NAME:RUPASRI A(717823E245)

MERN STACK TRAINING TASKS

TASK 1:

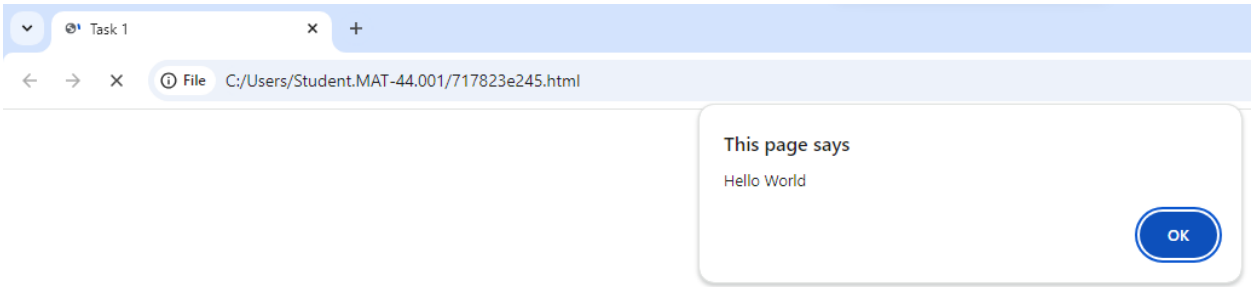
Write a simple script that displays “Hello, World!” on the web page using

an alert box.

CODE:

```
<!DOCTYPE html>

<html>

        <title> Task 1</title>

    <body>

        <script>

            alert("Hello World");

        </script>

    </body>

</html>
```

OUTPUT:



TASK 2:

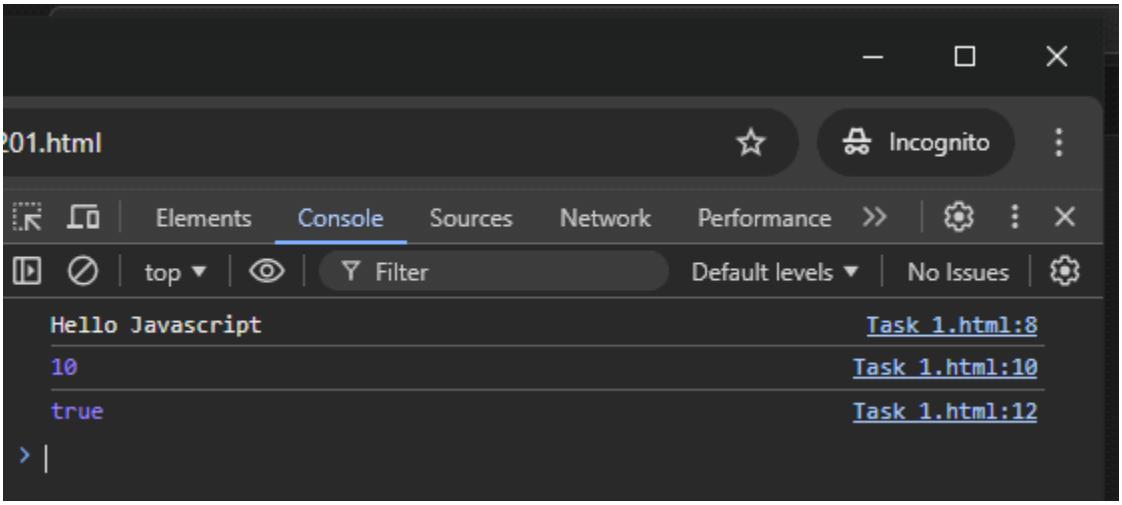
Experiment with different data types in JavaScript (e.g., string, number,boolean) by declaring and logging them in the console.

CODE:

```
<!DOCTYPE html>

<html>
```

```
        <title> Task 2</title>
    <body>
        <script>
            let Str="Hello Javascript";
            console.log(Str);
            let Num= 10;
            console.log(Num);
            let Bool= true;
            console.log(Bool);
        </script>
    </body>
</html>
```

OUTPUT:



TASK 3:

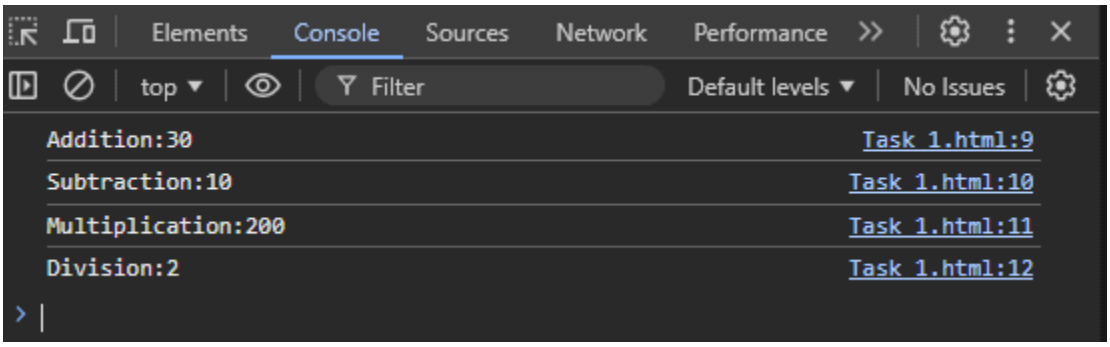
Use the console to perform basic math operations like addition, subtraction, multiplication, and division.

CODE:

```
<!DOCTYPE html>
```

```
<html>
        <title> Task 3</title>
    <body>
        <script>
            let a=20;
            let b=10;
            console.log("Addition:"+(a+b) );
            console.log("Subtraction:"+(a-b));
            console.log("Multiplication:"+(a*b));
            console.log("Division:"+(a/b));
        </script>
    </body>
</html>
```

OUTPUT:



TASK 4:

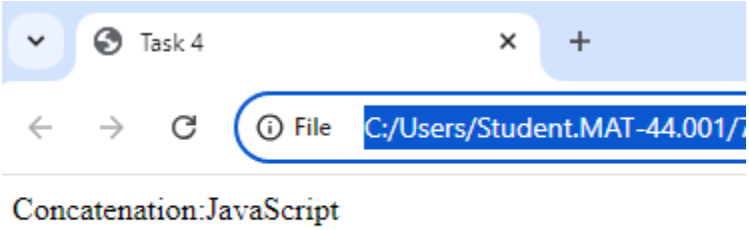
Declare two strings and concatenate them using the + operator

CODE:

```
<!DOCTYPE html>
<html>
        <title> Task 4</title>
```

```
    <body>
        <script>
            let str1="Java";
            let str2="Script";
            document.writeln("Concatenation:"+ (str1+str2));
        </script>
    </body>
</html>
```

OUTPUT:



TASK 5:

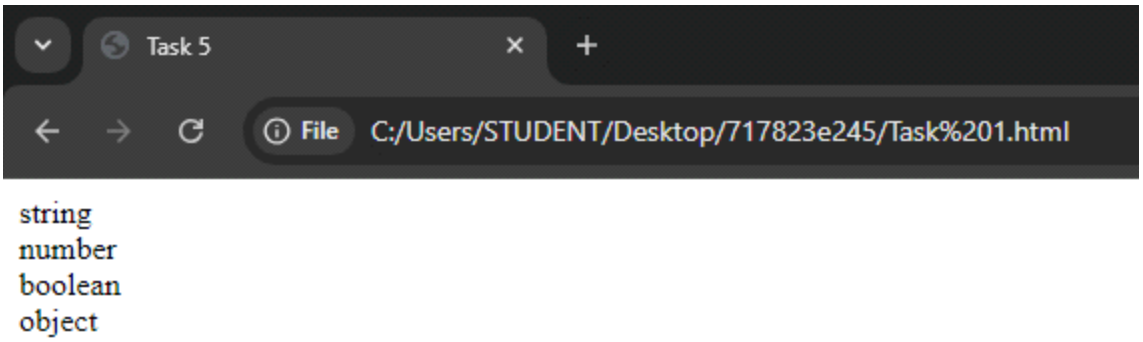
Use the typeof operator to check the data type of various variables.

CODE:

```
<!DOCTYPE html>
<html>

    <title> Task 5</title>
    <body>
        <script>
            let str="Java";
            let num=10;
```

```
            let bool=true;
            let arr=[1,2,3,4];
            document.writeln(typeof str+"<br>");
            document.writeln(typeof num+"<br>");
            document.writeln(typeof bool+"<br>");
            document.writeln(typeof arr+"<br>");
        </script>
    </body>
</html>
```

OUTPUT:



TASK 6:

Write a multi-line JavaScript comment and a single-line comment.

Explain the difference.

CODE and OUTPUT:

erminal Help Search

Task 1.html 1

C: > Users > STUDENT > Desktop > 717823e245 > Task 1.html > html

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

    <title> Task 6</title>
    <body>
        <script>
//single line command
let a = 5;
let b = 10;
let result = a + b;
/*console.log(result);
let x = 5
let y = 10
let sum = x + y
console.log(sum)
/*
        </script>
    </body>
</html>
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (e.g. text, !exclude, \esca

```
15
15
```

Single-Line Comment: Uses // to comment out the rest of the line.

