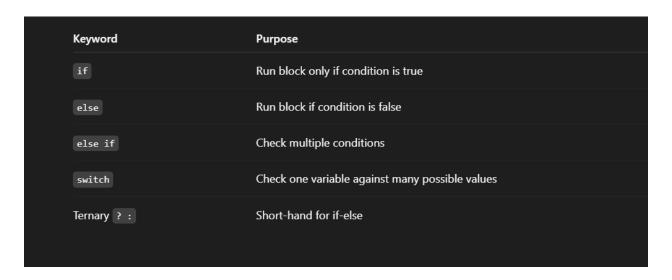


Conditionals



If / else if / else

- if = agar ye condition true hai
- else if = warna agar dusri condition true hai
- else = sab galat ho gaya toh ye karo

Nullish coalescing

The **nullish coalescing operator (??)** is used to provide a **default value** when the original value is either **null** or **undefined**.

It is **not** triggered by false, o, or (empty string) — only by **null or undefined**

```
// nullish coalescing ooperator (??): null undefined (fallback)
// do safety check if null or undeinfed then null ?? 18; consider 18
let val1
val1= 5 ?? 10
console.log(val1) //5

let val2
val2= null ?? 10
console.log(val2) //10

let val3
val3= undefined ?? 15
console.log(val3);
```

Ternary operator

condition? expressionIfTrue: expressionIfFalse;

```
// ternary operator
const teaPrice= 78
teaPrice >= 80 ? console.log("less than 80"): console.log("more than 80");
```

Switch

- switch is used to test one variable or expression against multiple possible values.
- It's an alternative to multiple if...else if blocks making code cleaner and easier when you have many conditions on the same variable.
- 1. Always use break or it may fall through to the next case.

- 2. Always end with a default to handle unexpected inputs.
- 3. You can group multiple case s to avoid duplicate code.

```
a= 10;
b= 15;
op= "*"
switch(op){
    case "+": {
       console.log(a+b) //break nahi add kara toh saree run hote
       break;
    case "-":{
       console.log(a-b)
       break;
   case "*":{
       console.log(a*b)
        break;
   case "/": {
       console.log(a/b)
        break;
   default: {
       console.log("Invalid operation") //no break
// return meh console.log matt karna ; return "message"
```