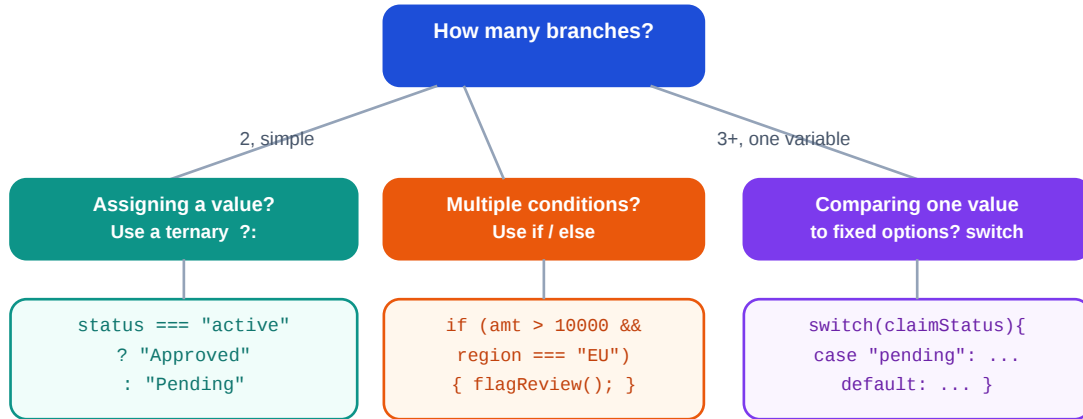


JavaScript Building Blocks: Decision Tree & QA Checklist

If/else vs. switch vs. ternary, the == vs === coercion table, and a role-by-role review checklist.

1. Which Control Flow Structure Do I Use?



2. == vs === Coercion Reference

Expression	Result	Why
"0" == 0	true	String coerced to number before comparing
"0" === 0	false	Different types, no coercion — not equal
0 == false	true	Both coerce to a falsy comparison
0 === false	false	Number is never strictly equal to boolean

3. Building Blocks Hierarchy

Block	Definition	Example
Value	A piece of data	"pending", 42, true
Variable	Named container for a value	let status = 'pending'
Operator	Acts on values to produce a result	+ === &&
Expression	Code that resolves to a value	totalDue > 0
Statement	A complete instruction	if (totalDue > 0) {...}
Function	Reusable named block of statements	function flag() {...}

4. Review Checklist by Role

QA / Automation: ■ Flag bare == in validation logic; confirm assertions compare the right building-block layer.

Business Analyst: ■ Write acceptance criteria specifying strict-equality expectations for numeric fields.

IT Support / Ops: ■ Check switch statements for a missing default case during config-driven defects.

Rule of thumb: name the building block before you debug it. Value, operator, expression, statement, or function — each layer fails in a different, predictable way.