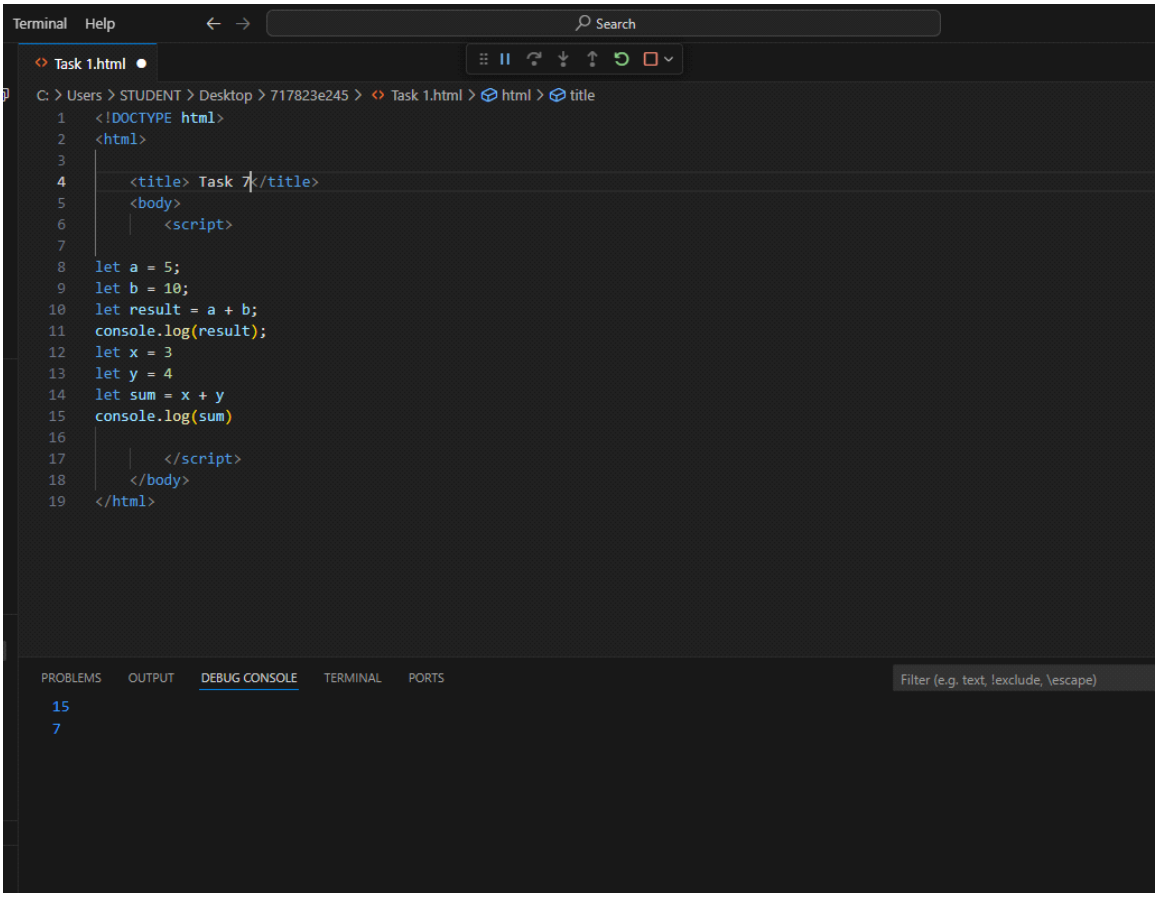
Multi-Line Comment: Uses /* to start and */ to end the comment, which allows for commenting over multiple line

TASK 7:

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

Note any differences in behavior

CODE and OUTPUT:



TASK 8:

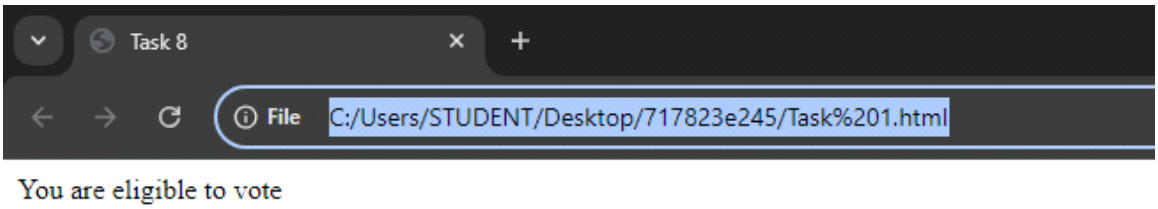
Use proper indentation to format a nested loop.

CODE:

```
<!DOCTYPE html>

<html>


    <title> Task 8</title>

    <body>

        <script>

let age=18;
```

```
        if(isNaN()){
            if(age>=18){
                document.writeln("You are eligible to vote");
            }
        else{
alert("You are not eligible");
        }}
        else{
document.writeln("Please enter a valid number");
        }


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

OUTPUT:



TASK 9:

Declare multiple variables in a single line.

CODE:

```
C: > Users > STUDENT > Desktop > 717823e245 > <> T
<!DOCTYPE html>
<html>

    <title> Task 9</title>
    <body>
        <script>
let a = 5,b=6,c=7;
let result = a+b+c;
console.log(result);
        </script>
    </body>
</html>
```

OUTPUT:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

18
```

TASK 10:

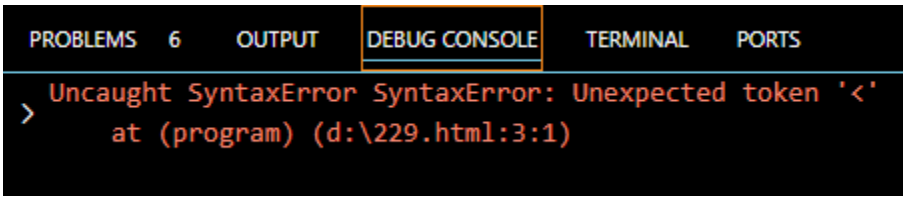
Place a script tag at the top and bottom of an HTML document. Note any differences in behavior.

CODE:

```
<!DOCTYPE html>
<script>
<html>
    <body>
        let a,b,c;
        a=20,b=23,c=24;
        document.writeln(a+"<br>"+b+"<br>"+c+"<br>");
```

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

OUTPUT:



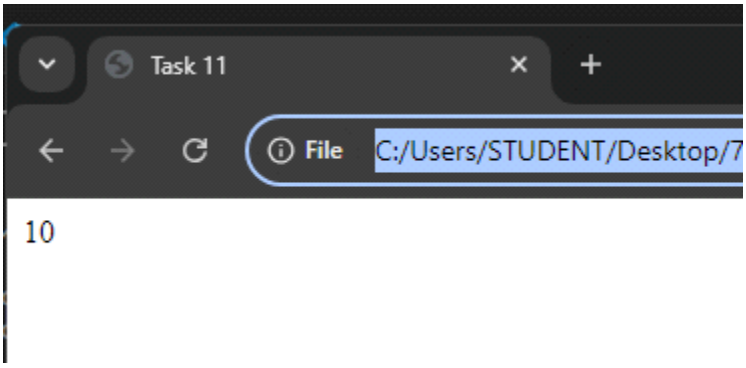
TASK 11:

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

CODE:

```
 <!DOCTYPE html>
<html>
        <title> Task 11</title>
        <body>
        <script>
a=10;
document.writeln(a);
                    </script>
            </body>
</html>
```

OUTPUT:



TASK 12:

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

CODE:

```
<!DOCTYPE html>

<html>

    <title> Task 12</title>

    <body>

            <script>

              "use strict";

a=10;

document.writeln(a);

            </script>

        </body>

</html>
```

OUTPUT:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

> Uncaught ReferenceError ReferenceError: a is not defined
    at <anonymous> (c:\Users\STUDENT\Desktop\717823e245\Task 1.html:7:2)
```

TASK 13:

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

parameter.

CODE:

```
<!DOCTYPE html>

<html>

    <title> Task 13</title>
```

```
        <body>
                <script>
                  "use strict";
let fun={
        age:"18",
        birth:"11052006",
};
document.writeln(fun.delete(age));
document.writeln(fun.delete(birth));
                </script>
            </body>
</html>
```

OUTPUT:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
> Uncaught ReferenceError ReferenceError: age is not defined
      at <anonymous> (c:\Users\STUDENT\Desktop\717823e245\Task 1.html:11:29)
```

TASK 14:

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

CODE:

//with "use strict"

```
<!DOCTYPE html>
<html>
        <title> Task 14</title>
        <body>
                <script>
                  "use strict";
```

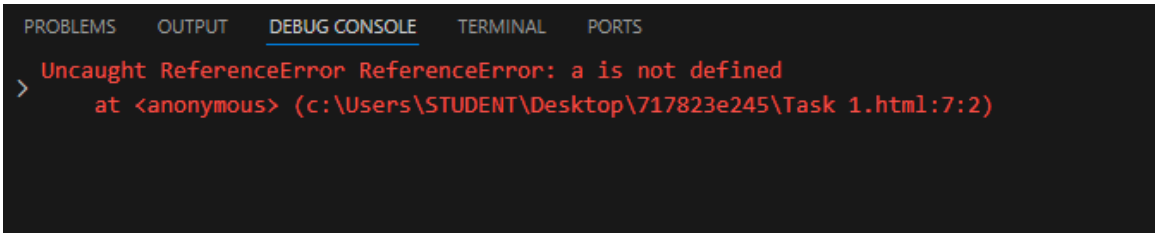
```
  a=10;

document.writeln(a);

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

//without "use strict"

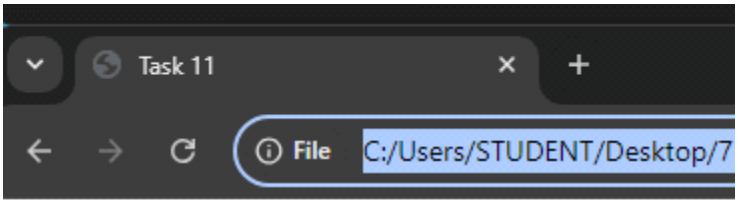
```
<!DOCTYPE html>

