CODEMIND
Your Success Partner

Control Flow

By CODEMIND Technology

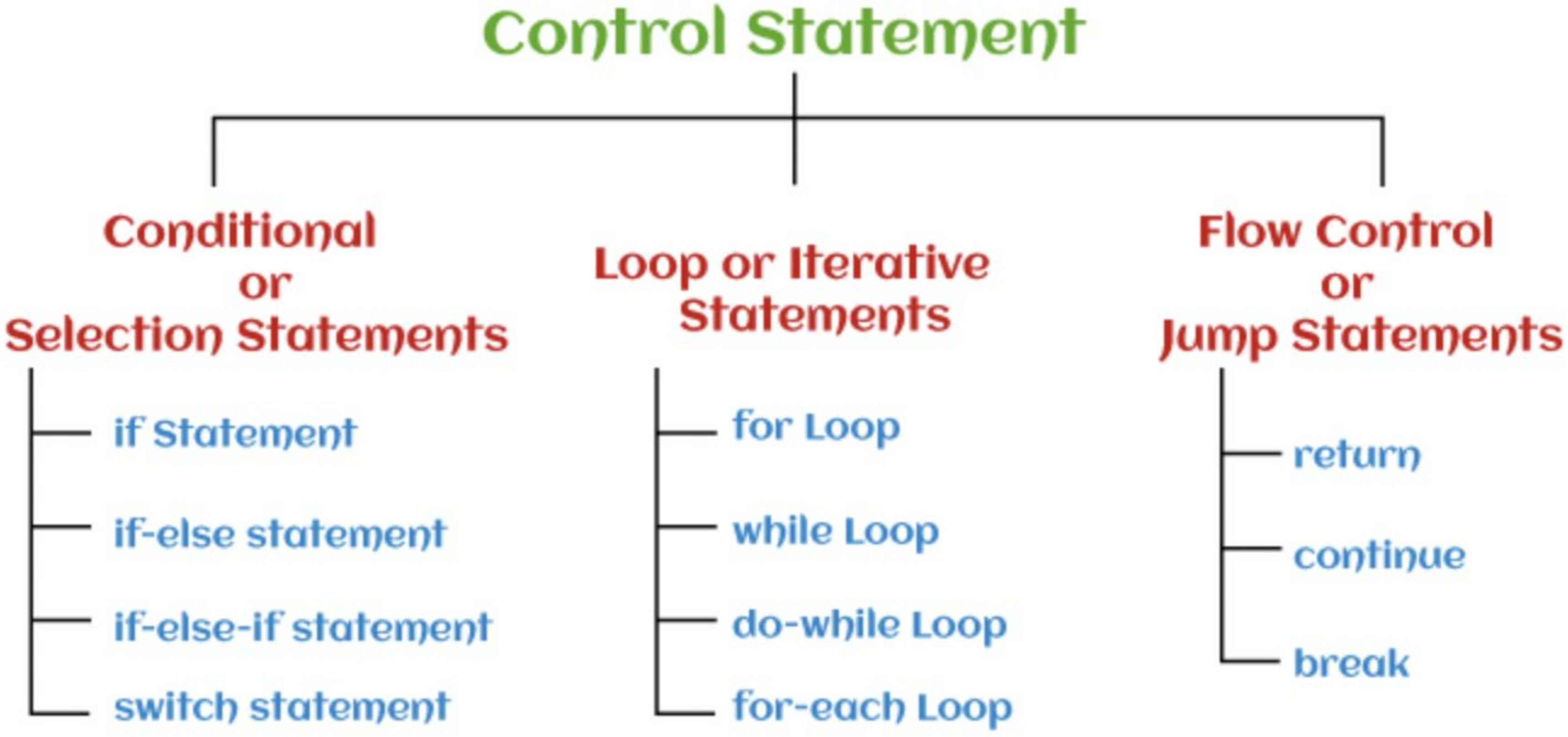
Contact us 966 5044 698
966 5044 598

Control Flow

Control flow in JavaScript is how our JS engine runs code from top to bottom.

