to determine the type of various variables.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <script>

        let num="twenty";

        let age=20;

        let a=Boolean(age==20);

        let data={

            name:"Rahini",

            dept:"ECE",

            rollno:242,

        };

        let b=null;

        let c;

        document.write(typeof(data)+"<br>");

        document.write(typeof(num)+"<br>");

        document.write(typeof(age)+"<br>");

        document.write(typeof(b)+"<br>");

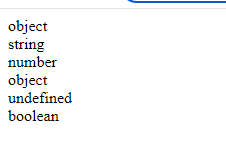
        document.write(typeof(c)+"<br>");

        document.write(typeof(a));

    </script>

</body>

</html>



1. Declare a symbol and print its type.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <script>

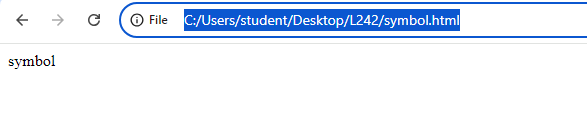
        let a=Symbol("#");

        document.write(typeof(a));

    </script>

</body>

</html>



1. Assign the value null to a variable and check its type using typeof.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <script>

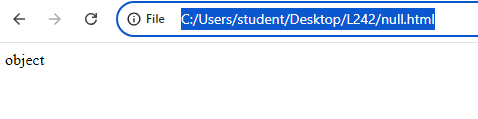
        let b=null;

        document.write(typeof(b));

    </script>

</body>

</html>



1. Differentiate between declaring a variable using var and let in terms of scope.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <script>

        var name="Rahini";//function scope

        let age=20;//window scope

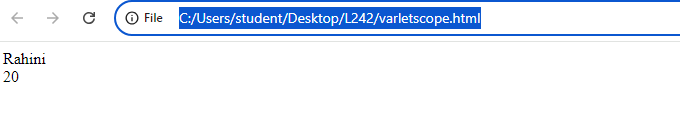
        document.write(name+"<br>");

        document.write(age);

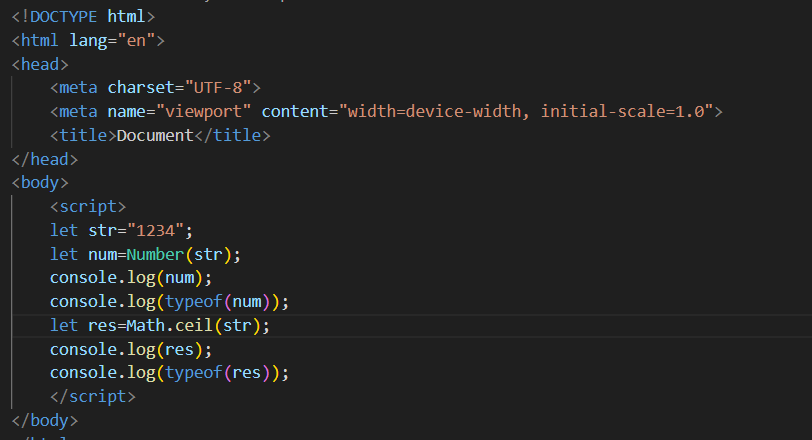
    </script>

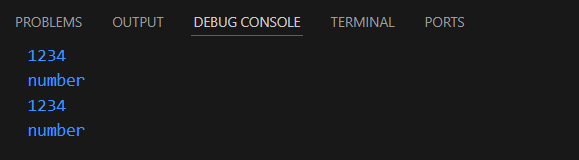
</body>

</html>



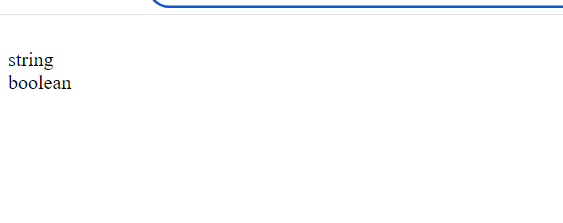
1. Convert a string to a number using both implicit and explicit conversion.





