Variable Difference:

1) Scope
   var has global scope
   Let and const have block scope

2) Re declaration
   Var can be re declared
   Let and const can't be re declared

3) Re assignment
   Var and let can be re assigned
   Const can't be re assigned

Operators

Javascript operators are used to perform different types of mathematical and logical computations.

(or)

In JavaScript, an **operator** is a symbol that performs an operation on one or more operands, such as variables or values, and returns a result. Let us take a simple expression **4 + 5** is equal to 9. Here 4 and 5 are called **operands**, and '+' is called the **operator**.

Types:

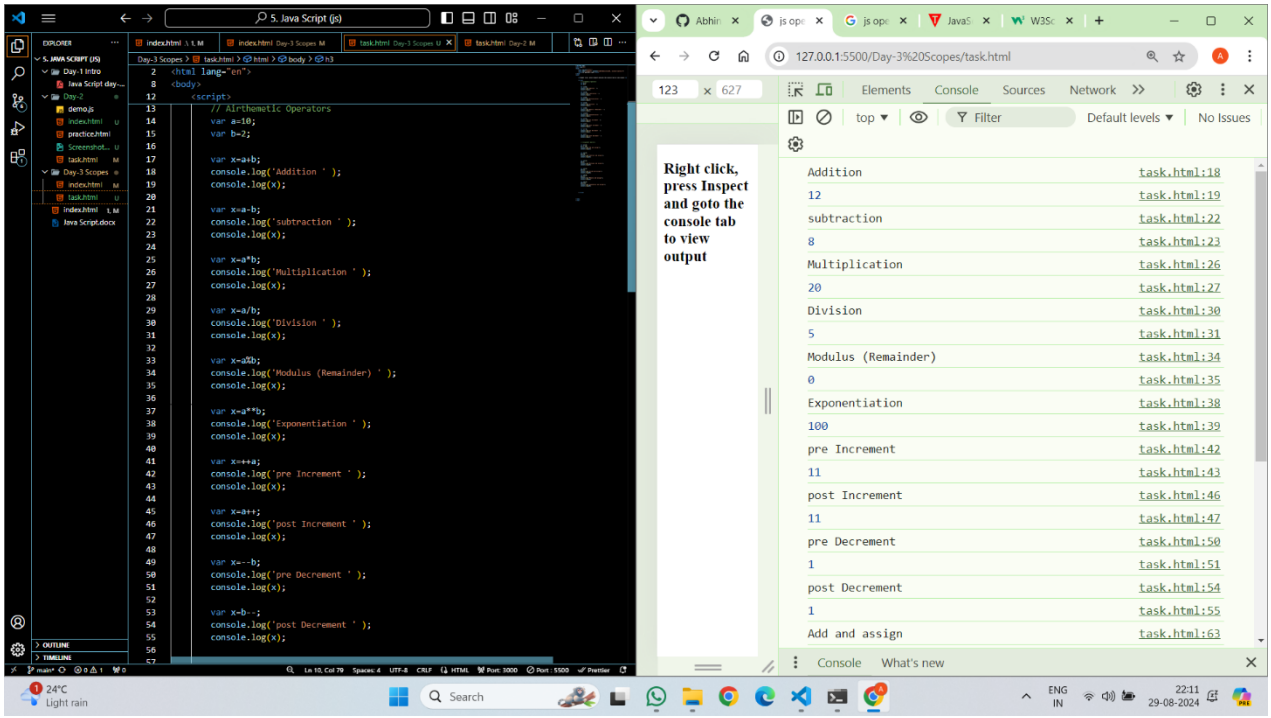
1. Airthmetic Operators
2. Assignment Operator
3. Comparision Operator
4. Logical Operator
5. Ternary Operator
6. Bitwise Oberator
7. String Operator
8. Typeof Operator

Arithmetic operators

Arithmetic operators are used to perform **arithmetic operations** between variables or values.

Operator	Name	Example
+	Addition	3 + 4 // 7
-	Subtraction	5 - 3 // 2
*	Multiplication	2 * 3 // 6
/	Division	4 / 2 // 2
%	Remainder	5 % 2 // 1
++	Increment (increments by **1**)	++5 or 5++ // 6
--	Decrement (decrements by **1**)	--4 or 4-- // 3
**	Exponentiation (Power)	4 ** 2 // 16

Example:

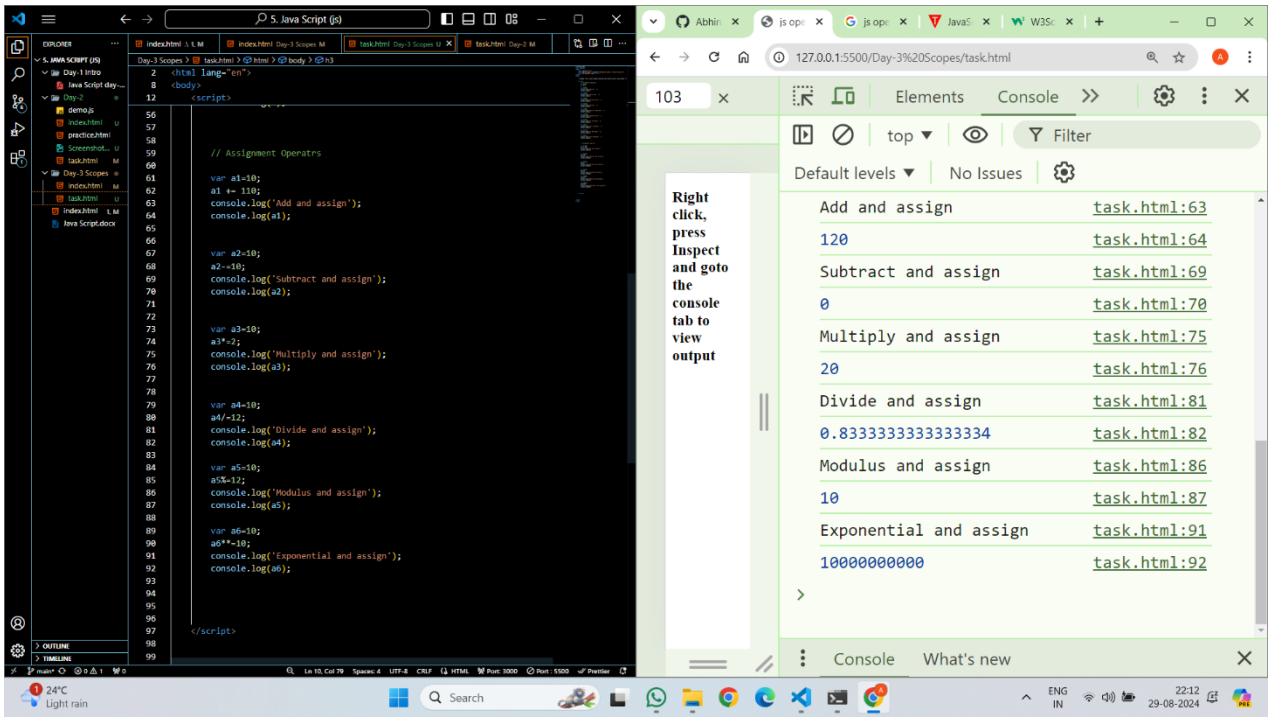


Assignment Operators:

We use assignment operators to **assign** values to variables.

Operator	Name	Example
=	Assignment Operator	a = 7;
+=	Addition Assignment	a += 5; // a = a + 5
-=	Subtraction Assignment	a -= 2; // a = a - 2
*=	Multiplication Assignment	a *= 3; // a = a * 3
/=	Division Assignment	a /= 2; // a = a / 2
%=	Remainder Assignment	a %= 2; // a = a % 2
**=	Exponentiation Assignment	a **= 2; // a = a**2

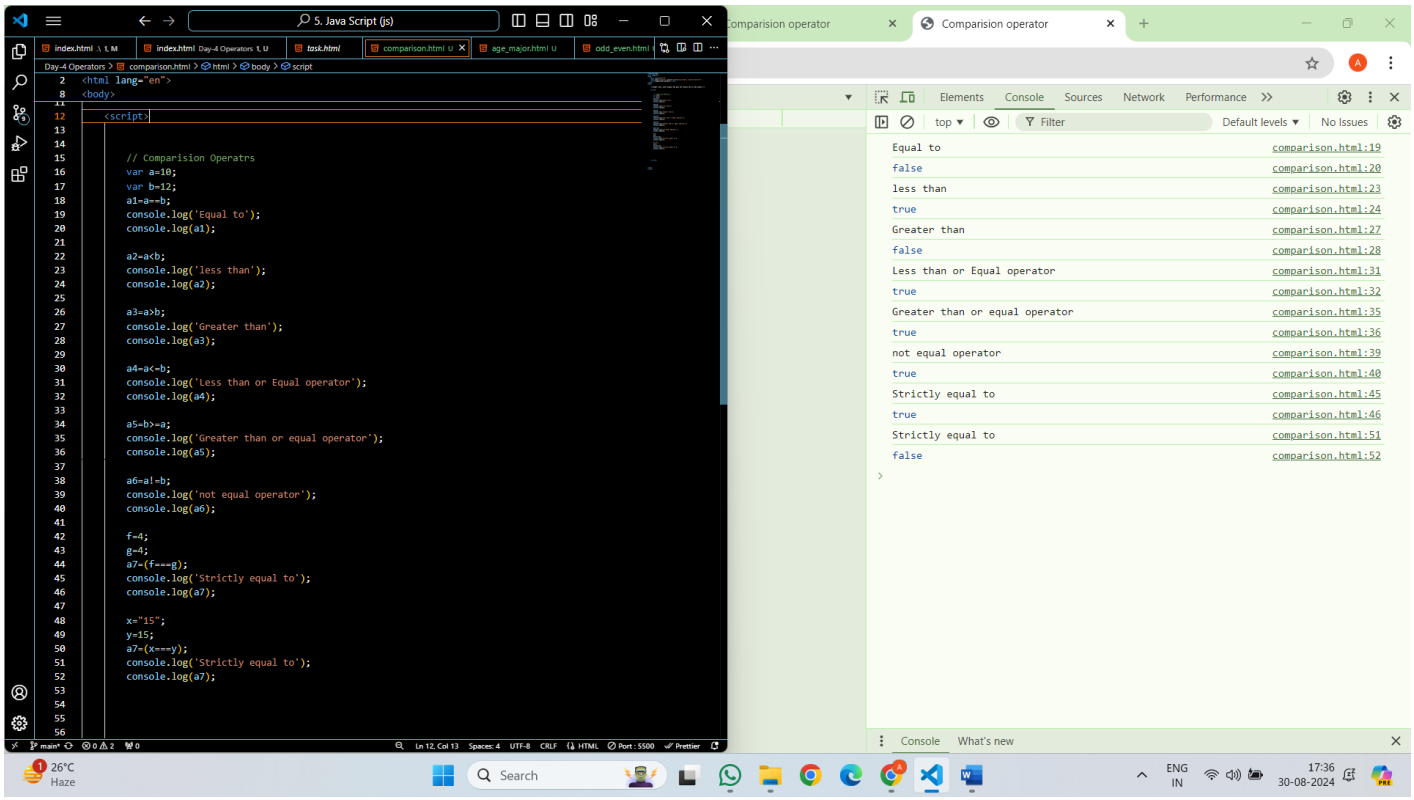
Example:



Comparision Operators

Comparison operators are used in logical statements to determine equality or difference between variables or values.