It starts from the first line and ends at the last line, unless it hits any statement that changes the control flow of the program such as loops, conditionals, or functions

Control Flow statements



Control Statement

Conditional or Selection Statements
- if Statement
- if-else statement
- if-else-if statement
- switch statement

Loop or Iterative Statements
- for Loop
- while Loop
- do-while Loop
- for-each Loop

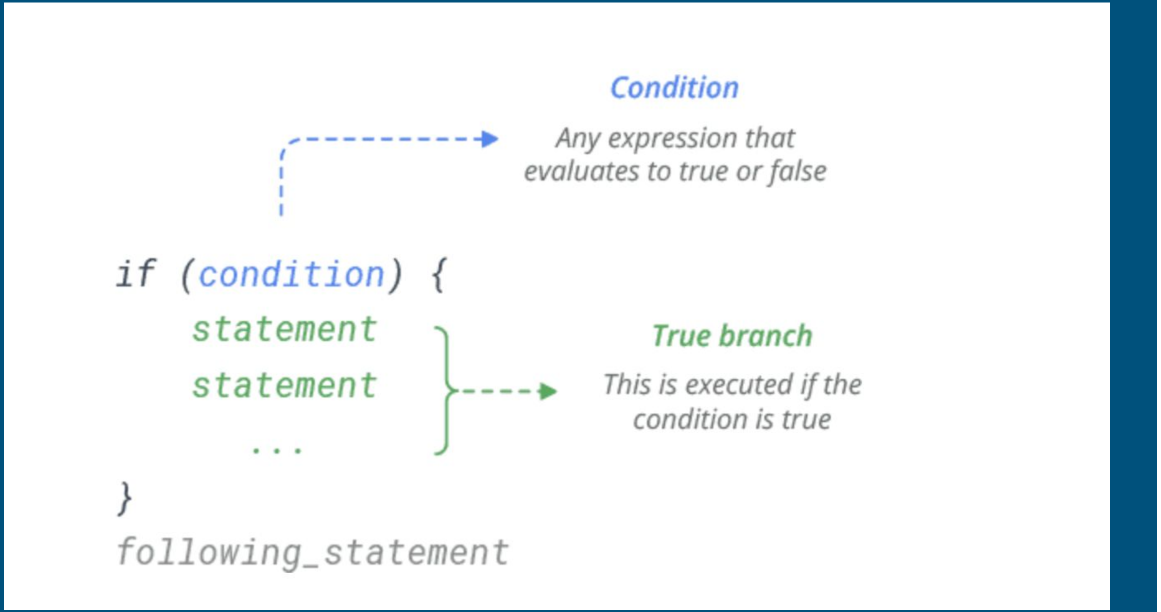
Flow Control or Jump Statements
- return
- continue
- break

if statement: Syntax and Flow chart

Condition can be checked using operators like

==, ===, !=, <, <=, >, &&, || and many more

Syntax:



```
if (condition) {
    statement
    statement
    ...
}
following_statement
```