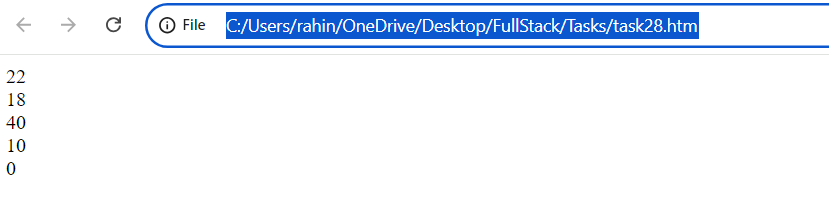
1. Convert a boolean to a string and vice versa.



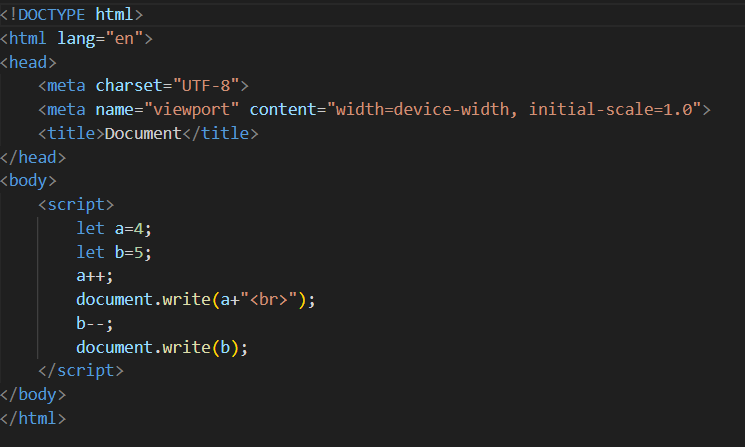


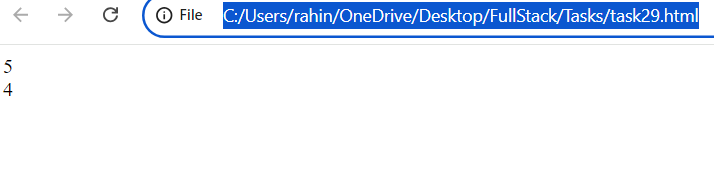
1. Practice basic arithmetic operators (+, -, \*, /, %).





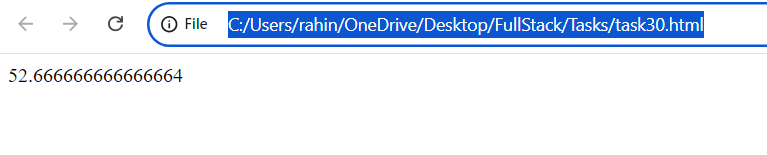
1. Use the ++ and -- operators on a numeric variable.

****

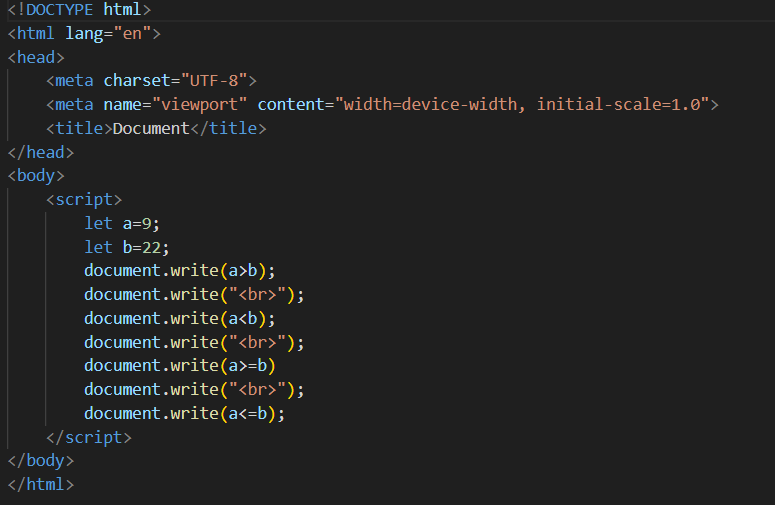
****

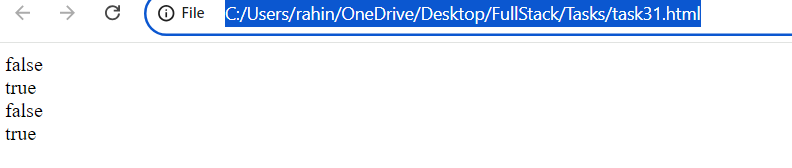
1. Explore the precedence of operators by combining multiple operators in a single expression.