Operator	Meaning	Example
==	Equal to	3 == 5 // false
!=	Not equal to	3 != 4 // true
===	Strictly equal to	3 === "3" // false
!==	Strictly not equal to	3 !== "3" // true
>	Greater than	4 > 4 // false
<	Less than	3 < 3 // false
>=	Greater than or equal to	4 >= 4 // true
<=	Less than or equal to	3 <= 3 // true

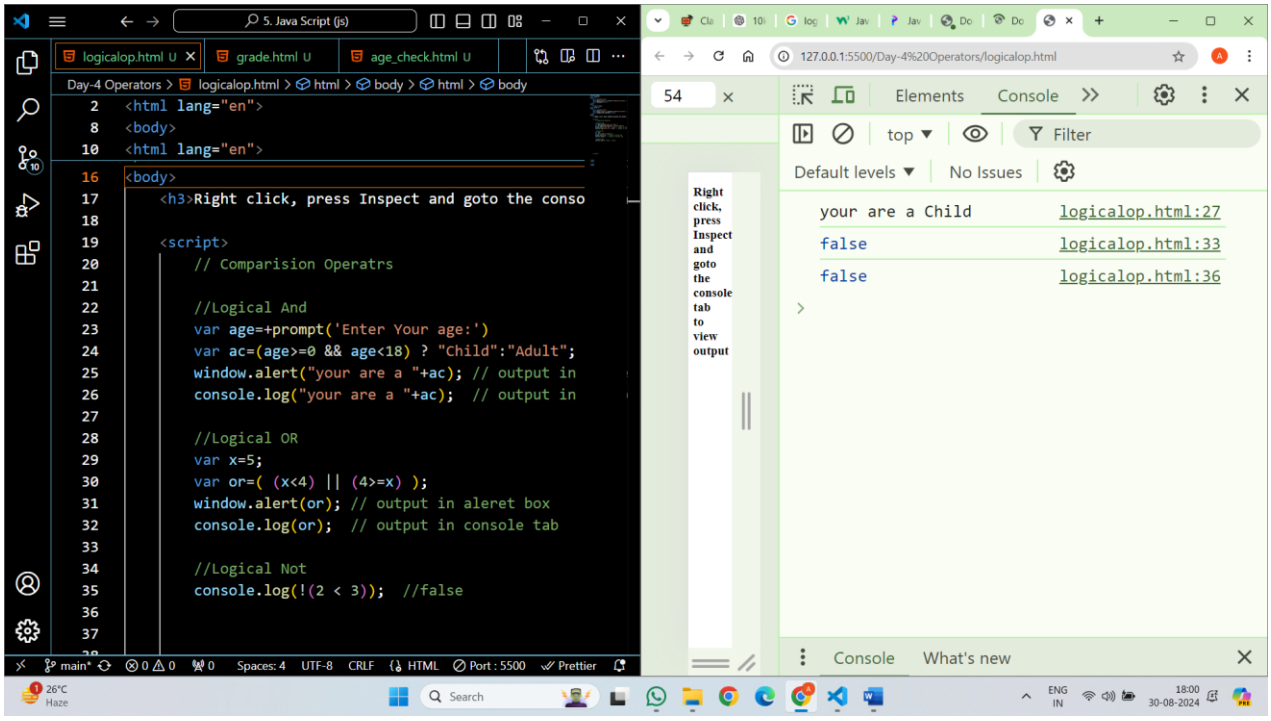


Logical Operator:

Logical operators return a boolean value by evaluating boolean expressions.

1. Logical And Operator: The logical AND operator **&&** returns `true` if both the expressions are `true`.
2. Logical OR Operator: The logical OR operator || returns true if at least one expression is true.
3. Logical Not Operator: The logical NOT operator **!** returns true if the specified expression is false and vice versa.

Operator	Syntax	Description
`&&` (Logical AND)	`expression1 && expression2`	`true` only if both `expression1` and `expression2` are `true`
`||` (Logical OR)	`expression1 || expression2`	`true` if either `expression1` or `expression2` is `true`
`!` (Logical NOT)	`!expression`	`false` if `expression` is `true` and vice versa



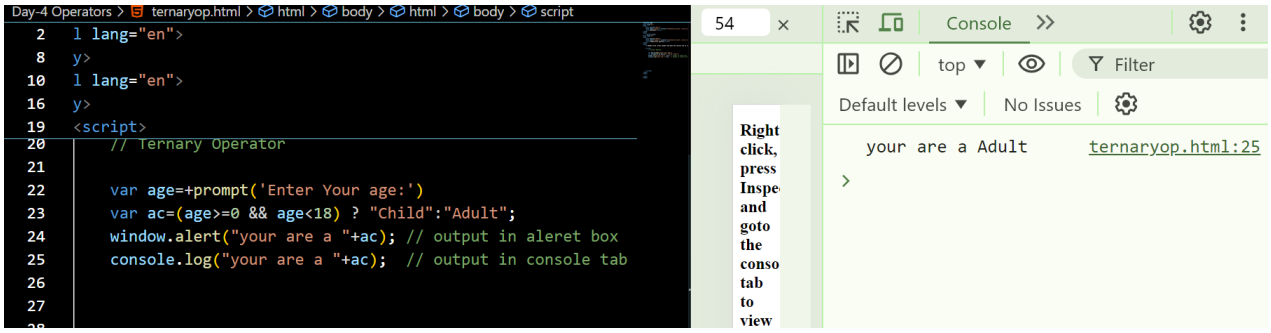
Ternary operator:

The Ternary Operator in JavaScript is a shortcut for writing simple if-else statements. It's also known as the Conditional Operator because it works based on a condition. The ternary operator allows you to quickly decide between two values depending on whether a condition is true or false.