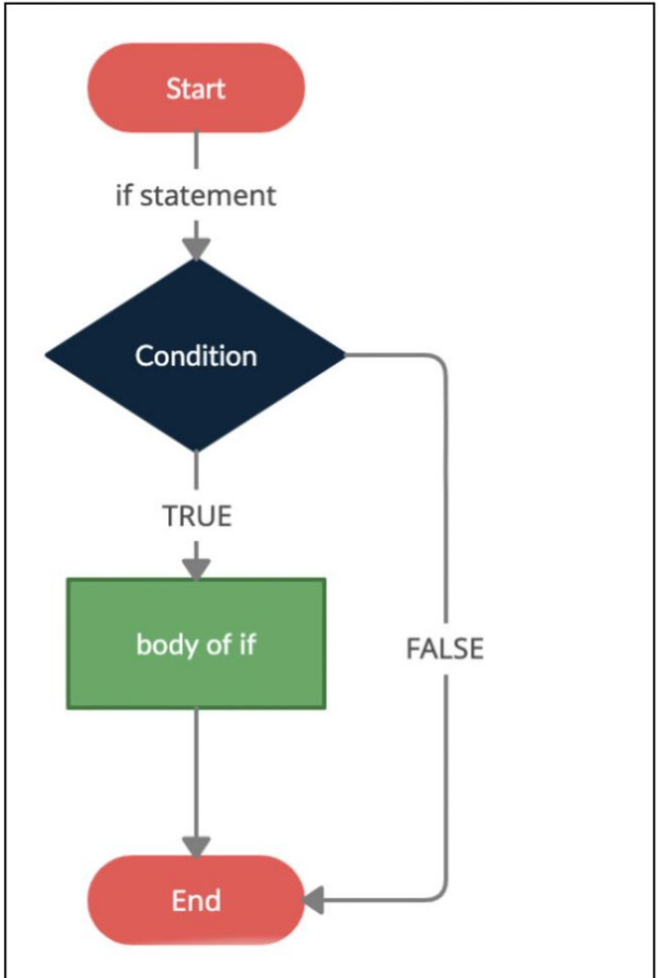
Condition

Any expression that evaluates to true or false

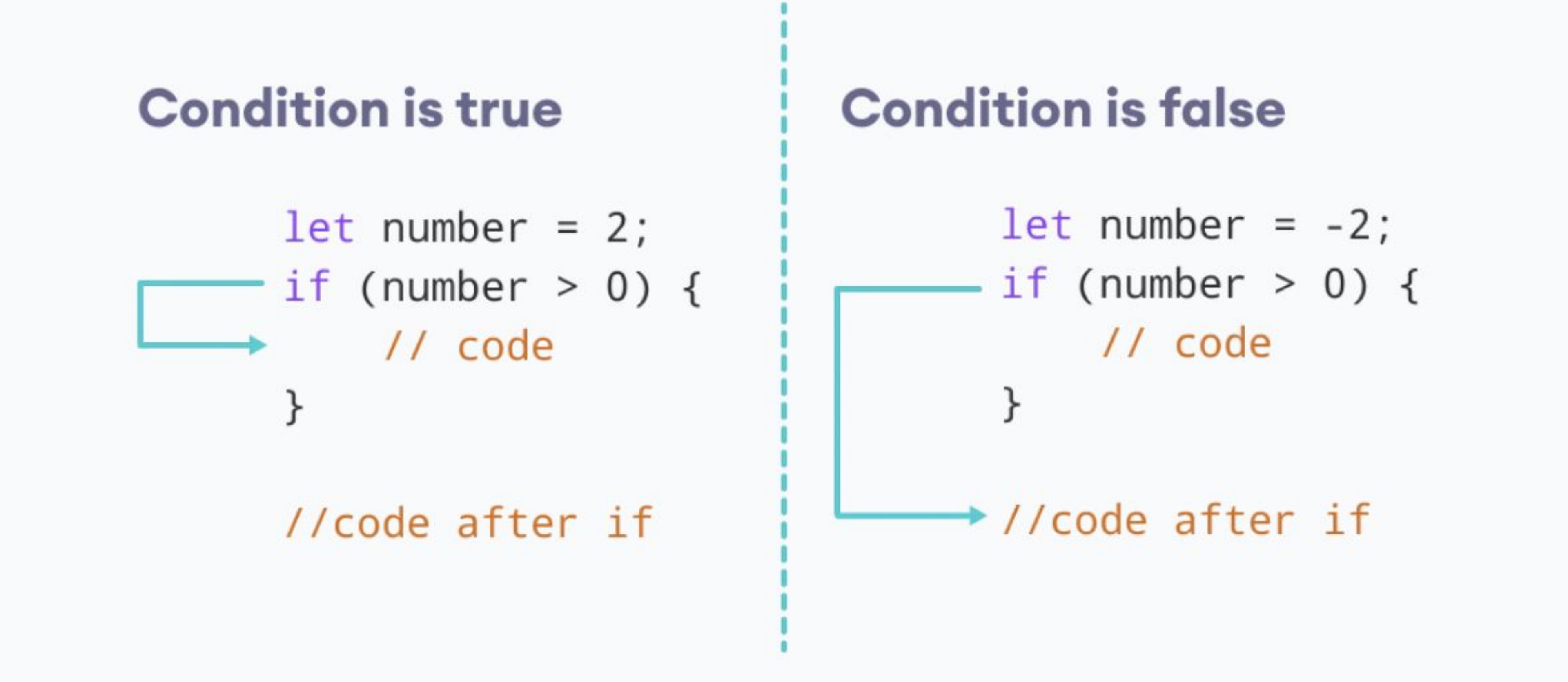
True branch

This is executed if the condition is true

Flow Chart



Conditional execution with true and false



Condition is true

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

//code after if
```

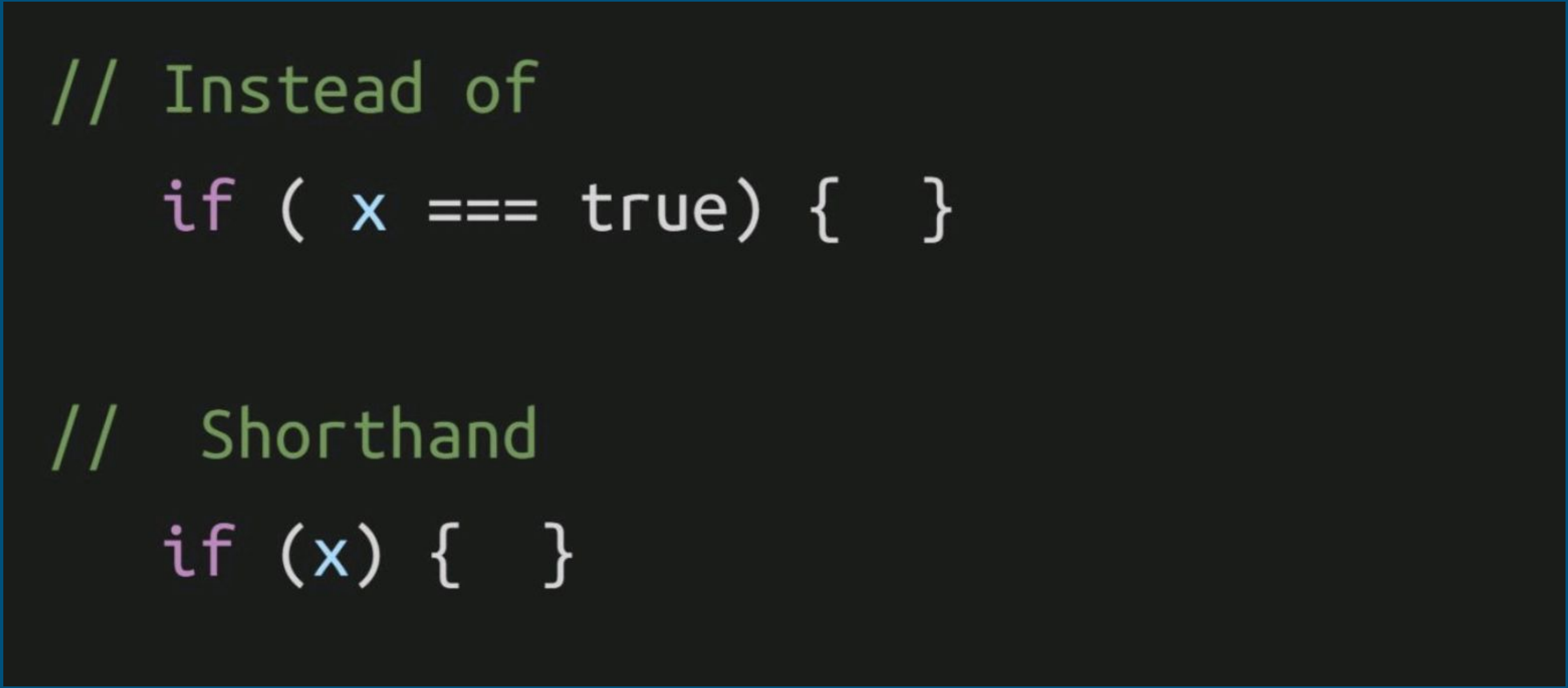
