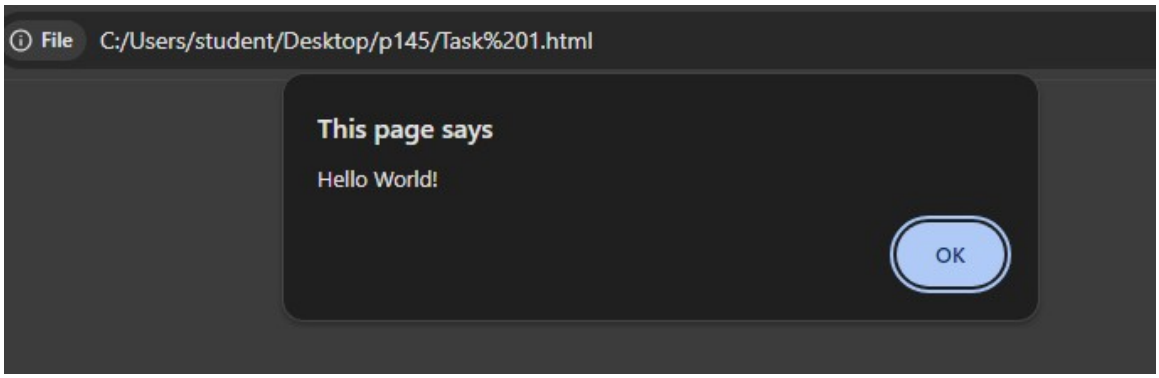
INTRODUCTION TO JS

Task 1:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Document</title>
        <body>
            <script>
                alert("Hello World!");
            </script>
        </body>
    </head>
</html>
```

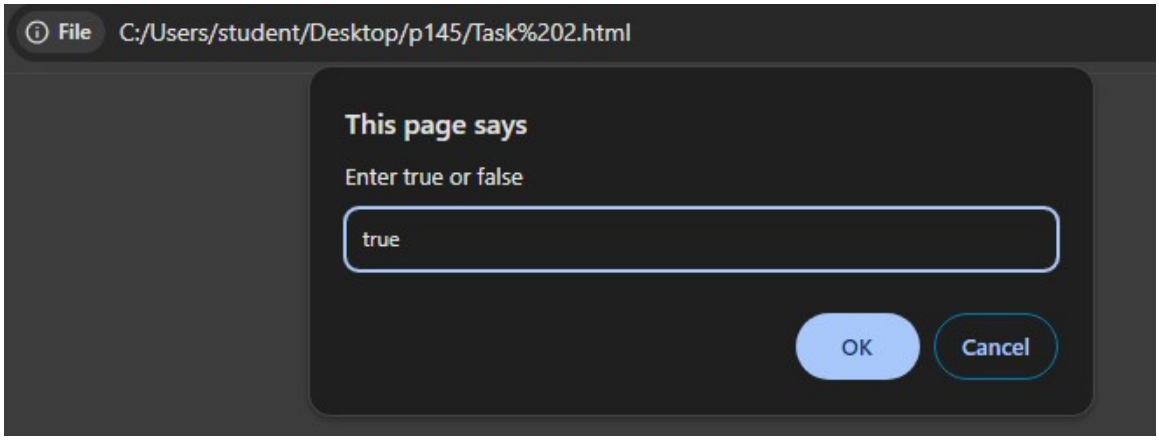
Output:

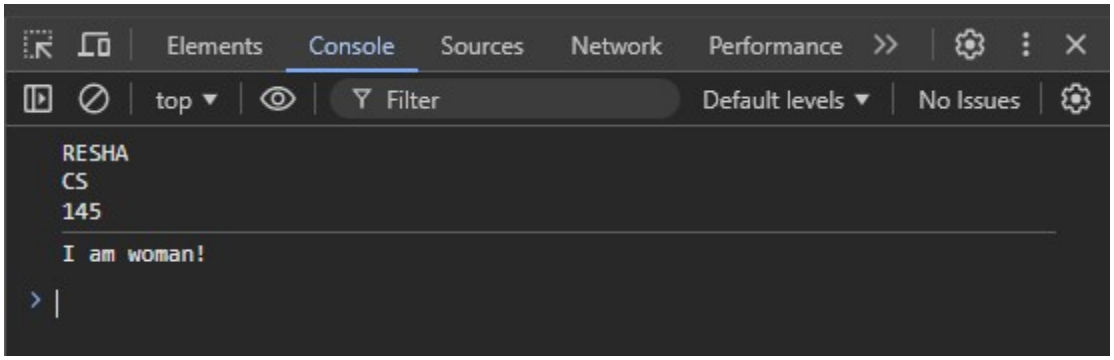


Task 2:

```
<!DOCTYPE html>
<html lang="en">
   <head>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <title>Document</title>
      <body>
         <script>
               var Name= "RESHA";
               var Dept= "CS";
               const RegNo= 145;
            console.log(Name+"\n"+Dept+"\n"+RegNo);
            let isfemale= prompt("Enter true or false");
            if(isfemale==true)
         {
            console.log("I am woman!");
         }else{
            console.log("I am male!");
         }
         </script>
      </body>
   </head>
</html>
```

Output:



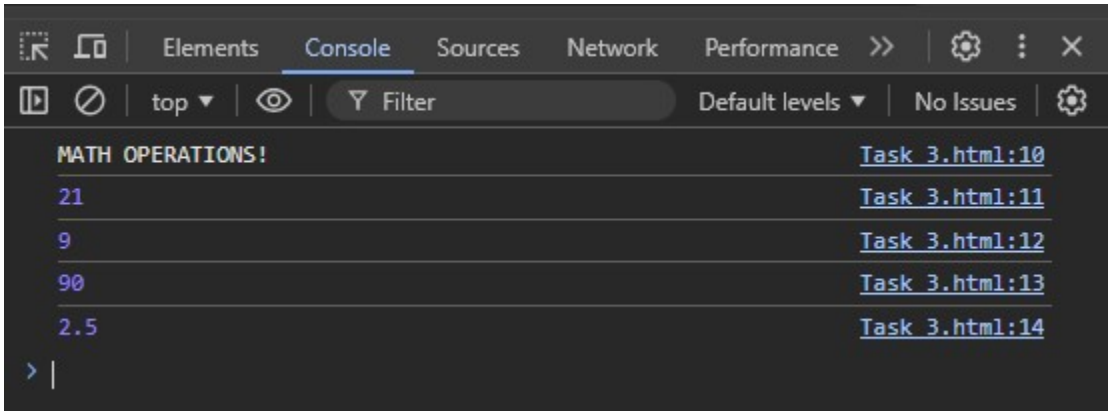


Task 3:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <body>
      <script>
        let r=15,v=6
        console.log("MATH OPERATIONS!")
```

```
            console.log(r+v);
            console.log(r-v);
            console.log(r*v);
            console.log(r/v);
        </script>
    </body>
  </head>
</html>
```

Output:

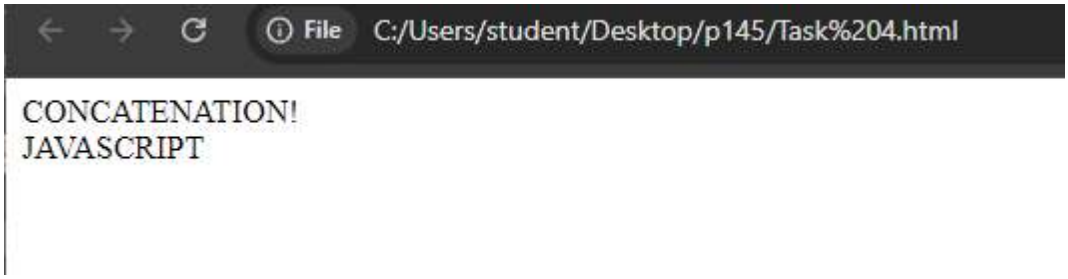


Task 4:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
```

```
<body>
    <script>
        document.writeln("CONCATENATION!"+"<br>");
        document.writeln("JAVA"+"SCRIPT");
    </script>
</body>
</html>
```

Output:

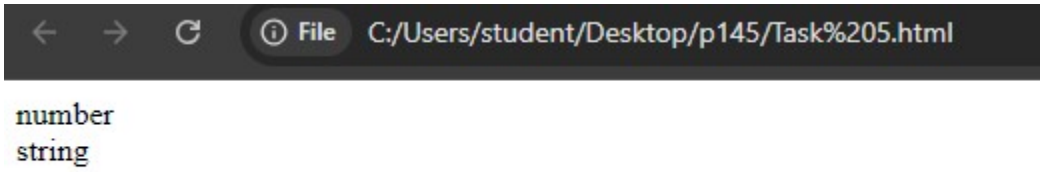


Task 5:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        document.writeln(typeof 12345+"<br>");
```

```
        document.writeln(typeof "RESHA");
    </script>
</body>
</html>
```

