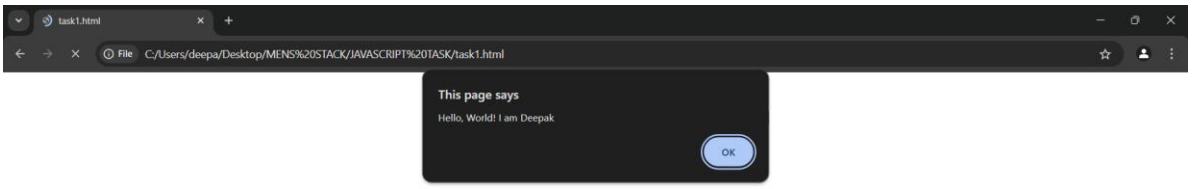
Task 1: Write a simple script that displays “Hello, World!” on the web page using an alert box

```
<html>
  <head>
  </head>
  <body>
    <script>
      window.alert("Hello, World! I am Deepak")
    </script>
  </body>
</html>
```

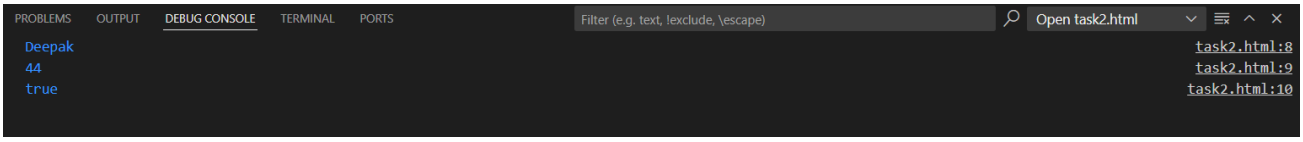
OUTPUT



Task 2: Experiment with different data types in JavaScript (e.g., string, number, boolean) by declaring and logging them in the console.

```
<html>
  <head></head>
  <body>
    <script>
      let a = "Deepak"
      let b = 44
      let c = true;
      console.log(a);
      console.log(b);
      console.log(c);
    </script>
  </body>
</html>
```

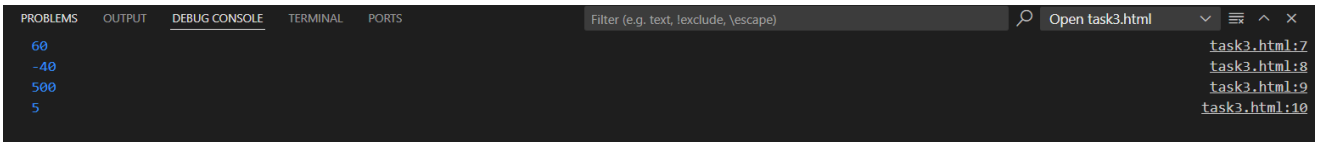
OUTPUT



Task 3: Use the console to perform basic math operations like addition, subtraction, multiplication, and division.

```
<html>
  <head></head>
  <body>
    <script>
      let a = 10;
      let b = 50;
      console.log(a+b);
      console.log(a-b);
      console.log(a*b);
      console.log(b/a);
    </script>
  </body>
</html>
```

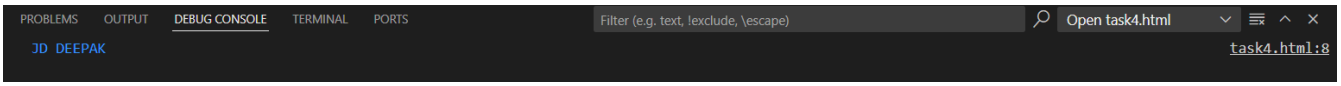
OUTPUT



Task 4: Declare two strings and concatenate them using the + operator.

```
<html>
  <head></head>
  <body>
    <script>
      let a = "JD";
      let b = "DEEPAK";
      let c = a+" "+b;
      console.log(c);
    </script>
  </body>
</html>
```