****

****

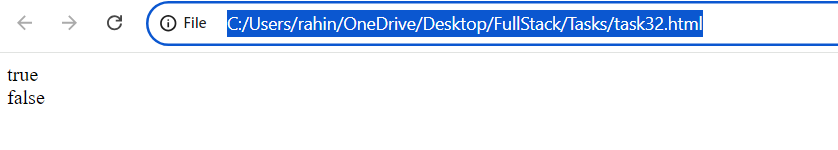
1. Compare two numbers using relational operators (>, <, >=, <=).

****

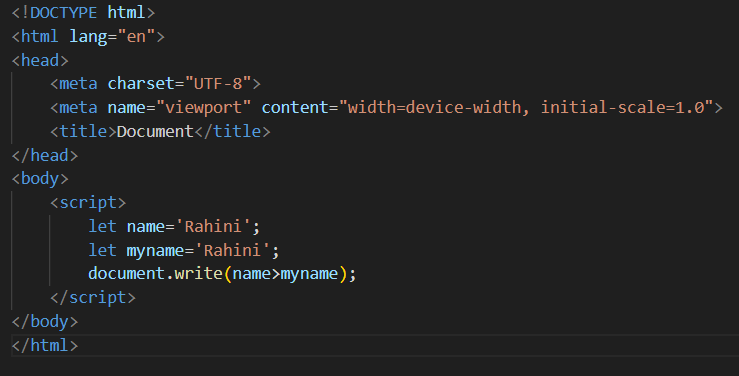
****

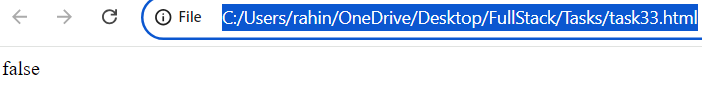
1. Use equality () and strict equality (=) operators to compare different data types and note the differences.

****

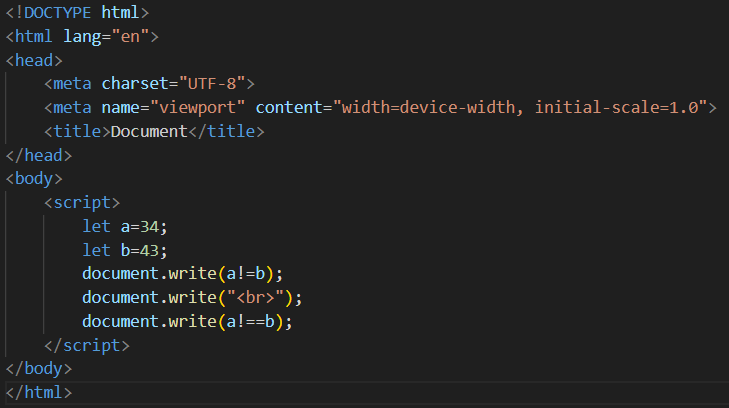
****

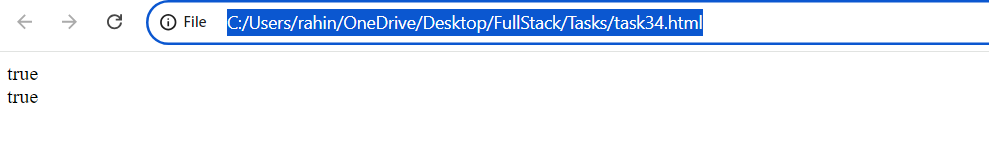
1. Compare two strings lexicographically.

****

****

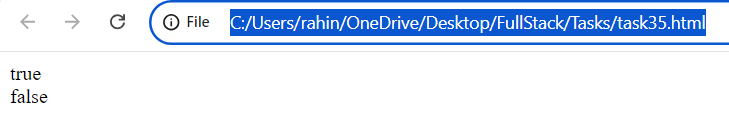
1. Use the inequality (!=) and strict inequality (!==) operators to compare values.

