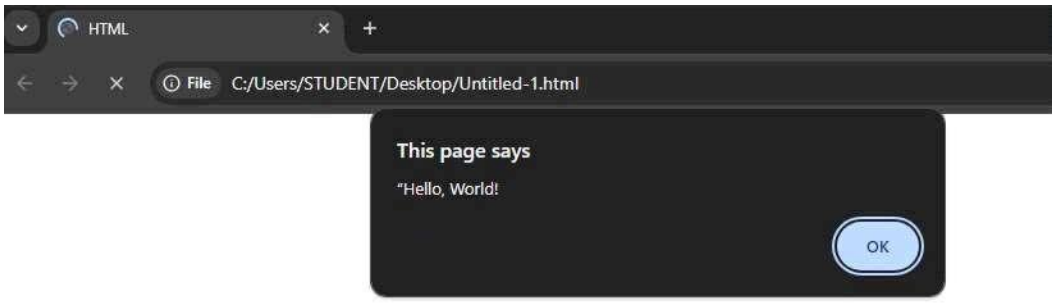
Task-1

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

```
<!Doctype html>

<html>

   <head>

   <title>HTML</title>

   <body>

      <script>

     alert("Hello, World!");

      </script>

      </body>

   </head>

</html>
```

OUTPUT:

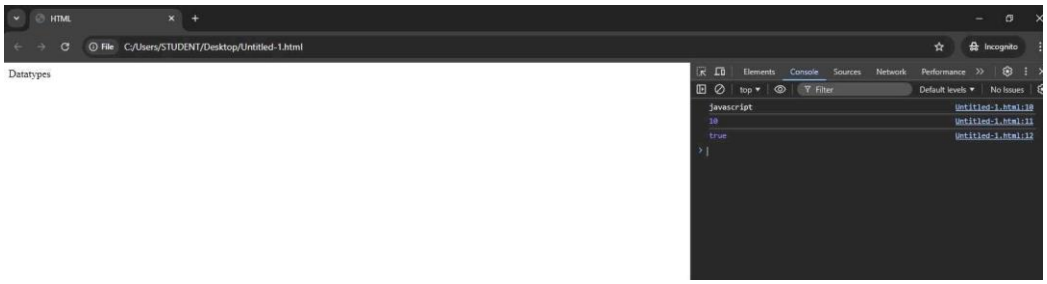


2.

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

```
<!Doctype html>
<html>
  <head>Datatypes
  <title>HTML</title>
  <body>
    <script>

   let str="javascript";
   let  num=10;
   let bool=true;
   console.log(str);
   console.log(num);
   console.log(bool);
    </script>
    </body>

  </head>
```

OUTPUT:



3.

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

```
<!Doctype html>
<html>
  <head>
  <title>MATHAMATICAL OPERATIONS</title>
  <body>

    <script>
   let a=100;
   let b=200;
   console.log(a+b);
   console.log(a-b);
   console.log(a*b);
   console.log(a/b);
    </script>
    </body>
  </head>
```

OUTPUT:

4.

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

```
<!Doctype html>

<html>

   <head>

   <title>CONCATE</title>

   <body>

      <script>


    let str1="Java";

    let str2="script";
    console.log("concate of two string: "+str1+str2);

      </script>

      </body>

   </head>

</html>
```

OUTPUT:

5.

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

```
<!Doctype html>

<html>

   <head>

   <title></title>

   <body>

      <script>
```

```
    let str1="Java";
    let num=40;
    let bool=true;
    let dou;
    let
      person={ name:"a
      run",
      dept:"IT",
    };

  console.log(typeof(str1));
  console.log(typeof(num));
  console.log(typeof(bool));
  console.log(typeof(person));
  console.log(typeof(dou));


      </script>
      </body>
    </head>

</html>
```

OUTPUT:

6.

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

Explain the difference.

```
<!Doctype html>

<html>

  <head>

  <title>MATHAMATICAL OPERATIONS</title>

  <body>

    <script>


   /*let str1="Java";

   let num=40;

   let bool=true;*/

    let dou;
   let

     person={ name:"a

     run",
     dept:"IT",

   };


 //single line command console.log(typeof(str1));

 console.log(typeof(num));


 console.log(typeof(bool));

 console.log(typeof(person));

 console.log(typeof(dou));



    </script>

    </body>

  </head>

</html>
```

OUTPUT:

SINGLE-LINE COMMAND:

A single-line command is one that is written and executed in a single line of code

MULTI-LINE COMMAND:

A multi-line command refers to a block of code that spans across multiple lines.

7.

Task 7: Create a script with both semicolon-separated and not separated lines.

Note any differences in behavior.

```
<!Doctype html>

<html>

   <head>

   <title></title>

   <body>


      <script>

   let a = 10;
let b = 20;


let sum = a + b;

console.log(sum);
```

```
let x = 10
let y = 20


let total = x + y

console.log(total)
      </script>

      </body>

   </head>

</html>
```

OUTPUT:

There is no difference in their behaviour

8.