OUTPUT



Task 5: Use the typeof operator to check the data type of various variables.

```
<html>
  <head></head>
  <body>
    <script>
      let a = "Deepak"
      let b = 44
      let c = true;
      console.log(typeof a);
      console.log(typeof b);
      console.log(typeof c);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS        Filter (e.g. text, !exclude, \escape)        Open task5.html
string                                                                                                  task5.html:8
number                                                                                                  task5.html:9
boolean                                                                                                 task5.html:10
```

Task 6: Write a multi-line JavaScript comment and a single-line comment. Explain the difference.

```
<html>
  <head></head>
  <body>
    <script>
      //single line comment
      /* multi
        line
        comment*/
      console.log("comments");
    </script>
  </body>
</html>
```

OUTOUT

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS        Filter (e.g. text, !exclude, \escape)        Open task6.html
comments                                                                                                task6.html:9
```

Task 7:Create a script with both semicolon-separated and not separated lines.Note any differences in behavior.

```
<html>
  <head></head>
  <body>
    <script>
      let a = 10;
      let b = 44;
      console.log(a+b);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS        Filter (e.g. text, !exclude, \escape)        Open task7.html
54                                                                                                      task7.html:7
```

```
<html>
  <head></head>
  <body>
    <script>
      let a = 10
      let b = 44
      console.log(a+b)
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS        Filter (e.g. text, !exclude, \escape)        Open task7_1.html
54                                                                                                      task7_1.html:7
```

Task 8: Use proper indentation to format a nested loop.

```
<html>
  <head></head>
  <body>
    <script>
      for(let i =1;i<3;i++){
        for(let j =1;j<11;j++){
          console.log(j*i);
        }
      }
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task8.html
1                                                                                              task8.html:7
2                                                                                              task8.html:7
3                                                                                              task8.html:7
4                                                                                              task8.html:7
5                                                                                              task8.html:7
6                                                                                              task8.html:7
7                                                                                              task8.html:7
8                                                                                              task8.html:7
9                                                                                              task8.html:7
10                                                                                             task8.html:7
2                                                                                              task8.html:7
4                                                                                              task8.html:7
6                                                                                              task8.html:7
8                                                                                              task8.html:7
10                                                                                             task8.html:7
12                                                                                             task8.html:7
14                                                                                             task8.html:7
16                                                                                             task8.html:7
18                                                                                             task8.html:7
20                                                                                             task8.html:7
```

Task 9: Declare multiple variables in a single line.

```
<html>
  <head></head>
  <body>
    <script>
      let a = 7, b = 4;
      console.log(a+b);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task9.html
11                                                                                             task9.html:6
```

Task 10: Place a script tag at the top and bottom of an HTML document. Note any differences in behavior.

```
<html>
  <head></head>
  <body>
    <script>
      let a = "DEEPAK";
      document.getElementById("para1").innerHTML = a;
    </script>
    <p id = "para1">HIIII HELLOOO</p>
  </body>
</html>
```

OUTPUT

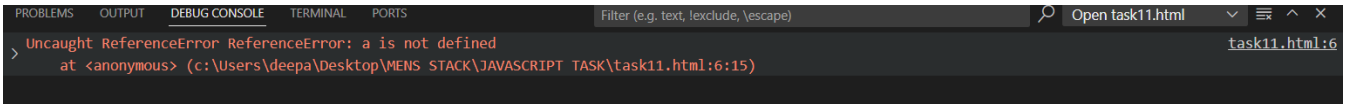

```
<html>
  <head></head>
  <body>
    <p id = "para1">HIIII HELLOOO</p>
    <script>
      let a = "DEEPAK";
      document.getElementById("para1").innerHTML = a;
    </script>
  </body>
</html>
```

OUTPUT

Task 11: Write a script without using "use strict" and try to assign a value to an undeclared variable. Note the result.

```
<html>
  <head></head>
  <body>
    <script>
      "use strict"
      a = 10;
      console.log(a);
    </script>
  </body>
</html>
```

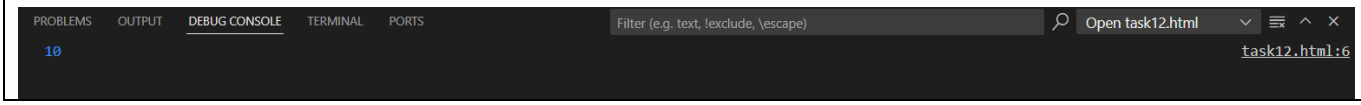
OUTPUT



Task 12: Enable "use strict" mode and repeat the above action, noting the difference.

