Precedence and Associativity

Precedence and associativity are important concepts in determining the order in which operators are evaluated in an expression in JavaScript.

1. Precedence:

- Precedence refers to the order of evaluation of operators, where operators with higher precedence are evaluated first.
- For example, in the expression 2 + 3 * 4, the multiplication operator has higher precedence than the addition operator, so it is evaluated first, resulting in the value 2 + 12, which equals 14.

2. Associativity:

- Associativity, on the other hand, refers to the order in which operators of the same precedence are evaluated.
- Operators with left-to-right associativity are evaluated from left to right, while operators with right-to-left associativity are evaluated from right to left.
- For example, the assignment operator has right-to-left associativity in the expression a = b = c, so c is assigned to b. Then the resulting value of that assignment (c) is assigned to a.

Here's a chart showing the operator precedence and associativity in JavaScript, along with example operators for each level:

Precedence	Operator	Associativity
20	()	Left-to-right
18	new	Right-to-left
17	.	Left-to-right
17	[]	
16	++ --	Right-to-left
15	! ~ + -	
14	**	
13	* / %	Left-to-right
12	+ -	
11	<< >> >>>	
10	< <= > >=	
9	== != === !==	
8	&	
7	^	
6	|	
5	&&	
4	||	
3	?:	Right-to-left
2	= += -= *= /= %= <<= >>= &= ^= |=	
1	,	Left-to-right