

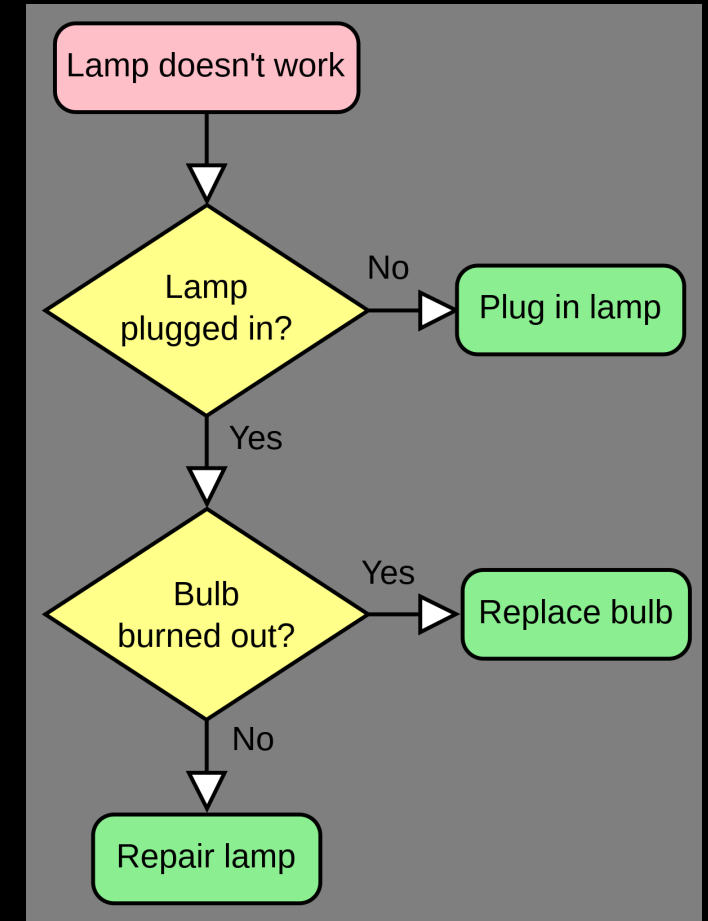
# Decision Control

- Comparison Operators
- if-else, ladder, nested-if
- Truthy and Falsy Values
- Logical Operator
- If alternates
- Scope
- Switch

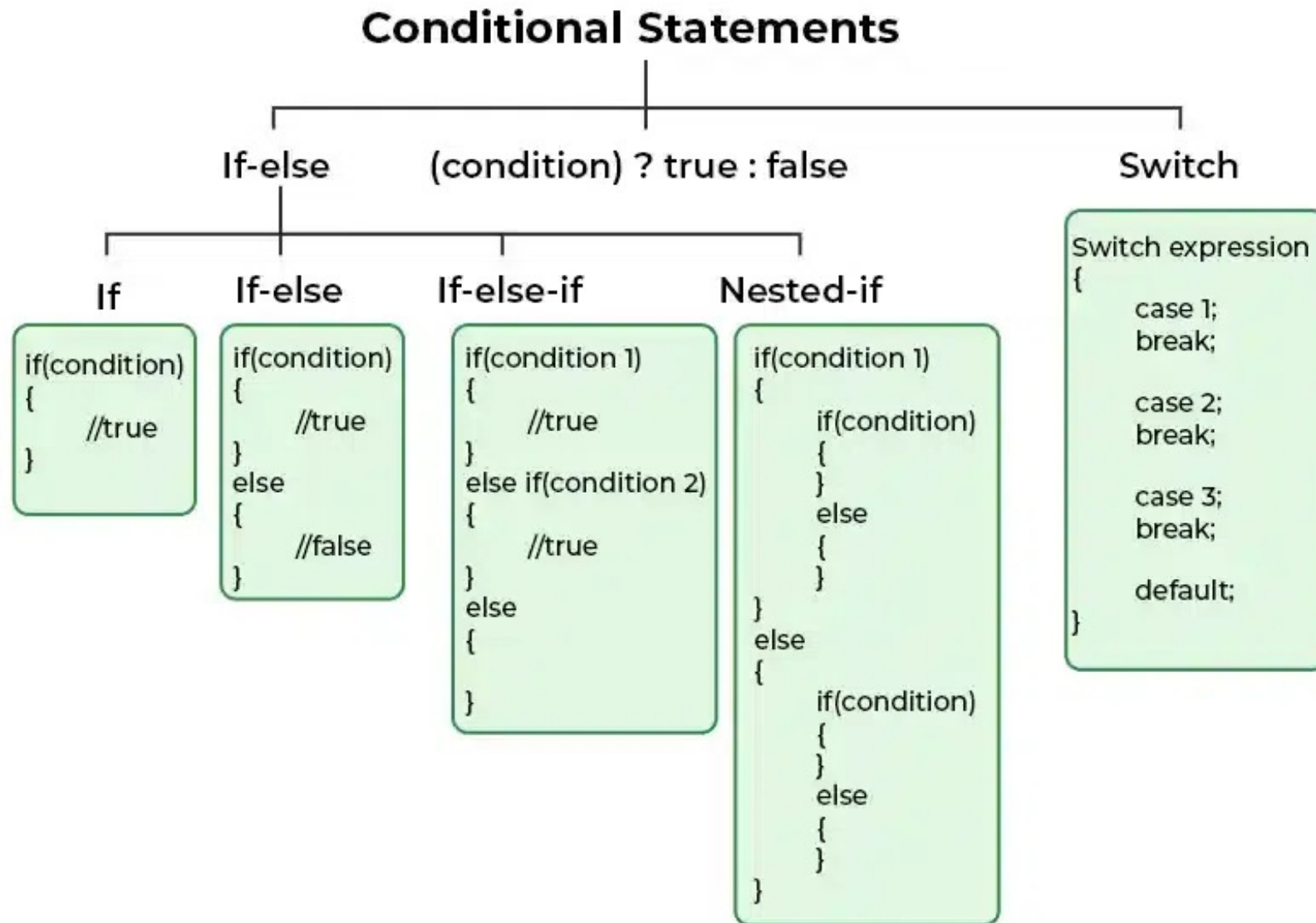
Create    Project

# What is Decision Control ?

1. **Conditional Execution:** They allow code to run **based on specific conditions**, making programs dynamic.
2. **Handles Complexity:** Enables handling **complex decisions** through nested statements.
3. **Enhances Flexibility:** Increases the adaptability of programs to **different scenarios**.



# What is Decision Control ?



# Comparison Operators

<, >, <=, >=, ==, !=

## •Equality

- == Checks value equality.
- === Checks value and type equality.

## •Inequality

- != Checks value inequality.
- !== Checks value and type inequality.

## •Relational

- > Greater than.
- < Less than.
- >= Greater than or equal to.
- <= Less than or equal to.

Order of comparison operators is less than arithmetic operators

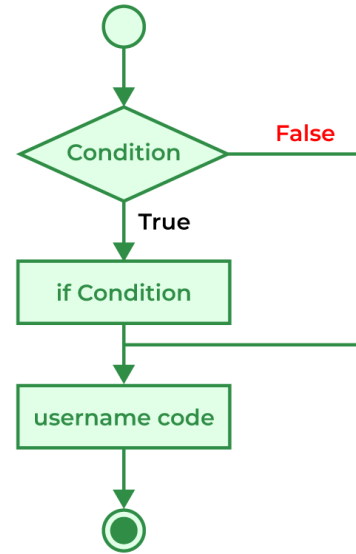
# if-else

## Condition is true

```
let number = 2;  
if (number > 0) {  
  // code  
}  
else {  
  // code  
}  
// code after if
```