```
<html>
  <head></head>
  <body>
    <script>
      a = 10;
      console.log(a);
    </script>
  </body>
</html>
```

OUTPUT



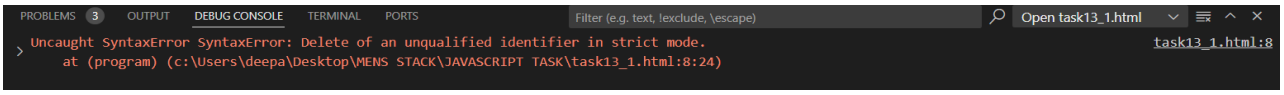
Task 13: In “use strict” mode, try to delete a variable, function, or function parameter.

```
<html>
  <head></head>
  <body>
    <script>
      "use strict"
      var a = 10;
      function function1(val1){
        delete x;
      }
      console.log(x);
    </script>
  </body>
</html>
```

OUTPUT


```
<html>
  <head></head>
  <body>
    <script>
      "use strict"
      var a = 10;
      function function1(val1){
        delete function1;
      }
      console.log(x);
    </script>
  </body>
</html>
```

OUTPUT




```
<html>
  <head></head>
  <body>
    <script>
      "use strict"
      var a = 10;
      function function1(val1){
        delete val1;
        return val1;
      }
      var result = function1(6);
      console.log(result);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 3   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS        Filter (e.g. text, !exclude, \escape)        Open task13_2.html
> Uncaught SyntaxError SyntaxError: Delete of an unqualified identifier in strict mode.                    task13_2.html:8
    at (program) (c:\Users\deepa\Desktop\MENS STACK\JAVASCRIPT TASK\task13_2.html:8:24)
```

Task 14: Assign a value to an undeclared variable without “use strict” and then with “use strict”.

```
<html>
  <head></head>
  <body>
    <script>
      a = 10;
      console.log(a);
      "use strict"
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 3   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS        Filter (e.g. text, !exclude, \escape)        Open task14.html
 10                                                                                                       task14.html:6
```

Task 15: Declare a variable with a reserved keyword in “use strict” mode.

```
<html>
  <head></head>
  <body>
    <script>
      "use strict"
      var break = 20;
      console.log(break);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 7   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS        Filter (e.g. text, !exclude, \escape)        Open task15.html
> Uncaught SyntaxError SyntaxError: Unexpected token 'break'                                              task15.html:6
    at (program) (c:\Users\deepa\Desktop\MENS STACK\JAVASCRIPT TASK\task15.html:6:17)
```

Task 16: Declare variables using let, const, and var. Discuss when each should be used.

```
<html>
  <head></head>
  <body>
    <script>
      let a = 10;
      var b = "DEEPAK";
      const pi = 3.14;
      console.log(a+" "+b+" "+pi);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 7   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS        Filter (e.g. text, !exclude, \escape)        Open task16.html
 10 DEEPAK 3.14                                                                                           task16.html:8
```

Task 17: Attempt to reassign a const variable and observe the result.

```
<html>
  <head></head>
  <body>
    <script>
      const pi = 3.14;
      const pi = 3;
      console.log(pi);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task17.html
> Uncaught SyntaxError SyntaxError: Identifier 'pi' has already been declared                    task17.html:6
    at (program) (c:\Users\deepa\Desktop\MENS STACK\JAVASCRIPT TASK\task17.html:6:19)
```

Task 18: Declare a variable without initializing it and print its value.

```
<html>
  <head></head>
  <body>
    <script>
      console.log(10);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task18.html
10                                                                                                task18.html:5
```

Task 19: Assign a number, string, and boolean value to a variable and print its type using typeof.

```
<html>
  <head></head>
  <body>
    <script>
      let a = 10;
      let b = "JD";
      let c = true;
      console.log(typeof a);
      console.log(typeof b);
      console.log(typeof c);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task19.html
number                                                                                            task19.html:8
string                                                                                            task19.html:9
boolean                                                                                           task19.html:10
```

Task 20: Rename a variable and observe the outcome.