****

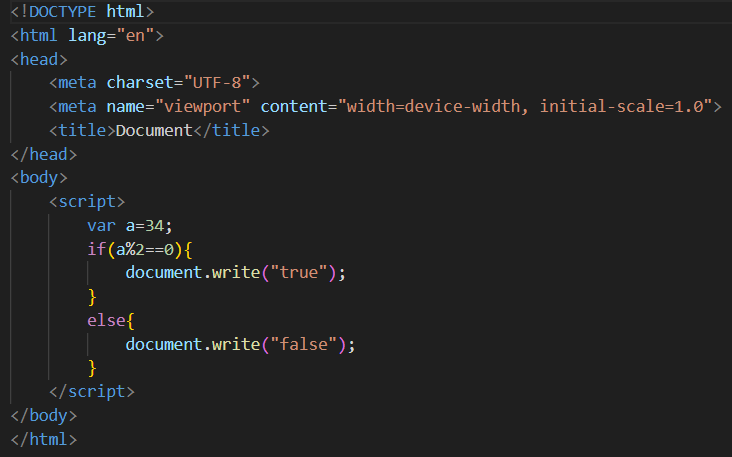
****

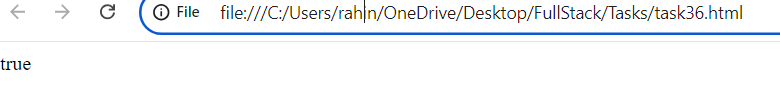
1. Compare null and undefined using both == and ===.

****

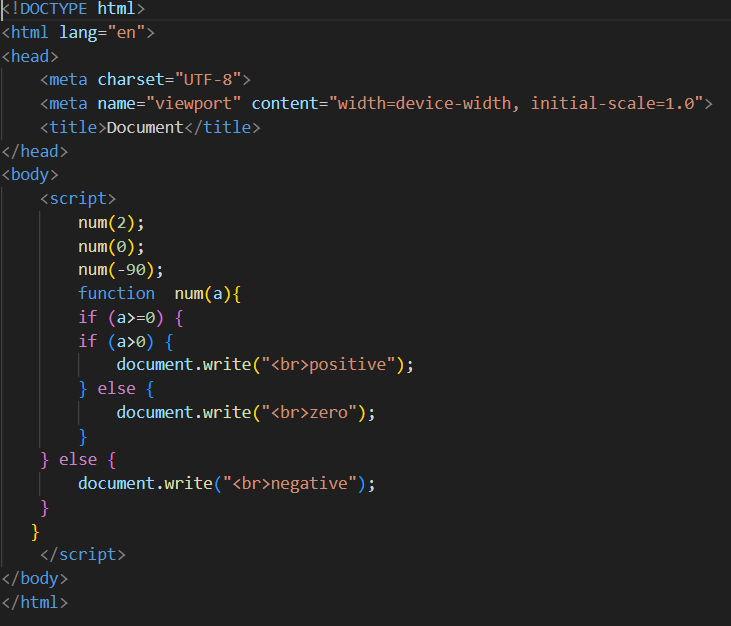
****

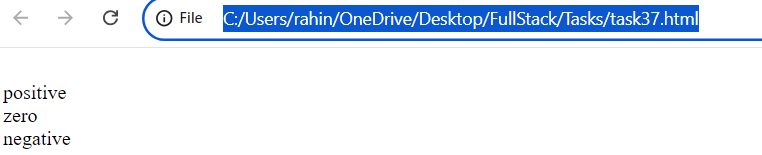
1. Write an if statement that checks if a number is even or odd.