Condition is false

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

//code after if
```

if statement

- If stmt without { } braces
    - In this case just one line is allowed inside if stmt body
    - Ex →

- Nested if stmt: We can write if statement inside if statement



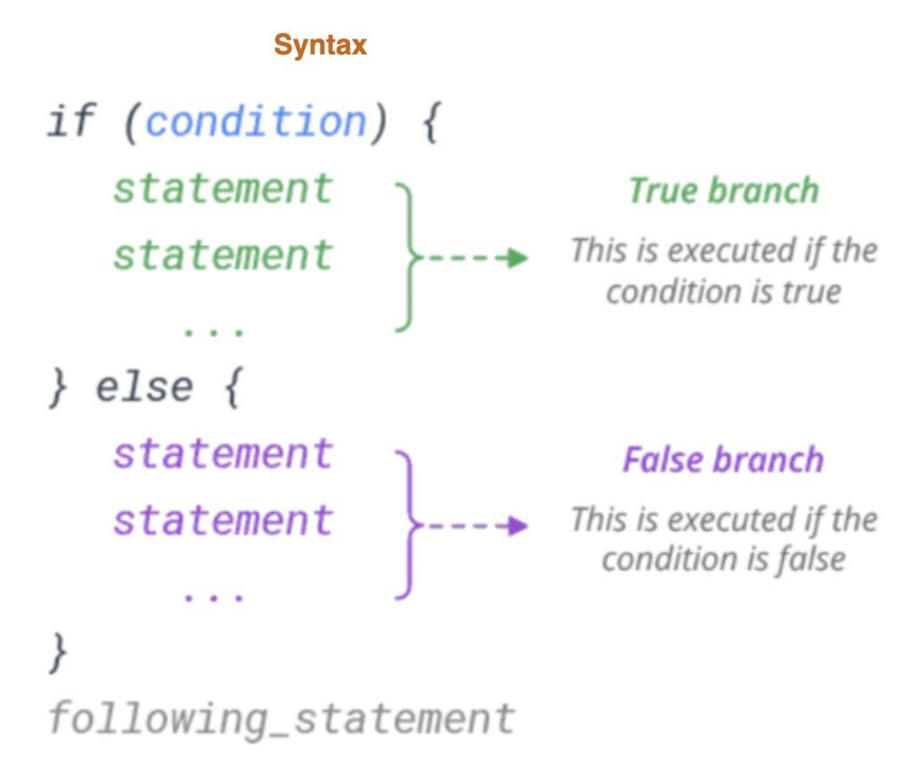
```
// Instead of
   if ( x === true) {  }


//  Shorthand
   if (x) {  }
```