```
<html>
  <head></head>
  <body>
    <script>
      let a = 101;
      let b = a;
      console.log(b);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task20.html
101                                                                                              task20.html:7
```

Task 21: Create variables of different data types (e.g., string, number, boolean, null, undefined, object).

```
<html>
  <head></head>
  <body>
    <script>
      let a = "JD";
      let b = 44;
      let c = true;
      let d = {name:"JD",age:21};
      let e = null;
      let f = undefined;
      console.log(typeof a)
      console.log(typeof b)
      console.log(typeof c)
      console.log(typeof d)
      console.log(typeof e)
      console.log(typeof f)
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task21.html
string                                                                                           task21.html:11
number                                                                                           task21.html:12
boolean                                                                                          task21.html:13
object                                                                                           task21.html:14
object                                                                                           task21.html:15
undefined                                                                                        task21.html:16
```

Task 22: Use the typeof operator to determine the type of various variables.

```
<html>
  <head></head>
  <body>
    <script>
      let a = "JD";
      let b = 44;
      let c = true;
```

```
        let d = {name:"JD",age:21};
        let e = null;
        let f = undefined;
        console.log(typeof a)
        console.log(typeof b)
        console.log(typeof c)
        console.log(typeof d)
        console.log(typeof e)
        console.log(typeof f)
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task22.html
string                                                                                            task22.html:1
number                                                                                            task22.html:1
boolean                                                                                           task22.html:1
object                                                                                            task22.html:1
object                                                                                            task22.html:1
undefined                                                                                         task22.html:1
```

Task 23: Declare a symbol and print its type.

```
<html>
  <head></head>
  <body>
    <script>
      let a = "@";
      console.log(a);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task23.html
@                                                                                                 task23.html:6
```

Task 24: Assign the value null to a variable and check its type using typeof.

```
<html>
  <head></head>
  <body>
    <script>
      let a = null;
      console.log(typeof a);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task24.html
object                                                                                            task24.html:6
```

Task 25: Differentiate between declaring a variable using var and let in terms of scope.

```
<html>
  <head></head>
  <body>
    <script>
      function add(b){
        let a = 10;
        console.log(a+b);
      }
      console.log(a);
      add(10);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task25.html
> Uncaught ReferenceError ReferenceError: a is not defined                                        task25.html:9
    at <anonymous> (c:\Users\deepa\Desktop\MENS STACK\JAVASCRIPT TASK\task25.html:9:25)
```

```
<html>
  <head></head>
  <body>
    <script>
      var e = 20;
      function sub(c){
        var e = 20;
        console.log(c-e);
      }
      let k = e+5;
      console.log(k);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task25_1.html
25                                                                                                task25_1.html:11
```

Task 26: Convert a string to a number using both implicit and explicit conversion.

```
<html>
  <head></head>
  <body>
    <script>
      var a = "108";
      console.log(typeof a);
      var a = Number(a);
      console.log(typeof a);
    </script>
  </body>
</html>
```

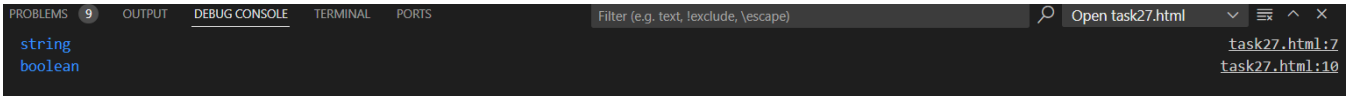
OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task26.html
string                                                                                            task26.html:6
number                                                                                            task26.html:8
```

Task 27: Convert a boolean to a string and vice versa.

```
<html>
  <head></head>
  <body>
    <script>
      var a = true;
      var a = String(a);
      console.log(typeof a);
      var b = "true";
      var b = Boolean(b);
      console.log(typeof b);
    </script>
  </body>
</html>
```

