

🔥 JavaScript Day 5 Notes – Comparisons

🎯 1. Basic Comparisons

In JavaScript, comparisons between different types are tricky because JS tries to convert values before comparing. Let's break it down:

```
console.log("2" > 1); // Output: true
```

✅ **Why?:** "2" (string) is converted to **2** (number), and **2 > 1 → true**.

```
console.log("02" > 1); // Output: true
```

✅ **Why?:** "02" becomes **2**, and **2 > 1 → true**.

🔍 2. Comparisons with null

```
console.log(null > 0); // Output: false
```

```
console.log(null == 0); // Output: false
```

```
console.log(null >= 0); // Output: true
```

✅ **Why?:**

- **null > 0 → false** (null converts to **0**, and **0 is not greater than 0**).
 - **null == 0 → false** (null is only equal to **undefined** in loose comparison).
 - **null >= 0 → true** (null converts to **0**, and **0 >= 0 → true**).
-

🔍 3. Comparisons with undefined

```
console.log(undefined == 0); // Output: false
```

```
console.log(undefined > 0); // Output: false
```

```
console.log(undefined < 0); // Output: false
```

✅ **Why?:**

- **undefined is not converted to 0** in comparisons. It remains **undefined**, which is **incomparable** with numbers. Hence, all results **→ false**.
-

🔥 4. Difference Between == and ===

✅ **== (Loose Equality):** Checks only **value**, converts types if needed.

```
console.log("5" == 5); // Output: true (string "5" converts to number 5)
```

✅ **=== (Strict Equality):** Checks **value + type**, no type conversion.

```
console.log("5" === 5); // Output: false (string vs number — different types)
```

✅ **Rule:** Always prefer `===` for better, bug-free code.