Understanding if-else vs. switch

When we have **multiple conditions to check**, we usually use if-else statements. However, when checking **one variable against many possible values**, a switch statement is **cleaner and easier to read**.

Example 1: Using if-else

```
if (dayOfWk === "Sat" || dayOfWk === "Sun") {
  alert("Whoopee!");
}
else if (dayOfWk === "Fri") {
  alert("TGIF!");
}
else {
  alert("Shoot me now!");
}
```

This works fine, but if there are **many conditions**, it gets messy.

Example 2: Using switch

```
switch (dayOfWk) {
  case "Sat":
  case "Sun":
    alert("Whoopee!");
    break;

  case "Fri":
    alert("TGIF!");
    break;

  default:
    alert("Shoot me now!");
}
```

How It Works:

1. We use switch(dayOfWk), where dayOfWk is the variable we are checking.
2. case "Sat": checks if dayOfWk is "Sat". If true, it runs alert("Whoopee!");
3. case "Sun": does the same, so both "Sat" and "Sun" show "Whoopee!"
4. case "Fri": runs alert("TGIF!");
5. default: runs if none of the cases match (like else in if-else).

Why Use switch Instead of if-else?

✓ Easier to read when there are many conditions.
✓ Cleaner than using multiple if-else statements.
✓ More efficient in some cases.

Key Things to Remember

- Each case **must end with break;** to stop checking further cases.
- The default: case **runs if no other case matches.**
- Multiple cases **can share the same block** (like "Sat" and "Sun" above).

When to Use switch?

Use switch when **checking one variable** against **many possible values**.

Use if-else when checking **complex conditions** (like x > 10 or y !== 5).