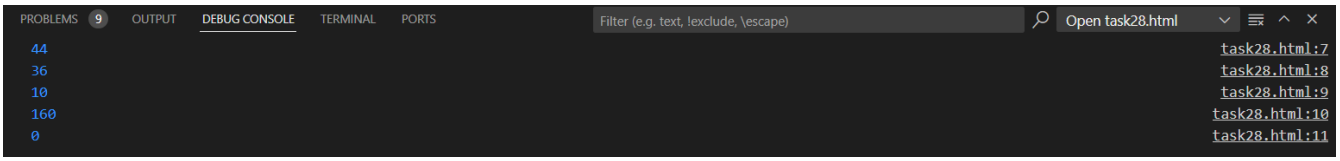
OUTPUT



Task 28: Practice basic arithmetic operators (+, -, *, /, %).

```
<html>
  <head></head>
  <body>
    <script>
      let a = 40;
      let b = 4;
      console.log(a+b);
      console.log(a-b);
      console.log(a/b);
      console.log(a*b);
      console.log(a%b);
    </script>
  </body>
</html>
```

OUTPUT



Task 29: Use the ++ and -- operators on a numeric variable.

```
<html>
  <head></head>
  <body>
    <script>
      let a = 40;
      if(a > 0){
        a++;
      }
      else{
        a--;
```

```
        }
        console.log(a);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task29.html
41                                                                                                task29.html:12
```

Task 30: Explore the precedence of operators by combining multiple operators in a single expression.

```
<html>
  <head></head>
  <body>
    <script>
      let a = 40;
      let b = 2;
      let c = 20;
      console.log(a*b-c);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task30.html
60                                                                                                task30.html:8
```

Task 31: Compare two numbers using relational operators (>, <, >=, <=).

```
<html>
  <head></head>
  <body>
    <script>
      let a = 40;
      let b = 2;
      if(a > b){
        console.log("a is greater");
      }
      else if (a < b) {
        console.log("b is greater");
      }
      else {
        console.log("Both are equal");
      }
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task31.html
a is greater                                                                                      task31.html:8
```

Task 32: Use equality () and strict equality (=) operators to compare different data types and note the differences.

```
<html>
  <head></head>
  <body>
    <script>
      let a = 4;
      let b = '4';
      console.log(a === b);
      console.log( a == b);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task32.html
false                                                                                            task32.html:7
true                                                                                             task32.html:8
```

Task 33: Compare two strings lexicographically.

```
<html>
  <head></head>
  <body>
    <script>
      let a = "apple";
      let b = "banana";
      console.log(a < b);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task33.html
true                                                                                             task33.html:7
```

Task 34: Use the inequality (!=) and strict inequality (!==) operators to compare values.

```
<html>
  <head></head>
  <body>
    <script>
      let a = 3;
      let b = "3";
      console.log(a != b);
      console.log(a !== b);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task34.html
false                                                                                            task34.html:7
true                                                                                             task34.html:8
```

Task 35: Compare null and undefined using both == and ===.

```
<html>
  <head></head>
  <body>
    <script>
      let a;
      let b = undefined;
      console.log(a == b);
      console.log( a === b);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task35.html
true                                                                                              task35.html:7
true                                                                                              task35.html:8
```

Task 36: Write an if statement that checks if a number is even or odd.

```
<html>
  <head></head>
  <body>
    <script>
      let a = 4;
      if( a %2 == 0 ){
        console.log("a is EVEN");
      }
      else{
        console.log("a is ODD");
      }
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task36.html
a is EVEN                                                                                         task36.html:7
```

Task 37: Use nested if statements to classify a number as negative, positive, or zero.

```
<html>
  <head></head>
  <body>
    <script>
      let a = -4;
      if( a > 0){
        console.log("a is POSITIVE");
      }
      else if( a < 0){
        console.log("a is NEGATIVE");
      }
      else{
        console.log("a is ZERO");
      }
    </script>
```

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

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task37.html
a is NEGATIVE                                                                                    task37.html:10
```

Task 38: Use the conditional (ternary) operator '?' to rewrite a simple if…else statement.

