Operators

- **Arithmetic Operators**: `+`, `-`, `*`, `/`, `%`, `**`.
- **Comparison Operators**: `==`, `!=`, `===`, `!==`, `>`, `<`, `>=`, `<=`.
- **Logical Operators**: `&&`, `||`, `!`.
- **Assignment Operators**: `=`, `+=`, `-=`, `*=`, `/=`.
- **Bitwise Operators**: `&`, `|`, `^`, `~`, `<<`, `>>`.

== vs ===

- `==` only compares values.
- `===` compares both value and data type.

Examples of ==

```
console.log(5 == "5");     // true (string "5" is converted to number 5)
console.log(0 == false);   // true (false is converted to 0)
console.log(null == undefined); // true
console.log("" == false);  // true ("" is treated as 0, false is also 0)
```

Examples of ===

```
console.log(5 === "5");    // false (number !== string)
console.log(0 === false);  // false (number !== boolean)
console.log(null === undefined); // false (different types)
console.log("" === false); // false (string !== boolean)
console.log(5 === 5);      // true (same type and value)
```

Null vs Undefined

Expression	Result	Explanation
`typeof null`	"object"	Historical JavaScript bug
`typeof undefined`	"undefined"	Correct type check
`null == null`	`true`	Null is equal to itself
`null === null`	`true`	Strict equality check passes

Expression	Result	Explanation
`null == undefined`	`true`	Loose equality treats them as equal
`null === undefined`	`false`	Different types (null vs undefined)
`!null`	`true`	`null` is falsy
`!!null`	`false`	Explicit boolean conversion of `null`
`1 + null`	`1`	`null` is treated as `0` in arithmetic
`1 + undefined`	`NaN`	`undefined` is not a number