What are Comparison Operators?

Comparison operators **compare two values** and return a **true** or **false** result.
They are mostly used in **if statements** and **loops**.

---

1. `===` (Strict Equality)

☞ Checks if **both value and type** are the same.
✅ **No type conversion** happens.

Example:

```
console.log(5 === 5);    // true (same value, same type)
console.log(5 === "5");  // false (same value, different type)
console.log(true === 1); // false (boolean vs number)
```

---

2. `==` (Loose Equality)

☞ Checks if values are the same, but **allows type conversion**.
❌ **Not recommended** because it can give unexpected results.

Example:

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

● **Avoid using `==` because it can cause confusion.** Use `===` instead.

---

3. `!==` (Strict Inequality)

☞ Checks if **either value or type is different**.
✅ **No type conversion** happens.

Example:

```
console.log(10 !== "10"); // true (different types)
console.log(10 !== 10);   // false (same value, same type)
```

---

4. != (Loose Inequality)

☞ Checks if values are **not the same**, but **allows type conversion**.
✕ **Not recommended** because of unexpected type conversion.

Example:

```
console.log(5 != "5");  // false (converted, so considered equal)
console.log(false != 0); // false (converted, so considered equal)
console.log(null != undefined); // false (special case)
```

● **Use !== instead to avoid confusion.**

---

5. > (Greater Than)

☞ Checks if the **left value is greater** than the right value.

Example:

```
console.log(10 > 5);  // true
console.log(5 > 10);  // false
console.log("b" > "a"); // true (based on Unicode values)
```

📌 **Strings are compared based on their Unicode values**.
For example, `"b"` has a **higher Unicode value** than `"a"`.

---

6. < (Less Than)

☞ Checks if the **left value is smaller** than the right value.

Example:

```
console.log(5 < 10);  // true
console.log(10 < 5);  // false
console.log("apple" < "banana"); // true (compares first letter)
```

---

7. >= (Greater Than or Equal To)

☞ Checks if the **left value is greater than or equal** to the right value.

Example:

```
console.log(10 >= 10); // true (equal)
console.log(11 >= 10); // true (greater)
console.log(5 >= 10);  // false
```

---

8. `<=` (Less Than or Equal To)

☞ Checks if the **left value is less than or equal** to the right value.

Example:

```
console.log(5 <= 10); // true (less)
console.log(10 <= 10); // true (equal)
console.log(15 <= 10); // false
```

Comparison Operators Summary

Operator	Meaning	Example	Result
`===`	Strict equality (checks type & value)	`5 === "5"`	false
`==`	Loose equality (allows type conversion)	`5 == "5"`	true
`!==`	Strict inequality (checks type & value)	`10 !== "10"`	true
`!=`	Loose inequality (allows type conversion)	`10 != "10"`	false
`>`	Greater than	`10 > 5`	true
`<`	Less than	`5 < 10`	true
`>=`	Greater than or equal to	`10 >= 10`	true
`<=`	Less than or equal to	`5 <= 10`	true

Key Takeaways

✓ **Use `===` and `!==` instead of `==` and `!=`** to avoid type conversion issues.
✓ **String comparisons are case-sensitive** (`"Apple" !== "apple"`).
✓ **Avoid `==` and `!=` unless you know how type conversion works.**
✓ **Use `>` and `<` for number comparisons.**

Would you like some coding exercises to practice these? 🚀