```
<html>
  <head></head>
  <body>
    <script>
      let a = -4;
      let b = a > 0 ? "POSITIVE" : "NEGATIVE";
      console.log(b);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task38.html
NEGATIVE                                                                                         task38.html:7
```

Task 39: Check the validity of a variable using the ? operator.

```
<html>
  <head></head>
  <body>
    <script>
      let a ={name:"JD", age:21};
      console.log(a?.name);
      console.log(a?.job);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task39.html
JD                                                                                               task39.html:6
undefined                                                                                        task39.html:7
```

Task 40: Use the conditional operator to assign a value to a variable based on a condition.

```
<html>
  <head></head>
  <body>
    <script>
      var a = 4;
      var b = 3;
      if( a%2 == 0 && b%2 == 0){
        var a = 0;
        var b = 0;
      }
      else if( a%2 == 0 || b%2 == 0){
        var a =1;
        var b =1;
      }
```

```
        console.log("a is:"+a);
        console.log("b is:"+b)
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task40.html
 a is:1                                                                                          task40.html:1
 b is:1                                                                                          task40.html:1
```

Task 41: Evaluate various combinations of logical operators (&&, ||, !).

```
<html>
  <head></head>
  <body>
    <script>
      let a = 2;
      let b = 3;
      if( a %2 == 0 && b %2 == 0 ){
        console.log("both ar even");
      }
      else if( a %2 ==0 || b%2 == 0){
        console.log("one number is even");
      }
      else if( !a %2 == 0){
        console.log("a is odd");
      }
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task41.html
 one number is even                                                                              task41.html:11
```

Task 42: Use logical operators to write a condition that checks if a number is in a given range.

```
<html>
  <head></head>
  <body>
    <script>
      let a = 4;
      let sr = 1;
      let er = 10;
      if( a >= sr && a <= er){
        console.log("Present inside the range");
      }
      else{
        console.log("Present outside the range");
      }
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS      Filter (e.g. text, !exclude, \escape)      Open task42.html
Present inside the range                                                                              task42.html:9
```

Task 43: Use the NOT (!) operator to invert a boolean value.

```
<html>
  <head></head>
  <body>
    <script>
      var a = true;
      if( !a == false){
        var a = false;
      }
      console.log(a);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS      Filter (e.g. text, !exclude, \escape)      Open task43.html
false                                                                                                 task43.html:9
```

Task 44: Evaluate the short-circuiting nature of logical operators.

```
<html>
  <head></head>
  <body>
    <script>
      var a = false || 0 || "JD" || null;
      console.log(a);
      var b = false && 0 && "JD" && null;
      console.log(b);
      var c = false || 0 || null;
      console.log(c);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS      Filter (e.g. text, !exclude, \escape)      Open task44.html
JD                                                                                                    task44.html:6
false                                                                                                 task44.html:8
null                                                                                                  task44.html:10
```

Task 45: Compare two non-boolean values using logical operators and observe the result.

```
<html>
  <head></head>
  <body>
    <script>
      let a = 3;
      let b = 5;
      if( a==b && a!=b ){
        console.log("both the condition are true");
      }
      else if( a == b || a!=b){
        console.log("one condition is true");
```

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

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task45.html
one condition is true                                                                            task45.html:11
```

Task 46: Write a function that takes two numbers as arguments and returns their sum.

```
<html>
  <head></head>
  <body>
    <script>
      function sum(a,b){
        console.log(a+b);
      }
      sum(143,341);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task46.html
484                                                                                              task46.html:6
```

Task 47: Create a function that calculates the area of a rectangle.

```
<html>
  <head></head>
  <body>
    <script>
      function area(length,width){
        console.log("The area is:"+length*width);
      }
      area(17,10);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task47.html
The area is:170                                                                                  task47.html:6
```

Task 48: Declare a function without parameters and call it.

```
<html>
  <head></head>
  <body>
    <script>
      function noprameter(){
        var a = 2;
        var b = 47;
        console.log(a-b);
      }
      noprameter();
    </script>
```

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

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task48.html
-45                                                                                              task48.html:8
```

Task 49: Write a function that returns nothing and observe the default return value.