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

```
<!Doctype html>

<html>

   <head>

   <title></title>

   <body>
```

```
        <script>
    let age=18;
      if(isNaN())
       { if(age>=18){
          console.log("you are eligible for voting");

       }

       else{

        alert("you are not eligible for voting");

       }

        }

        else{

          console.log("enetr a valid number");

        }


       </script>

       </body>

    </head>

</html>
```

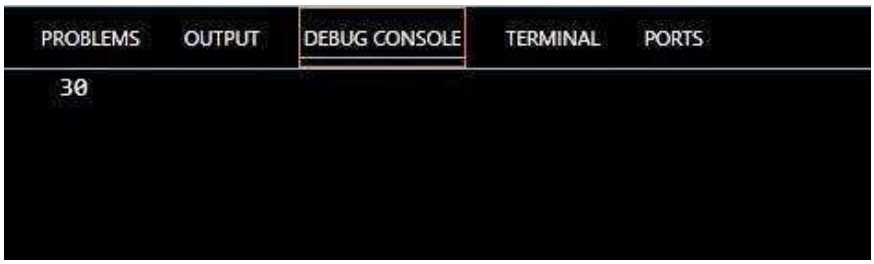
OUTPUT:

9.

Task 9: Declare multiple variables in a single line.

```
<!Doctype html>
<html>
  <head>
  <title></title>
  <body>
    <script>

  let num1,num2,sum; num1=
  10;


  num2=20;
  sum=num1+num2;
  console.log(sum);
    </script>
    </body>
  </head>
</html>
```

OUTPUT:



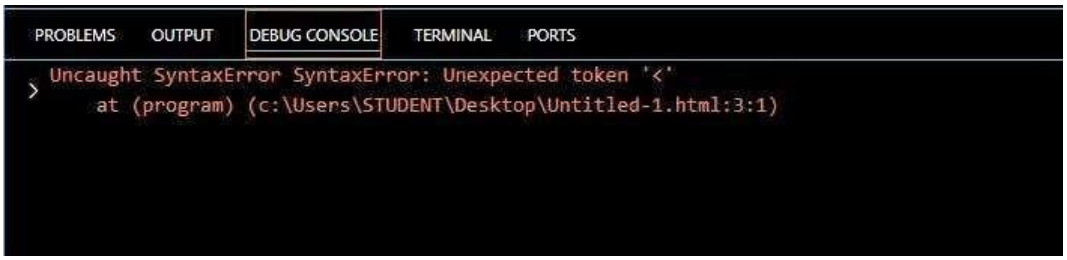
10.

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

```
<!Doctype html>
<script>
```

```
<html>
  <head>
  <body>
    let str1="java";
    let str2="script";
    console.log(str1+str2);
    <h1>Script tag before body</h1>
    </body>
  </head>
</html>

</script>
```

OUTPUT:



11.

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

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Javascript</title>
    </head>
    <body>
```

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

OUTPUT:

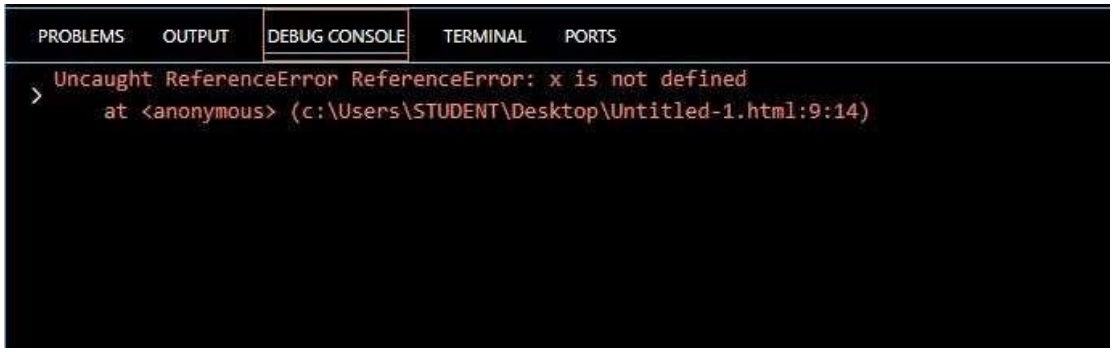
12.

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

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Javascript</title>
    </head>
    <body>
    <script>
      "use strict"; x=
      10;
      console.log(x);
```

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

OUTPUT:

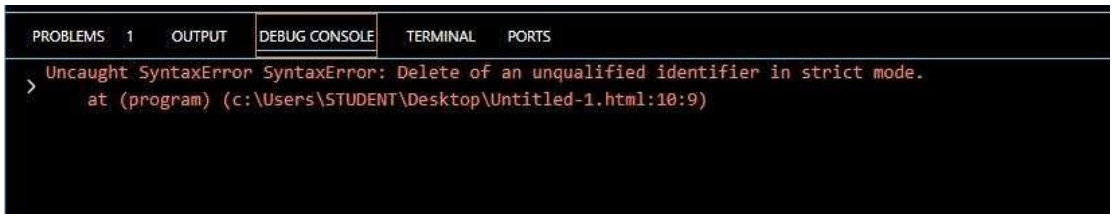


13.In "use strict" mode, try to delete a variable, function, or function parameter.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Javascript</title>
    </head>
    <body>
    <script>
```

