Operators in Type Script

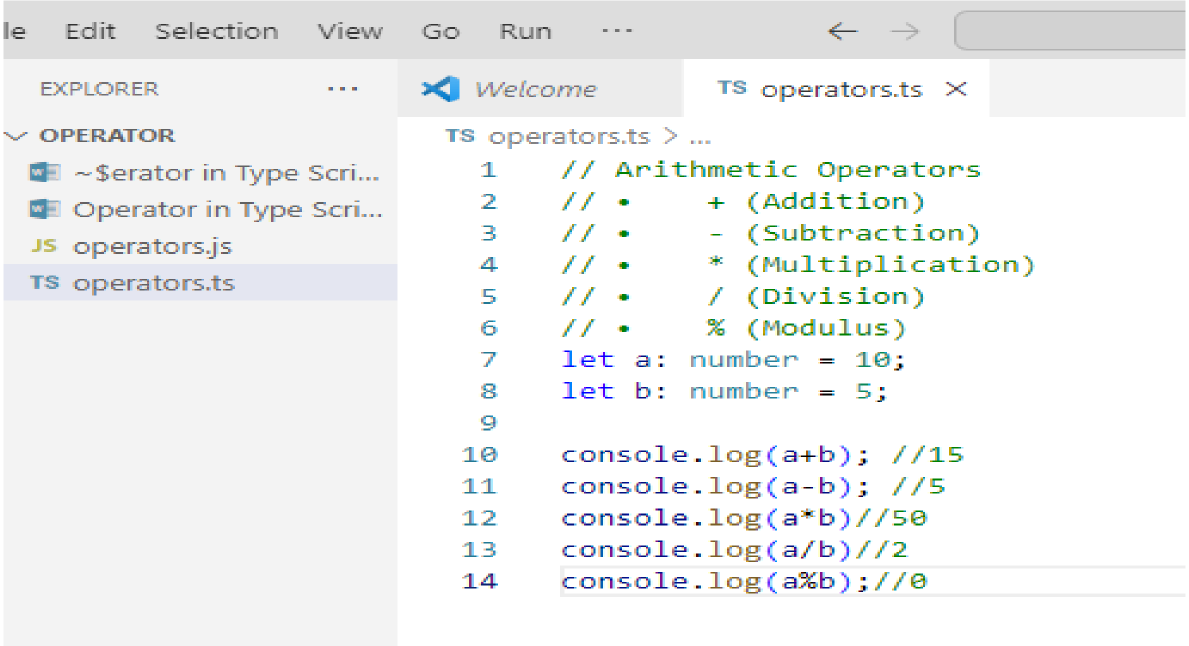
Operators:

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

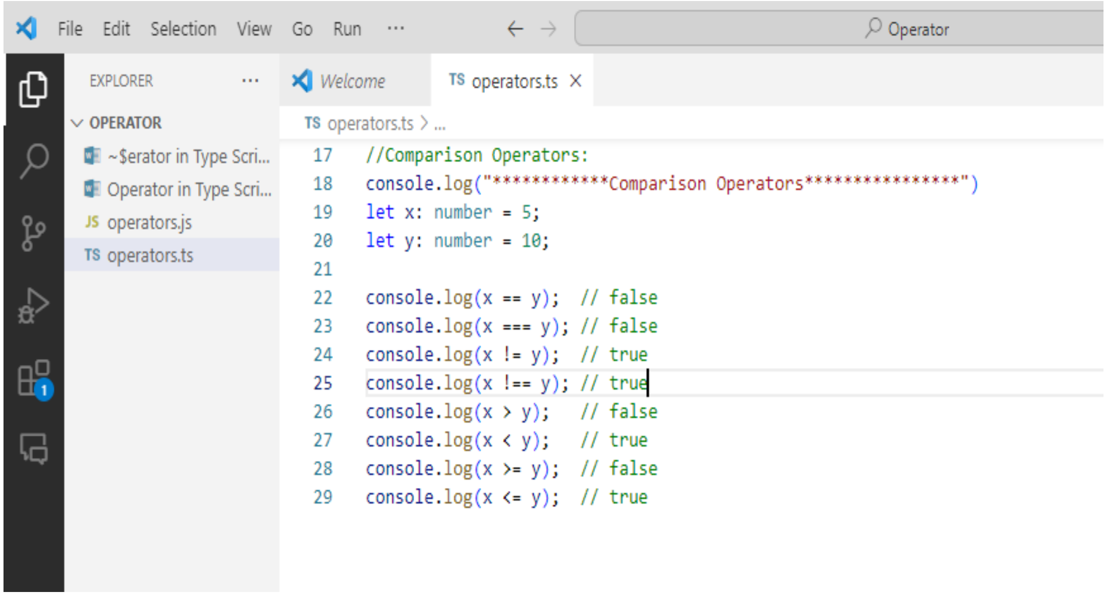
- + (Addition)
- - (Subtraction)
- * (Multiplication)
- / (Division)
- % (Modulus)



```
// Arithmetic Operators
// •     + (Addition)
// •     - (Subtraction)
// •     * (Multiplication)
// •     / (Division)
// •     % (Modulus)
let a: number = 10;
let b: number = 5;

console.log(a+b); //15
console.log(a-b); //5
console.log(a*b)//50
console.log(a/b)//2
console.log(a%b);//0
```

Comparison Operators

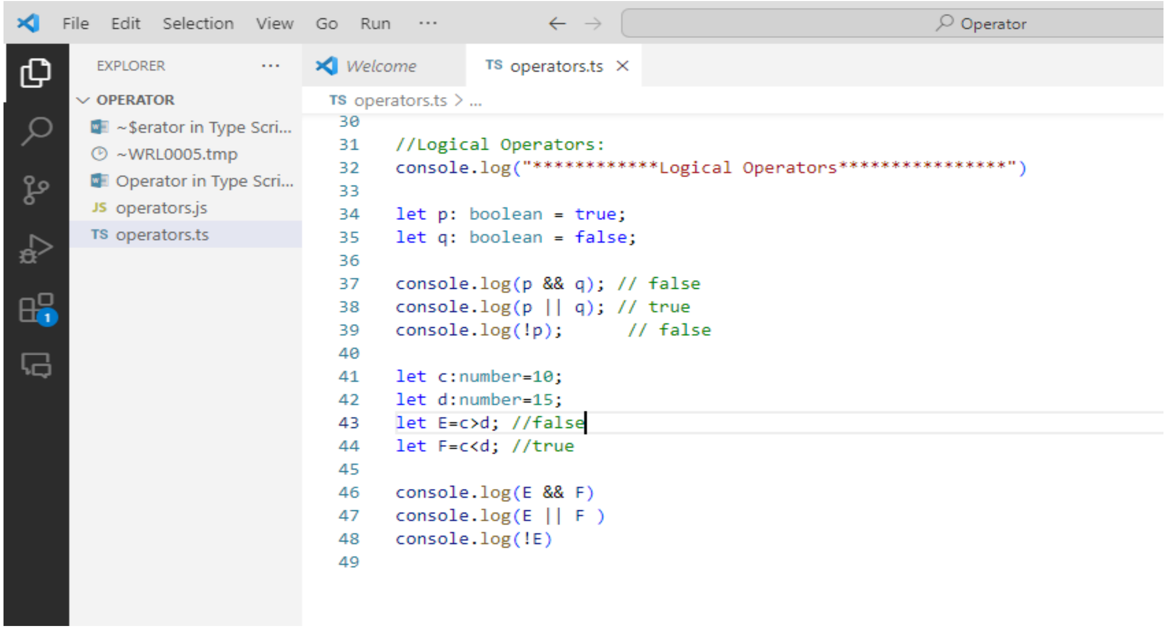
- == (Equality)
- === (Strict Equality) It ensures that both the values and types are identical for the comparison to be considered true
- != (Inequality)
- !== (Strict Inequality) This operator is useful when you want to ensure both value and type differences in your comparison, similar to the way ===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



```
//Comparison Operators:
console.log("************Comparison Operators****************")
let x: number = 5;
let y: number = 10;

console.log(x == y);  // false
console.log(x === y); // false
console.log(x != y);  // true
console.log(x !== y); // true
console.log(x > y);   // false
console.log(x < y);   // true
console.log(x >= y);  // false
console.log(x <= y);  // true
```

Logical Operators

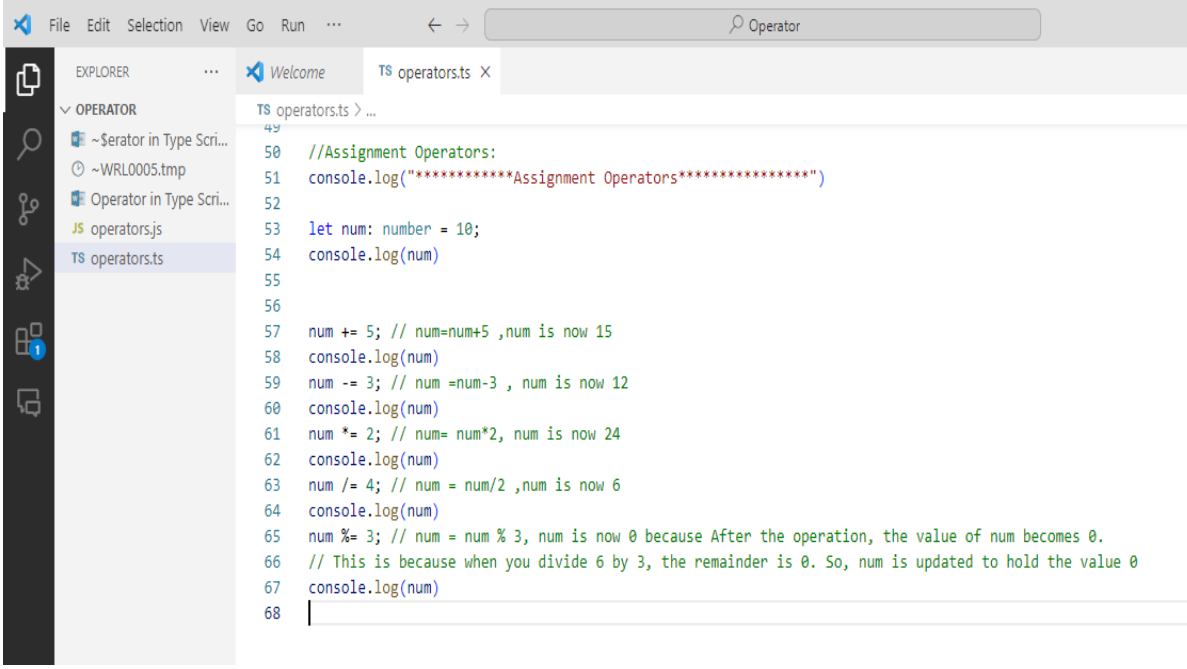
- && (Logical AND)
- || (Logical OR)
- ! (Logical NOT)



```
//Logical Operators:
console.log("************Logical Operators****************")

let p: boolean = true;
let q: boolean = false;

console.log(p && q); // false
console.log(p || q); // true
console.log(!p);      // false

let c:number=10;
let d:number=15;
let E=c>d; //false
let F=c<d; //true

console.log(E && F)
console.log(E || F )
console.log(!E)
```

Assignment Operators:

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

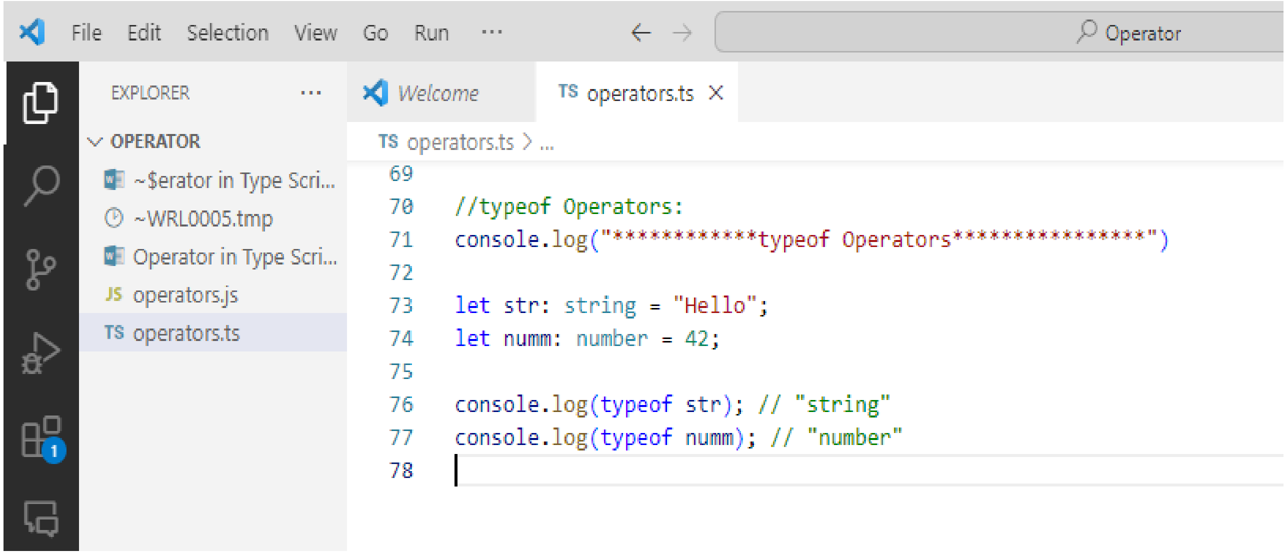


```
//Assignment Operators:
console.log("************Assignment Operators****************")

let num: number = 10;
console.log(num)


num += 5; // num=num+5 ,num is now 15
console.log(num)
num -= 3; // num =num-3 , num is now 12
console.log(num)
num *= 2; // num= num*2, num is now 24
console.log(num)
num /= 4; // num = num/2 ,num is now 6
console.log(num)
num %= 3; // num = num % 3, num is now 0 because After the operation, the value of num becomes 0.
// This is because when you divide 6 by 3, the remainder is 0. So, num is updated to hold the value 0
console.log(num)
```

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.



```
//typeof Operators:
console.log("************typeof Operators****************")

let str: string = "Hello";
let numm: number = 42;

console.log(typeof str); // "string"
console.log(typeof numm); // "number"
```