```
<html>
  <head></head>
  <body>
    <script>
      function function1(){
        var a = 27;
        var k = 3;
      }
      let v = function1();
      console.log(v);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task49.html
undefined                                                                                        task49.html:10
```

Task 50: Declare a function with default parameters and call it with different arguments.

```
<html>
  <head></head>
  <body>
    <script>
      function defaultparameter(a = "JD",b = "Boss"){
        var res = a+" "+b;
        return res;
      }
      var result = defaultparameter("Deepak","Vijay");
      console.log(result);
    </script>
  </body>
</html>
```

OUTPUT

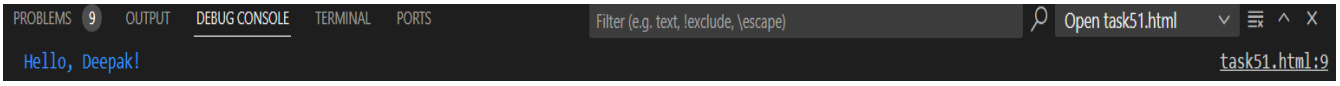
```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Filter (e.g. text, !exclude, \escape)   Open task50.html
Deepak Vijay                                                                                     task50.html:10
```

Task 51: Declare a simple arrow function named greet that takes one parameter name and returns the string "Hello, name!". Test your function with various names.

```
<html>
  <head></head>
  <body>
    <script>
      var greet = (name) =>{
        return "Hello, "+name+"!";
      }
      var result = greet("Deepak");
```

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

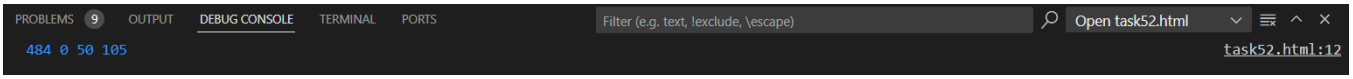
OUTPUT



Task 52: Write an arrow function named add that takes two parameters and returns their sum. Validate your function with several pairs of numbers.

```
<html>
  <head></head>
  <body>
    <script>
      var add = (num1,num2) =>{
        return num1 + num2;
      }
      var obj1 = add(143,341);
      var obj2 = add(0,0);
      var obj3 = add(10,40);
      var obj4 = add(78,27);
      console.log(obj1+" "+obj2+" "+obj3+" "+obj4);
    </script>
  </body>
</html>
```

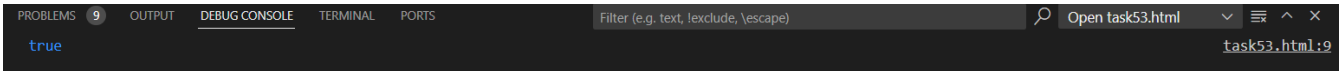
OUTPUT



Task 53: 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.

```
<html>
  <head></head>
  <body>
    <script>
      var isEven = (num1) =>{
        return num1 %2 == 0 ? "true":"false";
      }
      var result = isEven(4);
      console.log(result);
    </script>
  </body>
</html>
```

OUTPUT



Task 54: 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.

```
<html>
  <head></head>
  <body>
    <script>
      var maxValue =(num1,num2) =>{
        console.log("num1: "+num1+" and "+"num2: "+num2);
        return Math.max(num1,num2);
      }
      let result = maxValue(15,22);
      console.log("The larger number is: "+result);
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS        Filter (e.g. text, !exclude, \escape)        Open task54.html
num1: 15 and num2: 22                                                                                   task54.html:6
The larger number is: 22                                                                                task54.html:10
```

Task 55: 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.

```
<html>
  <head></head>
  <body>
    <script>
      var myObject ={
        value :47,
        multiplyArrow:(num1) =>{
          return this.value * num1;
        },
        multiplyTraditional : function(num2){
          return this.value * num2;
        }
      };
      console.log(myObject.multiplyArrow(4));
      console.log(myObject.multiplyTraditional(4));
    </script>
  </body>
</html>
```

OUTPUT

```
PROBLEMS 9   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS        Filter (e.g. text, !exclude, \escape)        Open task55.html
NaN                                                                                                     task55.html:14
188                                                                                                     task55.html:15
```