```
        "use strict";
function x(p1, p2) {};
delete x;
        </script>
        </body>
        </html>
```

OUTPUT:



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

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Javascript</title>
    </head>
    <body>

    <script>
        x = 10;
console.log(x);
```

```
    </script>

    </body>

    </html>
```

OUTPUT:


```
<!DOCTYPE HTML>

<html>


  <head>

    <title>Javascript</title>

    </head>

    <body>


    <script>

      "use strict";

      x=10;
      console.log(x);

      </script>

      </body>

      </html>
```

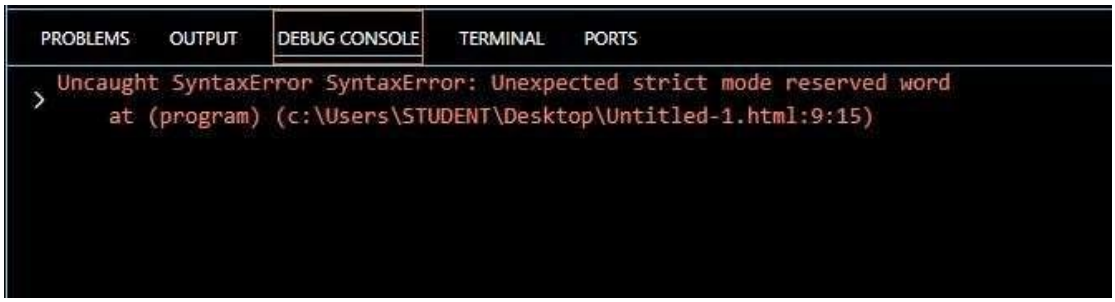
OUTPUT:

15.

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

```
<!DOCTYPE HTML>
<html>
   <head>
      <title>Javascript</title>
      </head>
      <body>
      <script>

        "use strict";
        var public=10;
        console.log(x);

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

OUTPUT:



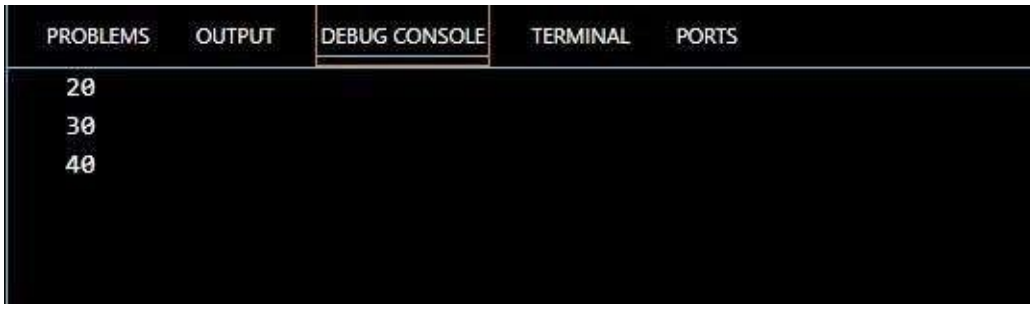
16

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

```
<!DOCTYPE HTML>

<html>

  <head>

    <title>Javascript</title>

    </head>

    <body>


    <script>

     var x=10;

     var x=20;
     console.log(x);


     let y=20;

     y=30;
```

```
console.log(y);

const z=40;

console.log(z);
  </script>

  </body>

  </html>
```

OUTPUT:



LET:we can redeclare the variable and It is not accessible outside the block where it is defined.

VAR:we can reassigning the values but we cannot redeclared and The variable declared with var is function-scoped or globally-scoped (if declared outside any function).

CONST: Use const when the variable's value will not change after it is initialized

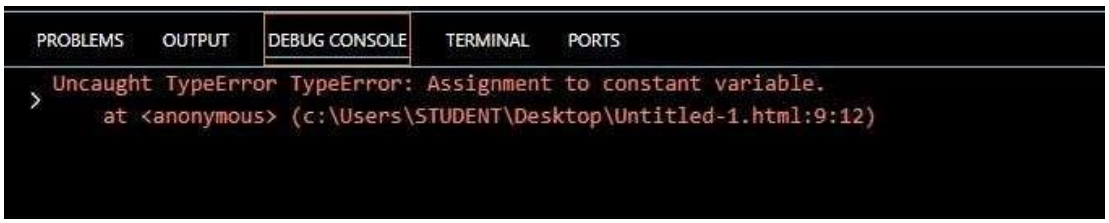
17.

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

```
<!DOCTYPE HTML>

<html>

  <head>

    <title>Javascript</title>

    </head>

    <body>


    <script>

     const n1=100;

     n1=200;
```

```
console.log(n1);
  </script>
  </body>
  </html>
```

OUTPUT:



18.

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

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Javascript</title>
    </head>
    <body>

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

OUTPUT:

19.

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

```
<!DOCTYPE HTML>

<html>

  <head>

    <title>Javascript</title>

    </head>

    <body>

    <script>

     let  num=100;

     let  str="sita";

     let bool=true;

     console.log(typeof(str));

     console.log(typeof(num));

     console.log(typeof(bool));
      </script>

      </body>

      </html>
```

OUTPUT:

20.

Rename a variable and observe the outcome.