Output:



CODE STRUCTURES

Task 6:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        document.writeln("hello!");//single line comment!
        /*document.writeln("multi line comment!");
        document.writeln("thank you!");*/
    </script>
</body>
</html>
```

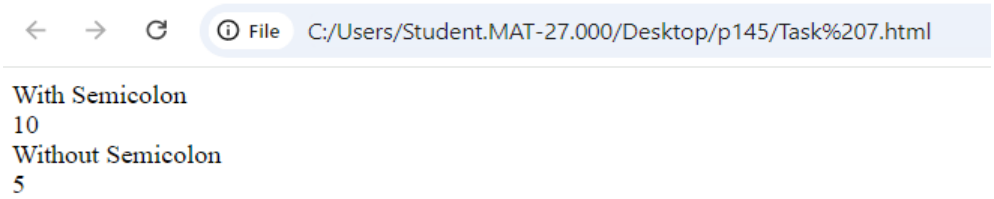
DIFFERENCE BETWEEN SINGLE AND MULTILINE COMMENT:

Starts with // and comments out everything on that line	Starts with /* and ends with */, allowing comments to span multiple lines.
Ideal for short comments or explanations on a single line.	Suitable for longer comments, descriptions, or commenting out blocks of code.
Affects only the current line	Can comment out multiple consecutive lines.
Cannot be nested within other single-line comments.	Cannot be nested within other multiline comments, but can contain single-line comments inside them

Task 7:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      x=10;
      document.writeln("With Semicolon"+"<br>");
      document.writeln(x+"<br>");
      y=5
      document.writeln("Without Semicolon"+"<br>");
      document.writeln(y)
   </script>
</body>
</html>
```

Output:



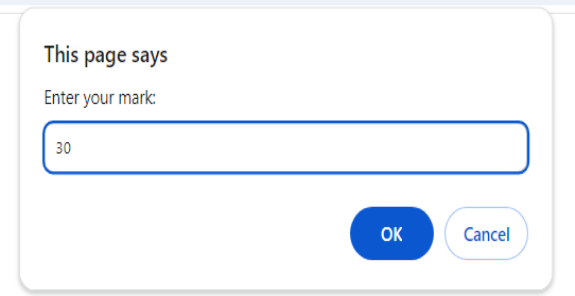
Task 8:

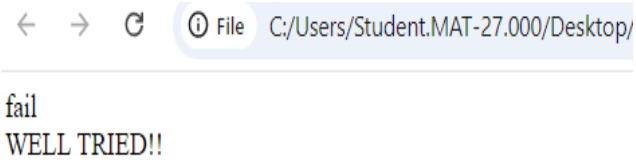
```
<html>
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      let marks=prompt("Enter your mark:")
```

```
        if(marks<=100){
            if(marks<35){
                document.writeln("fail <br>");
                document.writeln("WELL TRIED!!");}
            else{
                document.writeln("pass <br>");
                document.writeln("CONGRATULATIONS!!!");
            }
            }
        else{
        document.writeln("you done your best!");}


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

Output:





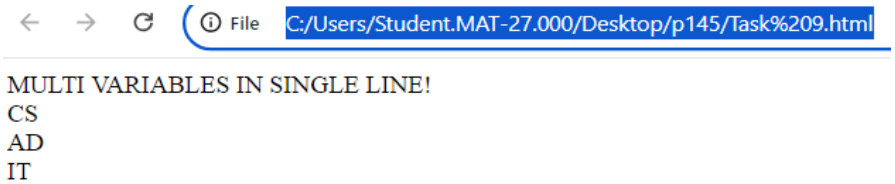
Task 9:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        document.writeln("MULTI VARIABLES IN SINGLE LINE!" +"<br>");

        let dept1="CS",dept2="AD",dept3="IT";
```

```
        document.writeln(dept1 +"<br>");
        document.writeln(dept2 +"<br>");
        document.writeln(dept3 +"<br>");
    </script>
</body>
</html>
```

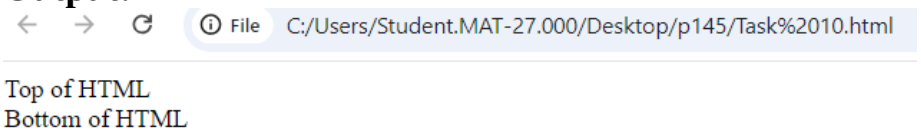
Output:



Task 10:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <script>
        document.writeln("Top of HTML"+"<br>");
    </script>
</head>
<body>
    <script>
        document.writeln("Bottom of HTML");
    </script>
</body>
</html>
```

Output:

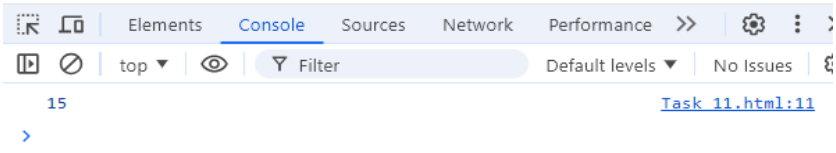


The modern mode, “use strict”, Variables

Task 11:

```
 <!DOCTYPE html>
    <html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <title>Document</title>
    </head>
    <body>
        <script>
            x=15;
            console.log(x);
        </script>
    </body>
    </html>
```

Output:

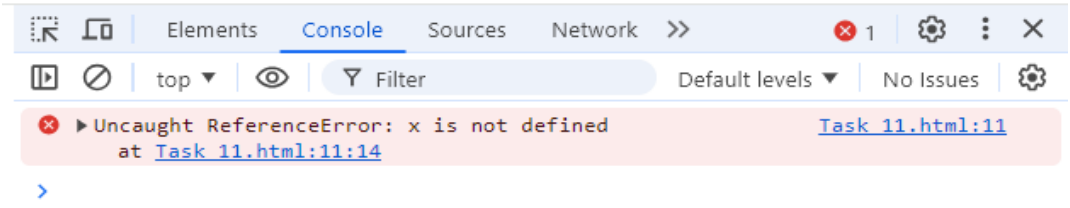


Task 12:

```
 <!DOCTYPE html>
    <html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <title>Document</title>
    </head>
    <body>
        <script>
            'use strict';
            x=15;
            console.log(x);
        </script>
```

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

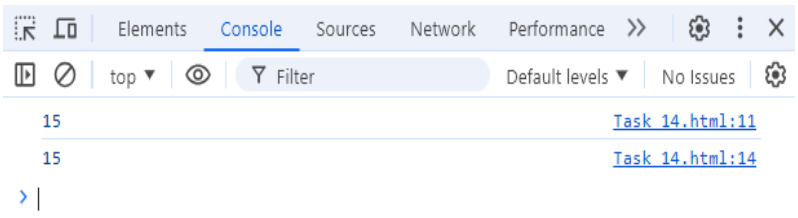
Output:



Task 14:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        x=15;
        console.log(x);
        'use srict';
        y=6;
        console.log(x);
    </script>
</body>
</html>
```

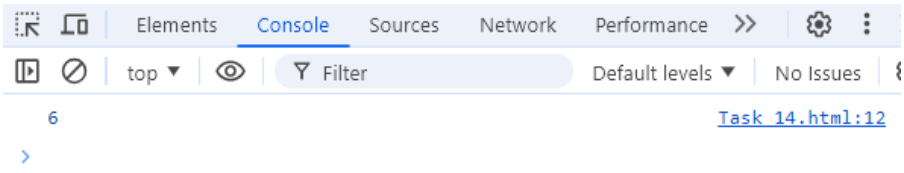
Output:



Task 15:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        'use srict';
        let y=6;
        console.log(x);
    </script>
</body>
</html>
```

Output:



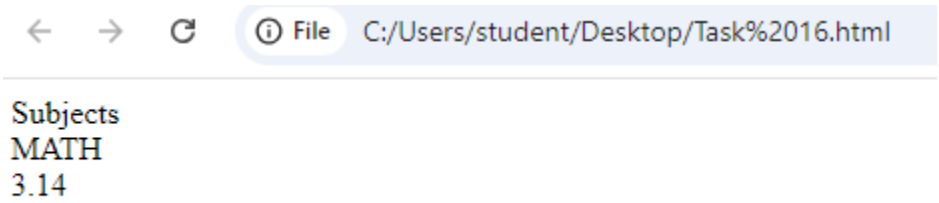
Variables

Task 16:

```
<html>
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      let ip="Subjects"+"<br>";
      document.writeln(ip);
      var subject1="MATH"+"<br>";
      document.writeln(subject1);
      const PI=3.14+"<br>";
      document.writeln(PI);

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

Output:



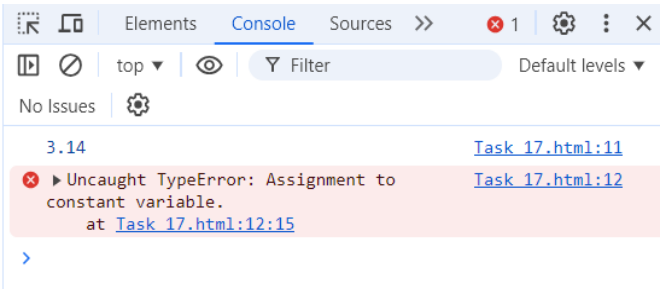
Task 17:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
```

```
<body>
   <script>
         const PI=3.14;
         console.log(PI);
         PI=3.12;
         console.log(PI);

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

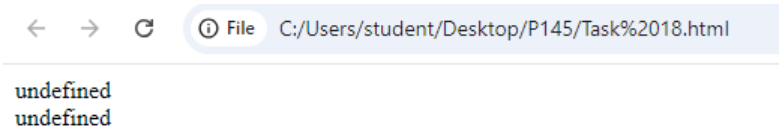
Output:



Task 18:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      let a;
      var b;
      document.writeln(a+"<br>");
      document.writeln(b);
      document.writeln(c);
   </script>
</body>
</html>
```

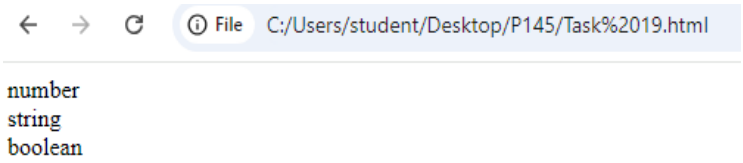
Output:



Task 19:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      document.writeln(typeof 123+"<br>");
      document.writeln(typeof "hello!"+"<br>");
      document.writeln(typeof true);
   </script>
</body>
</html>
```

Output:

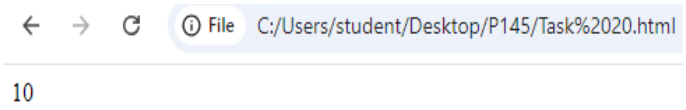


Task 20:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
```

```
  <script>
    var a=10;
    var b=a;
    document.writeln(b);
  </script>
</body>
</html>
```

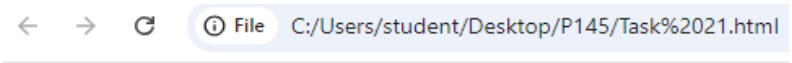
Output:



Task 21:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    let name="resha";
    document.writeln("String:"+name+"<br>");
    let age=19;
    document.writeln("Number:"+age+"<br>");
    let female=true;
    document.writeln("Boolean:"+female+"<br>");
    let notdefined;
    document.writeln("Undefined:"+undefined+"<br>");
    let object
      document.writeln("Object:"+object+"<br>");
  </script>
</body>
</html>
```

Output:

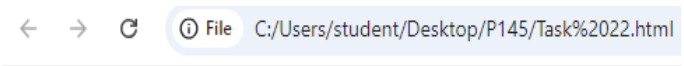


String:resha
Number:19
Boolean:true
Undefined:undefined
Object:undefined

Task 22:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      let a=2;
      b=a;
      document.writeln(typeof b+"<br>");
      let name="resha";
      harini=name;
      document.writeln(typeof harini+"<br>");
      let c=5;
      d=c;
      document.writeln(typeof d+"<br>");
      var apple="fruit";
      banana=apple;
      document.writeln(typeof banana);
   </script>
</body>
</html>
```

Output:

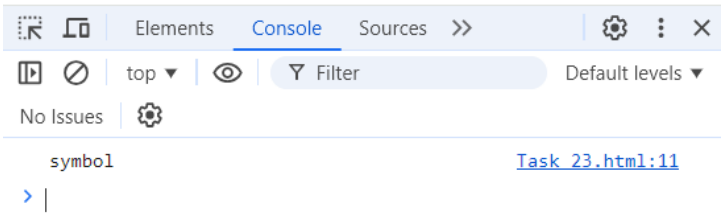


number
string
number
string

Task 23:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      let a=Symbol('&');
      console.log(typeof (a));
   </script>
</body>
</html>
```

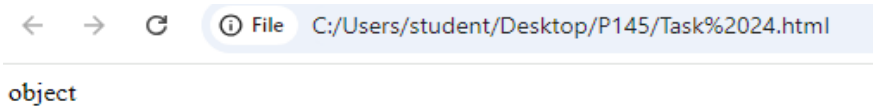
Output:



Task 24:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      let value="null";
      document.writeln(typeof null);
   </script>
</body>
</html>
```

Output:



Task 25:

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

var	let
Function or global scope.	Block scope (inside {}).
Hoisted, but starts as undefined.	Hoisted, but not initialized until declared
Can be redeclared in the same scope	Cannot be redeclared in the same block.

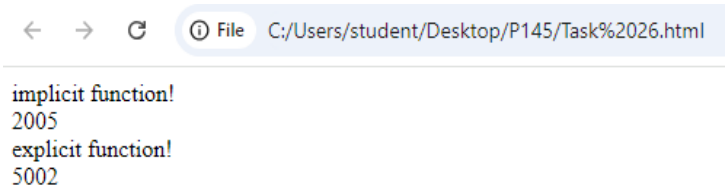
Example:
function testVar() {
if (true) {
var x = 10; // var is function-scoped
} console.log(x); // Works because var is function-scoped
} testVar();
	Example:
	function testLet() {
	if (true) {
	let y = 20; // let is block-scoped
	} console.log(y); // Error: y is not defined
	} testLet();

Task 26:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        document.writeln("implicit function!"+"<br>");
        let str1="2005"+"<br>";
        let num1=str1;
        document.writeln(num1);
        document.writeln("explicit function!"+"<br>");
        let str2=parseInt("5002");
        let num2=str2;
```

```
        document.writeln(num2);
    </script>
</body>
</html>
```

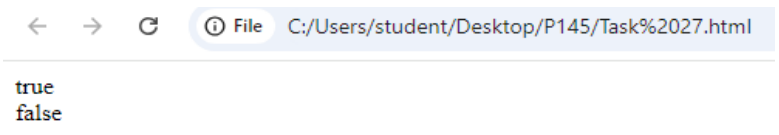
Output:



Task 27:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        let bool1=true;
        let str1=bool1+"";
        document.writeln(str1+"<br>");
        let bool2=false;
        let str2=bool2+"";
        document.writeln(str2);
    </script>
</body>
</html>
```

Output:



Task 28:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        var a=10,b=5;
        sum=a+b;
        document.writeln("Addition:"+sum+"<br>");
        sub=a-b;
        document.writeln("Subtraction:"+sub+"<br>");
        mul=a*b;
        document.writeln("Multiplication:"+mul+"<br>");
        div=a/b;
        document.writeln("Division:"+div+"<br>");
        mod=a%b;
        document.writeln("Modulus:"+mod+"<br>");
    </script>
</body>
</html>
```

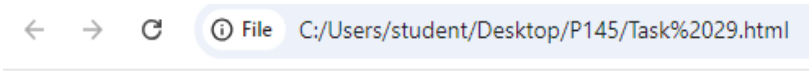
Output:

Task 29:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
```

```
<body>
    <script>
        document.writeln("++operator"+"<br>")
        r=15;
        r=r++ ;
        document.writeln(r+"<br>");
        r=++r ;
        document.writeln(r +"<br>");
        document.writeln("--operator"+"<br>")
        r1=25;
        r1=r1-- ;
        document.writeln(r+"<br>");
        r1=--r1 ;
        document.writeln(r1);
    </script>
</body>
</html>
```

Output:

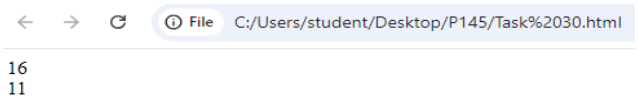


++operator
15
16
--operator
16
24

Task 30:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        document.writeln((5+3)*2+"<br>");
        document.writeln(5+3*2);
    </script>
</body>
</html>
```

Output:



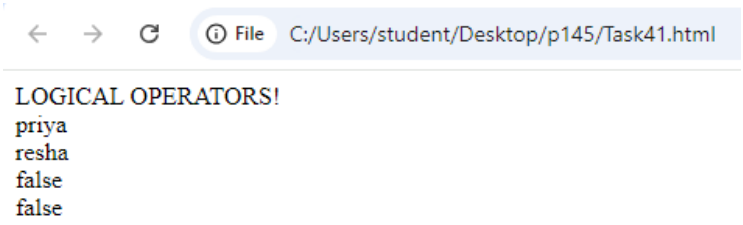
Logical operators, Functions

Logical operators:

Task 41:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      document.writeln("LOGICAL OPERATORS!"+"<br>")
      let c="resha";
      let d="priya";
      document.writeln((c && d)+"<br>")
      document.writeln((c || d)+"<br>")
      document.writeln((!c) +"<br>")
      document.writeln(!d)
   </script>
</body>
</html>
```

Output:

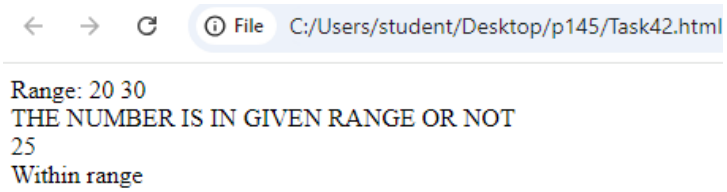


Task 42:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=<device-width>, initial-scale=1.0">
   <title>Document</title>
</head>
```

```
<body>
   <script>
      let v=20,r=30,b=25;
      document.writeln("Range: "+v+" "+r+"<br>")
      document.writeln("THE NUMBER IS IN GIVEN RANGE OR NOT"+"<br>"+b+"<br>")
      if((v<=b)&&(r>=b))
      document.writeln("Within range"+"<br>")
      else
      document.writeln("Without range")
   </script>
</body>
</html>
```

Output:

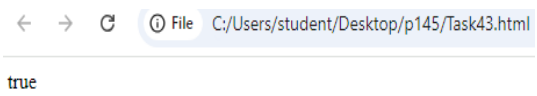


Task 43:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      let a=2;b=4;
      document.writeln(a!=b?true:false);

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

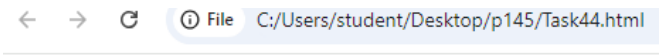
Output:



Task 44:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      let a=5;b=10;
      document.writeln(a&&b+"<br>")
      document.writeln(a||b)
   </script>
</body>
</html>
```

Output:

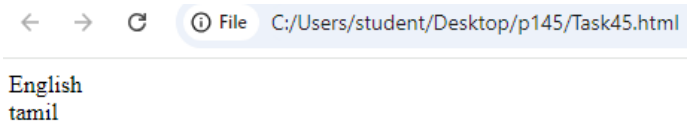



```
10
5
```

Task 45:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      let a="tamil",b="English";
      document.writeln(a&&b+"<br>")
      document.writeln(a||b)
   </script>
</body>
</html>
```

Output:



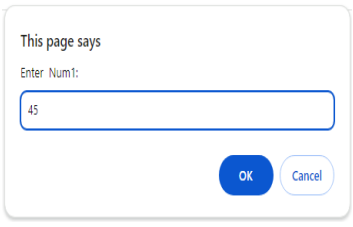
Functions:

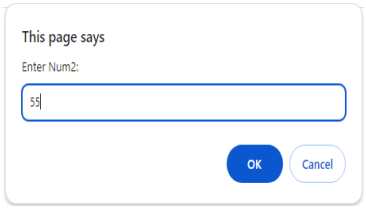
Task 46:

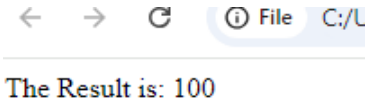
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      let num1=parseInt(prompt("Enter  Num1:"))
      let num2=parseInt(prompt("Enter Num2:"))
      function sum(num1,num2){
         return num1+num2;

      }
      let result=sum(num1,num2);
      document.writeln("The Result is: "+result);
   </script>
</body>
</html>
```

Output:





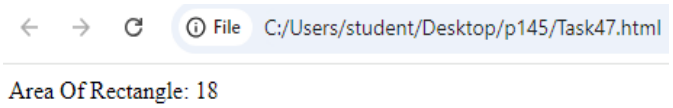


Task 47:

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      let l=2,b=9;
      function rectangle(l,b){
         return l*b;
      }
      let Area=rectangle(l,b)
      document.writeln("Area Of Rectangle: "+Area)
   </script>
</body>
</html>
```

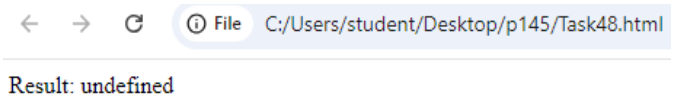
Output:



Task 48:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      function fu(){
         document.writeln("Result: ");
      }
      let result=fu()
      document.writeln(result)
   </script>
</body>
</html>
```

Output:

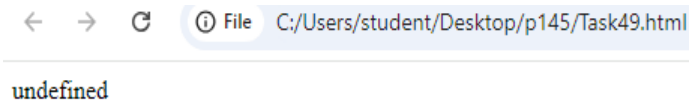


Task 49:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      function fu(){

      }
      let result=fu()
      document.writeln(result)
   </script>
</body>
</html>
```

Output:



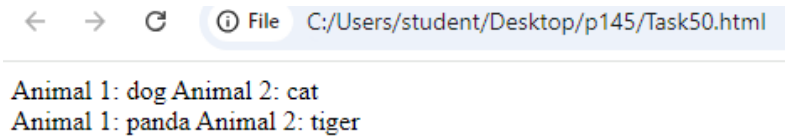
Task 50:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
```

```
</head>
<body>
  <script>
    var Animal1="dog";
    var Animal2="cat";
    function animals(Animal1,Animal2){
      document.writeln(`Animal 1: ${Animal1}  Animal 2: ${Animal2}`+"<br>");
    }
    animals(Animal1,Animal2)
    animals("panda","tiger")

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

Output:



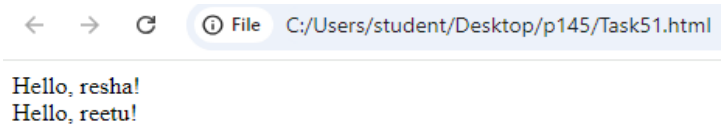
Arrow Functions:

Task 51:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    let greet=(name)=>{
      return `Hello, ${name}!<br>`
    }
      document.writeln(greet("resha"))
      document.writeln(greet("reetu"))
```

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

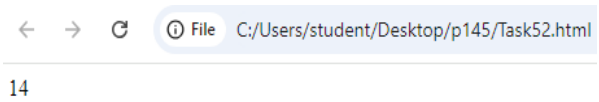
Output:



Task 52:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        let add,num1,num2;
        let greet=(num1,num2)=>{
            return num1+num2;
        }
        let res=greet(6,8)
        document.writeln(res)
    </script>
</body>
</html>
```

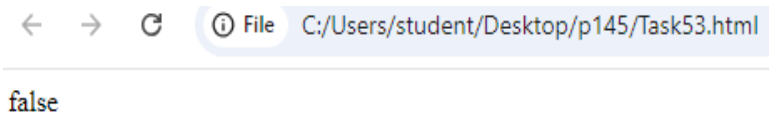
Output:



Task 53:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <script>
      let num=2;
      let greet=(num)=>{
         return num%2==0;
      }
      let res=greet(7)
      document.writeln(res)
   </script>
</body>
</html>
```

Output:



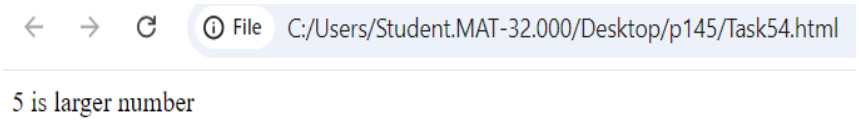
Task 54:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        let num1=3,num2=5;
        let maxValue=(num1,num2)=>{
            if (num1<num2)
            return num2;
            else
            return  num1;
```

```
        }
        document.writeln(maxValue(num1,num2)+"  is larger number ")

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

Output:

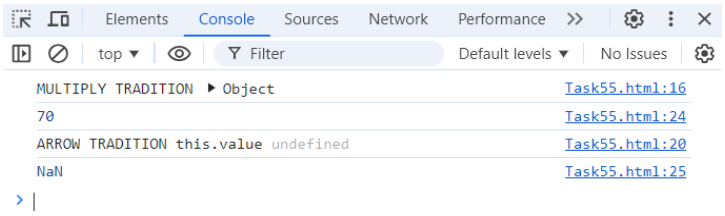


Task 55:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        let MultiplyTradition;
        let ArrowTradition;
        const myObject={
            value:10,

        multiplyTradition:function(num){
            console.log("MULTIPLY TRADITION",this)
            return this.value*num;
        },
        arrowTradition:(num)=>{
            console.log("ARROW TRADITION","this.value",this.value)
            return this.value*num;
        },
    };
        console.log(myObject.multiplyTradition(7));
        console.log(myObject.arrowTradition(9));
    </script>
</body>
</html>
```

Output:


Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
MULTIPLY TRADITION ▶ Object
Task55.html:16
70
Task55.html:24
ARROW TRADITION this.value undefined
Task55.html:20
NaN
Task55.html:25