<html>

        <title> Task 14</title>

        <body>

                    <script>

  a=10;

document.writeln(a);

                    </script>

               </body>

</html>
```

OUTPUT:

//with "use strict"



//without "use strict"



10

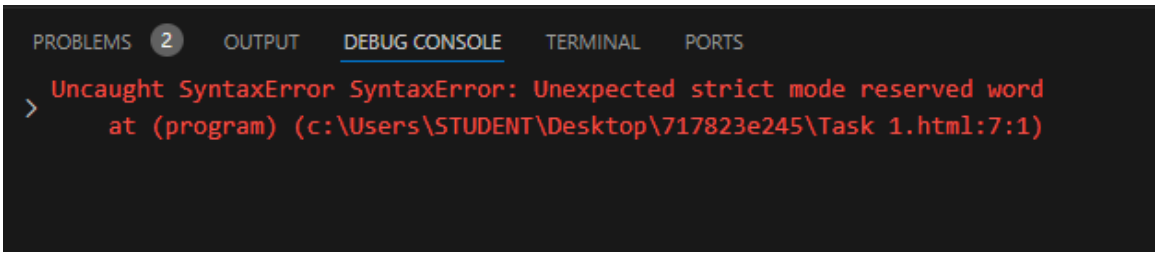
TASK 15:

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

CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 15</title>
    <body>
        <script>
          "use strict";
let private a=10;
let public b=20;
document.writeln(a);
document.writeln(b);
        </script>
    </body>
</html>
```

OUTPUT:



TASK 16:

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

CODE:

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

```
        <title> Task 16</title>
        <body>
                <script>
let age = 25;
age = 30;
console.log(age);
var name = "Alice";
  var name = "Bob";
console.log(name);
const pi = 3.14159;
console.log(pi);
                </script>
            </body>
</html>
```

OUTPUT:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
  30
  Bob
  3.14159
```

- In "let", Redeclaration of same variables is not allowed.But Reassigning is possible.
- In 'var', Redeclaration is possible.
- In 'const',Both Redeclaration and Reassigning is not possible.

TASK 17:

Attempt to reassign a const variable and observe the result.

CODE:

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

```
    <title> Task 17</title>
    <body>
        <script>
const pi = 3.14;
pi=3.24;
console.log(pi);
        </script>
    </body>
</html>
```

OUTPUT:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
> Uncaught TypeError TypeError: Assignment to constant variable.
    at <anonymous> (c:\Users\STUDENT\Desktop\717823e245\Task 1.html:7:3)
```

TASK 18:

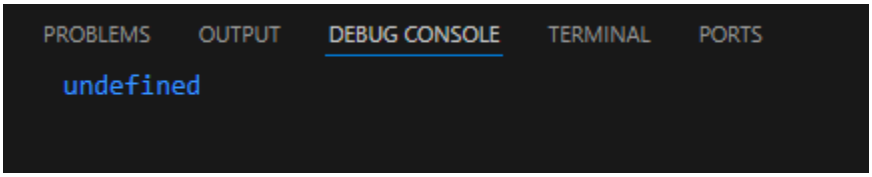
Declare a variable without initializing it and print its value.

CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 18</title>
    <body>
        <script>
let a;
console.log(a);

        </script>
    </body>
```

```
</html>
```

OUTPUT:



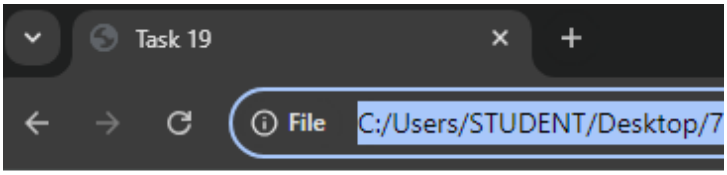
TASK 19:

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

CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 19</title>
    <body>

        <script>

      let str="Rupa";

        let num=10;

        let bool=true;

        document.writeln(typeof str+"<br>");

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

        document.writeln(typeof bool+"<br>");
      </script>
    </body>
</html>
```

OUTPUT:



string
number
boolean

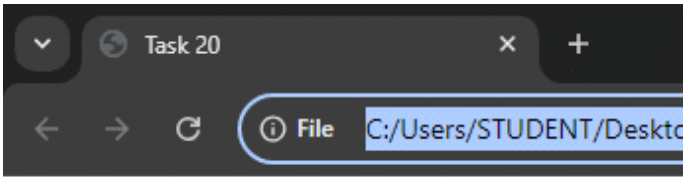
TASK 20:

Rename a variable and observe the outcome.

CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 20</title>
    <body>
            <script>
        let a=10;
        b=a;
        document.writeln(b);
          </script>
      </body>
</html>
```

OUTPUT:



10

TASK 21:

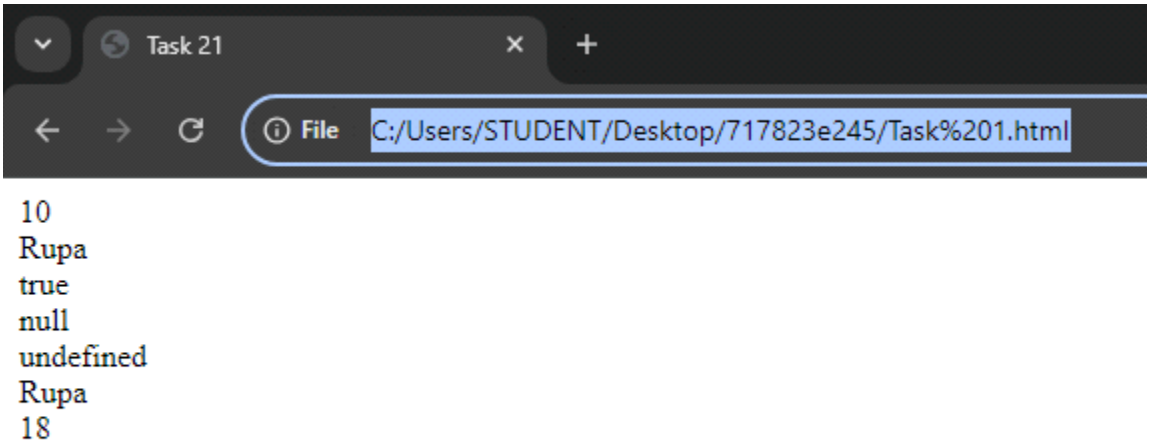
Create variables of different data types (e.g., string, number, boolean, null, undefined, object).

CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 21</title>
    <body>
            <script>
```

```
            let num=10;

            document.writeln(num+"<br>");

            let str="Rupa";

            document.writeln(str+"<br>");

            let bool=true;

            document.writeln(bool+"<br>");

            let a= null;

            document.writeln(a+"<br>");

            let address;

            document.writeln(address+"<br>");

            let obj={

                name:"Rupa",

                age:"18",

            };
            document.writeln(obj.name+"<br>");
            document.writeln(obj.age+"<br>");

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

OUTPUT:



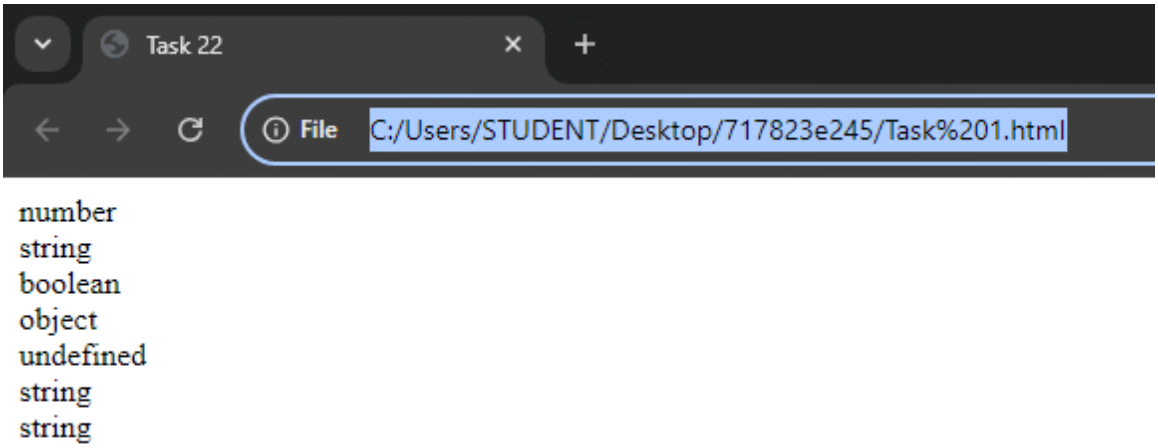
TASK 22:

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

CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 22</title>
    <body>
            <script>
        let num=10;
        document.writeln(typeof num+"<br>");
        let str="Rupa";
        document.writeln(typeof str+"<br>");
        let bool=true;
        document.writeln(typeof bool+"<br>");
        let a= null;
        document.writeln(typeof a+"<br>");
        let address;
        document.writeln(typeof address+"<br>");
        let obj={
            name:"Rupa",
            age:"18",
        };
        document.writeln(typeof obj.name+"<br>");
        document.writeln(typeof obj.age+"<br>");
          </script>
       </body>
</html>
```

OUTPUT:



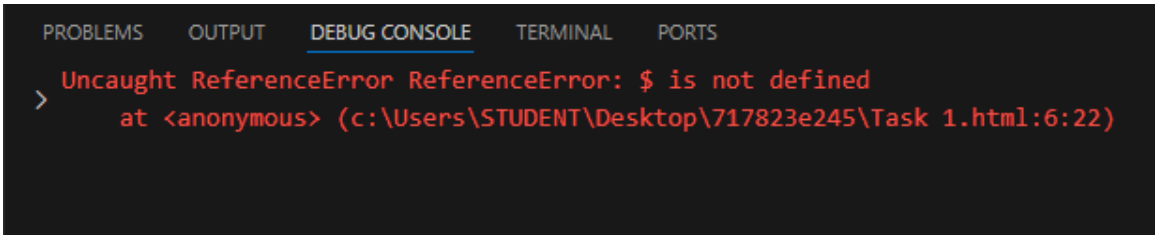
TASK 23:

Declare a symbol and print its type.