```
<!DOCTYPE HTML>

<html>

  <head>

    <title>Javascript</title>

    </head>

    <body>

    <script>

     let num1=100;

     let num2=num1;
     console.log(num2);

      </script>

      </body>

      </html>
```

OUTPUT:

21.

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

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Javascript</title>
    </head>
    <body>
    <script>
     let Str="ram";

     let num=123456;
     let bool=true;
     let sum;

     let car=null;
     let fun={
       name:"sita",
       age:18,
     };

     console.log(typeof(str));
     console.log(typeof(num));
     console.log(typeof(bool));
     console.log(typeof(sum));
     console.log(typeof(car));
     console.log(typeof(fun));
      </script>
```

```
</body>

</html>
```

OUTPUT:

22.

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

```
<!DOCTYPE HTML>

<html>

  <head>

    <title>Javascript</title>

    </head>

    <body>

    <script>

     let Str="ram";


     let num=123456;

     let bool=true;
     let sum;
```

```
let
  fun={ name:"sit
  a", age:18,


};

console.log(typeof(str));
console.log(typeof(num));
console.log(typeof(bool));
console.log(typeof(sum));
console.log(typeof(fun));
  </script>
  </body>
  </html>
```

OUTPUT:

23.

Task 23: Declare a symbol and print its type.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Javascript</title>
```

```
</head>
<body>
<script>

 let symbol1=Symbol('js');
 console.log(typeof(symbol1));
  </script>
  </body>
  </html>
```

OUTPUT:

24.

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

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Javascript</title>
    </head>
    <body>
    <script>

    let num=null;
    console.log(typeof(num));
```

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

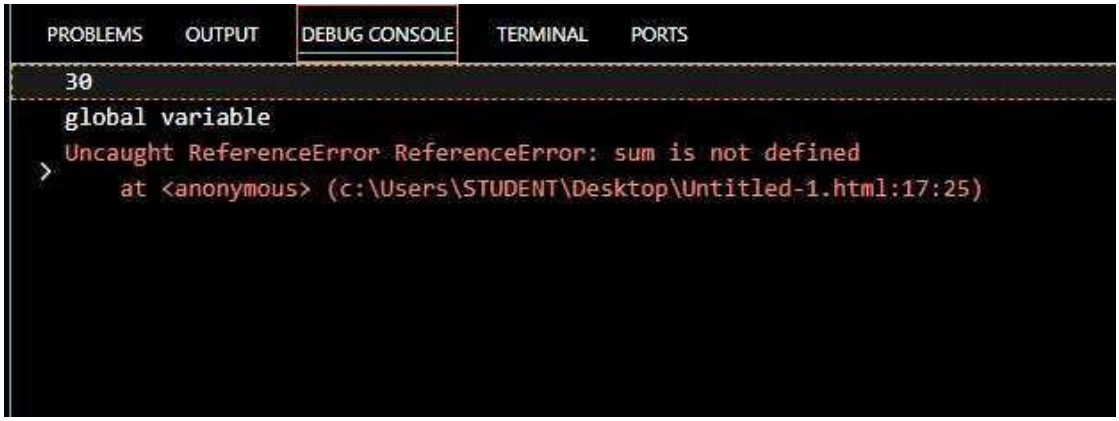
OUTPUT:

25.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Javascript</title>
  </head>
  <body>
    <script>

      var a = 10;
      var b = 20;
      var c = "global variable";
      fun();
      function fun()
        { let sum = a +
        b;
        console.log(sum);
      }
```

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

OUTPUT:



- var a , b , c are global variable...are all declared globally (outside of any function). These variables are accessible anywhere within the script.
- let sum is declared inside the function fun(). Since sum is declared with let, it only accessible inside the fun() function. Trying to access it outside fun() will result in an error.

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

```
<!DOCTYPE HTML>
<html>
   <head>
      <title>Javascript</title>
   </head>
   <body>
      <script>

      let str1 = "1234";
let res = str1 * 1;
console.log(res);
let str2 = "456";
let res2 = Number(str2);


console.log(res2);
</script>
   </body>
</html>
```

OUTPUT:

27.

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

```
<!DOCTYPE HTML>
<html>
   <head>
      <title>Javascript</title>
   </head>
   <body>
      <script>

         console.log("boolean to string");
      let bool=true;
      let res=String(bool);
      console.log(res);
      console.log("string to boolean");
      let str1="false";
      let res2=Boolean(str1);


      console.log(res2);
</script>
   </body>
</html>
```

OUTPUT:

28.

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

```
<!DOCTYPE HTML>
<html>
   <head>
      <title>Javascript</title>
   </head>
   <body>
      <script>

         let a=100;
         let b=200;
         let sum=a+b;
         let sub=a-b;
         let multi=a*b;
         let div=a/b;


         let mod=a%b;
```

```
        console.log(sum);
        console.log(sub);
        console.log(multi);
        console.log(div);
        console.log(mod);
</script>
  </body>
</html>
```

OUTPUT:

29.

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

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Javascript</title>
  </head>
  <body>
    <script>
      let num = 10;
```

```
        let prein = ++num;
        let posin = num++;
        let prede = --num;
        let postde = num--;
        console.log(prein);
        console.log(posin);
        console.log(prede);
        console.log(postde);
    </script>
  </body>
</html>
```