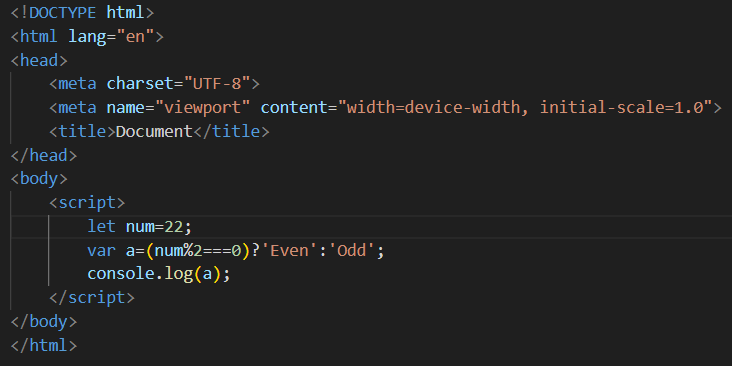
****

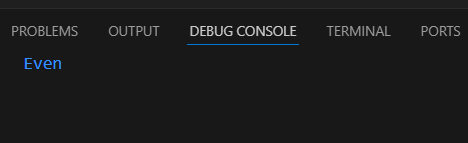
1. Use nested if statements to classify a number as negative, positive, or zero.

****

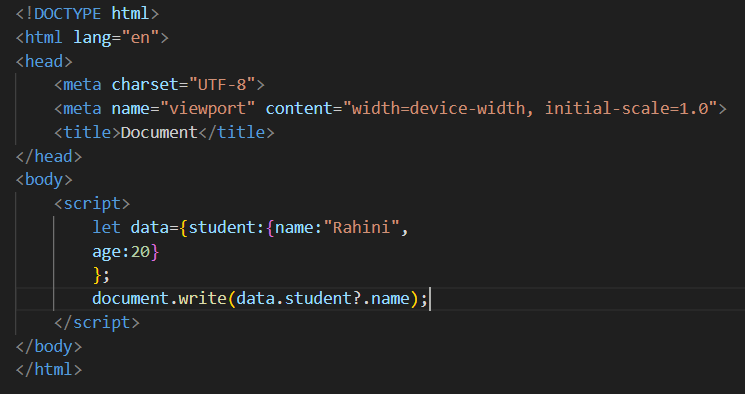
****

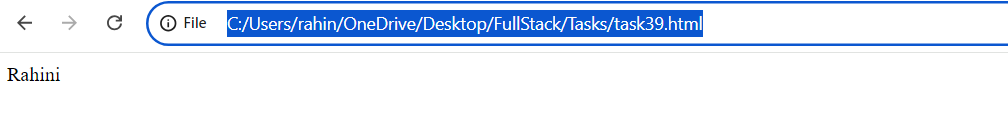
1. Use the conditional (ternary) operator ‘?’ to rewrite a simple if…else statement.

****

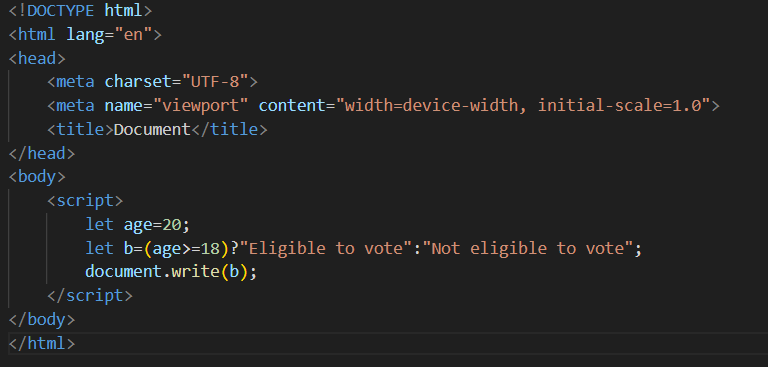
****

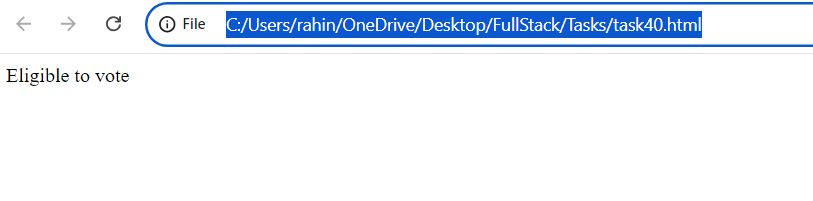
1. Check the validity of a variable using the ? operator.

****

****

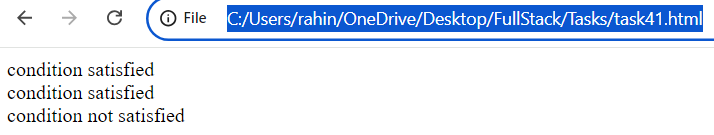
1. Use the conditional operator to assign a value to a variable based on a condition.