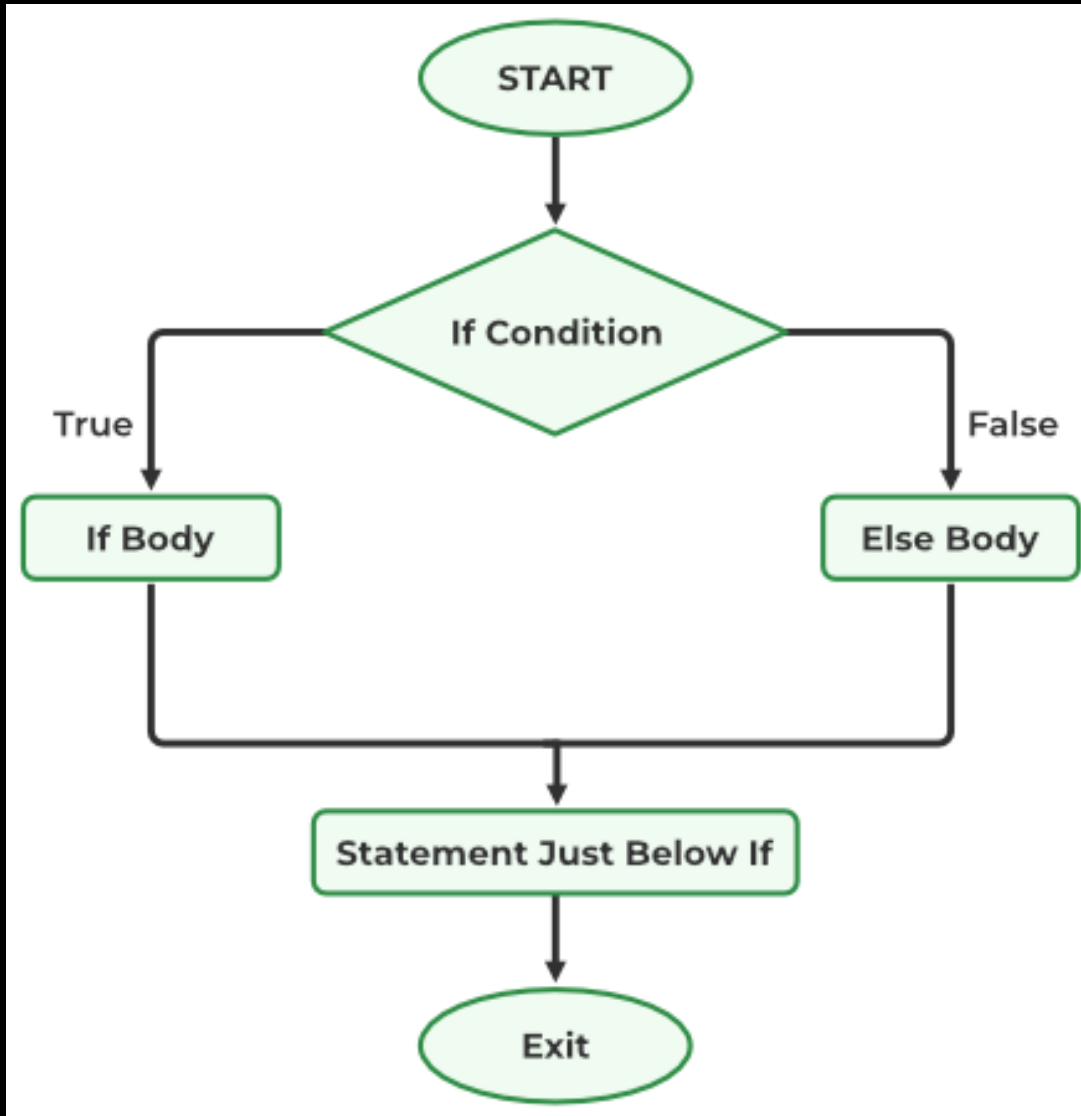
## Condition is false

```
let number = -2;  
if (number > 0) {  
  // code  
}  
else {  
  // code  
}  
// code after if
```



1. **Syntax:** Uses `if () {}` to check a condition.
2. **What is if:** Executes block if condition is **true**, skips if **false**.
3. **What is else:** Executes a block when the if condition is **false**.
4. **Curly Braces** can be omitted for single statements, but not recommended.
5. **Use Variables:** Can store **conditions** in variables for use in if statements.

# if-else



```
if thirsty {
```



```
} else {
```



```
}
```

**What** is **else**: Executes a block when the **if** condition is **false**.