OUTPUT:

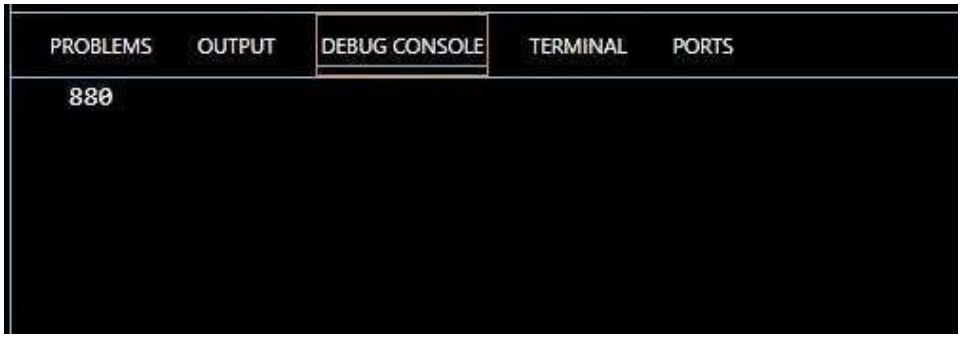
30.

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

```
<!DOCTYPE HTML>
<html>
  <head>
```

```
    <title>Javascript</title>
  </head>
  <body>

    <script>
      let a=10;
      let b=20;
      let c=30;
      let res=(a+b)*c-b;
      console.log(res);
    </script>
  </body>
</html>
```

OUTPUT:



31.

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

```
<!DOCTYPE HTML>
<html>
  <head></head>
  <title>Webpage</title>
  <body>
    <script>
    let a=10;
    let b=20;
    if(a>b){
      console.log("a is the biggest number");

    }
```

```
else if(b>a){
   console.log("b is the greatest number");

}
else if(a>=b){
   console.log9("a is the greatest number");

}
else{
   console.log("b is the greatest number");
}
</Script>
</body>
</html>
```

OUTPUT:

32.

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

```
<!DOCTYPE HTML>
<html>
  <head></head>
  <title>Webpage</title>
  <body>
    <script>
    let a=10;
    console.log(a=='10');
    console.log(a==='10');
    </Script>
    </body>
    </html>
```

OUTPUT:

33.

Task 33: Compare two strings lexicographically.

```
<!DOCTYPE HTML>
<html>
   <head></head>
   <title>Webpage</title>
   <body>
      <script>
   let str1="app";
   let str2="house";
   console.log(str1<str2);
   console.log(str1>str2);
   console.log(str1>=str2);
   console.log(str1<=str2);
      </Script>
      </body>
      </html>
```

OUTPUT:

34.

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

```
<!DOCTYPE HTML>
<html>
   <head></head>
   <title>Webpage</title>
   <body>
      <script>
   let a=10
   let b=20;
   let c='10';
   console.log(a!=b);
   console.log(a!==c);

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

OUTPUT:

35.

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

```
<!DOCTYPE HTML>
<html>
   <head></head>
   <title>Webpage</title>
   <body>
      <script>
   let str1=null;
   let str2;
   console.log(str1==str2);
   console.log(str1===str2);
```

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

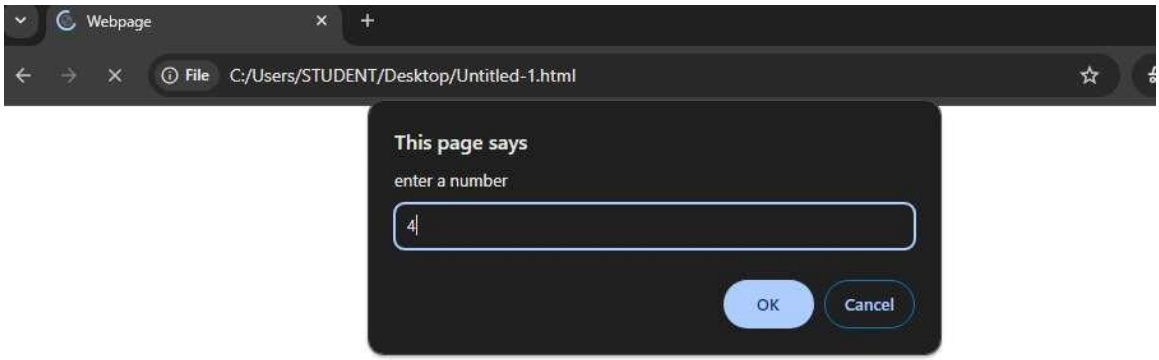
OUTPUT:

36.

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