****

****

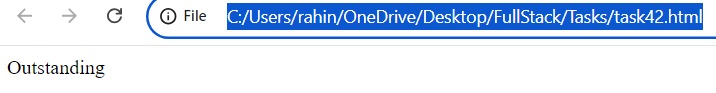
1. Evaluate various combinations of logical operators (&&, ||, !).

****

****

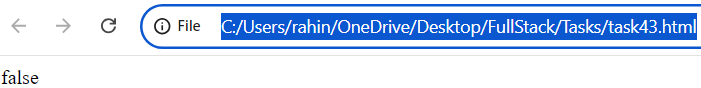
1. Use logical operators to write a condition that checks if a number is in a given range.

****

****

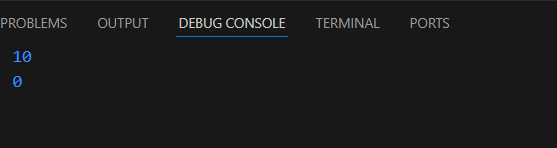
1. Use the NOT (!) operator to invert a boolean value.

****

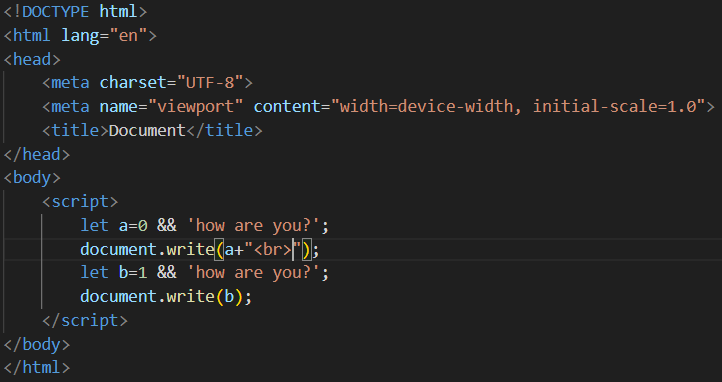
****

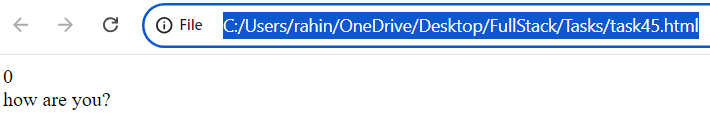
1. Evaluate the short-circuiting nature of logical operators.

****

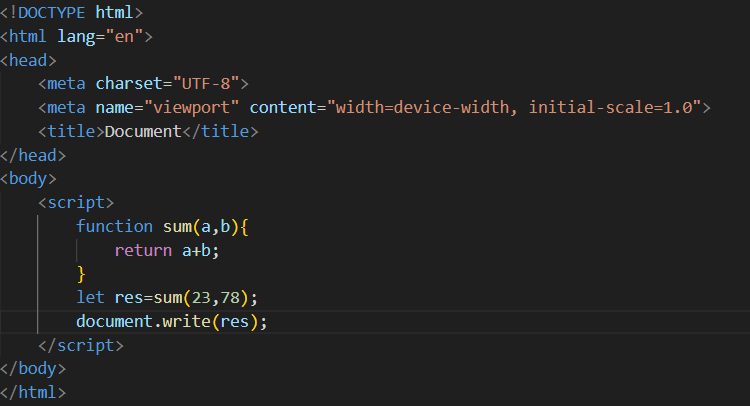
****

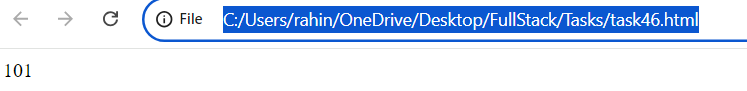
1. Compare two non-boolean values using logical operators and observe the result.

