

Scopes

A Scope in JS defines the **Accessibility or life or Visibility of Variables and Functions**.

1. Global Scope:

Variable declared globally (out side function) have globally. Scope means can be access from any where.

Var have global Scope and Function Scope.

2. Block Scope:

Variables declared in a block have block scope means that can't be accessed outside of the block.

Only var have global scope the remaining let and const have block scope.

3. Local Scope:

Variables declared within the function have local scope. They can only be accessed with in the function.

Example:-

Global:-

```
var a = 10;
console.log(a); // 10
```

block:-

```
{
  var a = 10;
  console.log(a); // 10
}
```

block:-

```
{
  let a = 10;
  console.log(a); // not defined
}
```

Block

```
{
  let a = 10;
  console.log(a); // 10
}
```

```
{
  {
    let a = 10;
    console.log(a); // not defined
  }
}
```

```
{
  let a = 10;
  {
    console.log(a); // 10
  }
}
```

```
var a = 10;
let b = 20;
const c = 30; } block scope
```

Debugger

We can check Execution line by line by using keyword debugger followed by semi colon (:)

syntax:

debugger;

Example:

<script>

debugger;

var a=10;

let b=20;

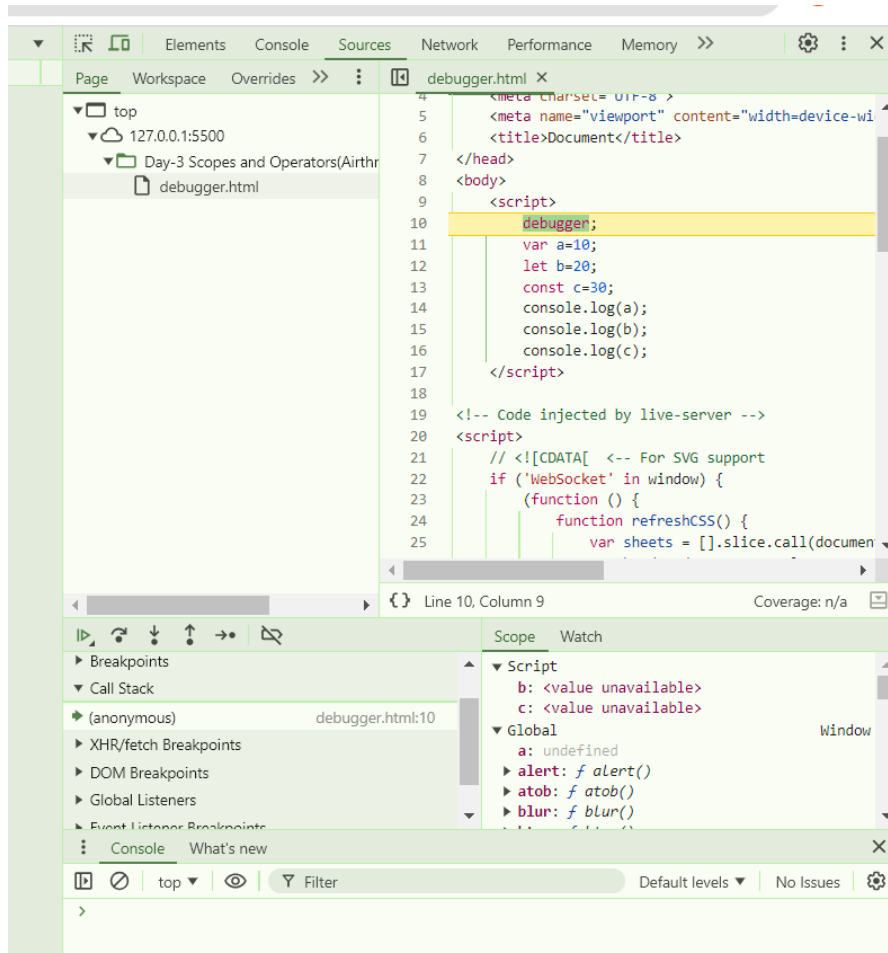
const c=30;

console.log(a);

console.log(b);

console.log(c);

</script>



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:

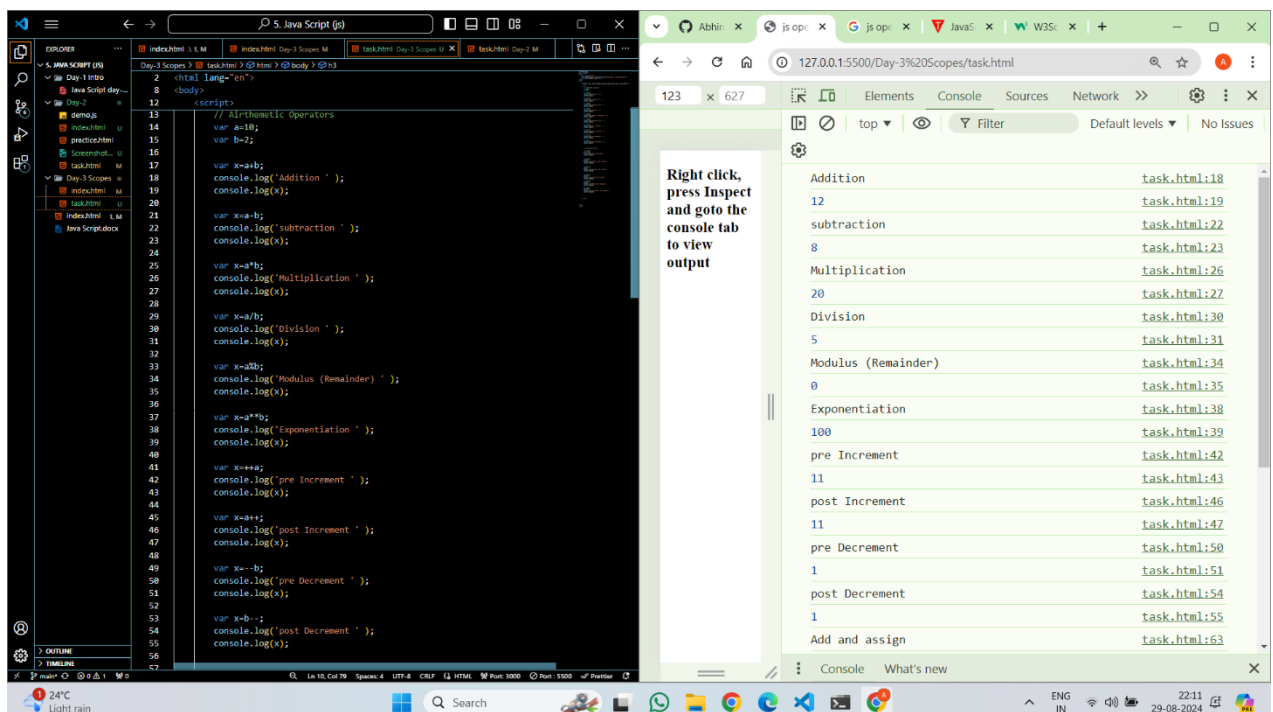
1. Airthmetic Operators
2. Assignment Operator
3. Comparision Operator
4. Logical Operator
5. Ternary Operator
6. Bitwise Oberator
7. String Operator
8. Typeof Operator

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

```
// Arithmetic Operators
var a=10;
var b=2;

var x=a+b;
console.log('Addition ');
console.log(x);

var x=a-b;
console.log('Subtraction ');
console.log(x);

var x=a*b;
console.log('Multiplication ');
console.log(x);

var x=a/b;
console.log('Division ');
console.log(x);

var x=a%b;
console.log('Modulus (Remainder) ');
console.log(x);

var x=a**b;
console.log('Exponentiation ');
console.log(x);

var x=++a;
console.log('pre Increment ');
console.log(x);

var x=a++;
console.log('post Increment ');
console.log(x);

var x=--b;
console.log('pre Decrement ');
console.log(x);

var x=b--;
console.log('post Decrement ');
console.log(x);
```

Addition	task.html:18
12	task.html:19
subtraction	task.html:22
8	task.html:23
Multiplication	task.html:26
20	task.html:27
Division	task.html:30
5	task.html:31
Modulus (Remainder)	task.html:34
0	task.html:35
Exponentiation	task.html:38
100	task.html:39
pre Increment	task.html:42
11	task.html:43
post Increment	task.html:46
11	task.html:47
pre Decrement	task.html:50
1	task.html:51
post Decrement	task.html:54
1	task.html:55
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 = a2</code>

Example:

The screenshot shows a web browser window with a JavaScript file loaded. The console displays the following 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	1000000000

The code in the browser's console is as follows:

```
// Assignment Operators
var a1=10;
a1 += 110;
console.log('Add and assign');
console.log(a1);

var a2=10;
a2 -= 10;
console.log('Subtract and assign');
console.log(a2);

var a3=10;
a3 *= 2;
console.log('Multiply and assign');
console.log(a3);

var a4=10;
a4 /= 12;
console.log('Divide and assign');
console.log(a4);

var a5=10;
a5 %= 12;
console.log('Modulus and assign');
console.log(a5);

var a6=10;
a6 **= 10;
console.log('Exponential and assign');
console.log(a6);
```