```
<!DOCTYPE HTML>
<html>
   <head></head>
   <title>Webpage</title>
   <body>
      <script>
   let num=prompt("enter a number");
   let res=Number(num);
   if(isNaN()){
   if(num%2==0){
      console.log("even number");
   }
   else{
      console.log("odd number");
   }
}
else{
   console.log("enter a valid number");
}
      </Script>
      </body>
      </html>
```

OUTPUT:



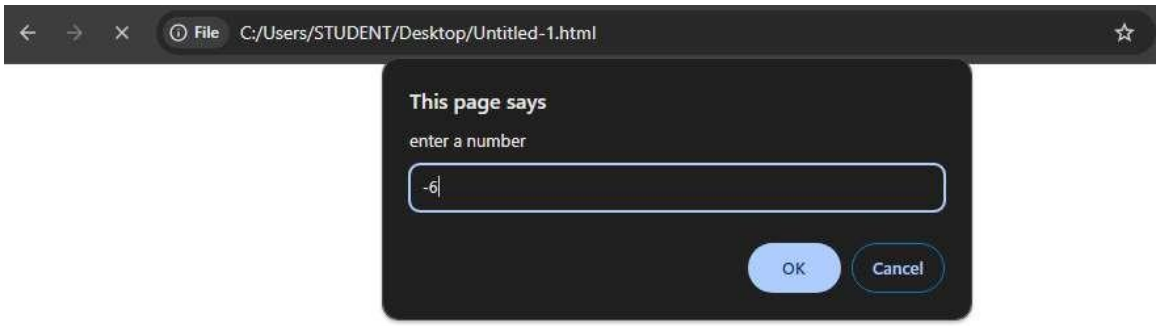
37.

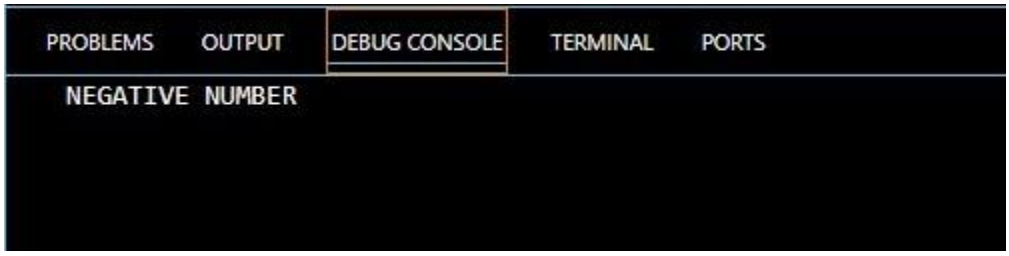
Task 37: Use nested if statements to classify a number as negative, positive, or zero.

```
<!DOCTYPE HTML>
<html>
   <head></head>
   <title>Webpage</title>
   <body>
      <script>
   let num=prompt("enter a number");
   let res=Number(num);
   if(isNaN()){
   if(num>0){
      console.log("POSITIVE NUMBER");
   }
   else if(num==0){
      console.log("GIVEN NUMBER IS 0");
   }
   else{
      console.log("NEGATIVE NUMBER")
   }
}
else{
   console.log("enter a valid number");
```

```
}
    </Script>
    </body>
    </html>
```

OUTPUT:



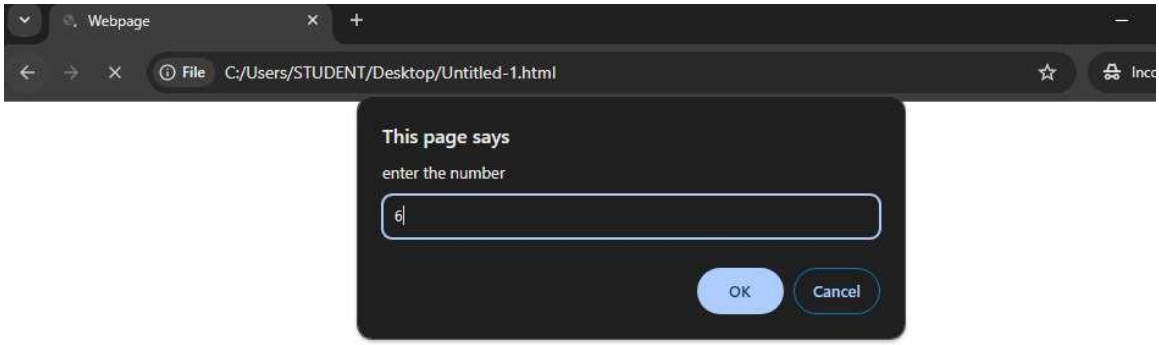


38.

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

```
<!DOCTYPE HTML>
<html>
  <head></head>
  <title>Webpage</title>
  <body>
    <script>
      let num=prompt("enter the number");
  let res=(num>=0?"POSITIVE NUMBER":"NEGATIVE NUMBER");
  console.log(res);
    </Script>
    </body>
    </html>
```

OUTPUT:

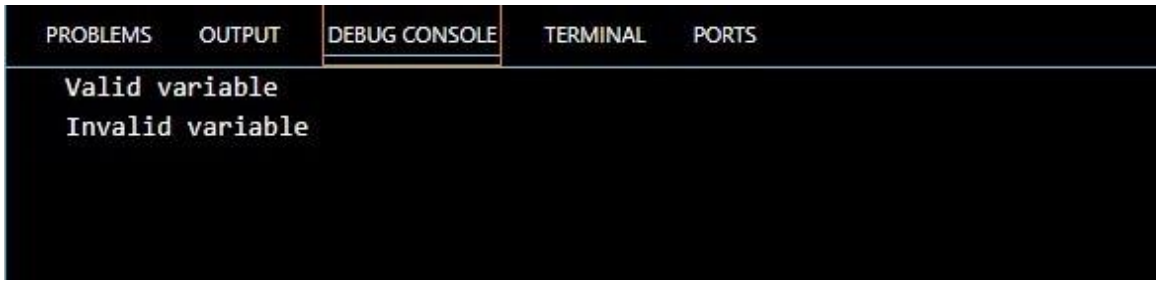


39.

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

