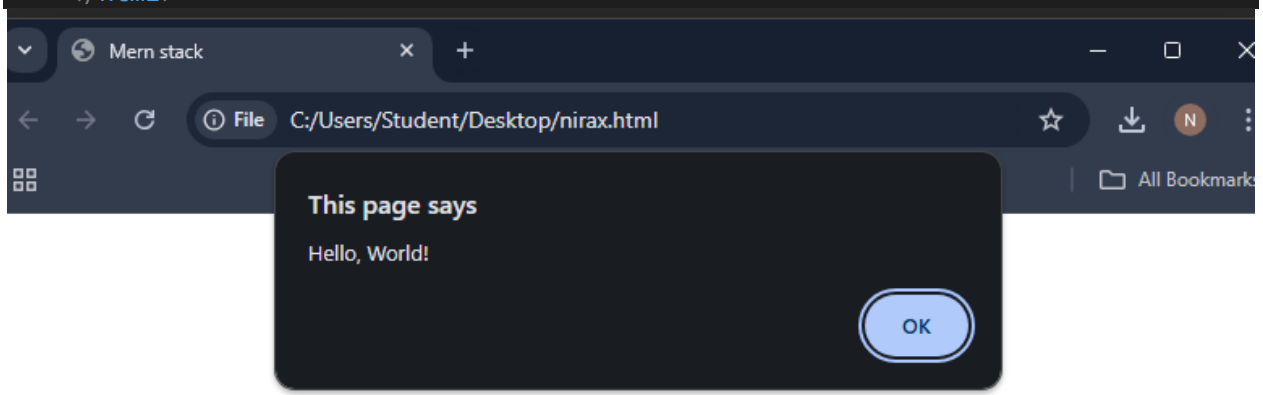
1.An Introduction to JavaScript

Task 1:

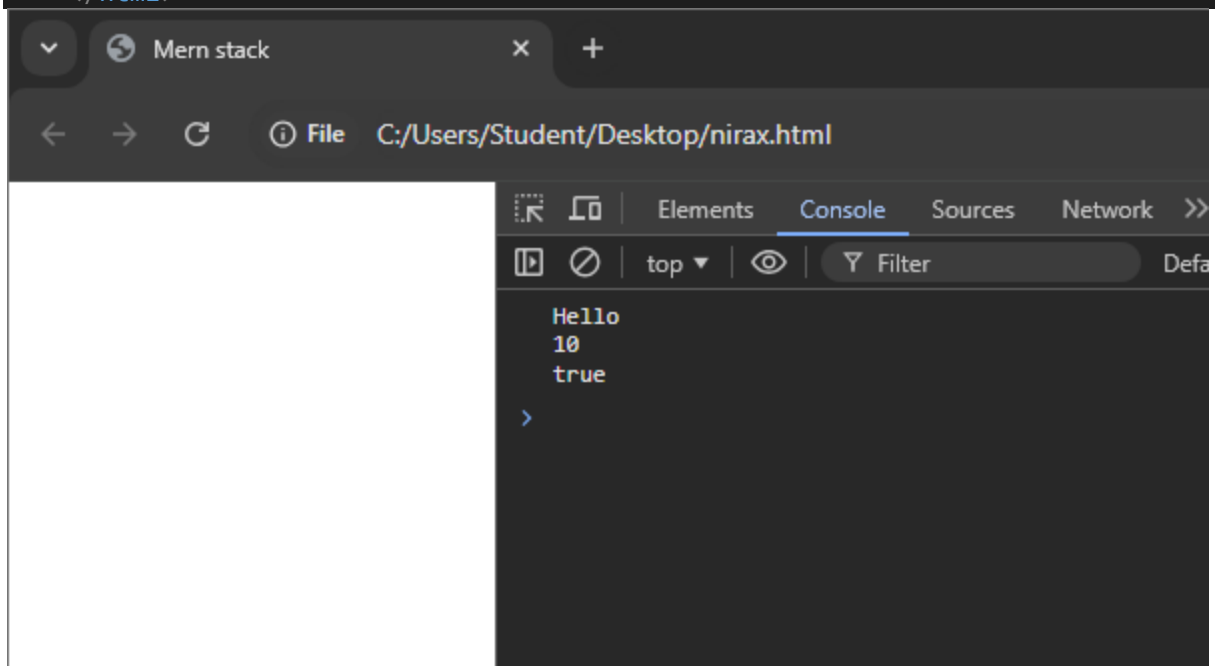
```
<DOCTYPE! html>
    <html>
        <head>
            <title>
                Mern stack
            </title>
        </head>
        <body>
            <script>
                alert("Hello, World!");
            </script>
        </body>
    </html>
```



Task 2:

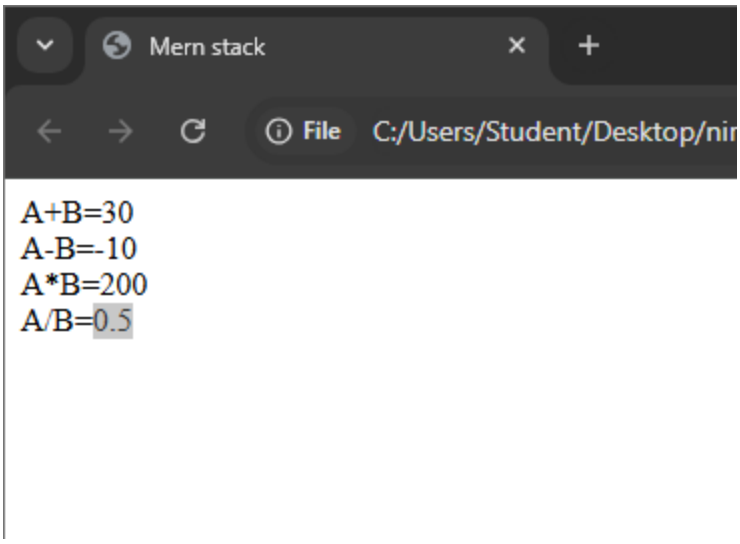
```
<DOCTYPE! html>
    <html>
        <head>
            <title>
                Mern stack
            </title>
        </head>
        <body>
            <script>
                var s="Hello";
                var x=10;
```

```
            var a=true;
            console.log(s+"\n"+x+"\n"+a+"\n");

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



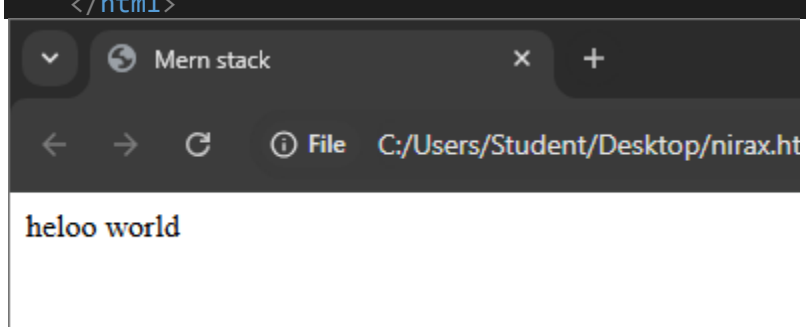
Task 3:

```
<DOCTYPE! html>
    <html>
        <head>
            <title>
                Mern stack
            </title>
        </head>
        <body>
            <script>
                let a=10;
                let b=20;
                document.writeln("A+B="+(a+b)+"<br>");
                document.writeln("A-B="+(a-b)+"<br>");
                document.writeln("A*B="+(a*b)+"<br>");
                document.writeln("A/B="+(a/b)+"<br>");

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



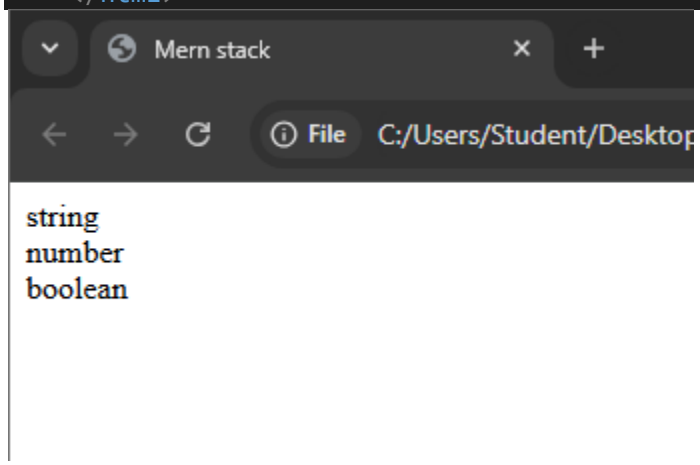
Task 4:

```
<DOCTYPE! html>
    <html>
        <head>
            <title>
                Mern stack
            </title>
        </head>
        <body>
            <script>
                let a="heloo";
                let b=" world";
                document.writeln(a+b);

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



Task 5:

```
<html>
    <head>
        <title>
            Mern stack
        </title>
    </head>
    <body>
        <script>
            var s="Hello";
            var x=10;
            var a=true;
            document.write(typeof s+"<br>");
            document.write(typeof x+"<br>");
            document.write(typeof a+"<br>");


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

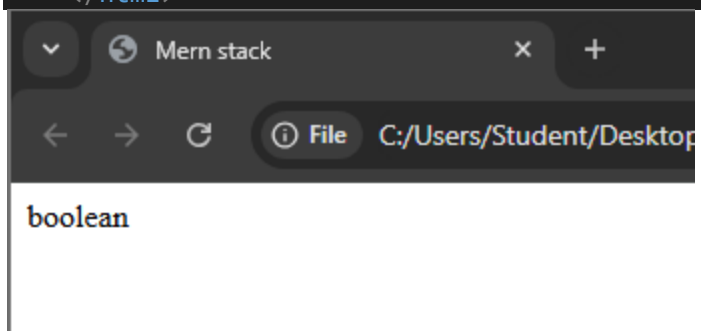


2.Code Structure:

Task 6:

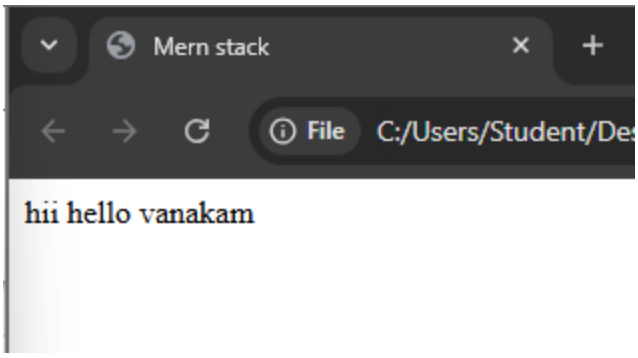
```
<DOCTYPE! html>
    <html>
        <head>
            <title>
                Mern stack
            </title>
        </head>
        <body>
```

```
            <script>
                //var s="Hello";
                //var x=10;
                var a=true;
                /*
                document.write(typeof s+"<br>");
                document.write(typeof x+"<br>");
                */
                document.write(typeof a+"<br>");


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



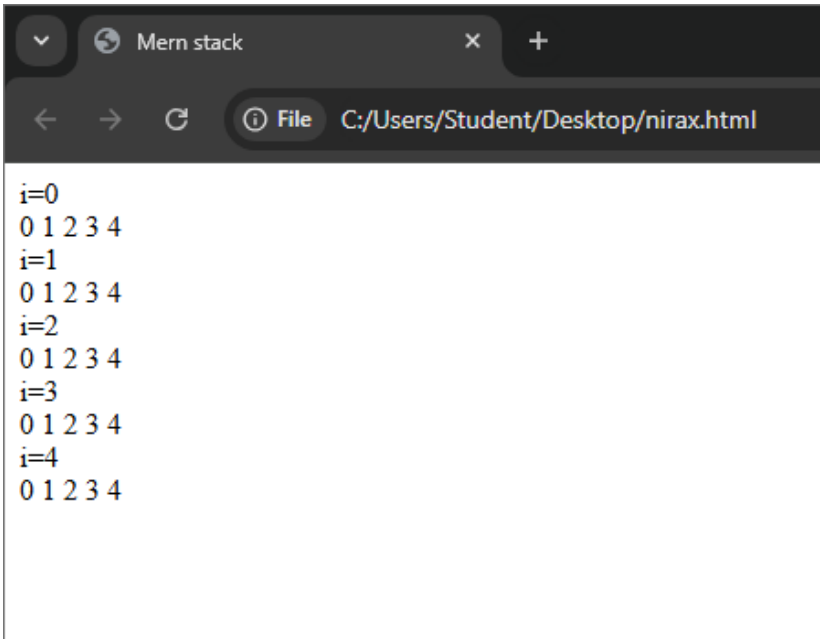
Task 7:

```
<DOCTYPE! html>
    <html>
        <head>
            <title>
                Mern stack
            </title>
        </head>
        <body>
            <script>
                document.write("hii")
                document.write(" hello");
                document.write(" vanakam");
            </script>
        </body>
    </html>
```



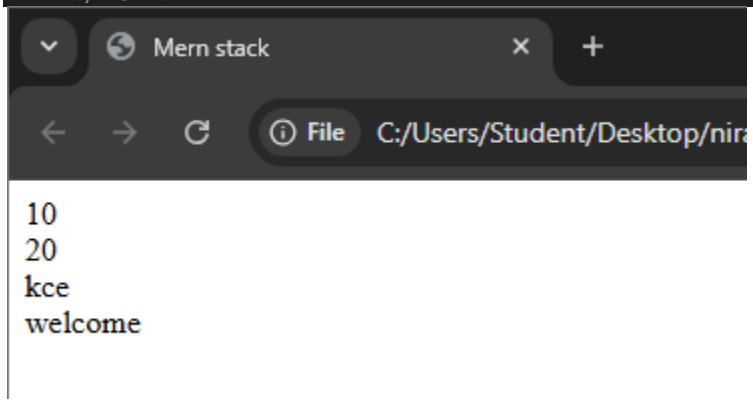
Task 8:

```
<DOCTYPE! html>
    <html>
        <head>
            <title>
                Mern stack
            </title>
        </head>
        <body>
            <script>
                for(var i=0;i<5;i++){
                    document.write("i="+i);
                    document.write("<br>");
                    for(var j=0;j<5;j++){
                        document.write(j+" ");
                    }
                    document.write("<br>");
                }
            </script>
        </body>
    </html>
```



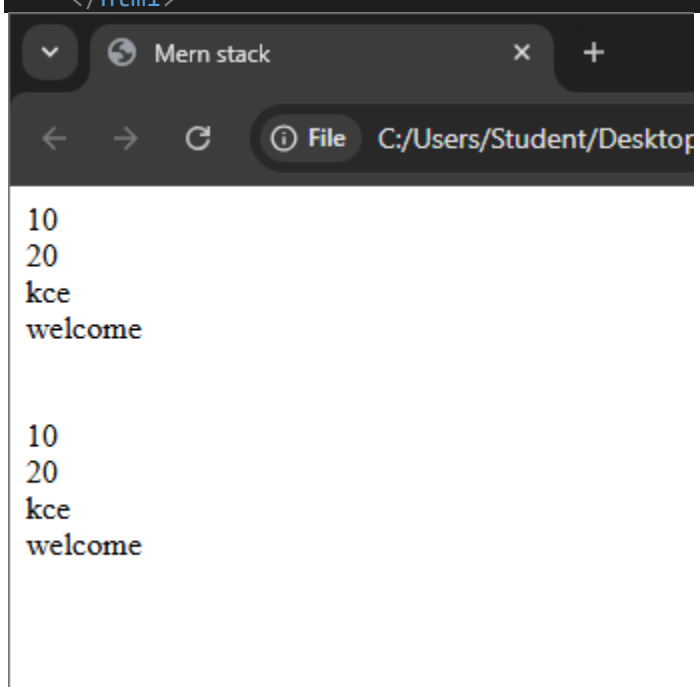
Task 9:

```
<DOCTYPE! html>
    <html>
        <head>
            <title>
                Mern stack
            </title>
        </head>
        <body>
            <script>
                var a=10,b=20,c="kce",d=" welcome";
                document.write(a+"<br>"+b+"<br>"+c+"<br>"+d+"<br>");

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



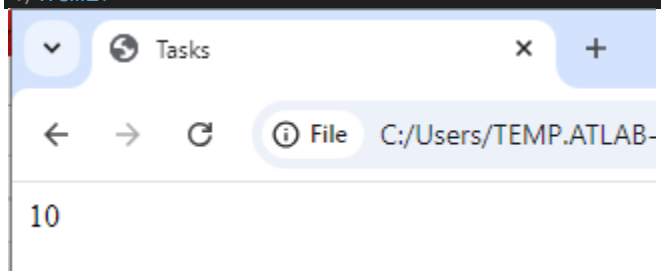
Task 10:

```
<DOCTYPE! html>
    <html>
        <script>
            var a=10,b=20,c="kce",d=" welcome";
            document.write(a+"<br>"+b+"<br>"+c+"<br>"+d+"<br>"+"<br>"+"<br>");

        </script>
        <head>
            <title>
                Mern stack
            </title>
        </head>
        <body>
        </body>
        <script>
            var a=10,b=20,c="kce",d=" welcome";
            document.write(a+"<br>"+b+"<br>"+c+"<br>"+d+"<br>"+"<br>"+"<br>");

        </script>
    </html>
```



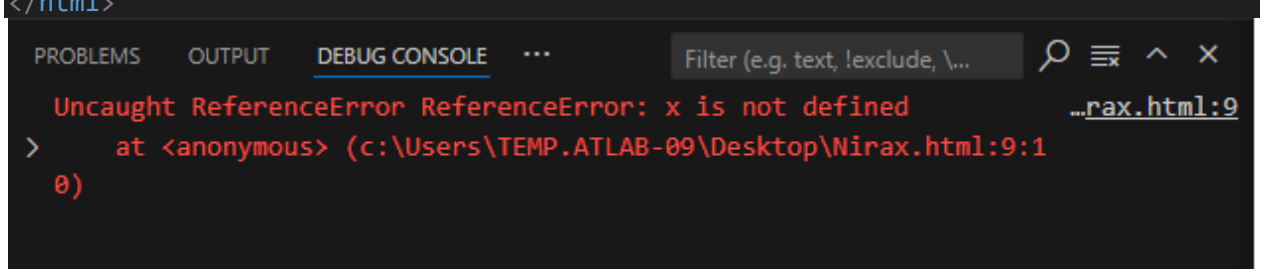
TASKS 11:

```html
<DOCTYPE! html>
<html>
    <head>
        <title>TASKS</title>
    </head>
    <body>
        <script>
        x=10;
        document.write(x);
        </script>
    </body>
</html>
```



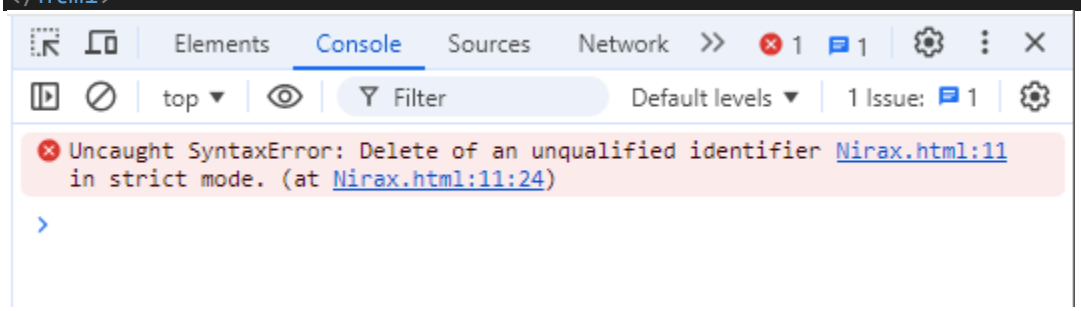
TASKS 12:

```html
<DOCTYPE! html>
<html>
    <head>
        <title>TASKS</title>
    </head>
    <body>
        <script>
        "use strict"
        x=10;
        document.write(x);
        </script>
    </body>
</html>
```



TASKS 13:

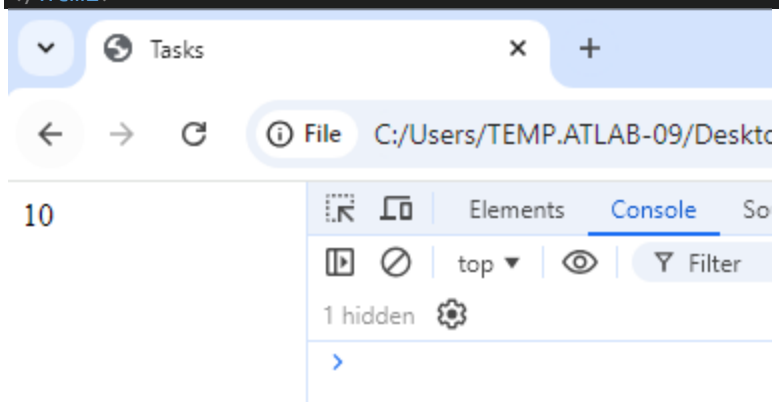
```
<DOCTYPE! html>
<html>
    <head>
        <title>TASKS</title>
    </head>
    <body>
        <script>
                "use strict"
                let x=10;
                document.write(x);
                function a(){
                    document.write("Function block");
                }
                delete x;
                delete a;

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



TASKS 14:

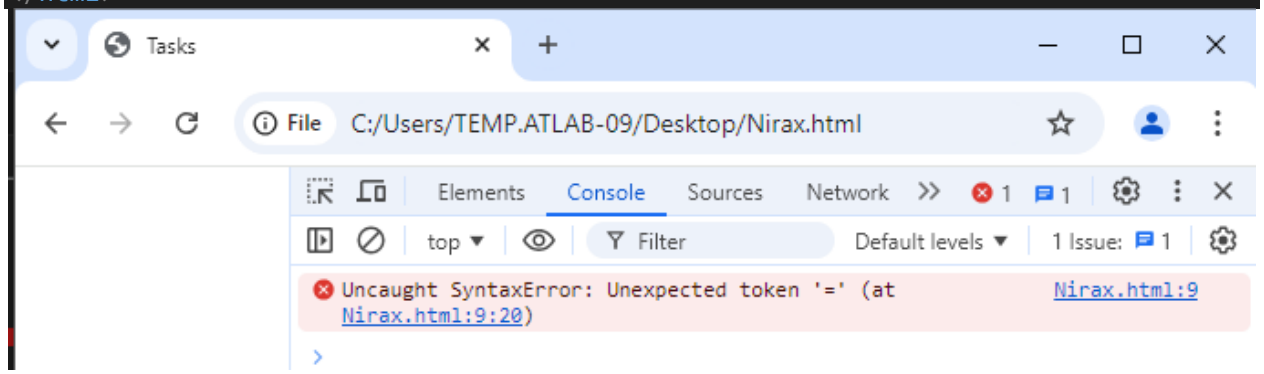
```
<DOCTYPE! html>
<html>
    <head>
        <title>TASKS</title>
    </head>
    <body>
        <script>
                x=20;
                "use strict"
                x=10;
                document.write(x);
                function a(){
                    document.write("Function block");
                }
```

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



TASKS 15:

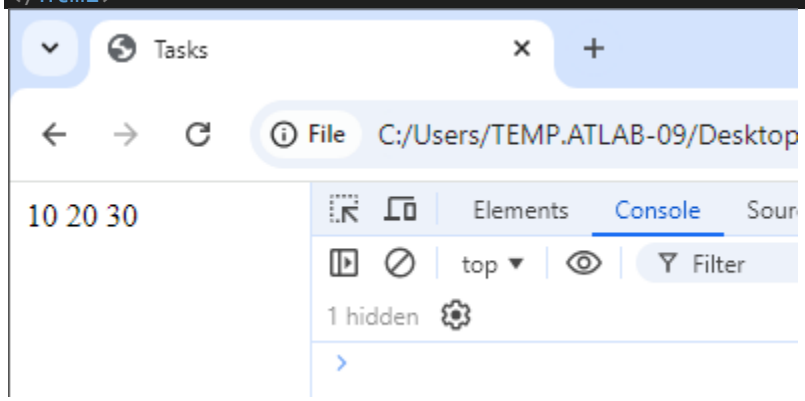
```
<DOCTYPE! html>
<html>
    <head>
        <title>TASKS</title>
    </head>
    <body>
        <script>
                "use strict"
                try=10;
                document.write(x);
                function a(){
                    document.write("Function block");
                }

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



2. Variables:

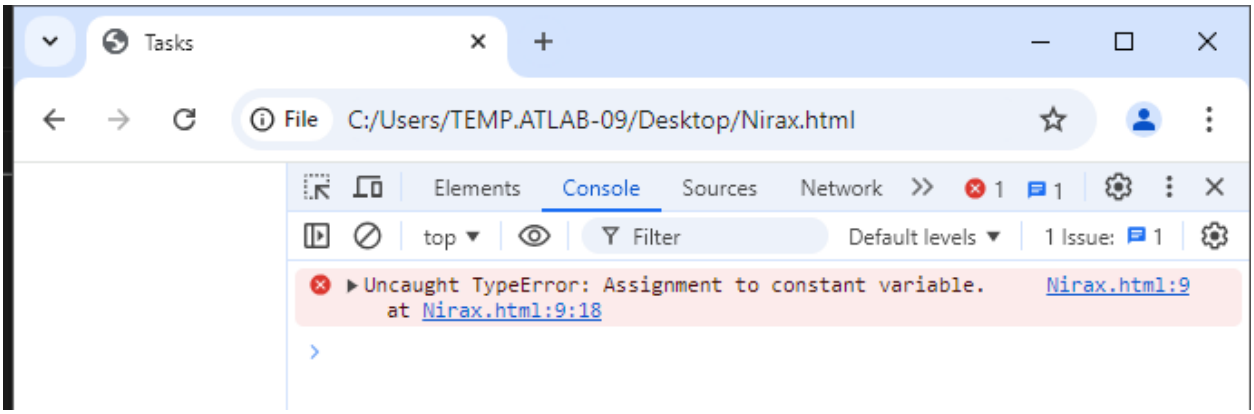
TASKS 16:

```
<DOCTYPE! html>
<html>
    <head>
        <title>TASKS</title>
    </head>
    <body>
        <script>
                let x=10;//x cannot be redeclared  but can be reassign
                var y=20;//y can both be redeclared and reassigned
                const z=30;//reaasign or redeclare is not possible
                document.write(x+" "+y+" "+z);
        </script>
    </body>
</html>
```



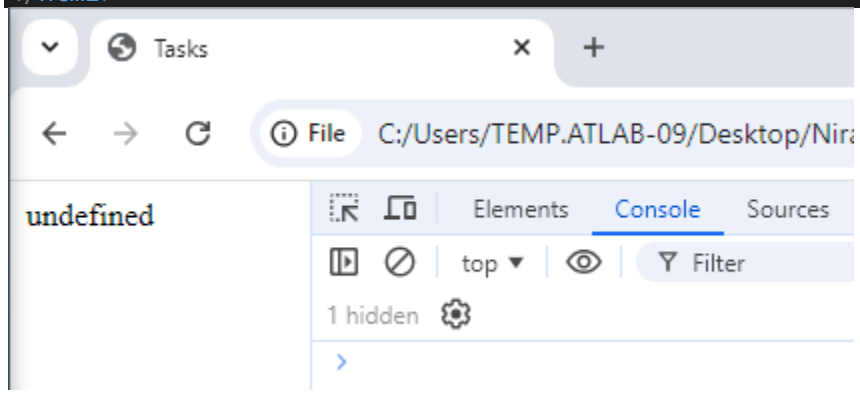
TASKS 17:

```
<DOCTYPE! html>
<html>
    <head>
        <title>TASKS</title>
    </head>
    <body>
        <script>
                const z=30;
                z=50;
                document.write(z);
        </script>
    </body>
</html>
```



TASKS 18:

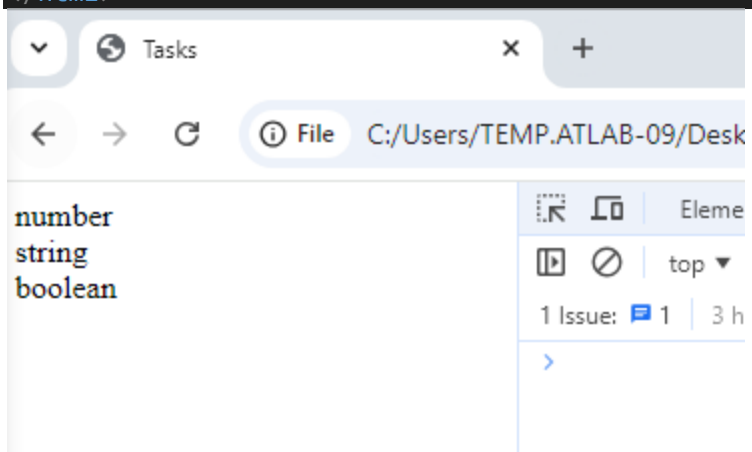
```
<DOCTYPE! html>
<html>
    <head>
        <title>TASKS</title>
    </head>
    <body>
        <script>
                let z;
                document.write(z);
        </script>
    </body>
</html>
```



TASKS 19:

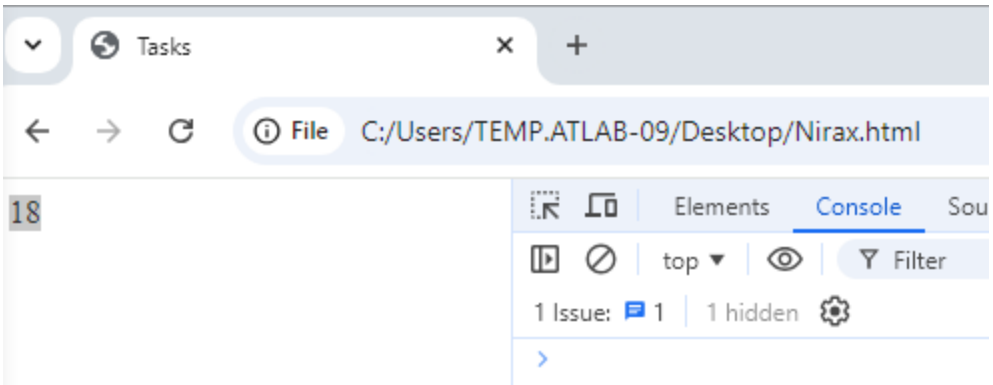
```
<DOCTYPE! html>
<html>
    <head>
        <title>TASKS</title>
    </head>
    <body>
        <script>
```

```
                let z=10;
                document.write(typeof z+"<br>");
                z="kce";
                document.write(typeof z+"<br>");
                z=true;
                document.write(typeof z+"<br>");
        </script>
    </body>
</html>
```



TASKS 20:

```
<DOCTYPE! html>
<html>
    <head>
        <title>TASKS</title>
    </head>
    <body>
        <script>
                const z={name:"niranjan",age:18};
                const {age:years}=z;
                document.write(years);
        </script>
    </body>
</html>
```