****

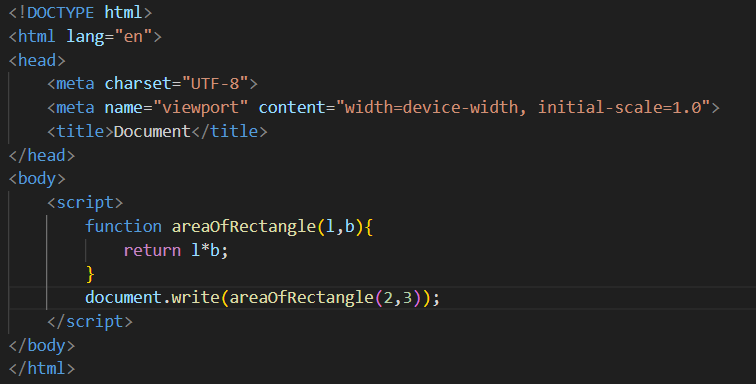
****

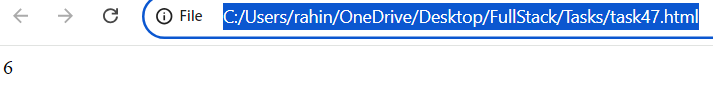
1. Write a function that takes two numbers as arguments and returns their sum.

****

****

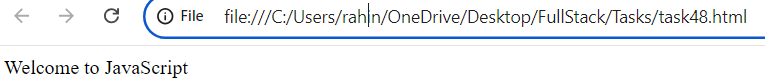
1. Create a function that calculates the area of a rectangle.

****

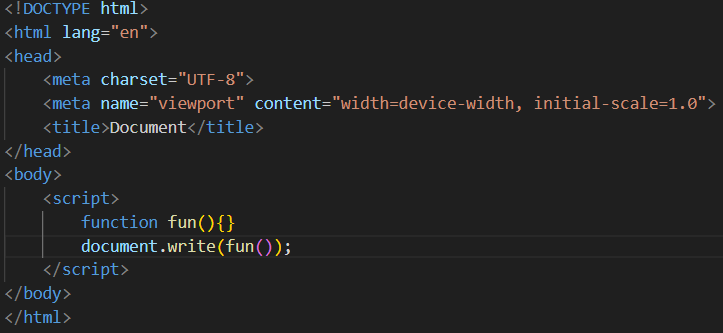
****

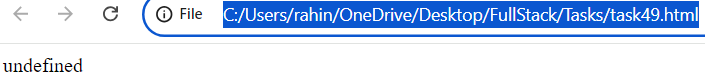
1. Declare a function without parameters and call it.

****

****

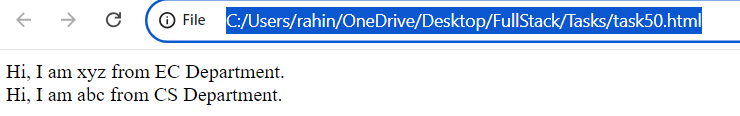
1. Write a function that returns nothing and observe the default return value.

****

****

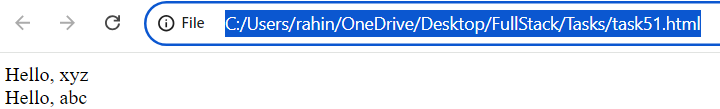
1. Declare a function with default parameters and call it with different arguments.



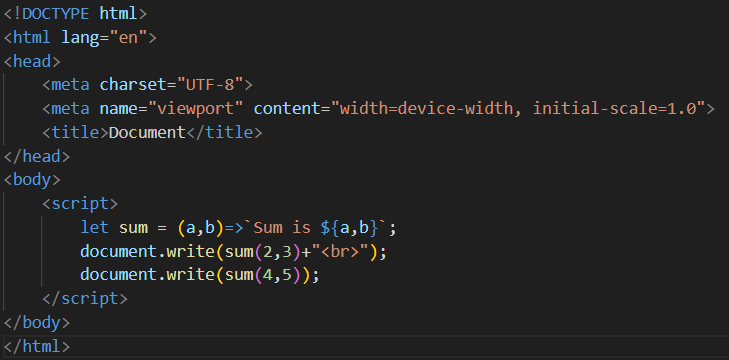
****

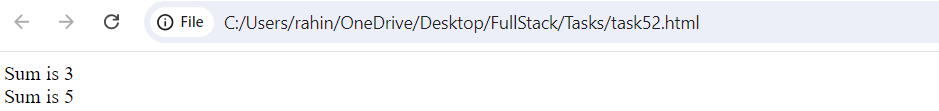
1. Declare a simple arrow function named greet that takes one parameter name and returns the string “Hello, name!”. Test your function with various names.