```
<!DOCTYPE HTML>
<html>
   <head></head>
   <title>Webpage</title>
   <body>
      <script>
         let str="shreya";
         let str1;
   let res=(str?"Valid variable":"Invalid variable");
   let res1=(str1?"valid variable":"Invalid variable");
   console.log(res);
   console.log(res1);
      </Script>
      </body>
      </html>
```

OUTPUT:

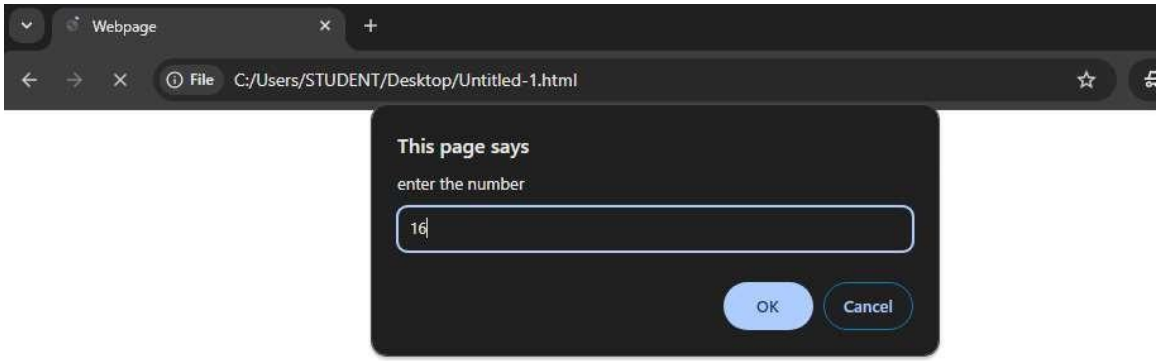


40.

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

```
<!DOCTYPE HTML>
<html>
  <head></head>
  <title>Webpage</title>
  <body>
    <script>
       let num=prompt("enter the number");
      let age=(num>=19?"ADULT AGE":"TEEN AGE");
      console.log(age);
    </Script>
    </body>
    </html>
```

OUTPUT:



41.

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

```
<!DOCTYPE HTML>
<html>
   <head></head>
   <title>Webpage</title>
   <body>
      <script>
        let  a=true;
        let  b=true;
        let  c=false;
        let d=false;
        let num=(a&&b)
        console.log(num);
        let res=(a||c)
        console.log(res);
        let res1=(a!=c)
        console.log(res1);
        let res2=(a&&c)
        console.log(res2);
        let res3=(c||d)
        console.log(res3);

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

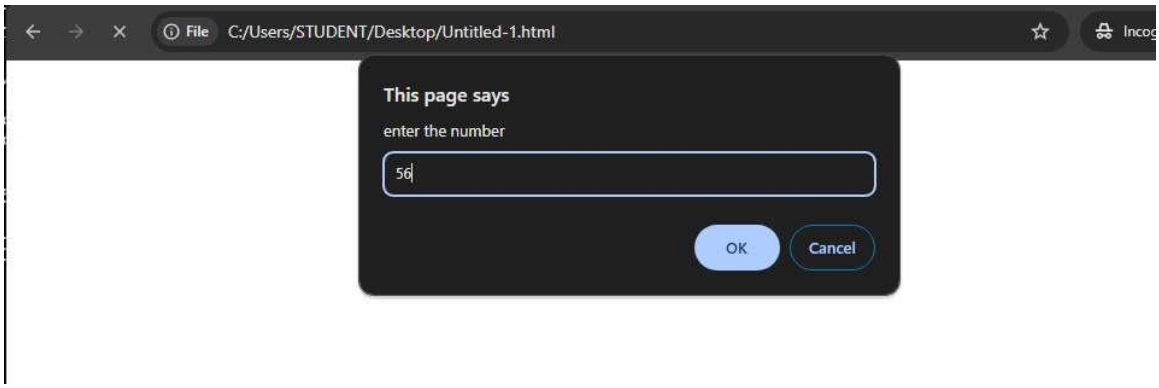
OUTPUT:

42.

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

```
<!DOCTYPE HTML>
<html>
  <head></head>
  <title>Webpage</title>
  <body>
    <script>

      let sel=prompt("enter the number");
       let res=Number(sel);
       if(isNaN()){
      if(sel>=0&&sel<=100){
       console.log("given number is within the range");
      }
      else {
       console.log("the number is not within the given range");
      }
    }
    else{
       console.log("enter the invalid number");
    }
    </Script>
    </body>
    </html>
```

OUTPUT:



43.