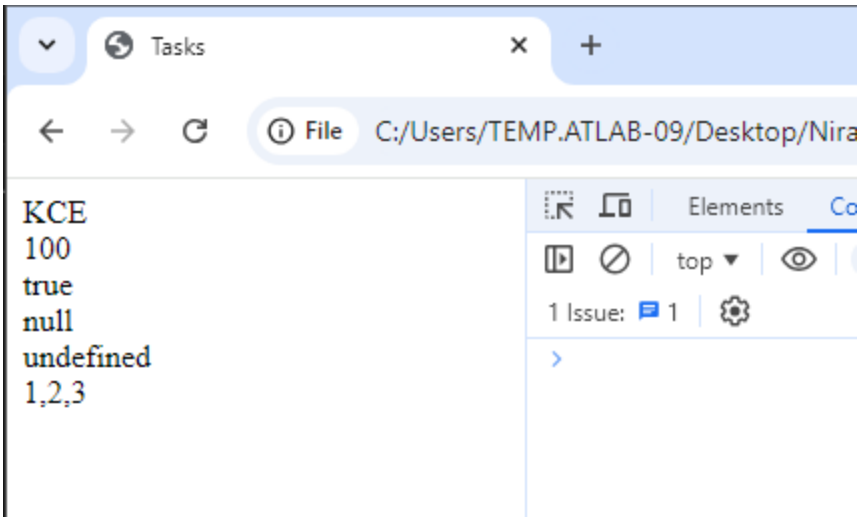


Data types, Basic operators, maths

1. Data types:

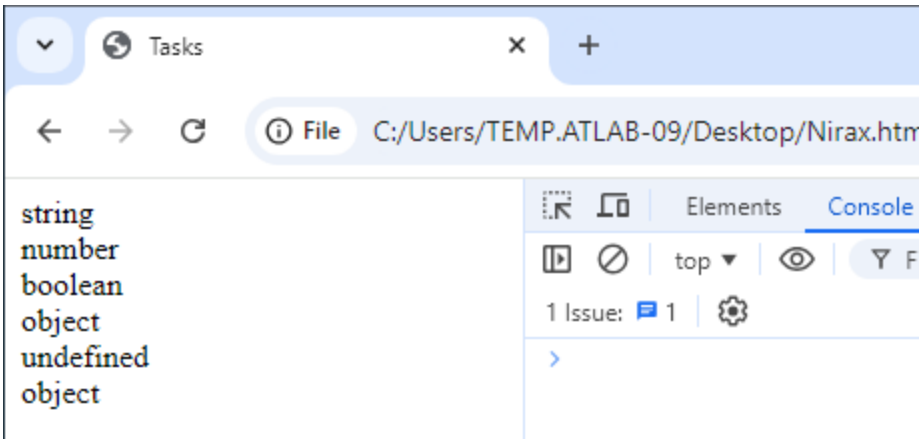
TASKS 21:

```
<DOCTYPE! html>
<html>
    <head>
        <title>Tasks</title>
    </head>
    <body>
        <script>
                let a="KCE";
                let b=100;
                let c=true;
                let d=null;
                let e;
                let f=[1,2,3];
                document.write( a+"<br>");
                document.write( b+"<br>");
                document.write( c+"<br>");
                document.write( d+"<br>");
                document.write( e+"<br>");
                document.write( f+"<br>");
        </script>
    </body>
</html>
```



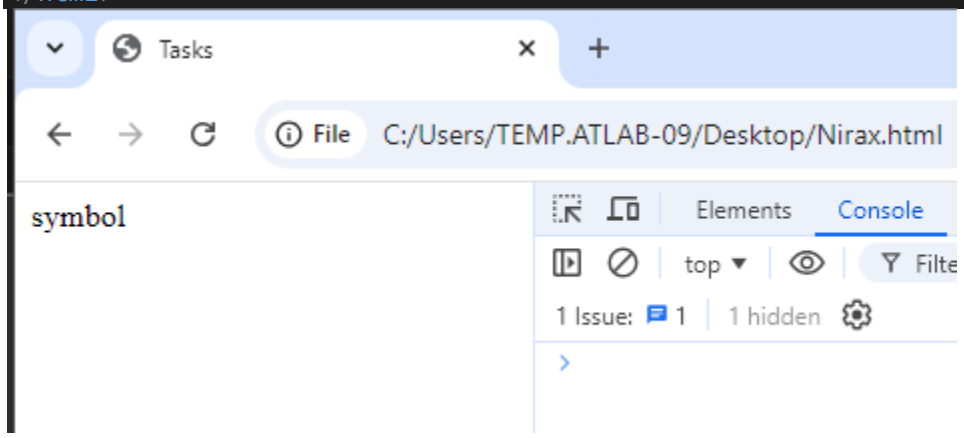
TASKS 22:

```
<DOCTYPE! html>
<html>
    <head>
        <title>TASKS</title>
    </head>
    <body>
        <script>
                let a="KCE";
                let b=100;
                let c=true;
                let d=null;
                let e;
                let f=[1,2,3];
                document.write(typeof a+"<br>");
                document.write(typeof b+"<br>");
                document.write(typeof c+"<br>");
                document.write(typeof d+"<br>");
                document.write(typeof e+"<br>");
                document.write(typeof f+"<br>");
        </script>
    </body>
</html>
```



TASK 23:

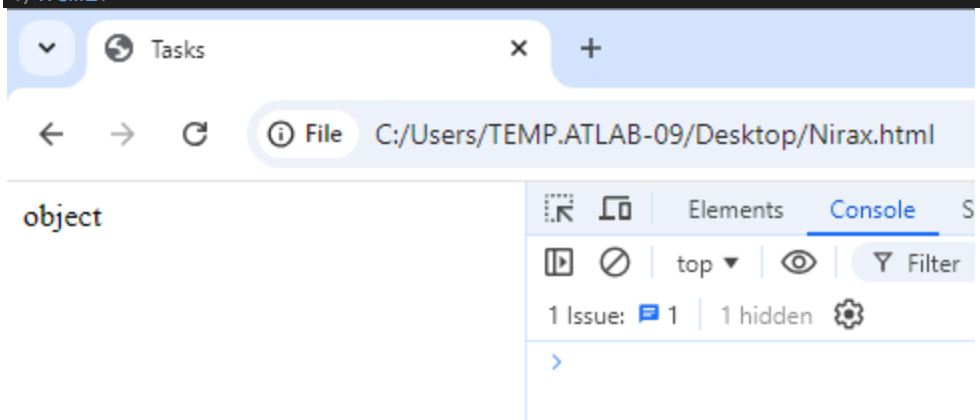
```
<DOCTYPE! html>
<html>
    <head>
        <title>Tasks</title>
    </head>
    <body>
        <script>
                let a=Symbol("123")
                document.write(typeof a);
        </script>
    </body>
</html>
```



TASK 24:

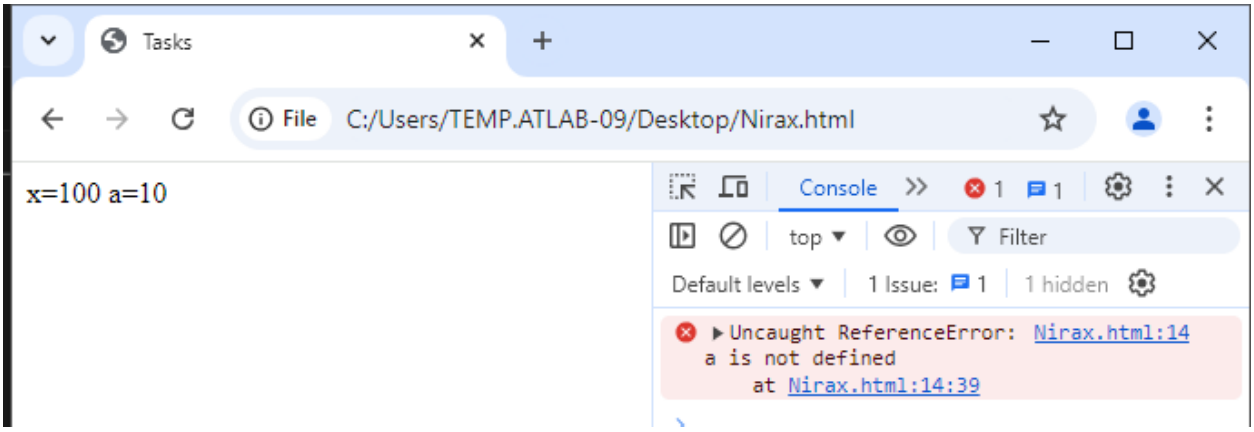
```
<DOCTYPE! html>
<html>
    <head>
        <title>Tasks</title>
    </head>
```

```
    <body>
        <script>
                let a=null;
                document.write(typeof a);
        </script>
    </body>
</html>
```



TASK 25:

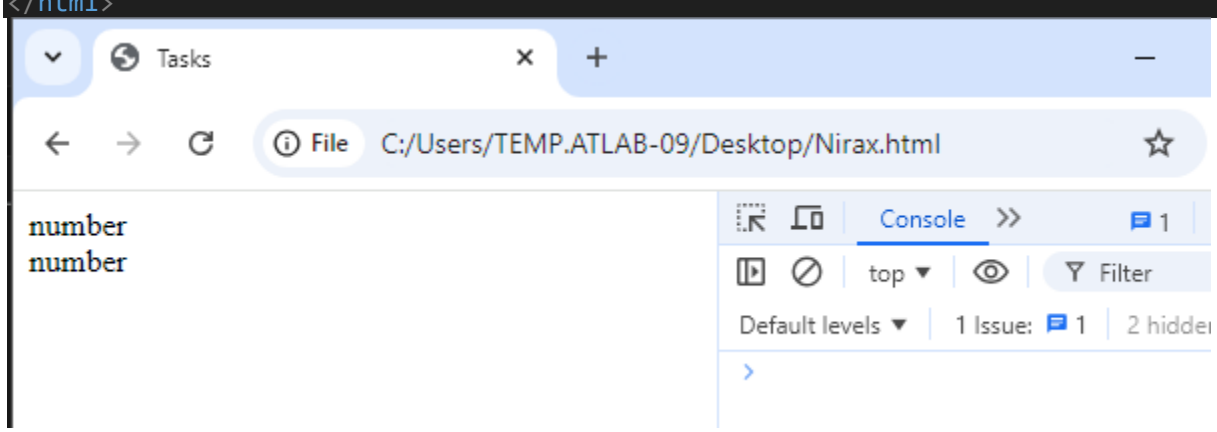
```
<DOCTYPE! html>
<html>
    <head>
        <title>Tasks</title>
    </head>
    <body>
        <script>
                var x=100;
                function y(){
                    let a=10;
                    document.write("x="+x+"\na="+a);
                }
                y();
                document.write("\na="+a);

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



Basic operators, maths:

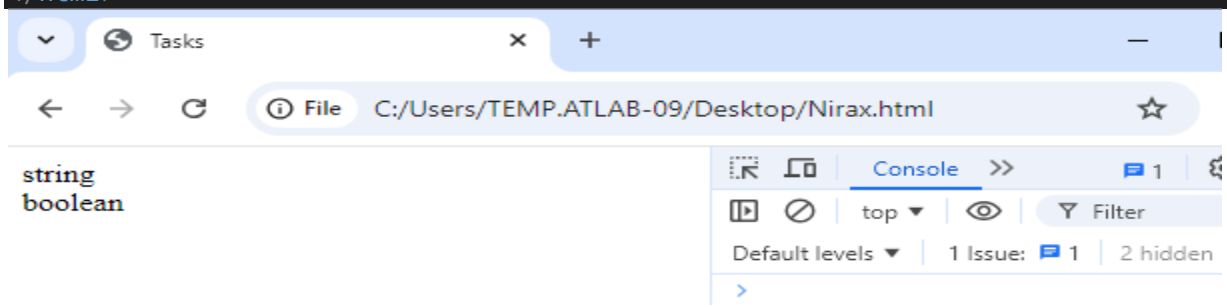
TASK 26:

```
<DOCTYPE! html>
<html>
    <head>
        <title>Tasks</title>
    </head>
    <body>
        <script>
                let z="123";
                document.write(typeof (z-1)+"<br>");
                z=Number(z);
                document.write(typeof z);


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



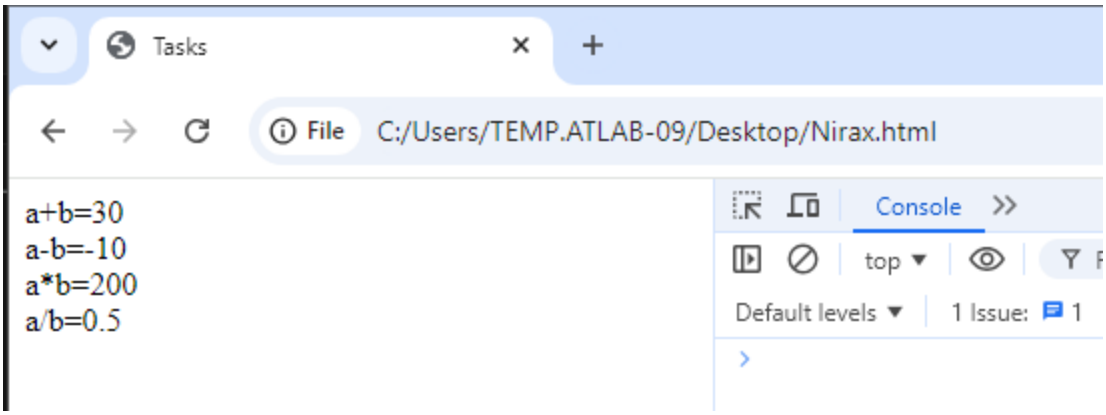
TASK 27:

```
<DOCTYPE! html>
<html>
    <head>
        <title>Tasks</title>
    </head>
    <body>
        <script>
                let z=true;
                y=String(z);
                document.write(typeof y+"<br>");
                x=Boolean(y);
                document.write(typeof x);


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



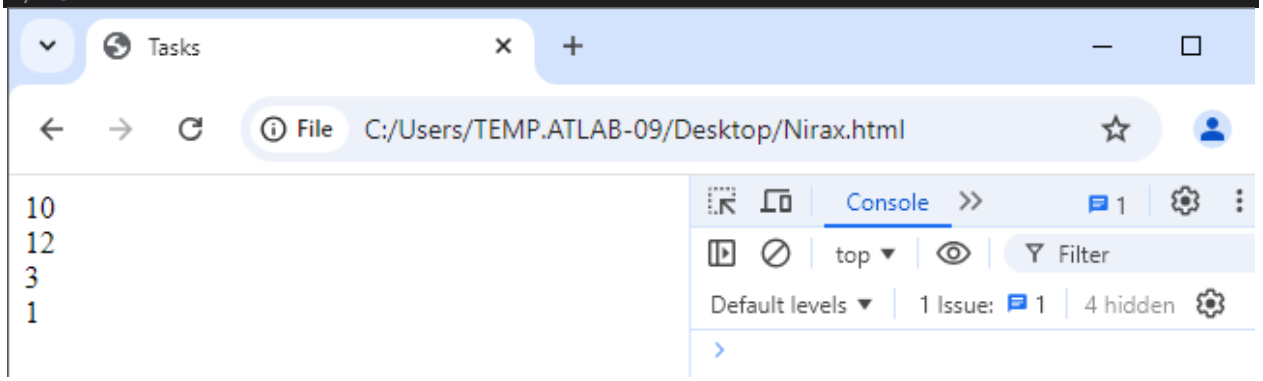
TASK 28:

```
<DOCTYPE! html>
<html>
    <head>
        <title>Tasks</title>
    </head>
    <body>
        <script>
                a=10;
                b=20
                document.write("a+b="+(a+b)+"<br>");
                document.write("a-b="+(a-b)+"<br>");
                document.write("a*b="+(a*b)+"<br>");
                document.write("a/b="+(a/b)+"<br>");
        </script>
    </body>
</html>
```



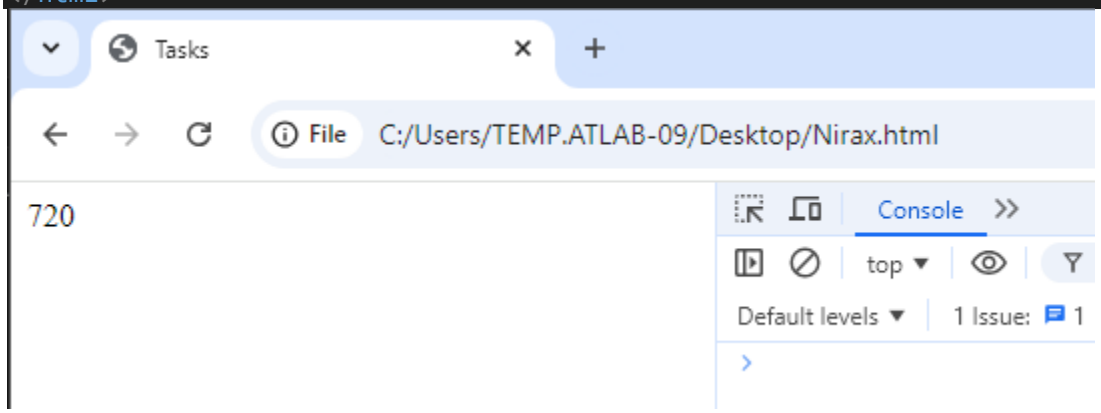
TASK 29:

```
<DOCTYPE! html>
<html>
    <head>
        <title>Tasks</title>
    </head>
    <body>
        <script>
                a=10;b=3;
                document.write(a++ +"<br>");
                document.write(++a +"<br>");
                document.write(b-- +"<br>");
                document.write(--b +"<br>");
        </script>
    </body>
</html>
```



TASK 30:

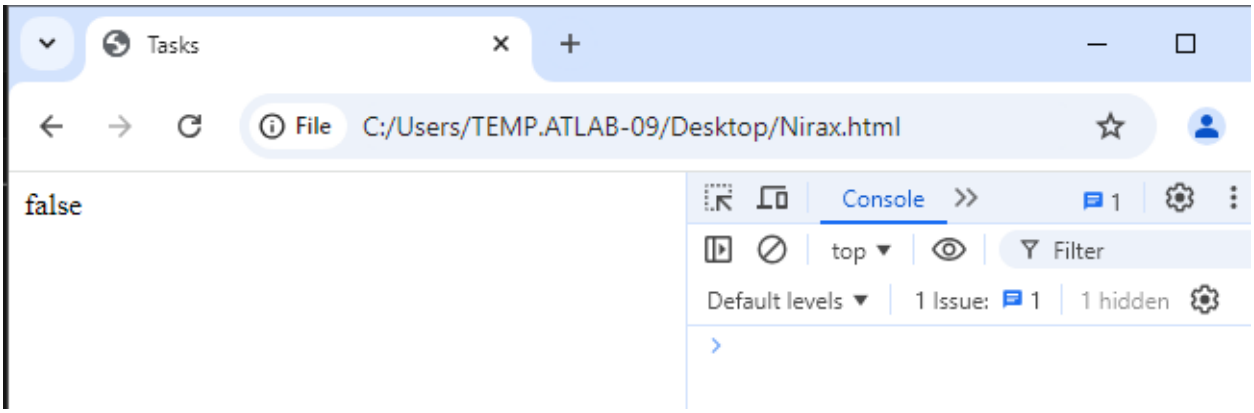
```
<DOCTYPE! html>
<html>
    <head>
        <title>Tasks</title>
    </head>
    <body>
        <script>
                a=(50-2)*15;
                document.write(a +"<br>");

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



Comparisons, Conditional branching: if, '?'

Task 31:

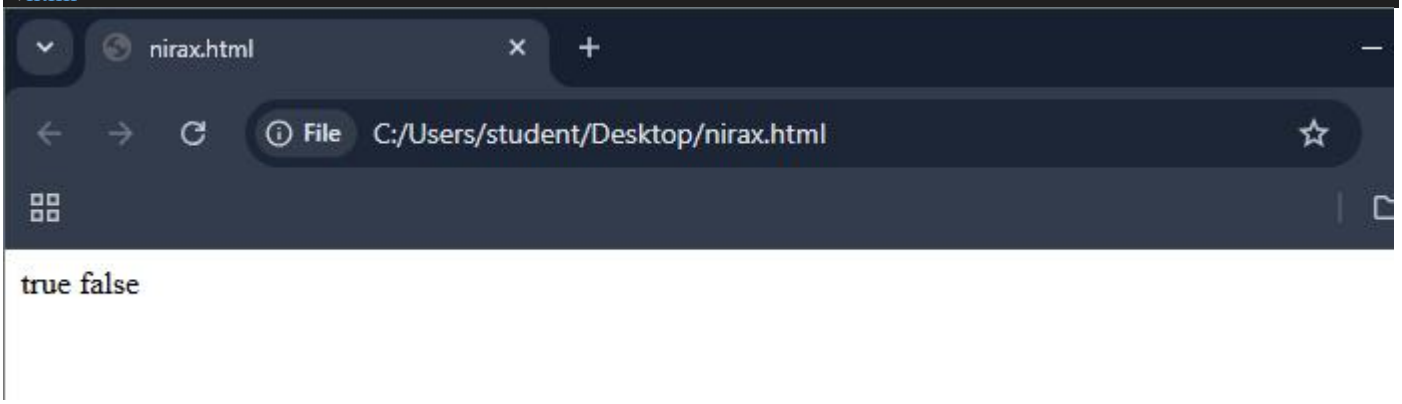
```
<DOCTYPE! html>
<html>
    <head>
        <title>Tasks</title>
    </head>
    <body>
        <script>
                a=27,b=35;
                a>=b?document.write('true'):document.write('false');
        </script>
    </body>
</html>
```


Tasks
File
C:/Users/TEMP.ATLAB-09/Desktop/Nirax.html
false
Console
1
top
Filter
Default levels
1 Issue: 1
1 hidden

Task 32:

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

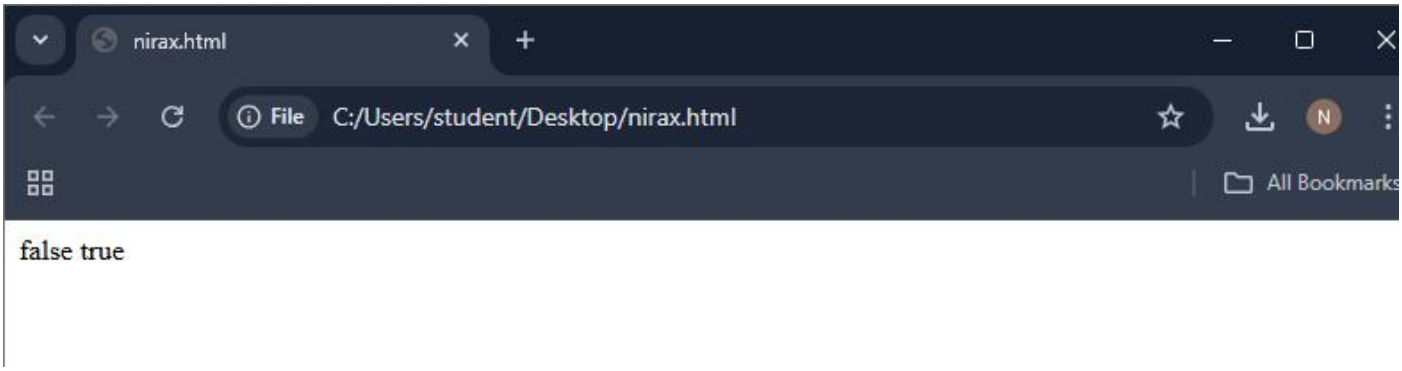
```
<html>
  <body>
   <script>
      document.writeln(10=='10'+"\n");
      document.writeln(10===='10'+"\n");
   </script>
  </body>
</html>
```



Task 33:

Compare two strings lexicographically.

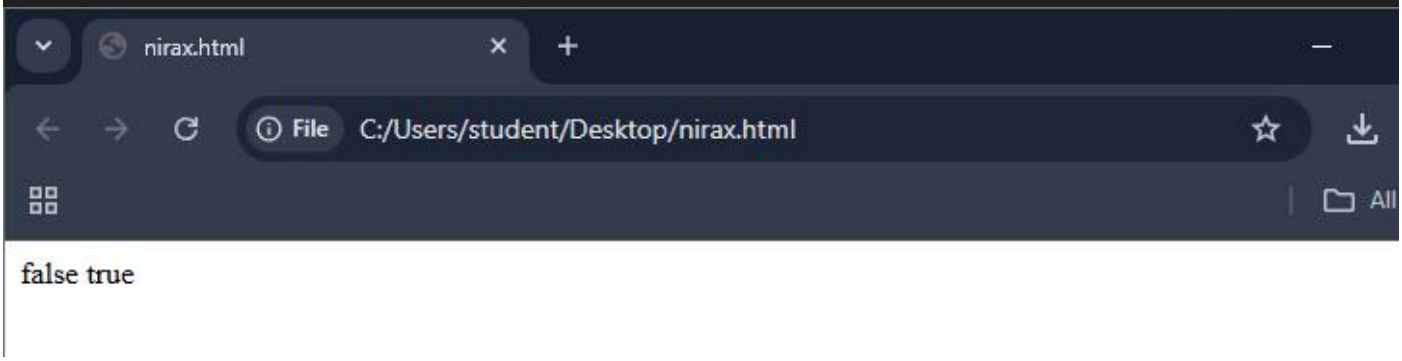
```
<html>
  <body>
   <script>
      document.writeln("abcd">"abde");
      document.writeln("abcd"<"abde");
   </script>
  </body>
</html>
```



Task 34:

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

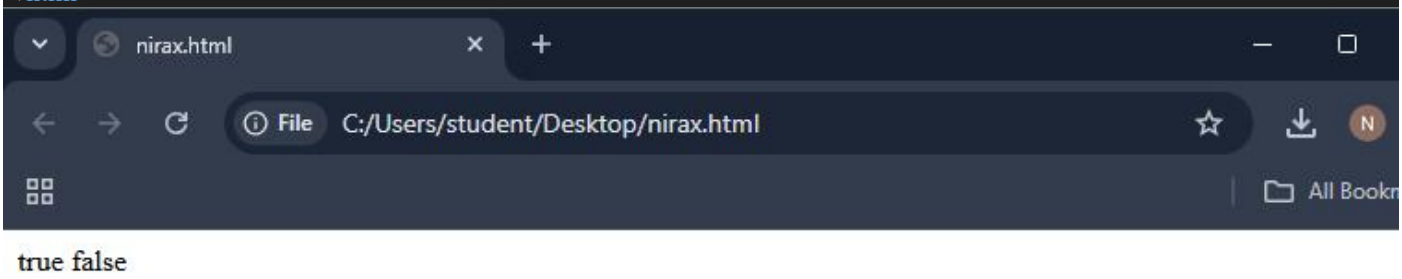
```
<html>
  <body>
   <script>
      document.writeln(55!='55');
      document.writeln(55!=='55');
   </script>
  </body>
</html>
```



Task 35:

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

```
<html>
  <body>
   <script>
      let x=null;
      let y;
      document.writeln(x==y);
      document.writeln(x===y);
   </script>
  </body>
</html>
```

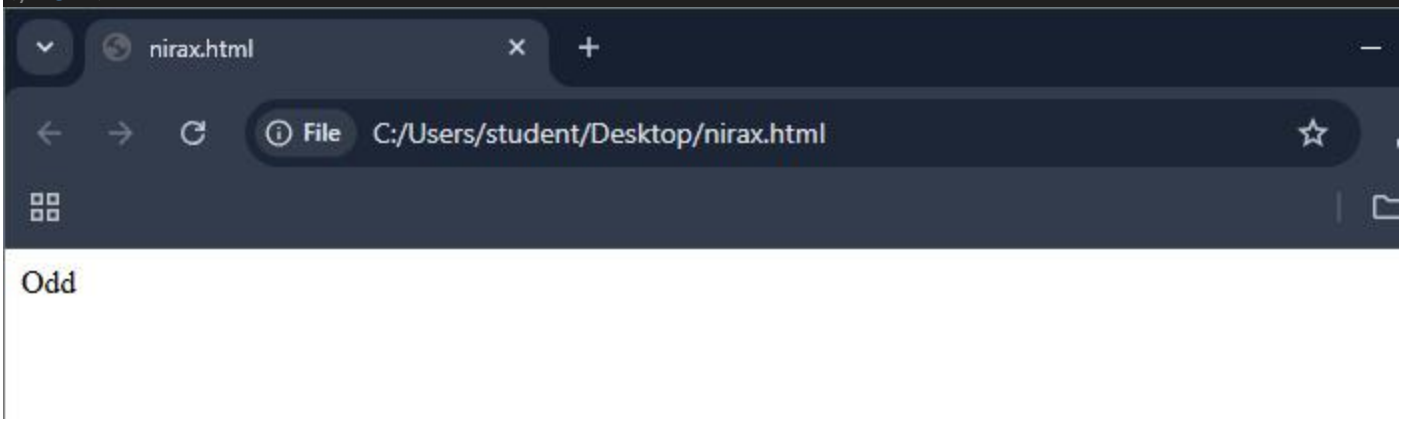


Conditional branching: if, ‘?’

Task 36:

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

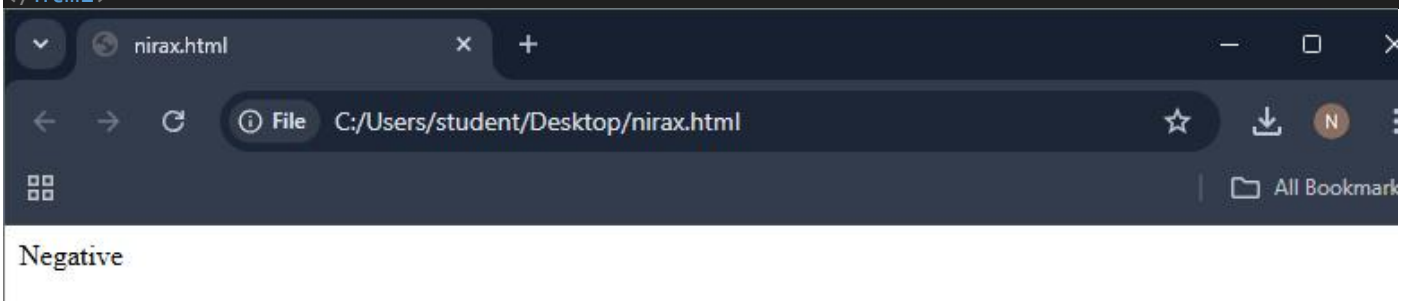
```
<html>
   <body>
    <script>
        x=45
        if(x%2==0)document.writeln("Even");
        else document.writeln("Odd");
    </script>
   </body>
</html>
```



Task 37:

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

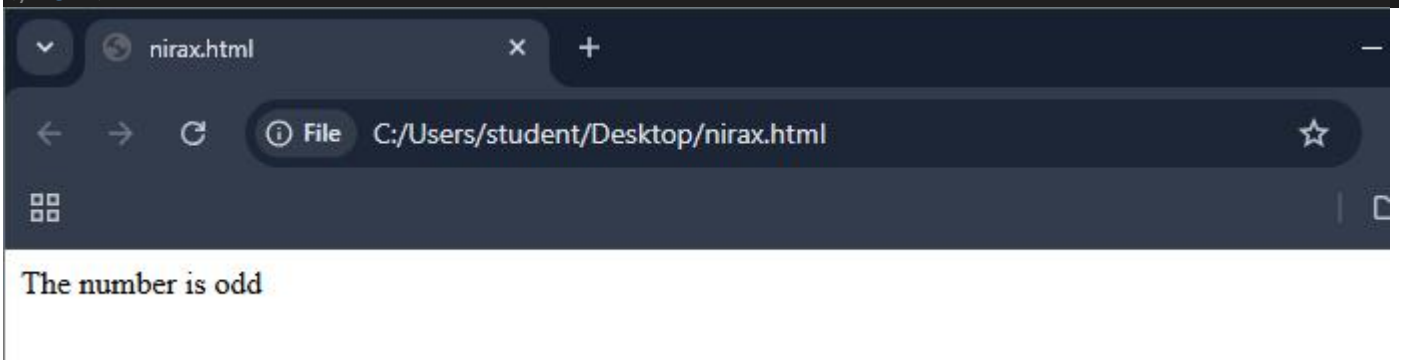
```
<html>
   <body>
    <script>
        x=-45
        if(x==0)
            document.writeln("Zero");
        else {
            if(x>=0)
                document.writeln("Positive");
            else
            document.writeln("Negative");
        }

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



Task 38:

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

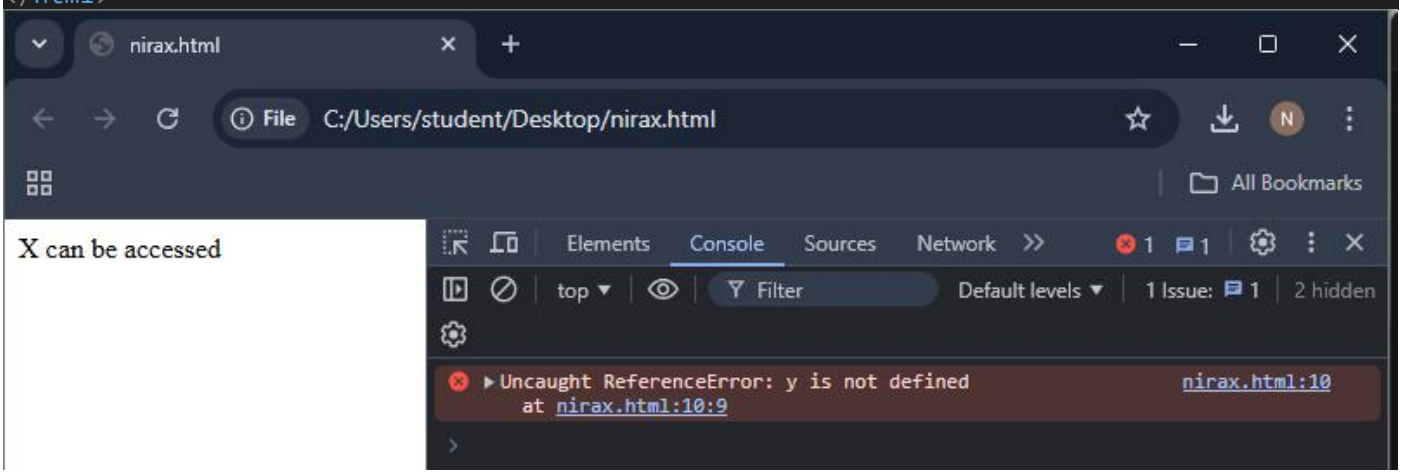
```
<html>
   <body>
    <script>
        x=23;
        x%2==0?document.writeln("The number is even"):document.writeln("The number is odd")
    </script>
   </body>
</html>
```



Task 39:

Check the validity of a variable using the ? operator

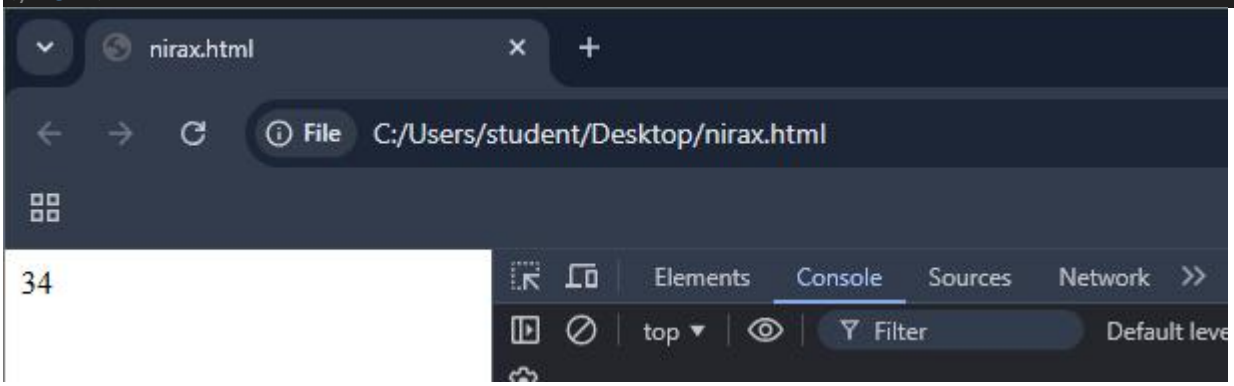
```
<html>
   <body>
    <script>
        var x=23;
        function f(){
            let y=89;
            x?document.writeln("X can be accessed"):document.writeln("X cannot be
accessed")
        }
        f();
        y?document.writeln("Y can be accessed"):document.writeln("Y cannot be accessed")
    </script>
   </body>
</html>
```



Task 40:

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

```
<html>
   <body>
    <script>
        var x;
        'Char'=='char'?x=45:x=34;
        document.writeln(x)
    </script>
   </body>
</html>
```



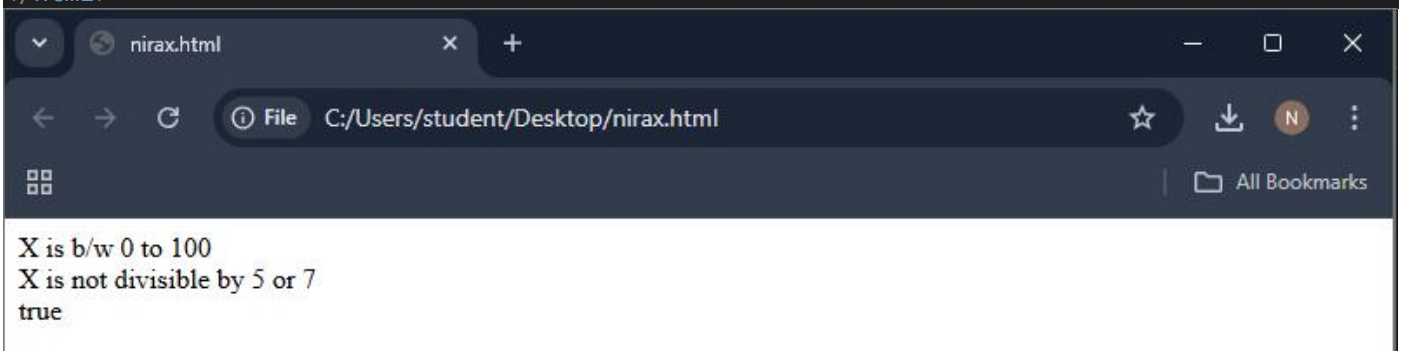
Logical operators, Functions

Logical operators:

Task 41:

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

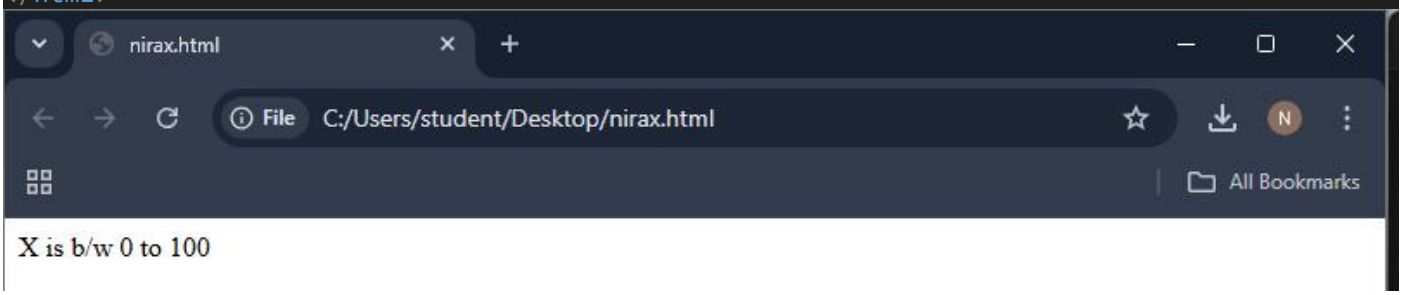
```
<html>
   <body>
    <script>
        x=67;
        (x>=0 && x<=100)?document.writeln("X is b/w 0 to 100<br>"):document.writeln("X is
not b/w 0 to 100<br>");
        (x%5==0 || x%7==0)?document.writeln("X is divisible by 5 or
7<br>"):document.writeln("X is not divisible by 5 or 7<br>");
        (!true)?document.writeln("false"):document.writeln("true");
    </script>
   </body>
</html>
```



Task 42:

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

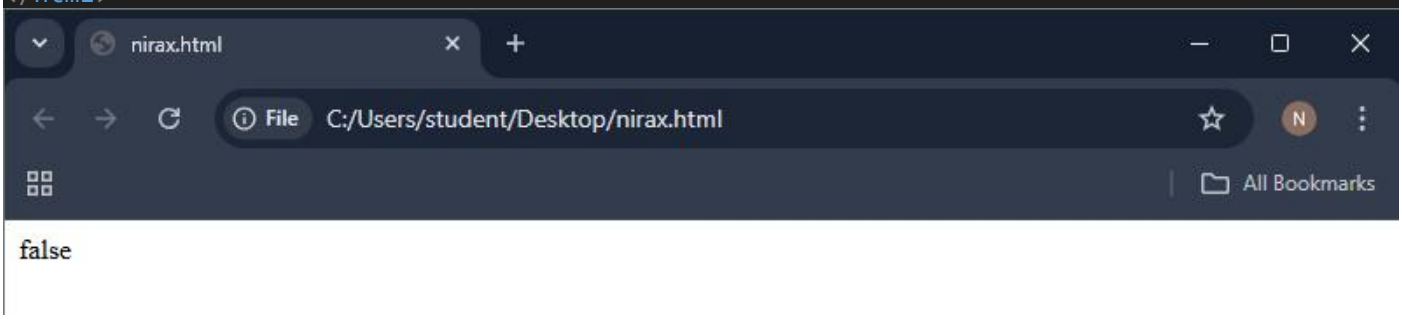
```
<html>
   <body>
    <script>
        x=67;
        (x>=0 && x<=100)?document.writeln("X is b/w 0 to 100<br>"):document.writeln("X is
not b/w 0 to 100<br>");
    </script>
   </body>
</html>
```



Task 43:

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

```
<html>
   <body>
    <script>
        (!true)?document.writeln("true"):document.writeln("false");
    </script>
   </body>
</html>
```



Task 44:

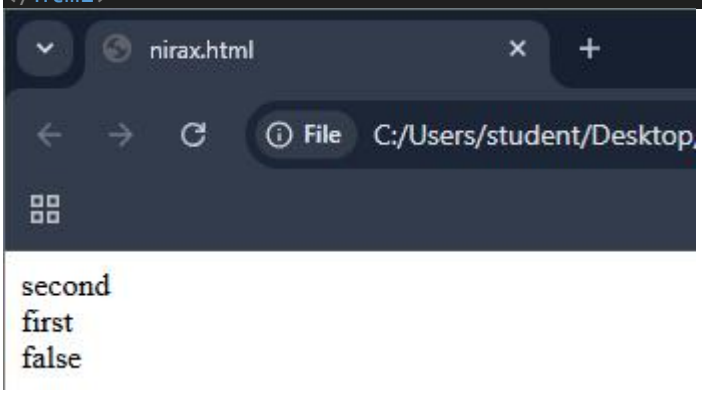
Evaluate the short-circuiting nature of logical operators.

```
<html>
   <body>
    <script>
        a=20;
        b=0;
        document.writeln(a&&b);
        document.writeln(a||b);
    </script>
   </body>
</html>
```

Task 45:

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

```
<html>
   <body>
    <script>
        a="first";
        b="second";
        document.writeln(a&&b+"<br>");
        document.writeln(a||b);
        document.writeln("<br>"+!b);
    </script>
   </body>
</html>
```

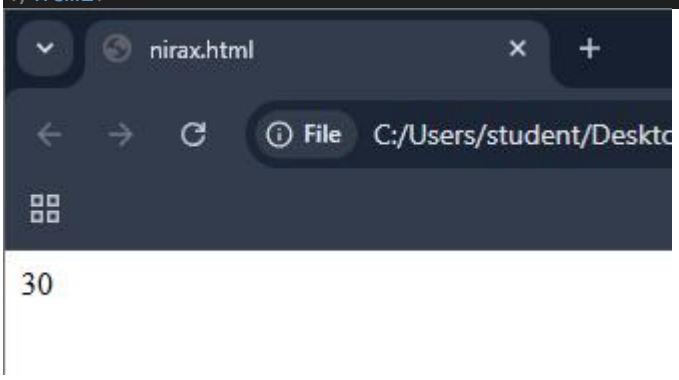


Functions:

Task 46:

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

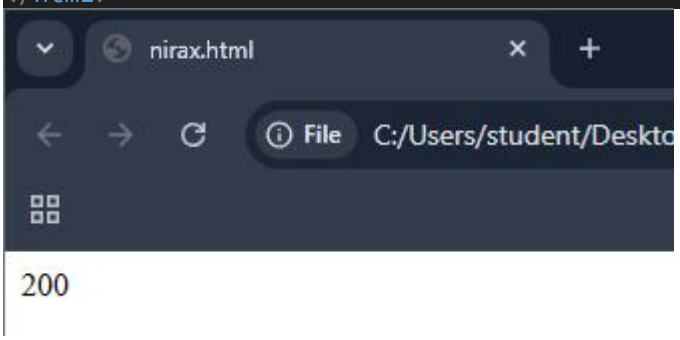
```
<html>
   <body>
    <script>
        function add(a,b){
            return a+b;
        }
        document.writeln(add(10,20));
    </script>
   </body>
</html>
```



Task 47:

Create a function that calculates the area of a rectangle.

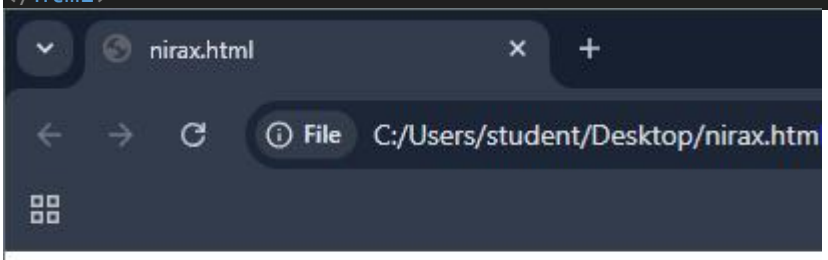
```
<html>
   <body>
    <script>
        function area(l,b){
            return l*b;
        }
        document.writeln(area(10,20));
    </script>
   </body>
</html>
```



Task 48:

Declare a function without parameters and call it.

```
<html>
   <body>
    <script>
        function call(){
            document.writeln("Heloo!!");
        }
        call();
    </script>
   </body>
</html>
```

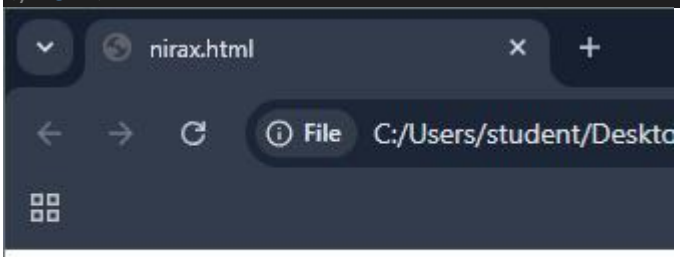


Heloo!!

Task 49:

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

```
<html>
   <body>
    <script>
        function call(){
            document.writeln("Heloo!!<br>");
        }
        document.writeln(call());
    </script>
   </body>
</html>
```

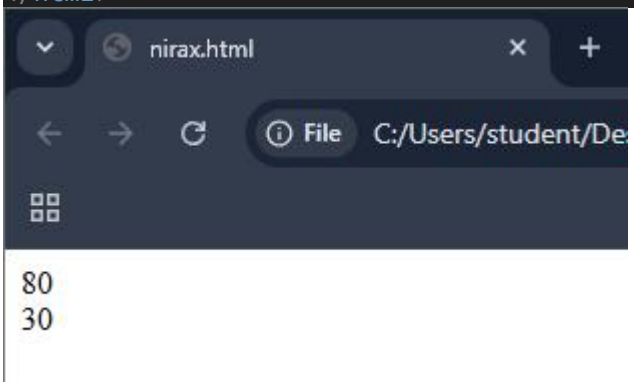


Heloo!!
undefined

Task 50:

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

```
<html>
   <body>
    <script>
        function call(a,b=8){
            return a*b;
        }
        document.writeln(call(10)+"<br>");
        document.writeln(call(5,6));
    </script>
   </body>
</html>
```

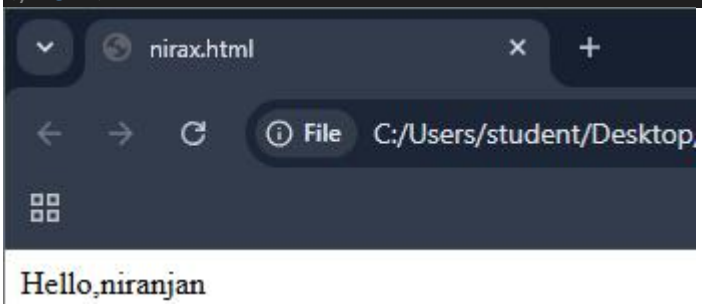


Arrow Functions:

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.

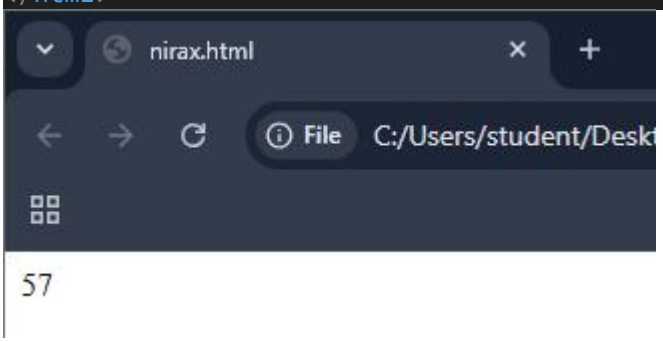
```
<html>
   <body>
    <script>
        let fun=(name)=>{
            document.writeln(`Hello,${name}`);
        };
        fun("niranjan");
    </script>
   </body>
</html>
```



Task 52:

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

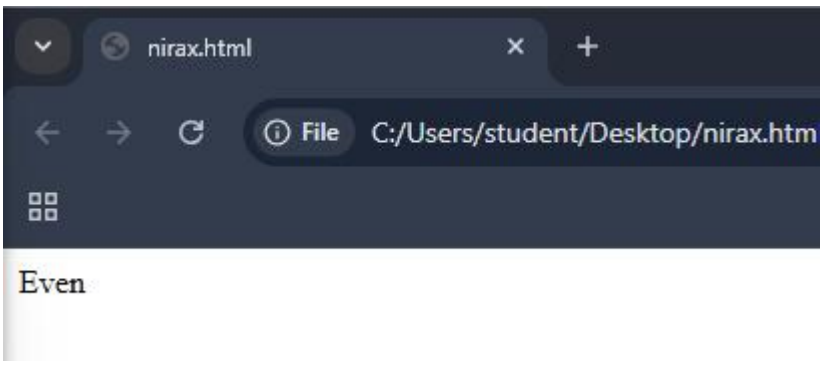
```
<html>
   <body>
    <script>
        let add=(a,b)=>{
            return a+b;
        };
        document.writeln(add(23,34));
    </script>
   </body>
</html>
```



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

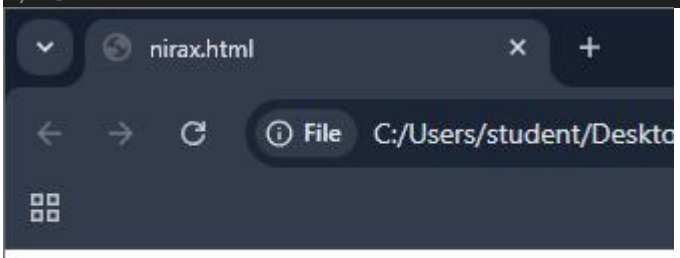
```
<html>
   <body>
    <script>
        let isEven=(a)=> (a%2==0)?"Even":"Odd";
        document.writeln(isEven(54));
    </script>
   </body>
</html>
```



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
<html>
   <body>
    <script>
        let maxValue=(a,b)=>{
            return (a>b)?"A is greatest":"B is greates";
        }
        document.writeln(maxValue(2,-52));
    </script>
   </body>
</html>
```

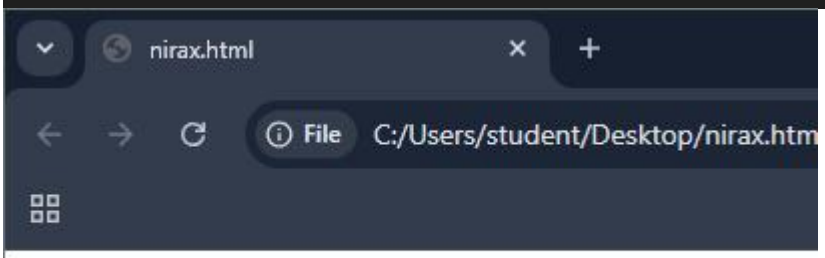


A is greatest

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
<html>
   <body>
    <script>
        let myObject={
            value:10,
            multiplyTraditional:function(x){
                return this.value*x;
            },
            multiplyArrow:(x)=>{
                return this.value*x;
            }
        };
        document.writeln("Traditional function="+myObject.multiplyTraditional(12)+"<br>");
        document.writeln("Arrow function="+myObject.multiplyArrow(12));
    </script>
   </body>
</html>
```



Traditional function=120
Arrow function=NaN