****

****

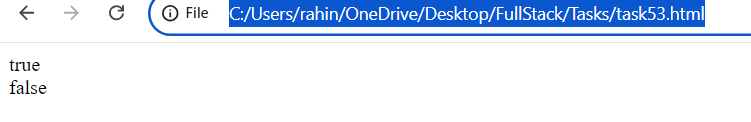
1. Write an arrow function named add that takes two parameters and returns their sum. Validate your function with several pairs of numbers.

****

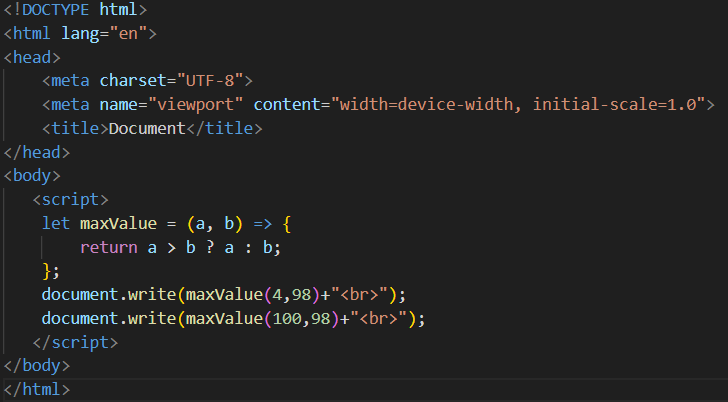
****

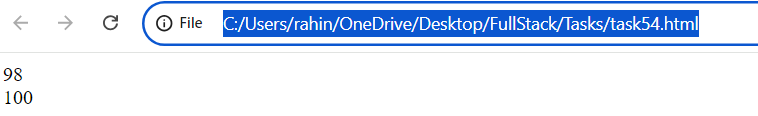
1. Declare an arrow function named isEven that checks if a number is even. If the number is even, it should return true; otherwise, false. Remember that if the arrow function body has a single statement, you can omit the curly braces.

****

****

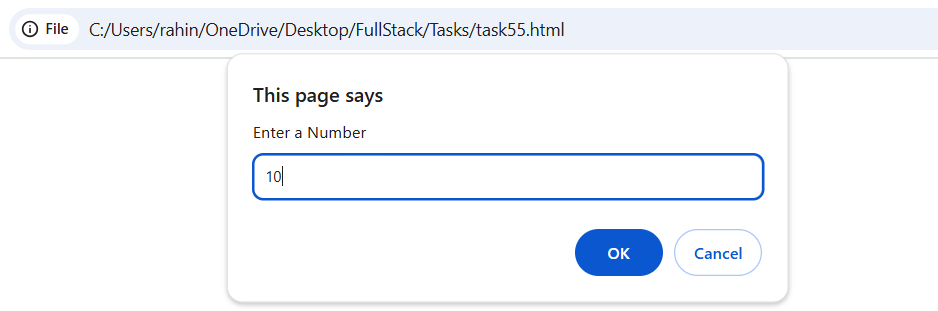
1. Implement an arrow function named maxValue that takes two numbers as parameters and returns the larger number. Here, you'll need to use curly braces for the function body and the return statement.

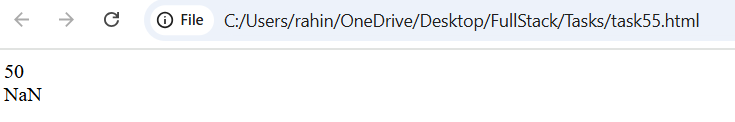
****

****

1. Examine the behavior of the *this* keyword inside an arrow function vs a traditional function. Create an object named myObject with a property value set to 10 and two methods: multiplyTraditional using a traditional function and multiplyArrow using an arrow function. Both methods should attempt to multiply the value property by a number passed as a parameter. Check the value of this inside both methods.

****

****

****