Syntax:

condition ? trueExpression : falseExpression

Example:



Nullish coalescing operator (??)

is a logical operator that returns its right-hand side operand when its left-hand side operand is null or undefined, and otherwise returns its left-hand side operand. It's commonly used to provide default values for variables.

Example:

```
<script>
      //Nulish Coalescing Operator
       var a=null;
       var b=a ?? "Some Content";
       console.log(b);     // Some Content
  </script>
```

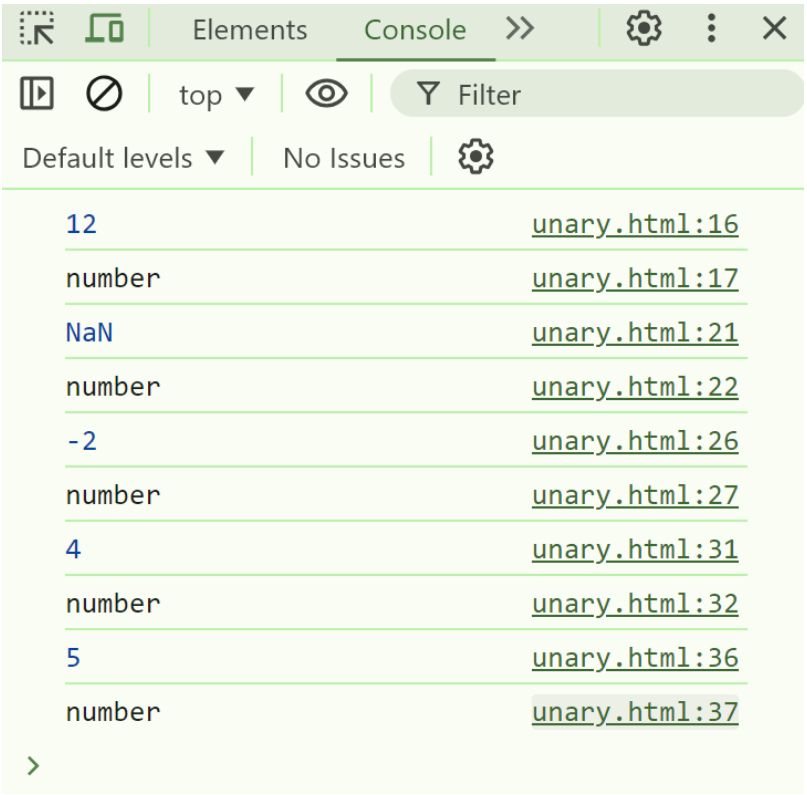
Unary Operator:

- Unary operators in JavaScript are unique operators that consider a single input and carry out all possible operations.
- The Unary plus, unary minus, prefix increments, postfix increments, postfix decrements, and prefix decrements are examples of these operators. These operators are either put before or after the operand.
- The unary operators are more effective in executing functions than JavaScript; they are more popular. Unary operators are flexible and versatile since they cannot be overridden.

Unary Operators	Operator's Name	Operators Description
+x	Unary Plus	The operator converts an input value into a number
-x	Unary Minus	The operator converts a value into a number and negates it
++x	Increment Operator (Prefix)	The operator uses to inserts one value before the incremental value by one
--x	Decrement Operator (Prefix)	The operator Subtracts one value from the given input value before
x++	Increment Operator (Postfix)	The operator uses to inserts one value after the incremental value by one
x--	Decrement Operator (Postfix)	The operator subtracts one value before the incremental value by one.

Example:

```
<script>
    // Using unary plus to convert string to number
    let str1 = "12";
    let num = +str1;
    console.log(num);
    console.log(typeof (num))  // Here we are using typeof operator

    // "Abhinav" cannot be converted to a number
    let str2 = +"Abhinav";
    console.log(str2);
    console.log(typeof (str2))

    let s1='2'
    let n1 = -s1;
    console.log(n1);
    console.log(typeof (n1))

    let s2='3'
    let n2 = ++s2;
    console.log(n2);
    console.log(typeof (n2))

    let s3='5'
    let n3 = s3++;
    console.log(n3);
    console.log(typeof (n3))
</script>
```



Type Coercion

Type coercion refers to the automatic or implicit conversion of values from one data type to another.

In programming, type conversion is the process of converting data of one type to another. For example, converting string data to number.

There are two types of type conversion in JavaScript:

- **Implicit Conversion** - Automatic type conversion.
- **Explicit Conversion** - Manual type conversion.

Explicit Type Conversion

JavaScript type conversion, allowing you to convert values from one data type to another.