CODE:

```
<!DOCTYPE html>

<html>

    <title> Task 23</title>

    <body>

            <script>

          let symbol=$;

          document.writeln(symbol);

          document.writeln(typeof (symbol));

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

OUTPUT:



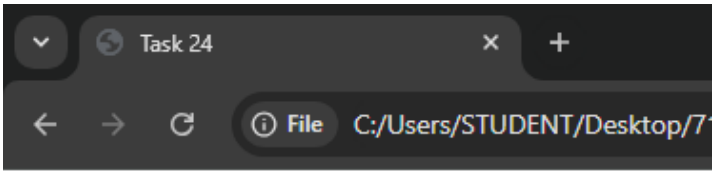
TASK 24:

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

CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 24</title>
    <body>
            <script>
          let a= null;
          document.writeln(a+"<br>");
          document.writeln(typeof a);
            </script>
        </body>
</html>
```

OUTPUT:



null
object

TASK 25:

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

CODE:

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

```
<title> Task 25</title>
<body>
<script>
var a=10;
if(a>5){
var b=20;
}
console.log(b);
let c=58;
if(c<100){
let d=90;
}
console.log(d);

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

OUTPUT:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
  20
  Uncaught ReferenceError ReferenceError: d is not defined
>     at <anonymous> (c:\Users\STUDENT\Desktop\717823e245\Task 1.html:15:13)
```

TASK 26:

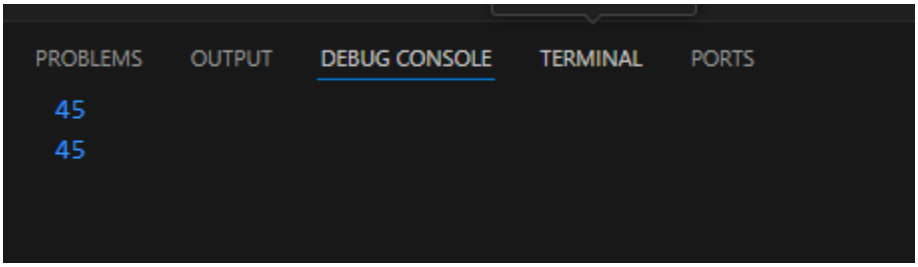
Convert a string to a number using both implicit and explicit conversion.

CODE:

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

```
        <title> Task 26</title>
        <body>
                <script>
let str1 = "45";
let result1 = str1 - 0;
console.log(result1);
let str2 = "45";
let result2 = Number(str2);
console.log(result2);
                </script>
        </body>
</html>
```

OUTPUT:



TASK 27:

Convert a boolean to a string and vice versa.

CODE:

```
<!DOCTYPE html>
<html>
        <title> Task 27</title>
        <body>
                <script>
let bool1 = true;
let str1 = String(bool1);
console.log(str1);
```

```
let str2 = "Rupa";
let bool2 = Boolean(str2);
console.log(bool2);
                </script>
        </body>
</html>
```

OUTPUT:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
  true
  true
```

TASK 28:

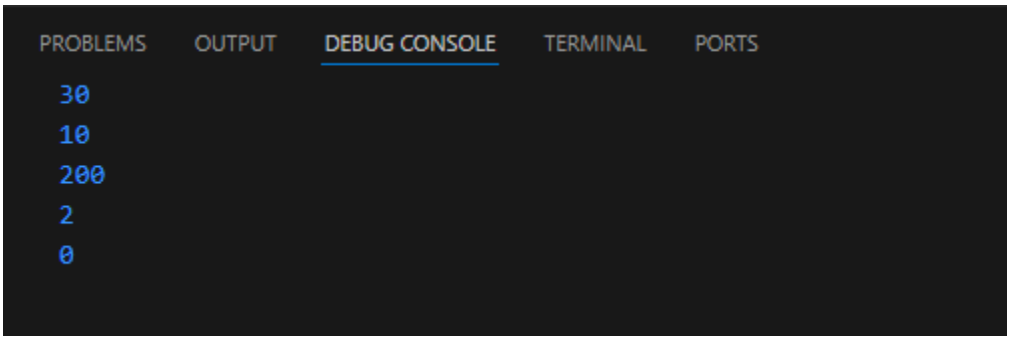
Practice basic arithmetic operators (+, -, *, /, %).

CODE:

```
<!DOCTYPE html>
<html>
        <title> Task 28</title>
        <body>
                <script>
  let calculator={
        sum(){
                return this.a+this.b;
        },
        sub(){
                return this.a-this.b;
        },
        mul(){
```

```
            return this.a*this.b;
        },
        div(){
            return this.a/this.b;
        },
        mod(){
            return this.a%this.b;
        },
        read(){
            this.a=+prompt('a?',0);
            this.b=+prompt('b?',0);
        },
    };
    calculator.read();
    console.log(calculator.sum());
    console.log(calculator.sub());
    console.log(calculator.mul());
    console.log(calculator.div());
    console.log(calculator.mod());
            </script>
        </body>
</html>
```

OUTPUT:



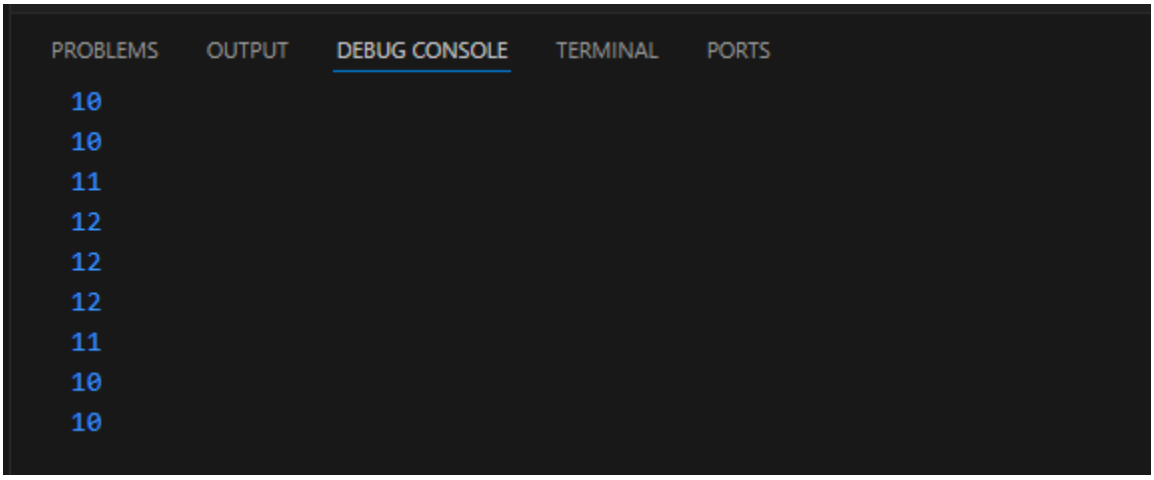
TASK 29:

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

CODE:

```
<!DOCTYPE html>

<html>

        <title> Task 29</title>

        <body>

                    <script>

  let a=10;

  console.log(a);

  console.log(a++);

  console.log(a);

  console.log(++a);

  console.log(a);

  console.log(a--);

  console.log(a);

  console.log(--a);

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

OUTPUT:



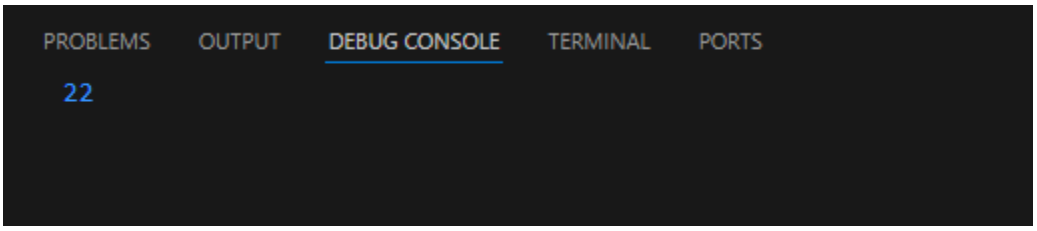
TASK 30:

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

CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 30</title>
    <body>
            <script>
 let a=10;
 let result=a++ + ++a;
 console.log(result);


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

OUTPUT:



TASK 31:

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

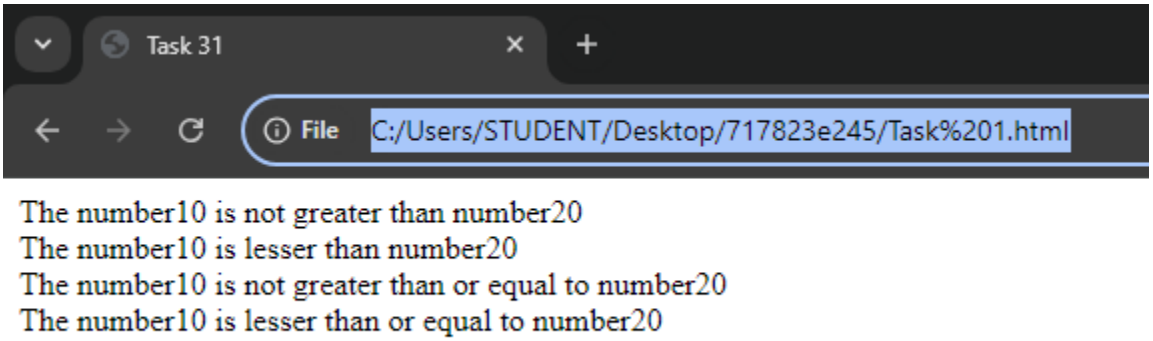
CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 31</title>
```

```
    <body>
            <script>
let a=10;
let b=20;
if(a>b){
    document.writeln("The number" +a+" is greater than number"+b +"<br>");
}
else{
    document.writeln("The number" +a+" is not greater than number"+b+"<br>");
}
if(a<b){
    document.writeln("The number" +a+" is lesser than number"+b+"<br>");
}
else{
    document.writeln("The number" +a+" is not lesser than number"+b+"<br>");
}
if(a>=b){
    document.writeln("The number" +a+" is greater than or equal to number"+b+"<br>");
}
else{
    document.writeln("The number" +a+" is not greater than or equal to number"+b+"<br>");
}
if(a<=b){
    document.writeln("The number" +a+" is lesser than or equal to number"+b+"<br>");
}
else{
    document.writeln("The number" +a+" is not less than or equal to number"+b+"<br>");
}
```

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

OUTPUT:



TASK 32:

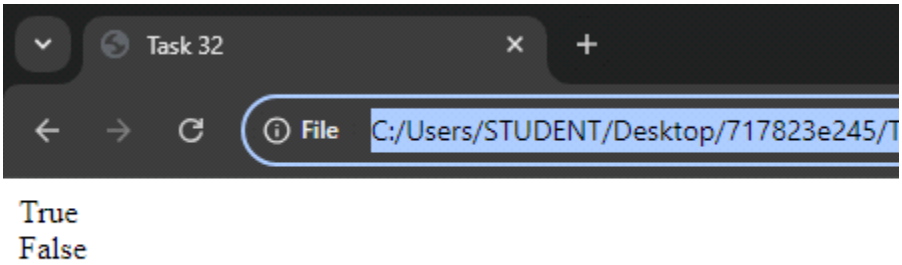
Use equality () and strict equality (=) operators to compare different data types and note the differences.

CODE:

```
<!DOCTYPE html>
<html>
        <title> Task 32</title>
        <body>
                <script>
                        let a=1;
                        let b='1';
                        if(a==b){
                                document.writeln("True"+"<br>");
                        }
                        else{
                                document.writeln("False"+"<br>");
                        }
```

```
                if(a===b){
                        document.writeln("True"+"<br>");
                }
                else{
                        document.writeln("False");
                }
        </script>
    </body>
</html>
```

OUTPUT:



TASK 33:

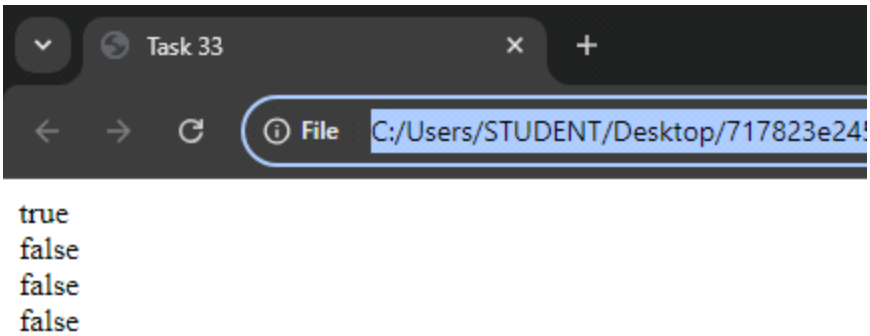
Compare two strings lexicographically.

CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 33</title>
    <body>
        <script>
                let a="Rupa";
                let b="Sri";
                let res=(a<b);
                let res1=(a>b);
```

```
                let res2=(a==b);
                let res3=(a===b);
                document.writeln(res+"<br>");
                document.writeln(res1+"<br>");
                document.writeln(res2+"<br>");
                document.writeln(res3+"<br>");
            </script>
    </body>
</html>
```

OUTPUT:



TASK 34:

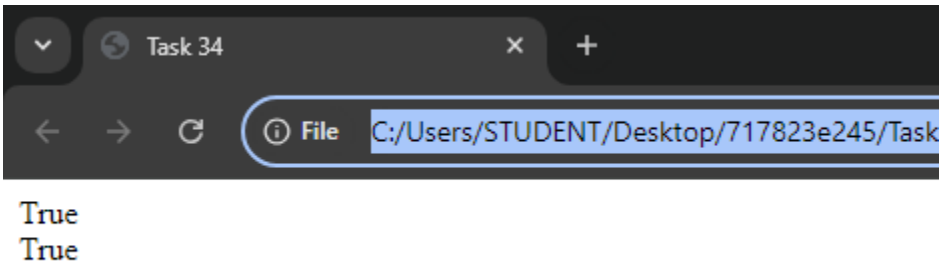
Use the inequality (!=) and strict inequality (!==) operators to compare values.

CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 34</title>
    <body>
        <script>
            let a=1;
            let b=2;
```

```
                if(a!=b){
                    document.writeln("True"+"<br>");
                }
                else{
                    document.writeln("False"+"<br>");
                }
                if(a!==b){
                    document.writeln("True"+"<br>");
                }
                else{
                    document.writeln("False"+"<br>");
                }
            </script>
    </body>
</html>
```

OUTPUT:



TASK 35:

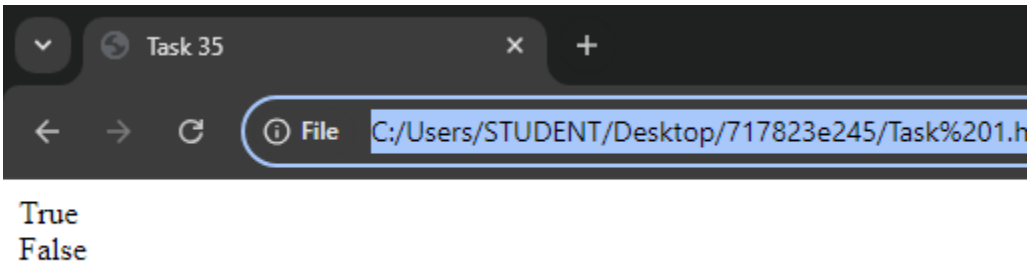
Compare null and undefined using both == and ===.

CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 35</title>
```

```
    <body>
        <script>
            let a=null;

            let b;

            if(a==b){

                document.writeln("True"+"<br>");

            }

            else{

                document.writeln("False"+"<br>");

            }

            if(a===b){

                document.writeln("True"+"<br>");

            }

            else{

                document.writeln("False"+"<br>");

            }

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

OUTPUT:



TASK 36:

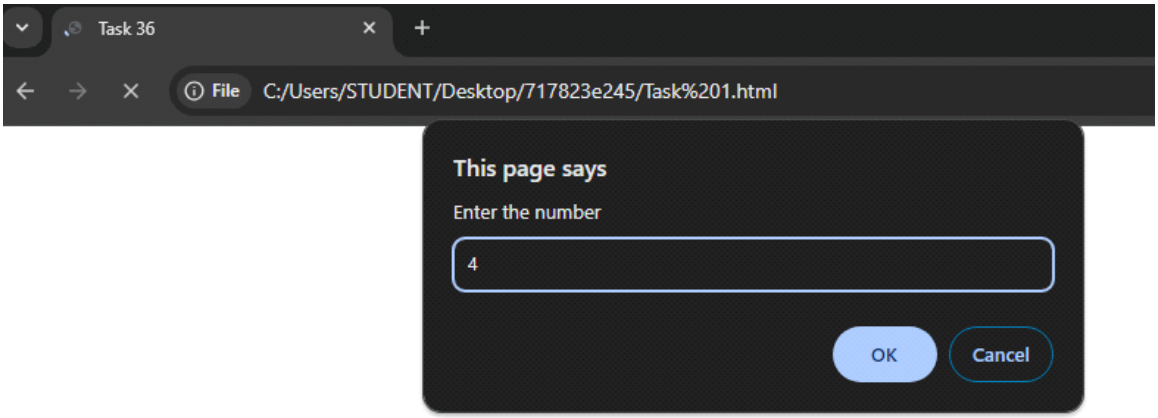
Write an if statement that checks if a number is even or odd.

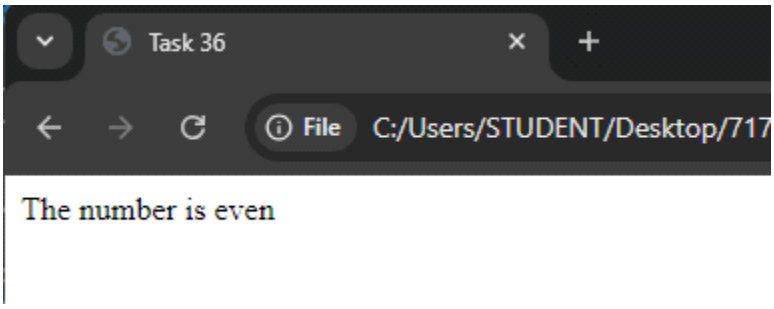
CODE:

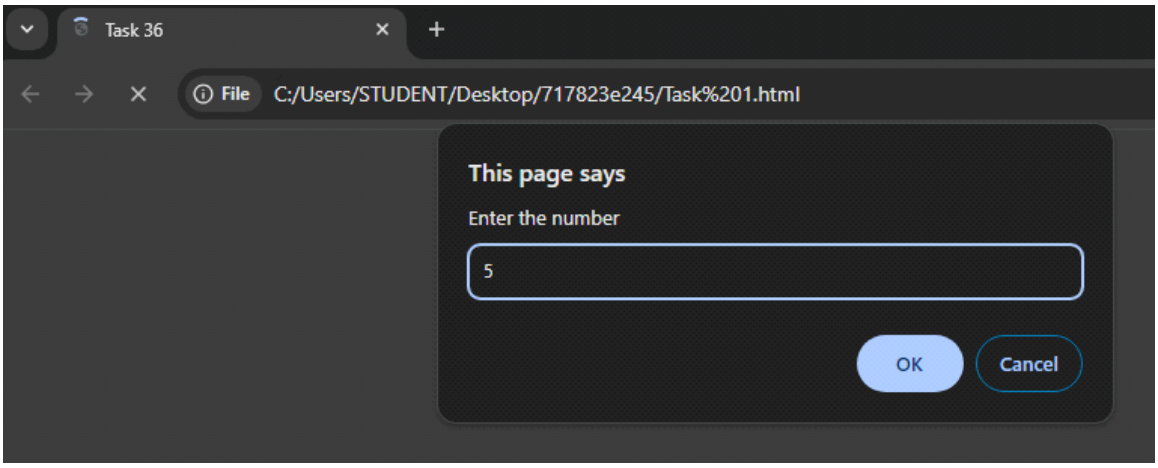
```
<!DOCTYPE html>
<html>
    <title> Task 36</title>
```

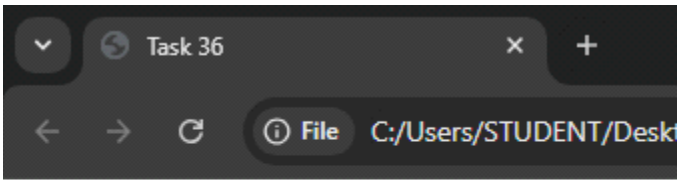
```
    <body>
        <script>
          let a=prompt("Enter the number",0);
                                    if(a%2==0){
                document.writeln("The number is even"+"<br>");
            }
            else{
                document.writeln("The number is odd"+"<br>");
            }
        </script>
    </body>
</html>
```

OUTPUT:









The number is odd

TASK 37:

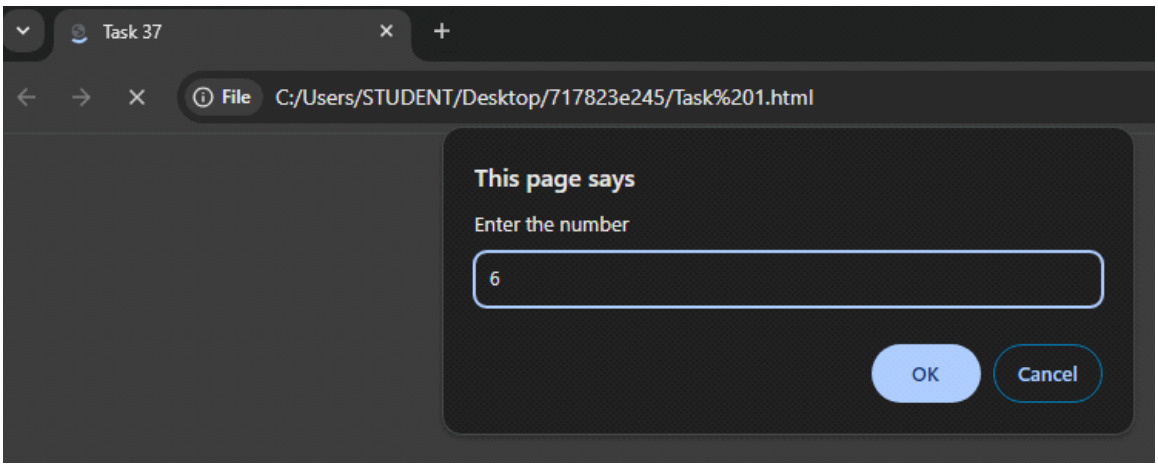
Use nested if statements to classify a number as negative, positive, or zero.

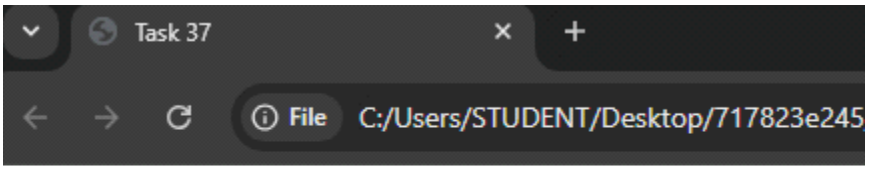
CODE:

```
<!DOCTYPE html>

<html>

    <title> Task 36</title>

    <body>

            <script>

              let a=prompt("Enter the number",0);

                                    if(a>0){

                        document.writeln("The number is positive"+"<br>");

                    }

                    else if(a<0){
```

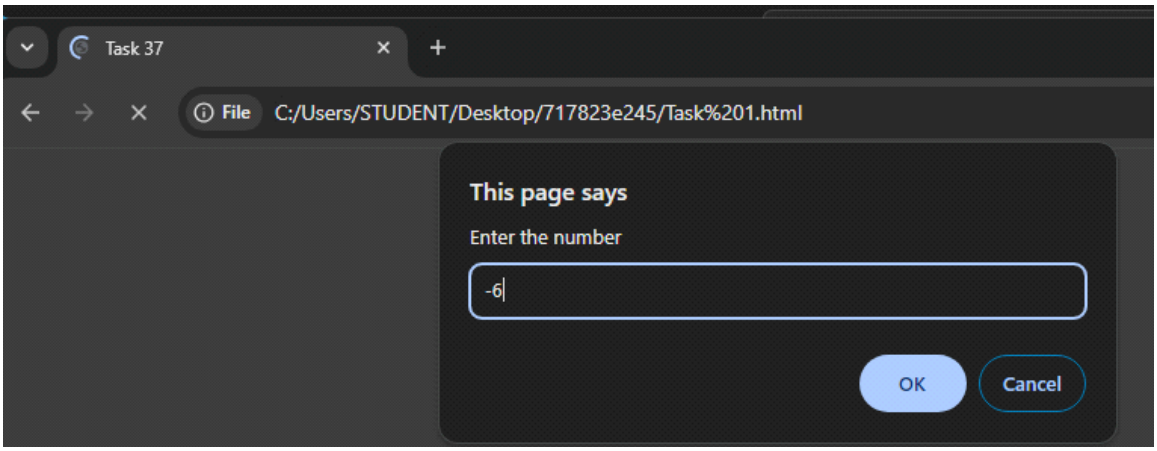
```
                    document.writeln("The number is negative"+"<br>");
                }
                else{
                    document.writeln("The number is zero"+"<br>");
                }
            </script>
    </body>
</html>
```

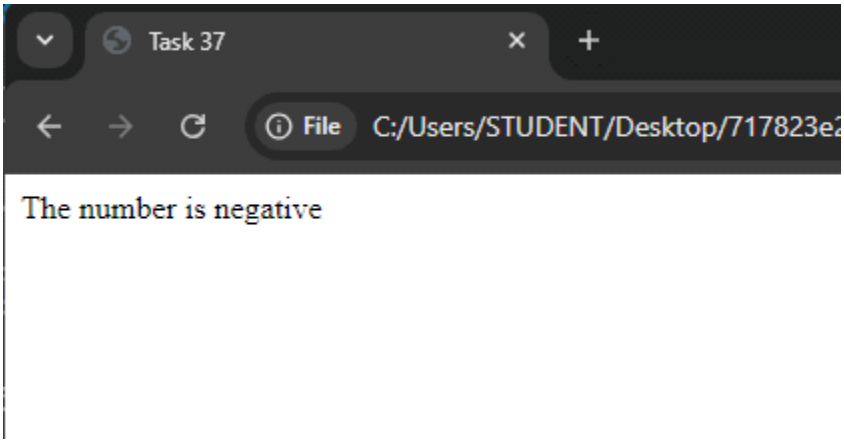
OUTPUT:





The number is positive





TASK 38:

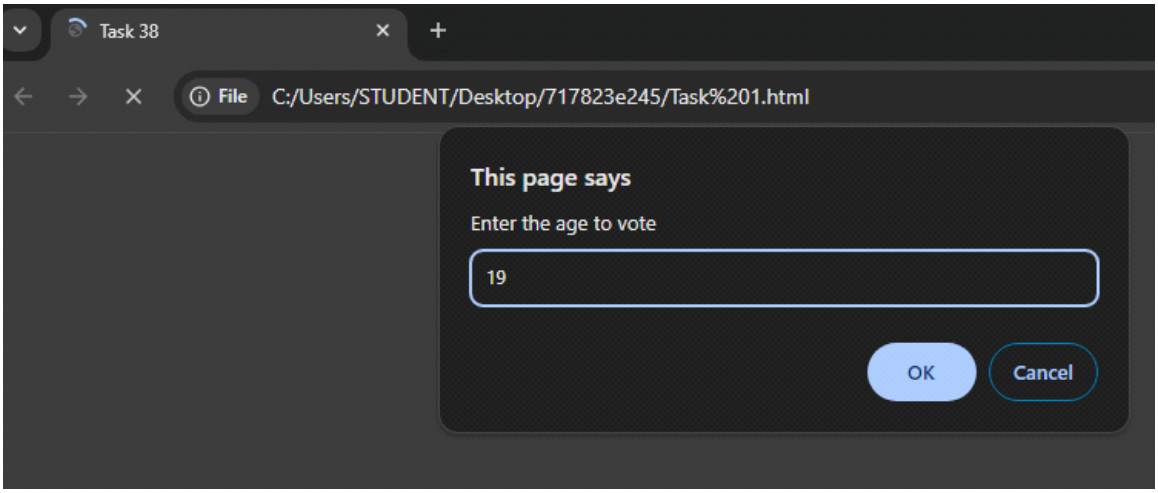
Use the conditional (ternary) operator '?' to rewrite a simple if…else statement.

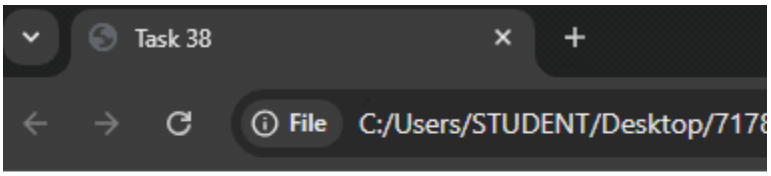
CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 38</title>
    <body>
            <script>
              let age=prompt("Enter the age to vote",0);
              let canVote = age >= 18 ? "Yes" : "No";
              document.writeln(canVote);
```

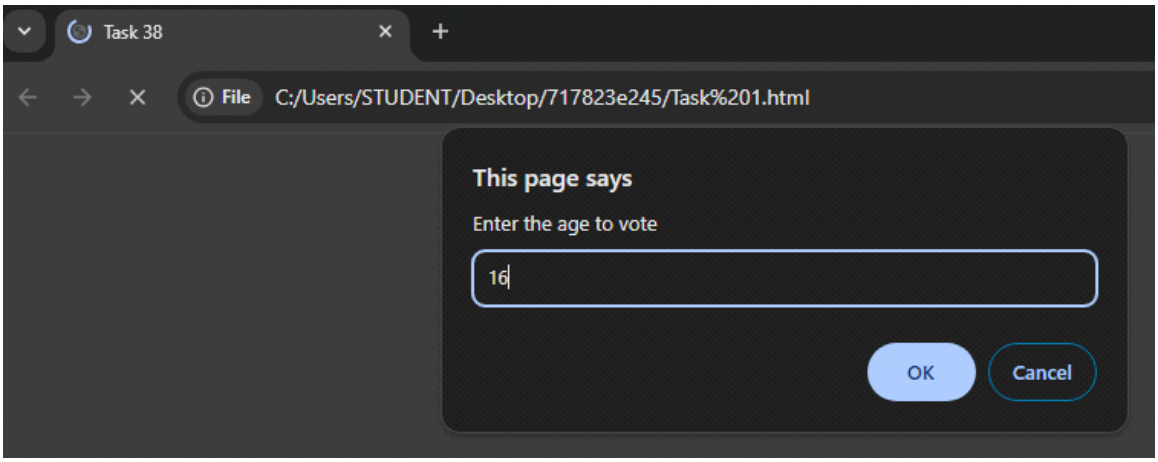
```
            </script>
        </body>
</html>
```

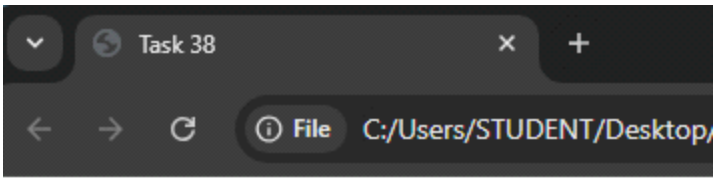
OUTPUT:





Yes





No

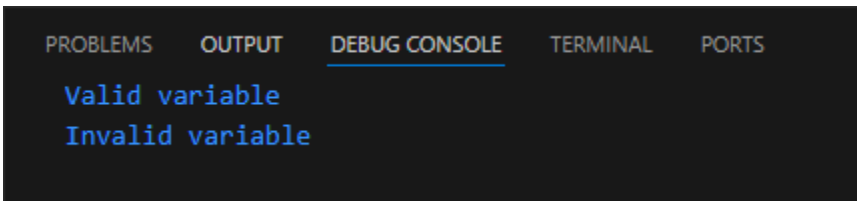
TASK 39:

Check the validity of a variable using the ? operator.

CODE:

```
<!DOCTYPE HTML>

<html>

<head></head>

<title>Task 39</title>

<body>

<script>

let str="Rupa";

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:



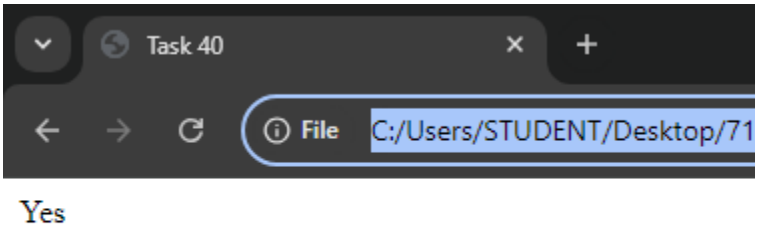
TASK 40:

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

CODE:

```
<!DOCTYPE html>

<html>

    <title> Task 40</title>

    <body>

            <script>

              let age=19;

              let canVote = age >= 18 ? "Yes" : "No";

              document.writeln(canVote);

            </script>

    </body>

</html>
```

OUTPUT:



TASK 41:

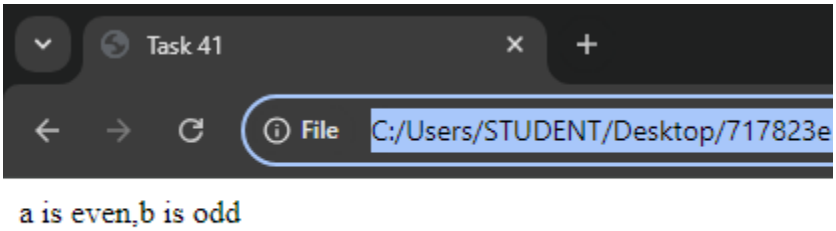
Evaluate various combinations of logical operators (&&, ||, !).

CODE:

```
<!DOCTYPE html>

<html>

    <title> Task 41</title>
```

```
<body>
        <script>
var a=46;
var b=57;
if(a!=b){
if(a%2==0 && b%2==0){
    document.writeln("Both are even ");
}
else if(a%2==0 || b%2==0){
  if(a%2==0){
    document.writeln("a is even,b is odd");
  }
  else{
        document.writeln("b is even,a is odd");
  }
}
else{
    document.writeln("Both are odd");
}}
     </script>
</body>
</html>
```

OUTPUT:



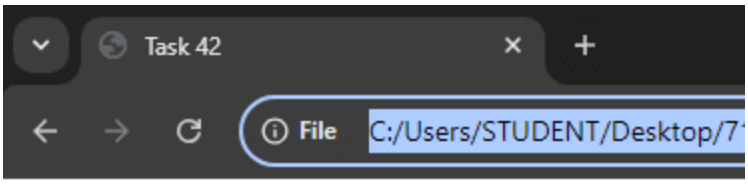
TASK 42:

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

CODE:

```
<!DOCTYPE html>

<html>

    <title> Task 42</title>

    <body>

            <script>

        let a=120;

        if(a>=0 && a<=100){

            document.writeln("The number lies between 0-100");

        }

        else if(a>100 && a<=200){

            document.writeln("The number lies between 101-200");

        }

        else{

            document.writeln("The number exceeds the limit");

        }

                    </script>

    </body>

</html>
```

OUTPUT:



The number lies between 101-200

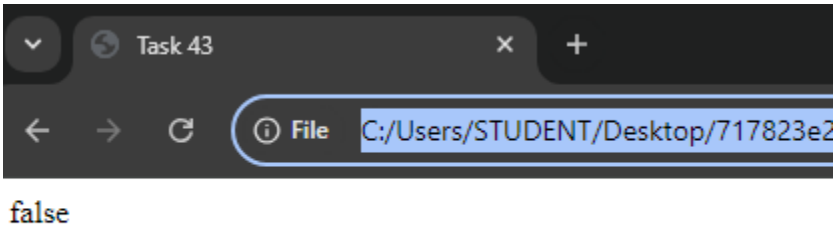
TASK 43:

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

CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 43</title>
    <body>
        <script>
        let a=true;
document.writeln(!a);
        </script>
    </body>
</html>
```

OUTPUT:



TASK 44:

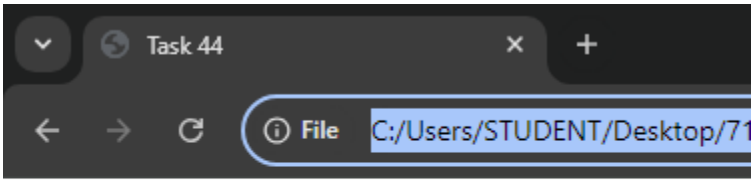
Evaluate the short-circuiting nature of logical operators.

CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 44</title>
    <body>
        <script>
        let a=2;
```

```
document.writeln(a>5 && a<10);

document.writeln("<br>");

  document.writeln(a<5 || a>5);

      document.writeln("<br>");

      document.writeln(!a);

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

OUTPUT:



false
true
false

TASK 45:

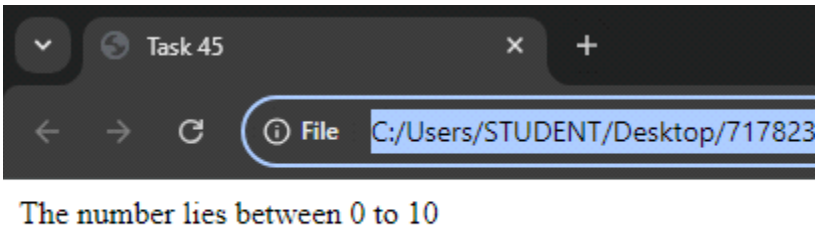
Compare two non-boolean values using logical operators and observe the result.

CODE:

```
<!DOCTYPE html>

<html>

        <title> Task 45</title>

        <body>

                    <script>

                    let a=2;

                    if(a>0 && a<10){

document.writeln("The number lies between 0 to 10");

document.writeln("<br>");
```

```
                    }
else if(a>10 || a==20){
  document.writeln("The number lies between 10 to 20");
      document.writeln("<br>");
}
else{
      document.writeln("The number is greater than 20");
}
                  </script>
        </body>
</html>
```

OUTPUT:



TASK 46:

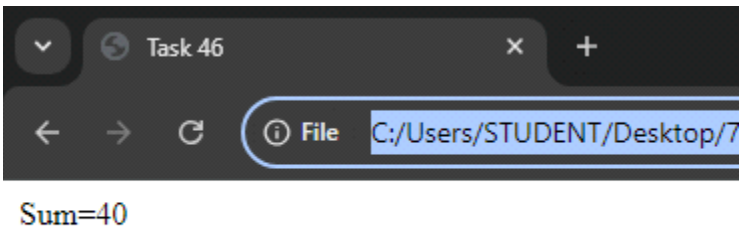
Write a function that takes two numbers as arguments and returns their sum.

CODE:

```
<!DOCTYPE html>
<html>
        <title> Task 46</title>
        <body>
                    <script>
```

```
            function sum(num1,num2){
            let sum=num1+num2;
            document.writeln("Sum="+ sum);
            }
            sum(20,20);
          </script>
    </body>
</html>
```

OUTPUT:



TASK 47:

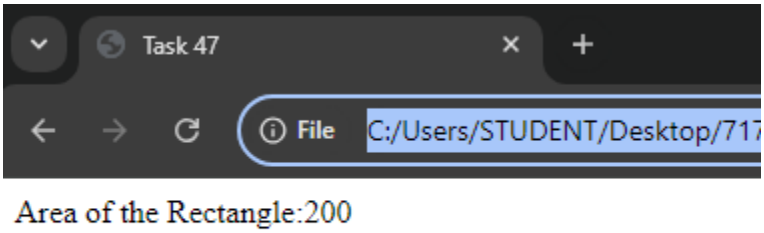
Create a function that calculates the area of a rectangle.

CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 47</title>
    <body>
            <script>
              function areaofRectangle(num1,num2){
                  let area=num1*num2;
                  document.writeln("Area of the Rectangle:"+area);
              }
          areaofRectangle(10,20);
```

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

OUTPUT:



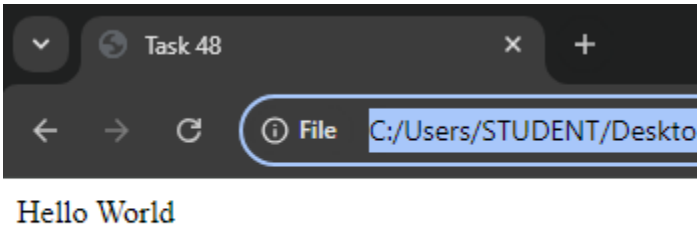
TASK 48:

Declare a function without parameters and call it.

CODE:

```
<!DOCTYPE html>
<html>
    <title> Task 48</title>
    <body>

            <script>

              function fun(){

                  document.writeln("Hello World");

              }

          fun();
            </script>
    </body>
</html>
```

OUTPUT:



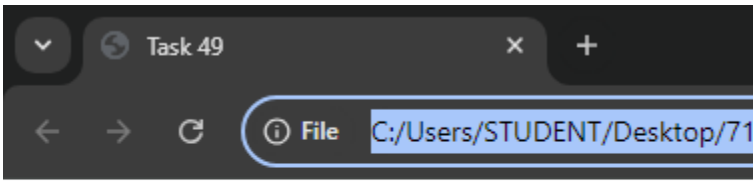
TASK 49:

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

CODE:

```
<!DOCTYPE html>

<html>

    <title> Task 49</title>

    <body>

            <script>

              function fun(){


              }

          document.writeln(fun());


           </script>

    </body>

</html>
```

OUTPUT:



undefined

TASK 50:

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

CODE:

```
<!DOCTYPE html>

<html>
```

```
<title> Task 50</title>
<body>
<script>
function fun(name = "Rupa", registerNum = "45") {
console.log(`${registerNum}, ${name}!`);
}
fun();
fun("Sri");
fun("Anu", "50");
</script>
</body>
</html>
```

OUTPUT:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
45, Rupa!
45, Sri!
50, Anu!
```

TASK 51:

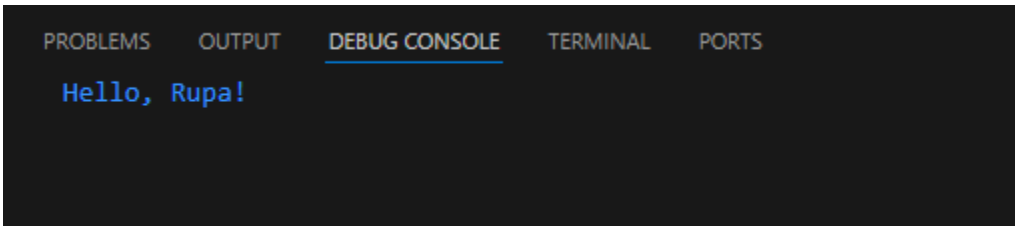
Declare a simple arrow function named greet that takes one parameter name and returns the string "Hello, name!". Test your function with various names.

CODE:

```
<!DOCTYPE html>
<html>
<title>Task 51</title>
<body>
<script>
```

```
const greet = (name) => {
        return "Hello, "+`${name}!`;
    }
    console.log(greet("Rupa"));
    </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.

CODE:

```
<!DOCTYPE html>
<html>
    <title>Task 51</title>
    <body>
        <script>
  const sum = (a,b) => a+b;
        console.log(sum(10,20));
        console.log(sum(35,45));
        console.log(sum(67,29));
        console.log(sum(88,12));
        </script>
```

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

OUTPUT:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
 30
 80
 96
 100
```

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.

CODE:

```
<!DOCTYPE html>
<html>
    <title>Task 53</title>
    <body>
        <script>
      const isEven = (a) => a%2==0;
        console.log(isEven(22));
        console.log(isEven(68));
        console.log(isEven(57));
        console.log(isEven(19));
        </script>
            </body>
</html>
```

OUTPUT:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
 true
 true
 false
 false
```

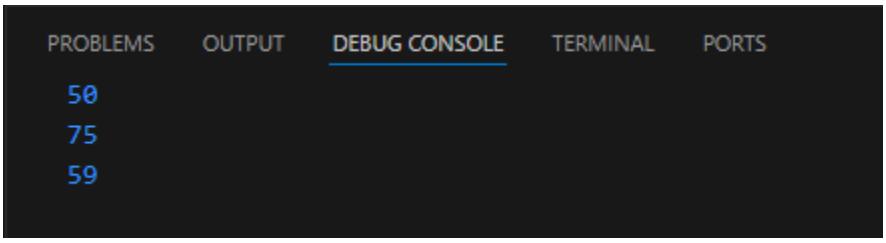
TASK 54:

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.

CODE:

```
<!DOCTYPE html>

<html>

    <title>Task 53</title>

    <body>

        <script>

      const maxValue= (a,b) => {

return a>b?a:b;

      };

        console.log(maxValue(20,50));

        console.log(maxValue(75,13));

        console.log(maxValue(31,59));

        </script>

            </body>
</html>
```

OUTPUT:



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.

CODE:

```
<!DOCTYPE html>
<html>
    <title>Task 53</title>
    <body>
        <script>
  const myObject = {
    value:10,
    multiplyTraditional:function (num) {
        console.log("Traditional function this:", this);
        return this.value*num;
    },
    multiplyArrow:(num) => {
        console.log("Arrow function this:",this);
        return this.value*num;
    }
};
console.log(myObject.multiplyTraditional(5));
console.log(myObject.multiplyArrow(5));
        </script>
            </body>
</html>
```

OUTPUT:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                    Filter (e
> Traditional function this: {value: 10, multiplyTraditional: ƒ, multiplyArrow: ƒ}
  50
> Arrow function this: Window {window: Window, self: Window, document: #document, name: '', location: Location, …}
  NaN
```