

Operators in Type Script

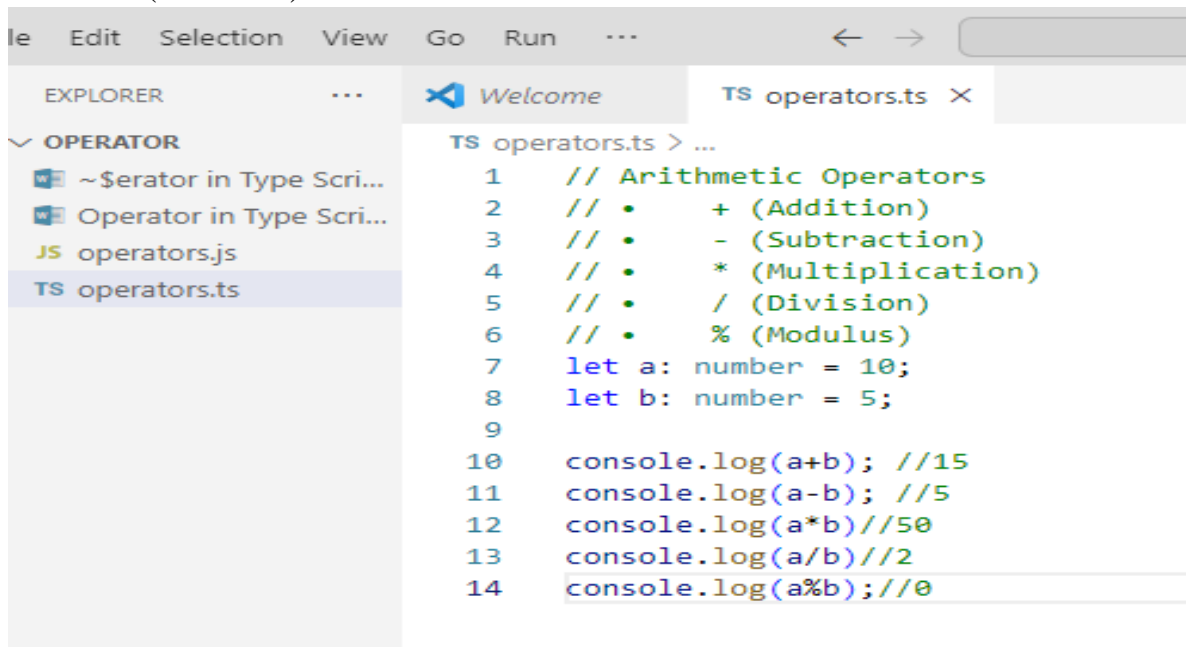
Operators:

Operators are symbols or keywords that perform operations on variables and values. TypeScript supports a variety of operators, which can be categorized into several groups. Some of the common types of operators in TypeScript are given below:

1. Arithmetic Operators
2. Comparison Operators
3. Logical Operators
4. Assignment Operators
5. Typeof Operator

Arithmetic Operators

- + (Addition)
- - (Subtraction)
- * (Multiplication)
- / (Division)
- % (Modulus)

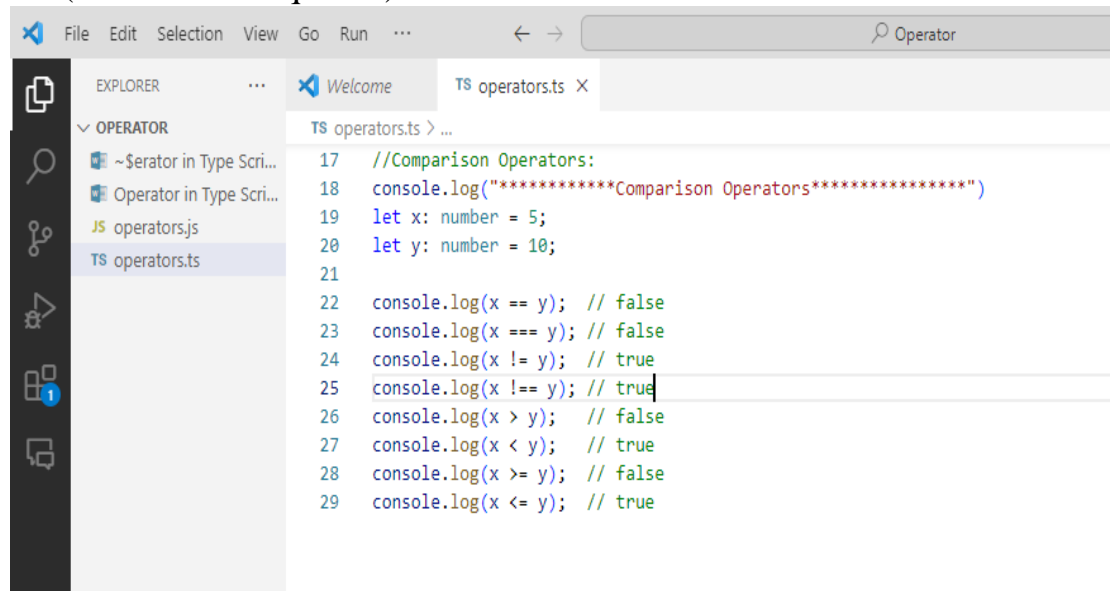


The screenshot shows the Visual Studio Code editor interface. The Explorer panel on the left shows a file named 'TS operators.ts' selected. The main editor area displays the following TypeScript code:

```
1 // Arithmetic Operators
2 // • + (Addition)
3 // • - (Subtraction)
4 // • * (Multiplication)
5 // • / (Division)
6 // • % (Modulus)
7 let a: number = 10;
8 let b: number = 5;
9
10 console.log(a+b); //15
11 console.log(a-b); //5
12 console.log(a*b)//50
13 console.log(a/b)//2
14 console.log(a%b);//0
```

Comparison Operators

- == (Equality)
- === (Strict Equality) It ensures that both the values and types are identical for the comparison to be considered true
- != (Inequality)
- !== (Strict Inequality) This operator is useful when you want to ensure both value and type differences in your comparison, similar to the way === ensures value and type equality.
- > (Greater than)
- < (Less than)
- >= (Greater than or equal to)
- <= (Less than or equal to)

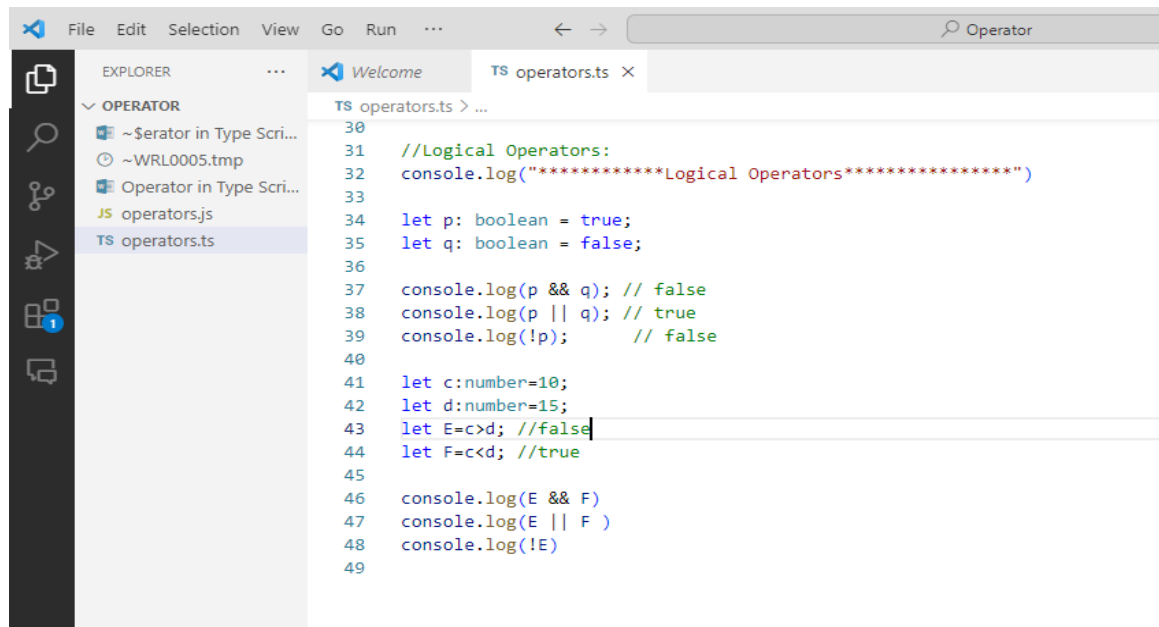


The screenshot shows the Visual Studio Code editor with a file named 'operators.ts' open. The file contains the following code:

```
17 //Comparison Operators:
18 console.log("*****Comparison Operators*****")
19 let x: number = 5;
20 let y: number = 10;
21
22 console.log(x == y); // false
23 console.log(x === y); // false
24 console.log(x != y); // true
25 console.log(x !== y); // true
26 console.log(x > y); // false
27 console.log(x < y); // true
28 console.log(x >= y); // false
29 console.log(x <= y); // true
```

Logical Operators

- && (Logical AND)
- || (Logical OR)
- ! (Logical NOT)

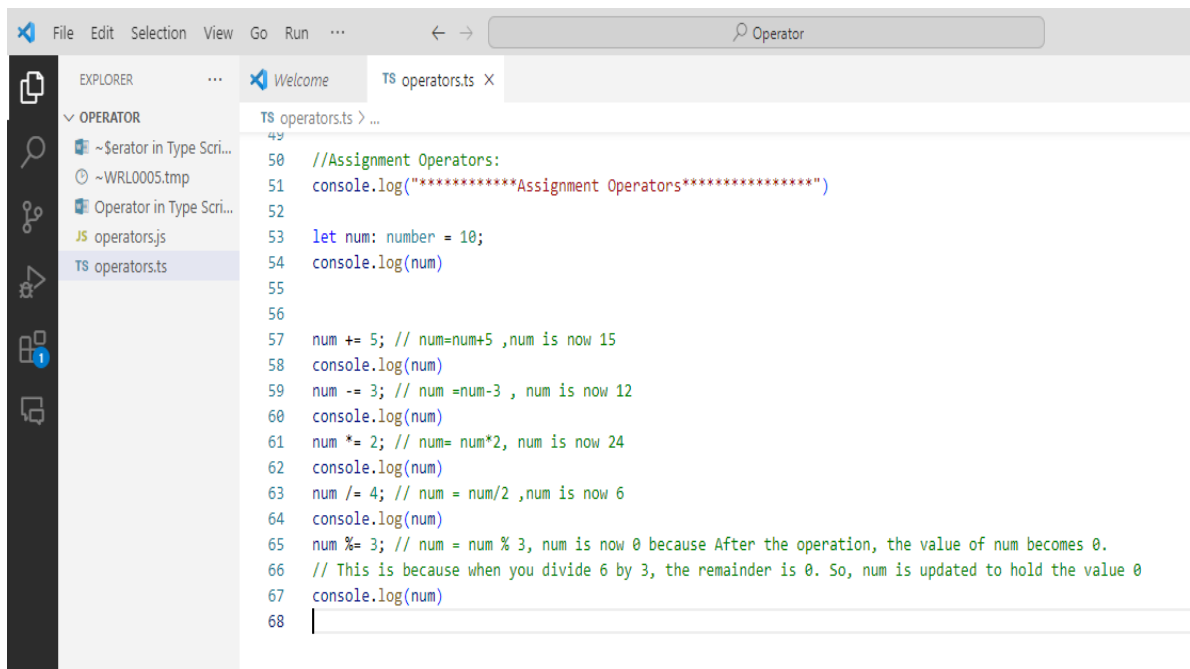


The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays a folder named 'OPERATOR' containing files: '~\$erator in Type Scri...', '~WRL0005.tmp', 'Operator in Type Scri...', 'JS operators.js', and 'TS operators.ts'. The 'TS operators.ts' file is selected and open in the main editor. The code in the file is as follows:

```
30
31 //Logical Operators:
32 console.log("*****Logical Operators*****")
33
34 let p: boolean = true;
35 let q: boolean = false;
36
37 console.log(p && q); // false
38 console.log(p || q); // true
39 console.log(!p);    // false
40
41 let c:number=10;
42 let d:number=15;
43 let E=c>d; //false
44 let F=c<d; //true
45
46 console.log(E && F)
47 console.log(E || F )
48 console.log(!E)
49
```

Assignment Operators:

- = (Assignment)
- += (Add and assign)
- -= (Subtract and assign)
- *= (Multiply and assign)
- /= (Divide and assign)
- %= (Modulus and assign)



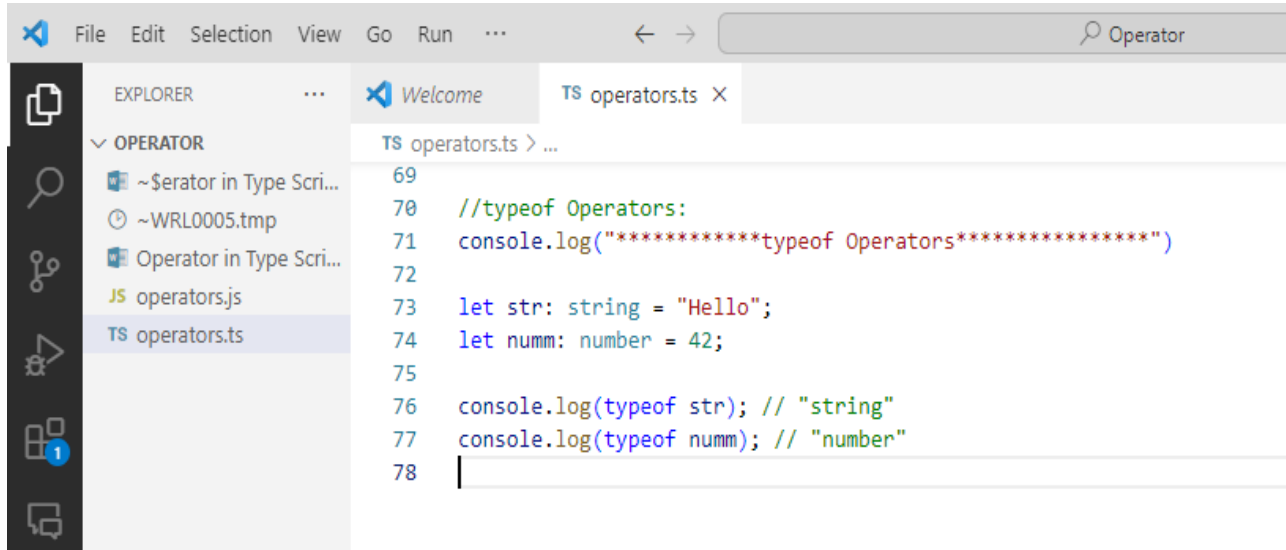
The screenshot shows a code editor with a sidebar on the left containing icons for Explorer, Search, Source Control, Run and Debug, and Testing. The Explorer sidebar is open, showing a file tree with folders 'OPERATOR' and 'JS', and files 'operators.js' and 'operators.ts'. The 'operators.ts' file is selected and its content is displayed in the main editor area. The code in 'operators.ts' demonstrates various assignment operators: basic assignment, addition, subtraction, multiplication, division, and modulus. Each operation is followed by a console log statement to show the result. A comment explains that the modulus operation results in 0 because 6 divided by 3 has a remainder of 0.

```
TS operators.ts > ...
49
50 //Assignment Operators:
51 console.log("*****Assignment Operators*****")
52
53 let num: number = 10;
54 console.log(num)
55
56
57 num += 5; // num=num+5 ,num is now 15
58 console.log(num)
59 num -= 3; // num =num-3 , num is now 12
60 console.log(num)
61 num *= 2; // num= num*2, num is now 24
62 console.log(num)
63 num /= 4; // num = num/2 ,num is now 6
64 console.log(num)
65 num %= 3; // num = num % 3, num is now 0 because After the operation, the value of num becomes 0.
66 // This is because when you divide 6 by 3, the remainder is 0. So, num is updated to hold the value 0
67 console.log(num)
68 |
```

Typeof Operator:

typeof (Returns a string representing the type of a variable).

the typeof operator is used to obtain a string representation of the type of a variable or an expression. It returns a string that represents the data type of the given operand.



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left and the editor window on the right. The Explorer sidebar shows a file named 'TS operators.ts' selected. The editor window displays the following TypeScript code:

```
69
70 //typeof Operators:
71 console.log("*****typeof Operators*****")
72
73 let str: string = "Hello";
74 let numm: number = 42;
75
76 console.log(typeof str); // "string"
77 console.log(typeof numm); // "number"
78 |
```