

## 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:

Right click, press Inspect and goto the console tab to view output

Operation	Result	File Location
Addition	12	task.html:18
subtraction	8	task.html:22
Multiplication	20	task.html:26
Division	5	task.html:30
Modulus (Remainder)	0	task.html:34
Exponentiation	100	task.html:38
pre Increment	11	task.html:42
post Increment	11	task.html:46
pre Decrement	1	task.html:50
post Decrement	1	task.html:54
Add and assign		task.html:63

# Assignment Operators:

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

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

## Example:

The screenshot shows a web browser window with a JavaScript file loaded. The console displays the results of various assignment operations. A text box on the left side of the console says: "Right click, press Inspect and goto the console tab to view output".

Operation	Result
Add and assign	120
Subtract and assign	0
Multiply and assign	20
Divide and assign	0.8333333333333334
Modulus and assign	10
Exponential and assign	10000000000

# Comparison Operators

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

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

The screenshot displays a web browser window with a single tab titled 'Comparison operator'. The browser's console shows the output of a JavaScript script. The script, visible in the background code editor, defines variables and uses various comparison operators to log results to the console. The console output is as follows:

```
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
```

## 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
<code>&amp;&amp;</code> (Logical AND)	<code>expression1 &amp;&amp; expression2</code>	<code>true</code> only if both <code>expression1</code> and <code>expression2</code> are <code>true</code>
<code>  </code> (Logical OR)	<code>expression1    expression2</code>	<code>true</code> if either <code>expression1</code> or <code>expression2</code> is <code>true</code>
<code>!</code> (Logical NOT)	<code>!expression</code>	<code>false</code> if <code>expression</code> is <code>true</code> and vice versa

The screenshot displays a web browser window with the address bar showing `127.0.0.1:5500/Day-4%20Operators/logicalop.html`. The browser's console shows the following output:

```
your are a Child logicalop.html:27
false logicalop.html:33
false logicalop.html:36
```

The code editor on the left shows the following JavaScript code:

```
2 <html lang="en">
8 <body>
10 <html lang="en">
16 <body>
17 <h3>Right click, press Inspect and goto the conso
18
19 <script>
20 // Comparision Operatrs
21
22 //Logical And
23 var age=prompt('Enter Your age:')
24 var ac=(age>=0 && age<18) ? "Child":"Adult";
25 window.alert("your are a "+ac); // output in
26 console.log("your are a "+ac); // output in
27
28 //Logical OR
29 var x=5;
30 var or=( (x<4) || (4>=x) );
31 window.alert(or); // output in aleret box
32 console.log(or); // output in console tab
33
34 //Logical Not
35 console.log(!(2 < 3)); //false
36
37
```

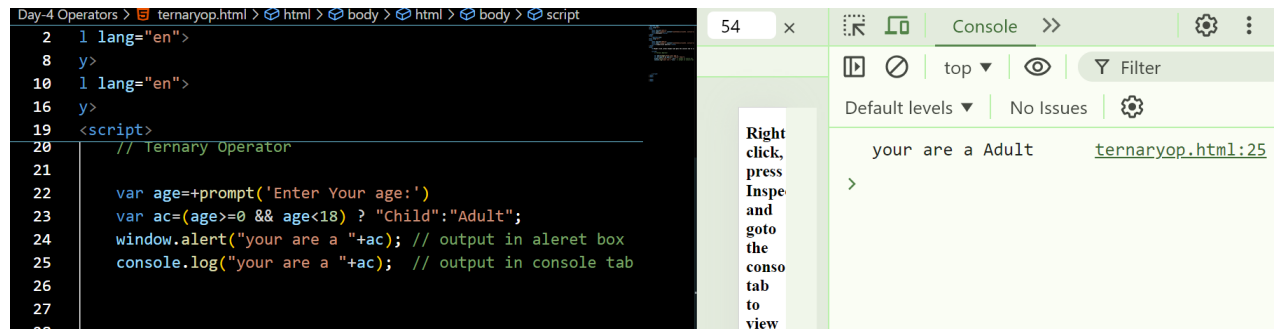
## Ternary operator:

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 (Prefix)	Operator	The operator uses to inserts one value before the incremental value by one
--x	Decrement (Prefix)	Operator	The operator Subtracts one value from the given input value before
x++	Increment (Postfix)	Operator	The operator uses to inserts one value after the incremental value by one
x--	Decrement (Postfix)	Operator	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>

```

12	<a href="#">unary.html:16</a>
number	<a href="#">unary.html:17</a>
NaN	<a href="#">unary.html:21</a>
number	<a href="#">unary.html:22</a>
-2	<a href="#">unary.html:26</a>
number	<a href="#">unary.html:27</a>
4	<a href="#">unary.html:31</a>
number	<a href="#">unary.html:32</a>
5	<a href="#">unary.html:36</a>
number	<a href="#">unary.html:37</a>

## 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.

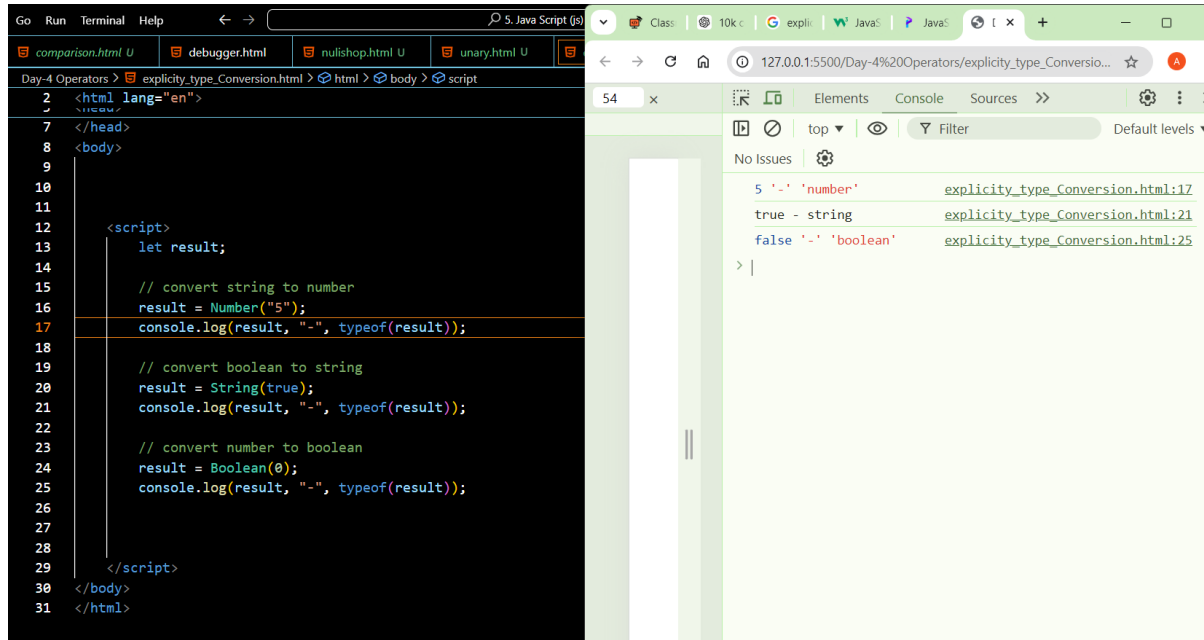
```
let str = "123";
let num = Number(str);
console.log(num); // Output: 123
```



### 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:

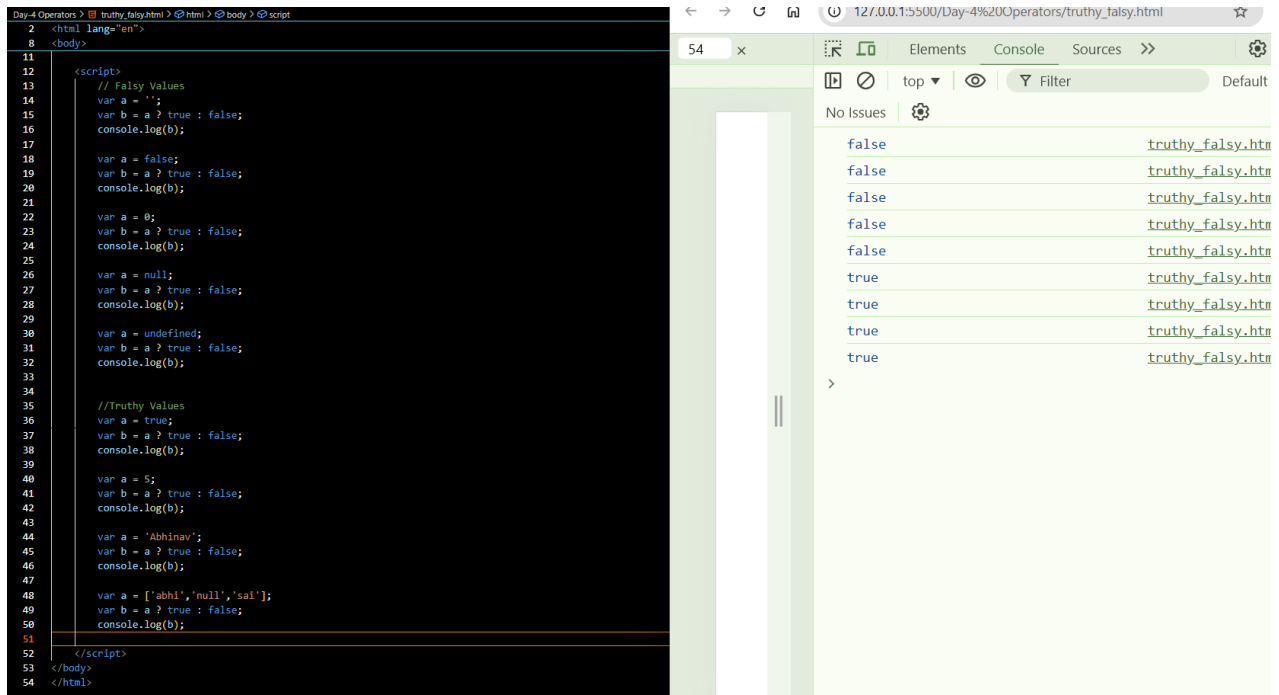
1. **false**: The boolean value false itself.
2. **0**: The number zero.
3. **""**: Empty string.
4. **null**: The absence of any value.
5. **undefined**: A variable that has not been assigned a value or a property that does not exist.
6. **NaN**: Not-a-Number.

### 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
2 <html lang="en">
8 <body>
12 <script>
13 // Falsy Values
14 var a = '';
15 var b = a ? true : false;
16 console.log(b);
17
18 var a = false;
19 var b = a ? true : false;
20 console.log(b);
21
22 var a = 0;
23 var b = a ? true : false;
24 console.log(b);
25
26 var a = null;
27 var b = a ? true : false;
28 console.log(b);
29
30 var a = undefined;
31 var b = a ? true : false;
32 console.log(b);
33
34 //Truthy Values
35 var a = true;
36 var b = a ? true : false;
37 console.log(b);
38
39 var a = 5;
40 var b = a ? true : false;
41 console.log(b);
42
43 var a = 'Abhinav';
44 var b = a ? true : false;
45 console.log(b);
46
47 var a = ['abhi','null','sai'];
48 var b = a ? true : false;
49 console.log(b);
50
51
52 </script>
53 </body>
54 </html>
```

54 x

Elements Console Sources >>

No Issues

Filter Default

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

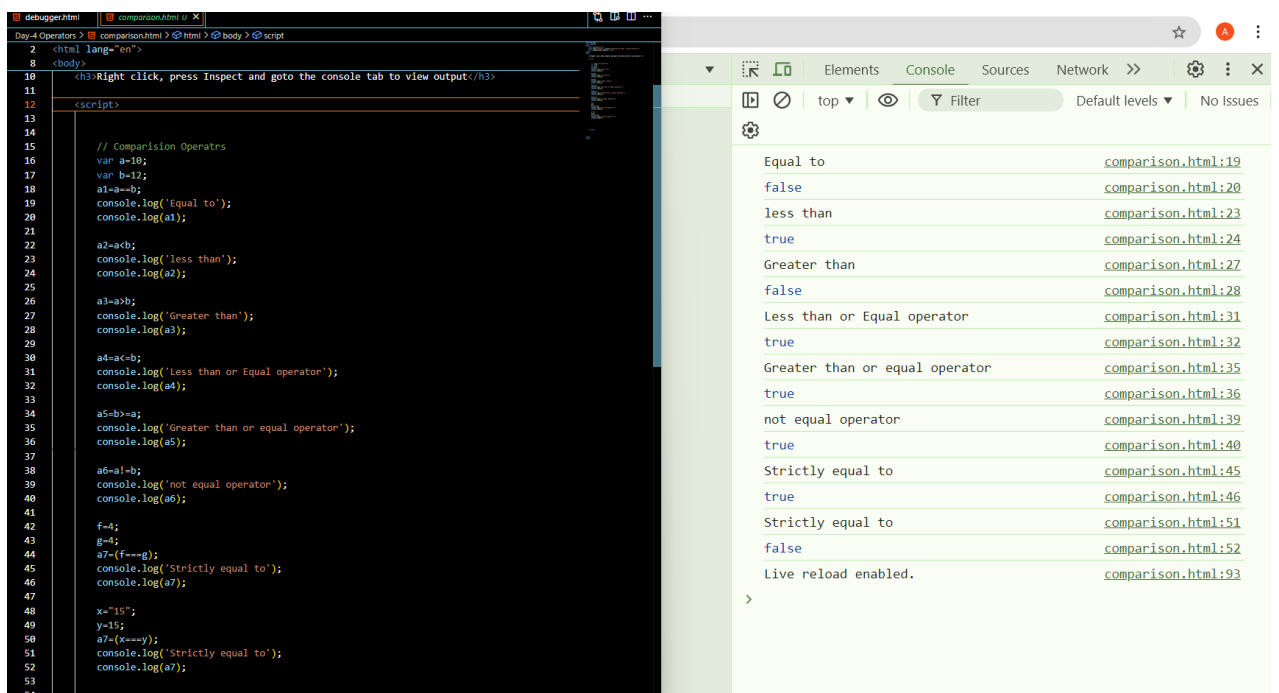
true truthy\_falsy.htm

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 > html > body > script
2 <html lang="en">
8 <body>
10 <h3>Right click, press Inspect and goto the console tab to view output</h3>
12 <script>
13
14 // Comparison Operators
15 var a=10;
16 var b=12;
17 a1=a==b;
18 console.log('Equal to');
19 console.log(a1);
20
21
22 a2=a<b;
23 console.log('less than');
24 console.log(a2);
25
26 a3=a>b;
27 console.log('Greater than');
28 console.log(a3);
29
30 a4=a<=b;
31 console.log('Less than or Equal operator');
32 console.log(a4);
33
34 a5=b>=a;
35 console.log('Greater than or equal operator');
36 console.log(a5);
37
38 a6=a!=b;
39 console.log('not equal operator');
40 console.log(a6);
41
42 f=4;
43 g=4;
44 a7=(f===g);
45 console.log('Strictly equal to');
46 console.log(a7);
47
48 x=15;
49 y=15;
50 a7=(x==y);
51 console.log('Strictly equal to');
52 console.log(a7);
53
54
```

Elements Console Sources Network >>

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

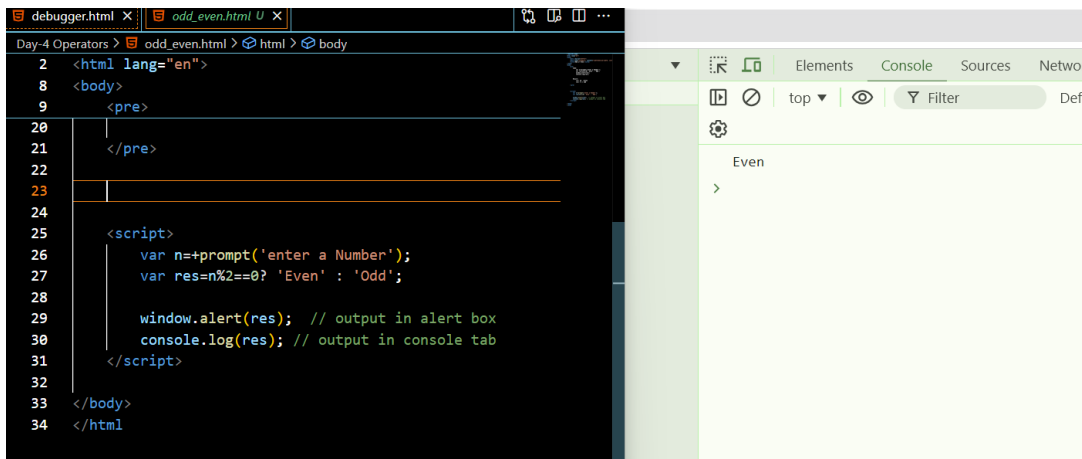
Strictly equal to comparison.html:51

false comparison.html:52

Live reload enabled. comparison.html:93

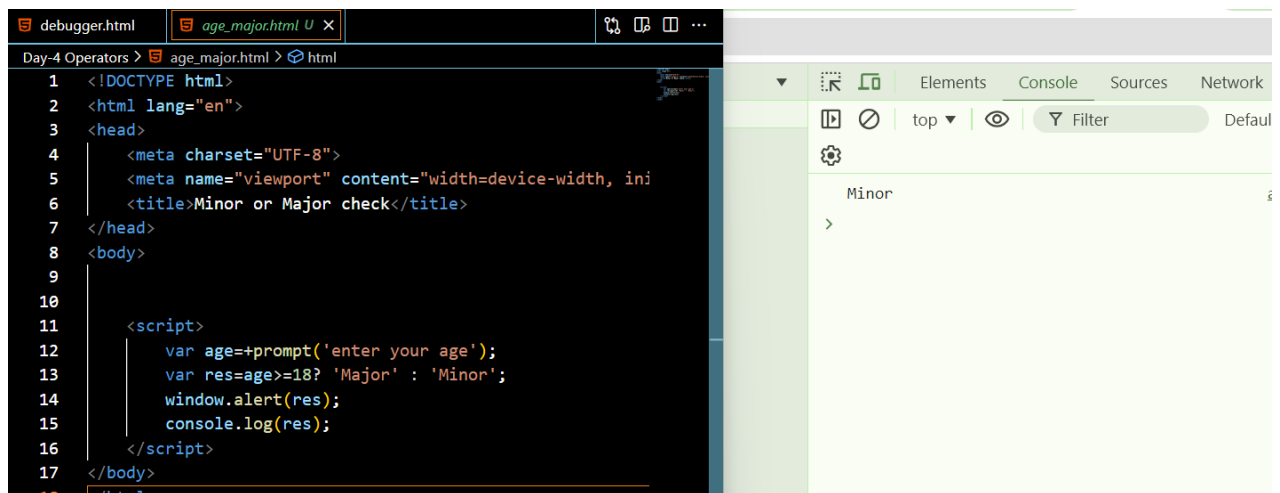
>

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



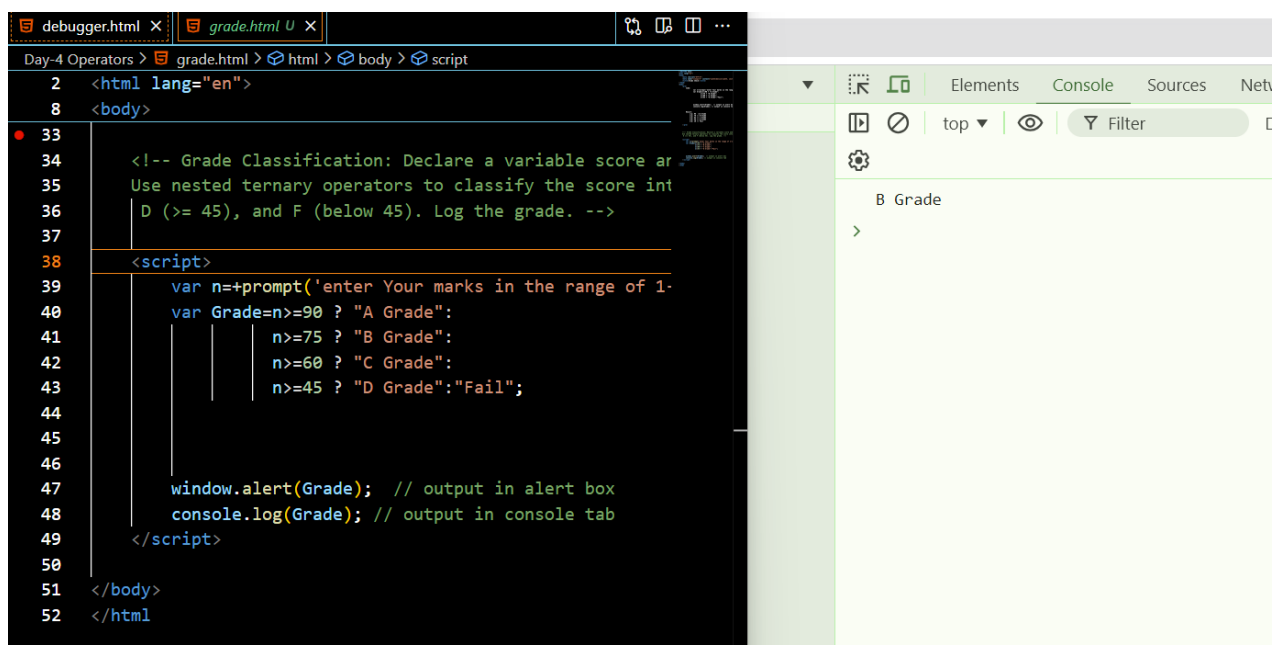
```
1 <html lang="en">
2 <body>
3   <pre>
20
21   </pre>
22
23
24
25   <script>
26     var n=prompt('enter a Number');
27     var res=n%2==0? 'Even' : 'Odd';
28
29     window.alert(res); // output in alert box
30     console.log(res); // output in console tab
31   </script>
32
33 </body>
34 </html>
```

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



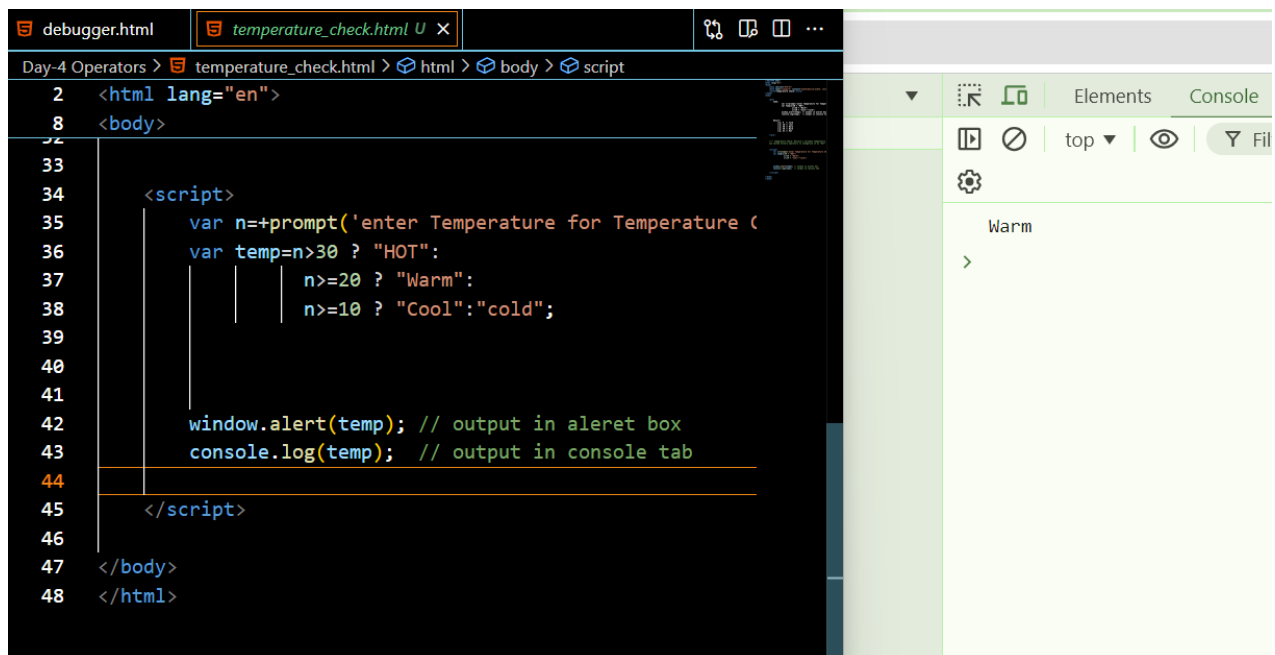
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, ini
6   <title>Minor or Major check</title>
7 </head>
8 <body>
9
10
11   <script>
12     var age=prompt('enter your age');
13     var res=age>=18? 'Major' : 'Minor';
14     window.alert(res);
15     console.log(res);
16   </script>
17 </body>
18 </html>
```

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 ( $\geq 90$ ), B ( $\geq 75$ ), C ( $\geq 60$ ), D ( $\geq 45$ ), and F (below 45). Log the grade.



```
1 <html lang="en">
2 <body>
33
34   <!-- Grade Classification: Declare a variable score ar
35   Use nested ternary operators to classify the score int
36   D ( $\geq 45$ ), and F (below 45). Log the grade. -->
37
38   <script>
39     var n=prompt('enter Your marks in the range of 1-
40     var Grade=n>=90 ? "A Grade":
41       | n>=75 ? "B Grade":
42       | n>=60 ? "C Grade":
43       | n>=45 ? "D Grade":"Fail";
44
45
46
47     window.alert(Grade); // output in alert box
48     console.log(Grade); // output in console tab
49   </script>
50
51 </body>
52 </html>
```

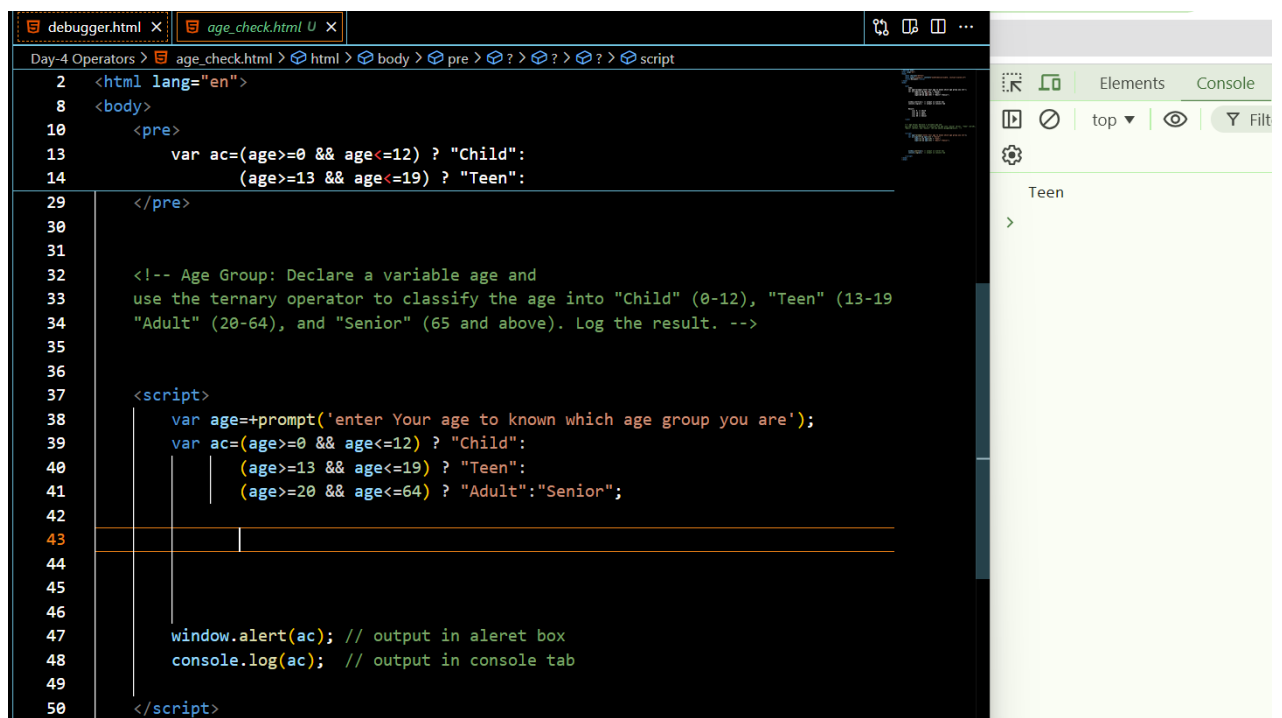
5. Temperature Check: Declare a variable temperature and use nested ternary operators to categorize it as "Hot" (above 30), "Warm" (20-30), "Cool" (10-19), and "Cold" (below 10). Log the result.



```
2 <html lang="en">
8 <body>
33
34 <script>
35   var n=+prompt('enter Temperature for Temperature (
36   var temp=n>30 ? "HOT":
37       |   n>=20 ? "Warm":
38       |   n>=10 ? "Cool":"cold";
39
40
41
42   window.alert(temp); // output in aleret box
43   console.log(temp); // output in console tab
44
45 </script>
46
47 </body>
48 </html>
```

Warm

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.



```
2 <html lang="en">
8 <body>
10 <pre>
13   var ac=(age>=0 && age<=12) ? "Child":
14       (age>=13 && age<=19) ? "Teen":
29 </pre>
30
31
32 <!-- Age Group: Declare a variable age and
33 use the ternary operator to classify the age into "Child" (0-12), "Teen" (13-19
34 "Adult" (20-64), and "Senior" (65 and above). Log the result. -->
35
36
37 <script>
38   var age=+prompt('enter Your age to known which age group you are');
39   var ac=(age>=0 && age<=12) ? "Child":
40       |   (age>=13 && age<=19) ? "Teen":
41       |   (age>=20 && age<=64) ? "Adult":"Senior";
42
43
44
45
46
47   window.alert(ac); // output in aleret box
48   console.log(ac); // output in console tab
49
50 </script>
```

Teen