1. **String()**: Converts a value to a string.

```
let num = 123;
let str = String(num);
console.log(str);
// Output: "123"
```

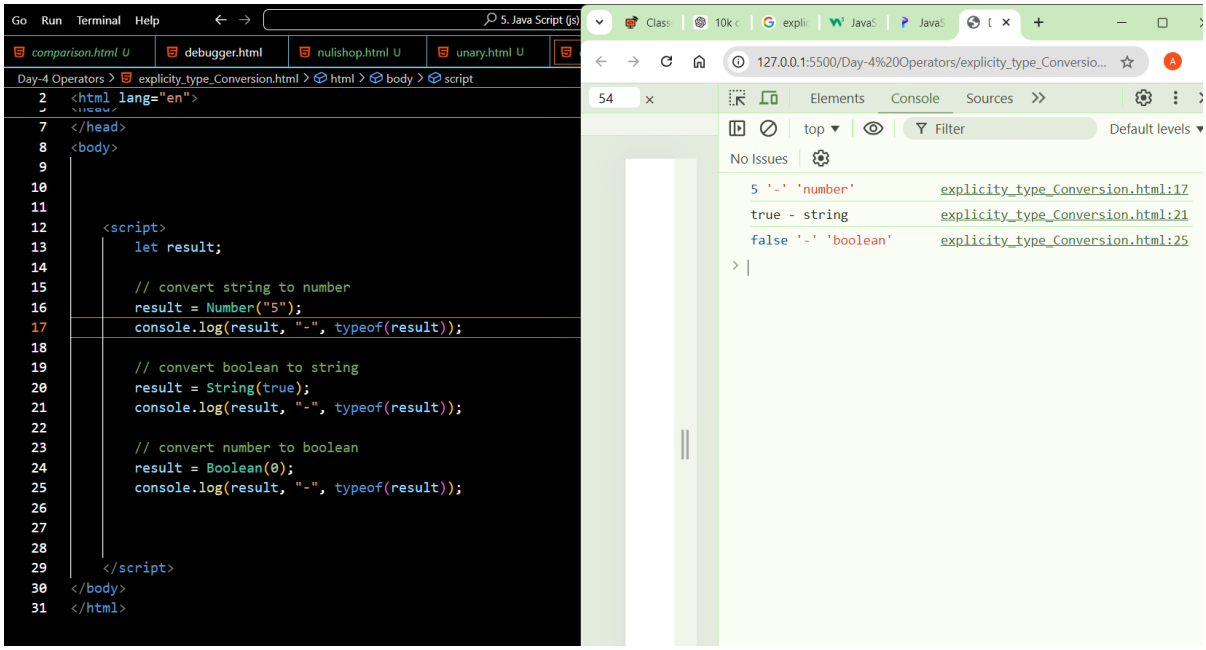
2. **Number()**: Converts a value to a number.

```
let str = "123";
let num = Number(str);
console.log(num); // Output: 123
```

3. **Boolean()**: Converts a value to a boolean.

```
let num = 0;
let bool = Boolean(num);
console.log(bool); // Output: false
```

Example:



How to take or get input from Users:

Var a= +prompt('Enter Your Data');

In JavaScript, values are categorized as either "truthy" or "falsy"

Falsy Values:

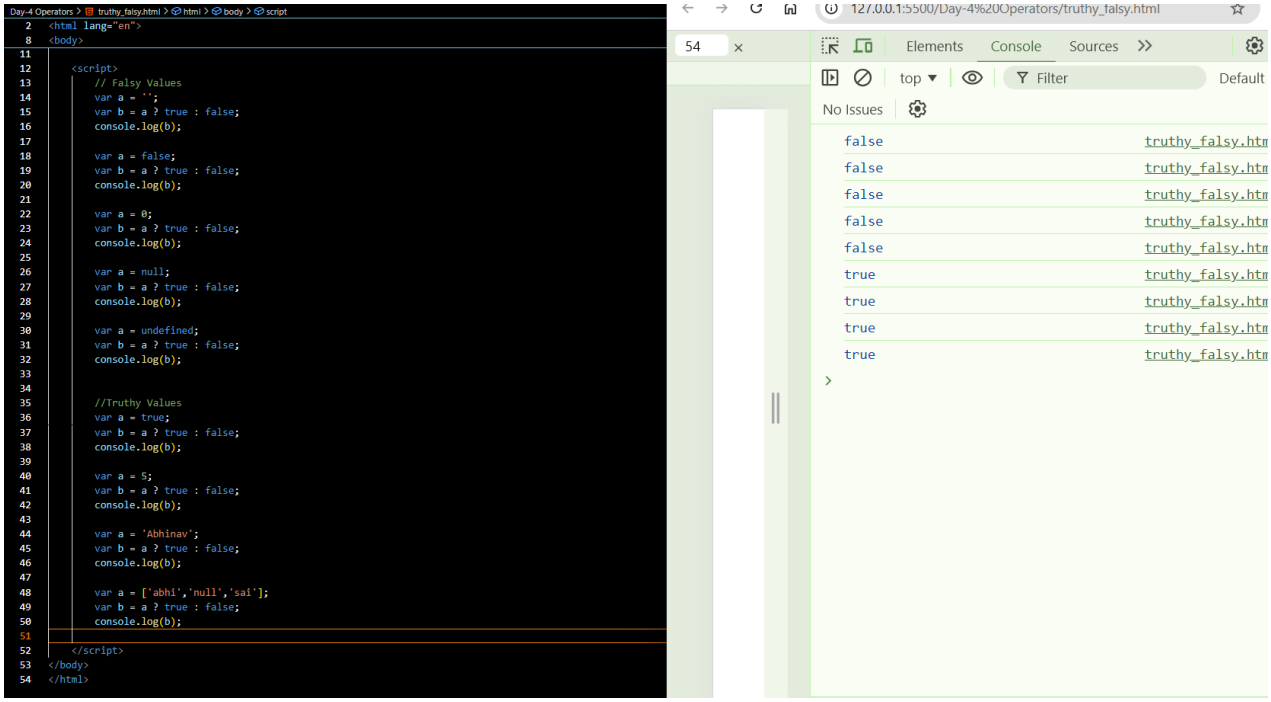
1. **false**: The boolean value false itself.
2. **0**: The number zero.
3. **""**: Empty string.
4. **null**: The absence of any value.
5. **undefined**: A variable that has not been assigned a value or a property that does not exist.
6. **NaN**: Not-a-Number.

Truthy Values:

1. **true**: The boolean value true itself.
2. **Non-zero numbers**: Any number other than 0 (including negative numbers and decimals).
3. **Non-empty strings**: Any string with at least one character.
4. **Non-empty arrays**: Arrays with at least one element.

5. **Objects**: Any object (including functions and arrays) is truthy, even if it's empty.
6. **Functions**: Any function is truthy, even if it doesn't return anything.

Check Truthy, Falsy values using ternary operator:

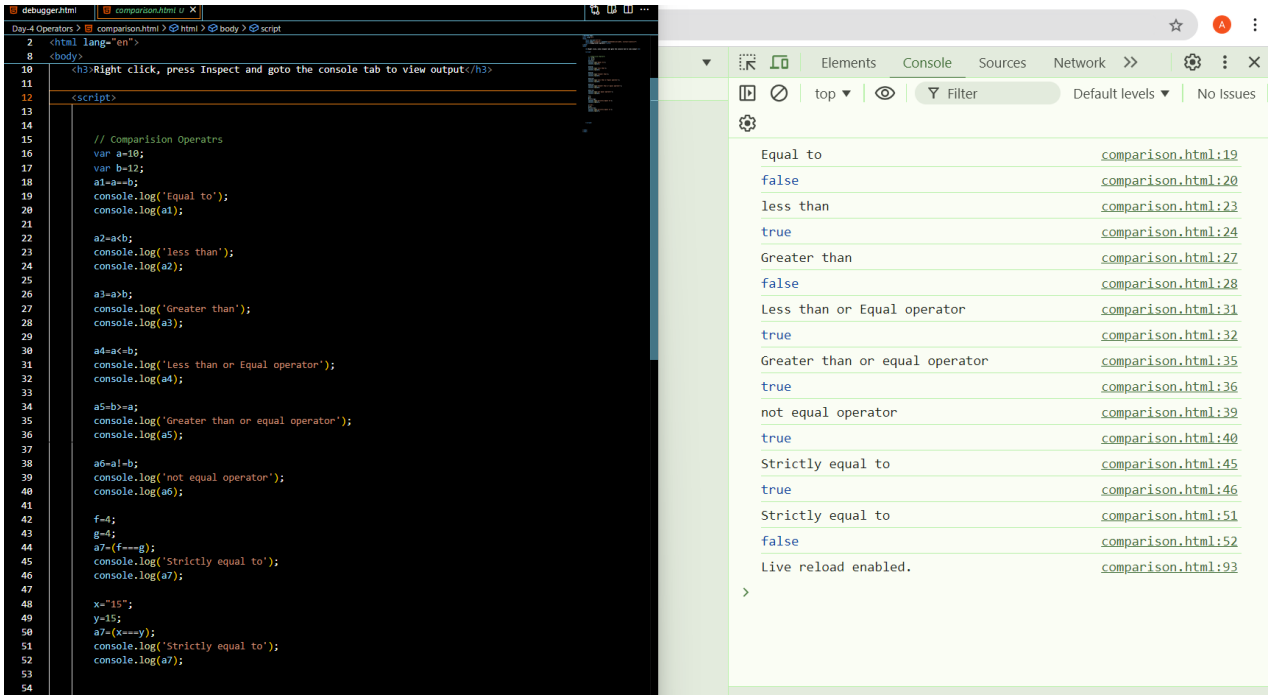


```
Day-4 Operators > truthy_falsy.html > html > body > script
<html lang="en">
<body>

    <script>
        // Falsy Values
        var a = '';
        var b = a ? true : false;
        console.log(b);

        var a = false;
        var b = a ? true : false;
        console.log(b);

        var a = 0;
        var b = a ? true : false;
        console.log(b);

        var a = null;
        var b = a ? true : false;
        console.log(b);

        var a = undefined;
        var b = a ? true : false;
        console.log(b);


        //Truthy Values
        var a = true;
        var b = a ? true : false;
        console.log(b);

        var a = 5;
        var b = a ? true : false;
        console.log(b);

        var a = 'Abhinav';
        var b = a ? true : false;
        console.log(b);

        var a = ['abhi','null','sai'];
        var b = a ? true : false;
        console.log(b);

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

```
127.0.0.1:5500/Day-4%20Operators/truthy_falsy.html
54
Elements  Console  Sources
top  Filter  Default
No Issues
false  truthy_falsy.htm
false  truthy_falsy.htm
false  truthy_falsy.htm
false  truthy_falsy.htm
false  truthy_falsy.htm
true  truthy_falsy.htm
true  truthy_falsy.htm
true  truthy_falsy.htm
true  truthy_falsy.htm
```

Tasks:

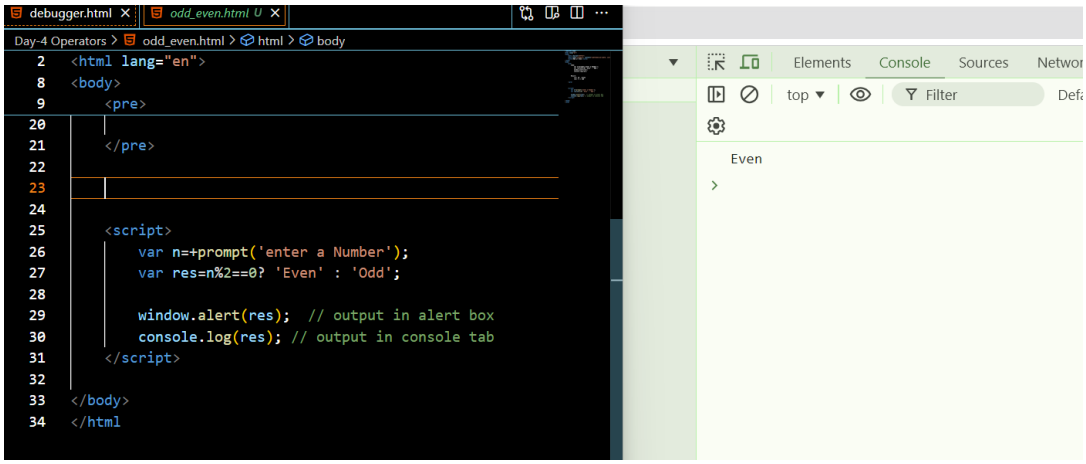
1. Write a JavaScript script that compares two variables using different comparison operators (==, ===, !=, !==, >, <, >=, <=) and prints the results.



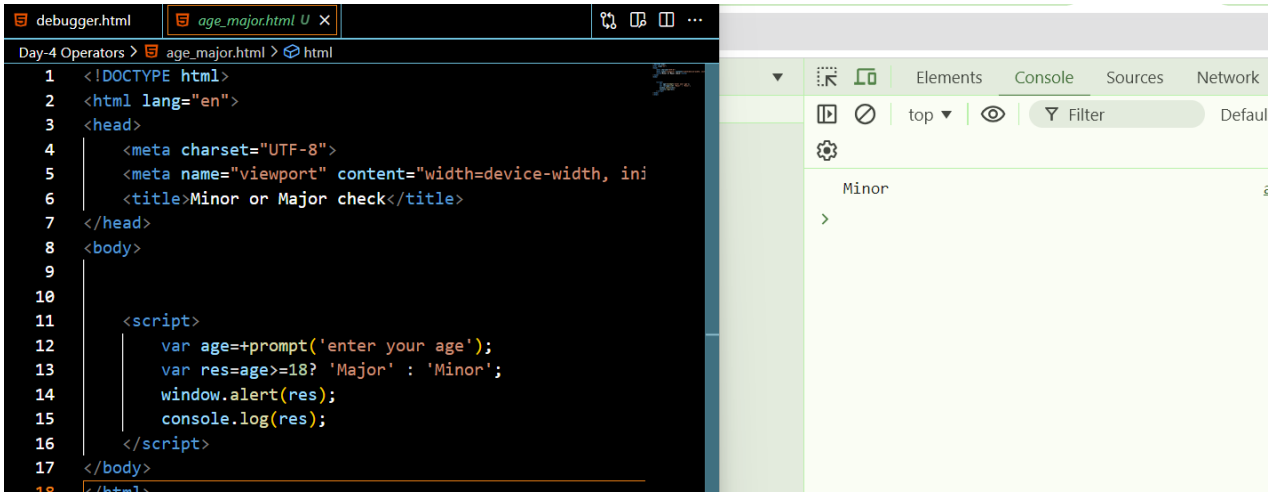
```
debugger.html  comparison.html U
Day-4 Operators > comparison.html > html > body > script
<html lang="en">
<body>
    <h3>Right click, press Inspect and goto the console tab to view output</h3>
    <script>


        // Comparision Operatrs
        var a=10;
        var b=12;
        a1=a==b;
        console.log('Equal to');
        console.log(a1);

        a2=a<b;
        console.log('less than');
        console.log(a2);

        a3=a>b;
        console.log('Greater than');
        console.log(a3);

        a4=a<=b;
        console.log('Less than or Equal operator');
        console.log(a4);

        a5=b>=a;
        console.log('Greater than or equal operator');
        console.log(a5);

        a6=a!=b;
        console.log('not equal operator');
        console.log(a6);

        f=4;
        g=4;
        a7=(f===g);
        console.log('Strictly equal to');
        console.log(a7);

        x="15";
        y=15;
        a7=(x===y);
        console.log('Strictly equal to');
        console.log(a7);
```

```
Elements  Console  Sources  Network
top  Filter  Default levels  No Issues
Equal to  comparison.html:19
false  comparison.html:20
less than  comparison.html:23
true  comparison.html:24
Greater than  comparison.html:27
false  comparison.html:28
Less than or Equal operator  comparison.html:31
true  comparison.html:32
Greater than or equal operator  comparison.html:35
true  comparison.html:36
not equal operator  comparison.html:39
true  comparison.html:40
Strictly equal to  comparison.html:45
true  comparison.html:46
Strictly equal to  comparison.html:51
false  comparison.html:52
Live reload enabled.  comparison.html:93
```

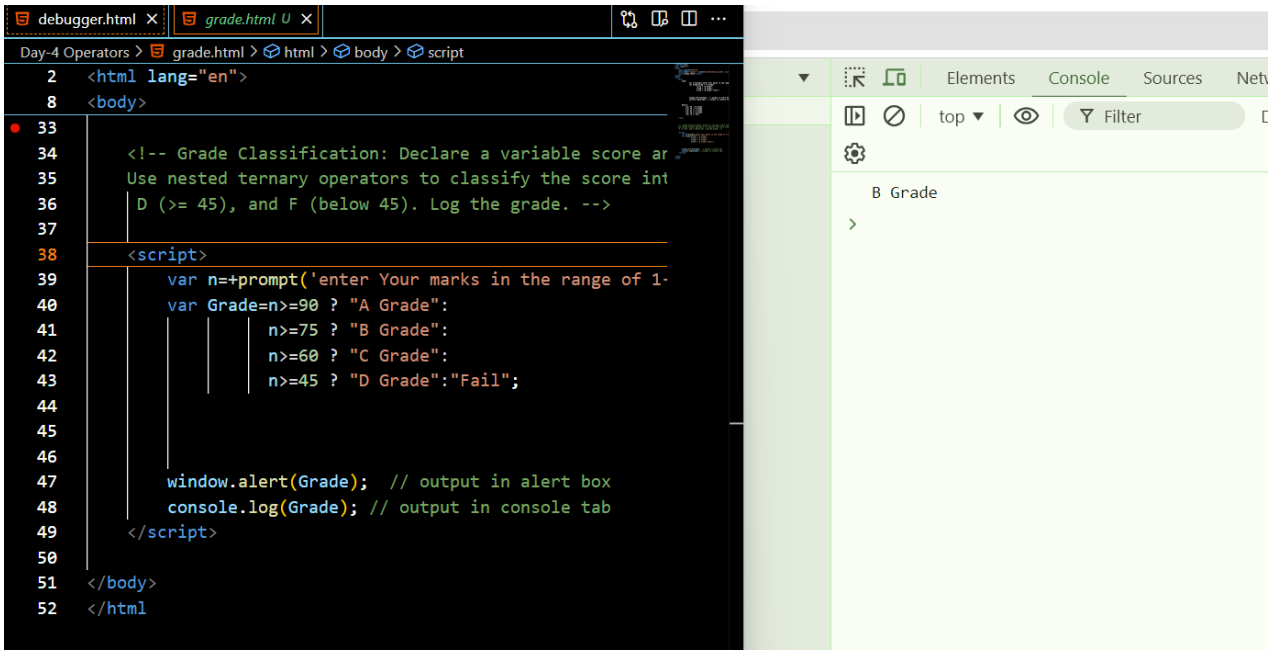
2. Write a JavaScript script that uses the ternary operator to determine if a number is even or odd.



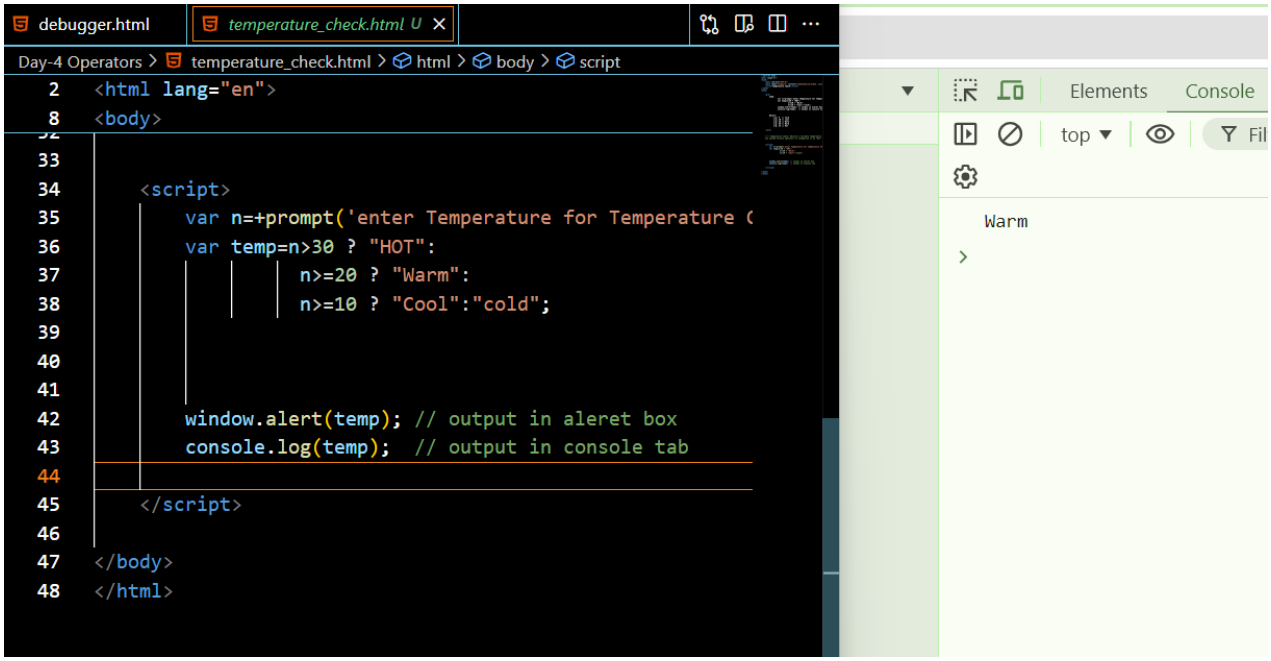
3. Expand the script to include a ternary operation that checks if a user is an adult (18+) or a minor.



4. Grade Classification: Declare a variable score and set it to a value between 0 and 100. Use nested ternary operators to classify the score into grades: A (>= 90), B (>= 75), C (>= 60), D (>= 45), and F (below 45). Log the grade.



5. Temperature Check: Declare a variable temperature and use nested ternary operators to categorize it as "Hot" (above 30), "Warm" (20-30), "Cool" (10-19), and "Cold" (below 10). Log the result.



6. Age Group: Declare a variable age and use the ternary operator to classify the age into "Child" (0-12), "Teen" (13-19), "Adult" (20-64), and "Senior" (65 and above). Log the result.