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

```
<!DOCTYPE HTML>
<html>
   <head></head>
   <title>Webpage</title>
   <body>
      <script>
        let bool=true;
        console.log(!bool);
        let bool2=false;
        console.log(!bool2);
      </Script>
      </body>
      </html>
```

OUTPUT:

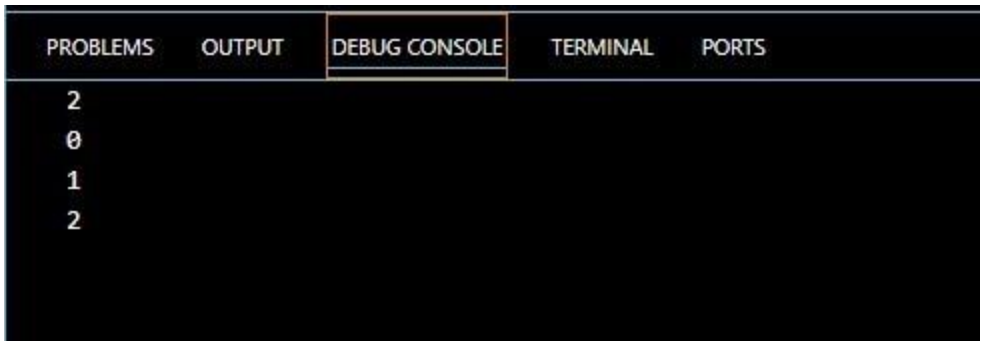
44.

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

```
<!DOCTYPE HTML>
<html>
  <head></head>
  <title>Webpage</title>
  <body>
    <script>
      let num1=1;
      let num2=2;
      let num3=0;
      let res=num1&&num2;
      console.log(res);
      let res1=num3&&num1;
      console.log(res1);
      let res2=num1||num2;
      console.log(res2);
      let res3=num3||num2;
      console.log(res3);
    </Script>
    </body>
    </html>
```

- **If the first operand is false, the second operand is not evaluated(for &&)**
- **If the first operand is true, the second operand is not evaluated(For ||)**

OUTPUT:



45.

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

```
<!DOCTYPE HTML>
<html>
   <head></head>
   <title>Webpage</title>
   <body>
      <script>
       let a=10;
       let b='10';
       let c=0;
      let res1=c&&b;
      console.log(res1);
      let res2=a||b;
      console.log(res2);
      </Script>
      </body>
      </html>
```

OUTPUT:

46.

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

```
<!DOCTYPE HTML>
<html>
   <head></head>
   <title>Webpage</title>
   <body>
      <script>
      function fun(num1,num2){
         return num1+num2;

      }

     let res=fun(10,20);
    console.log(res);
      </script>
      </body>
      </html>
```

OUTPUT:

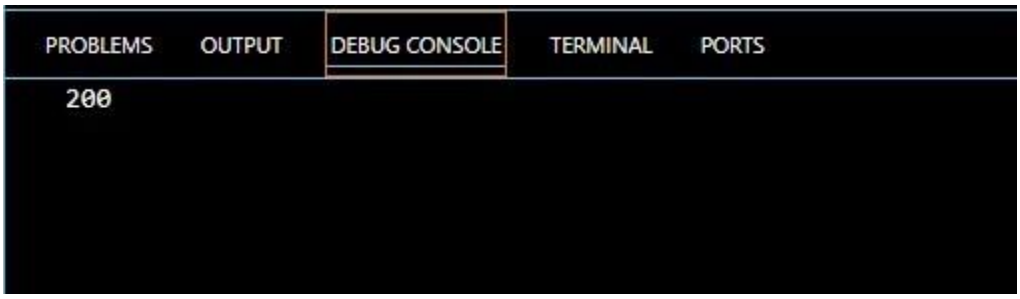
47.

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

```
<!DOCTYPE HTML>
<html>
   <head></head>
   <title>Webpage</title>
   <body>
```

```
    <script>
    function fun(length,width){
       return length*width;

    }
  let x=10;
  let y=20;
    let res=fun(x,y);
   console.log(res);
    </script>
    </body>
    </html>
```

OUTPUT:



48.

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

```
<!DOCTYPE HTML>
<html>
  <head></head>
  <title>Webpage</title>
  <body>
    <script>
    function fun(){
       console.log("hello,good  morning");
    }
    fun();
    </script>
    </body>
    </html>
```

OUTPUT:

49.

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

```
<!DOCTYPE HTML>
<html>
   <head></head>
   <title>Webpage</title>
   <body>
      <script>
      function fun(){

      }
     let res=fun();
     console.log(res);
      </script>
      </body>
      </html>
```

OUTPUT:

50.

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

```
<!DOCTYPE HTML>
<html>
```

```
<head></head>
<title>Webpage</title>
<body>
   <script>
   function fun(name="dename",age=14){
      console.log(name+" "+age);
   }
  fun("shreya",18);
  fun("dheresh",30);
  fun("naren",20);
  fun("riya",10);
 fun();
  </script>
  </body>
  </html>
```

OUTPUT: