Day 2

Operators and Expressions:

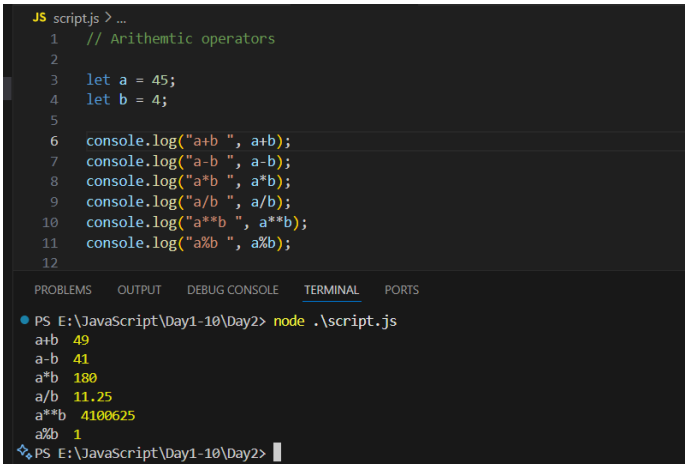
What Are Operators and Expressions?

- Operators are symbols that perform operations on values and variables.
- Expressions are combinations of values, variables, and operators that produce a result

Types of operators:

Category	Description	Example
1. Arithmetic Operators	Perform mathematical operations.	+, -, *, /, %, **, ++, --
2. Assignment Operators	Assign values to variables.	=, +=, -=, *=, /=, %=
3. Comparison Operators	Compare two values and return a boolean.	==, ===, !=, !==, >, <, >=, <=
4. Logical Operators	Combine or invert conditions.	&&, `
5. Bitwise Operators	Work on binary representations.	&, `
6. String Operators	Used to combine strings.	+, +=
7. Ternary Operator	Short form of if-else condition.	condition ? value1 : value2
8. Type Operators	Used to check or convert types.	typeof, instanceof

Arithmetic operator:



Increment operator and decrement operator:

Example: a++

```
13  let a = 45;
14  let b = 4;
15  console.log(a); //print original
16  console.log(a++); //first print then increase
17  console.log(a); //print a (which is now having incremented value)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\JavaScript\Day1-10\Day2> node .\script.js
45
45
46
PS E:\JavaScript\Day1-10\Day2>
```

Example: ++a

```
20  let a = 45;
21  let b = 4;
22  console.log(a); //print original a
23  console.log(++a); //increase by 1 then print
24  console.log(a); //print the value in a (which is incremented in last)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\JavaScript\Day1-10\Day2> node .\script.js
45
46
46
PS E:\JavaScript\Day1-10\Day2>
```

Example: b—

```
27  let b = 4;
28  console.log(b); //print the original
29  console.log(b--); //print b then decrement
30  console.log(b); //print the value in b(which is decremented )

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\JavaScript\Day1-10\Day2> node .\script.js
4
4
3
PS E:\JavaScript\Day1-10\Day2>
```

Example: --b

```
32  let b = 4;
33  console.log(b); //print the original
34  console.log(--b); //print b then decrement
35  console.log(b); //print the value in b(which is decremented )
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS E:\JavaScript\Day1-10\Day2> node .\script.js
4
3
3
PS E:\JavaScript\Day1-10\Day2>
```

So, basically:

Type	Syntax	Meaning	Example
Pre-increment	++x	Increases the value **before** using it in an expression.	let y = ++x;
Post-increment	x++	Increases the value **after** using it in an expression.	let y = x++;
Pre-decrement	--x	Decreases the value **before** using it.	let y = --x;
Post-decrement	x--	Decreases the value **after** using it.	let y = x--;

Assignment operator:

```
37  //Assignment operator
38
39  let a = 1;
40  a += 5;
41  console.log(a);
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS E:\JavaScript\Day1-10\Day2> node .\script.js
6
PS E:\JavaScript\Day1-10\Day2>
```

Comparison operator:

```
43  //Comparison operator
44
45  let a = 1;
46  let b = 2;
47  console.log(a==b); //false as 1 is not equal to 2
48  console.log(a!=b); //true
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS E:\JavaScript\Day1-10\Day2> node .\script.js
false
true
```

Comparison operator: === (strict equality) and !== (strict inequality)

```
51    let a = 1;
52    let b = 1;
53    console.log(a===b);
54    console.log(a!==b);
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\JavaScript\Day1-10\Day2> node .\script.js
true
false
PS E:\JavaScript\Day1-10\Day2>
```

```
51    let a = 1;
52    let b = "1";
53    console.log(a===b);
54    console.log(a!==b);
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\JavaScript\Day1-10\Day2> node .\script.js
false
true
PS E:\JavaScript\Day1-10\Day2>
```

Logical operator:

```
56    let a = 2;
57    let b = 7;
58    console.log(a>b && a==2);
59    console.log(a<b && a==2);
60    console.log(a<b && a!=2);
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\JavaScript\Day1-10\Day2> node .\script.js
false
true
false
```

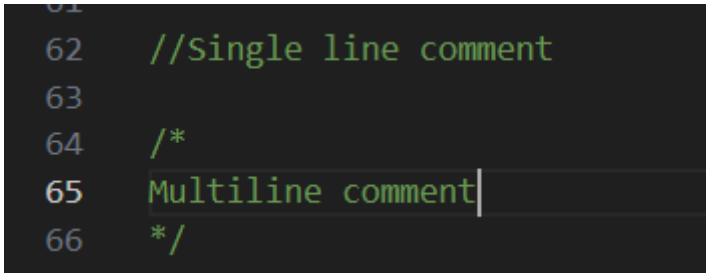
What are Comments in JavaScript?

Comments are notes written inside the code that the JavaScript engine ignores during execution. They are used to explain code, make it readable, or temporarily disable parts of code.

Types of Comments in JavaScript

Type	Syntax	Description	Example
Single-line Comment	//	Used for short notes on a single line.	// This is a single-line comment
Multi-line Comment	/* ... */	Used for longer explanations or to comment multiple lines.	/* This is a multi-line comment */

Example:



--The End--