Assignment 01: Please make sure to write function for each step and use the if block Of if else block

- File Name → 06_ifElseHandsOn.js

1. Create a function to check passed argument number value is even or odd and return the result as EVEN or ODD. Ex. 45, 70, 67, 98
2. Check if person is eligible for vote or not Ex→ age: 18, 20, 17, 40 → Ignore please
3. Check if string contains more than 10 character or not Ex → “JavaScript - ES6”
4. Check if string starts with “Java” Ex→ “JavaScript Language”

If else statement



Syntax

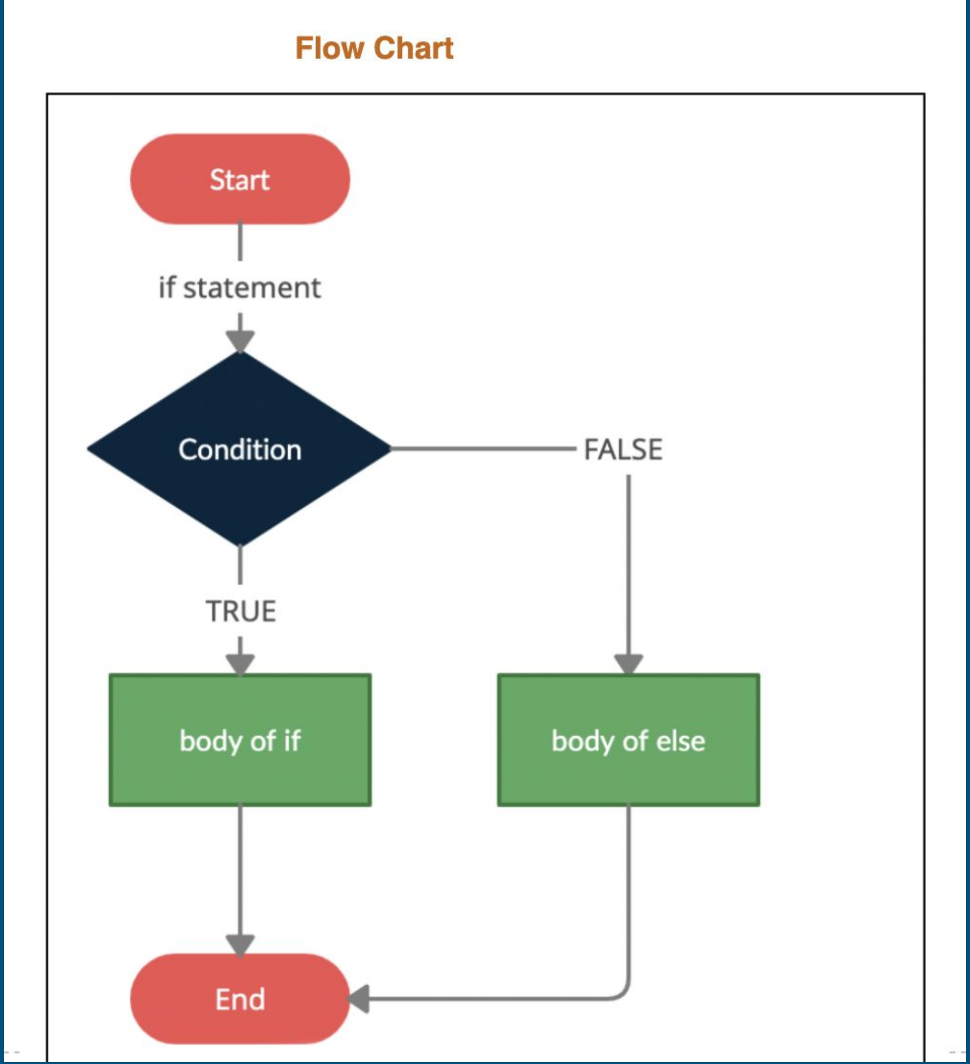
```
if (condition) {
    statement
    statement
     ...
} else {
    statement
    statement
     ...
}
following_statement
```

True branch

This is executed if the condition is true

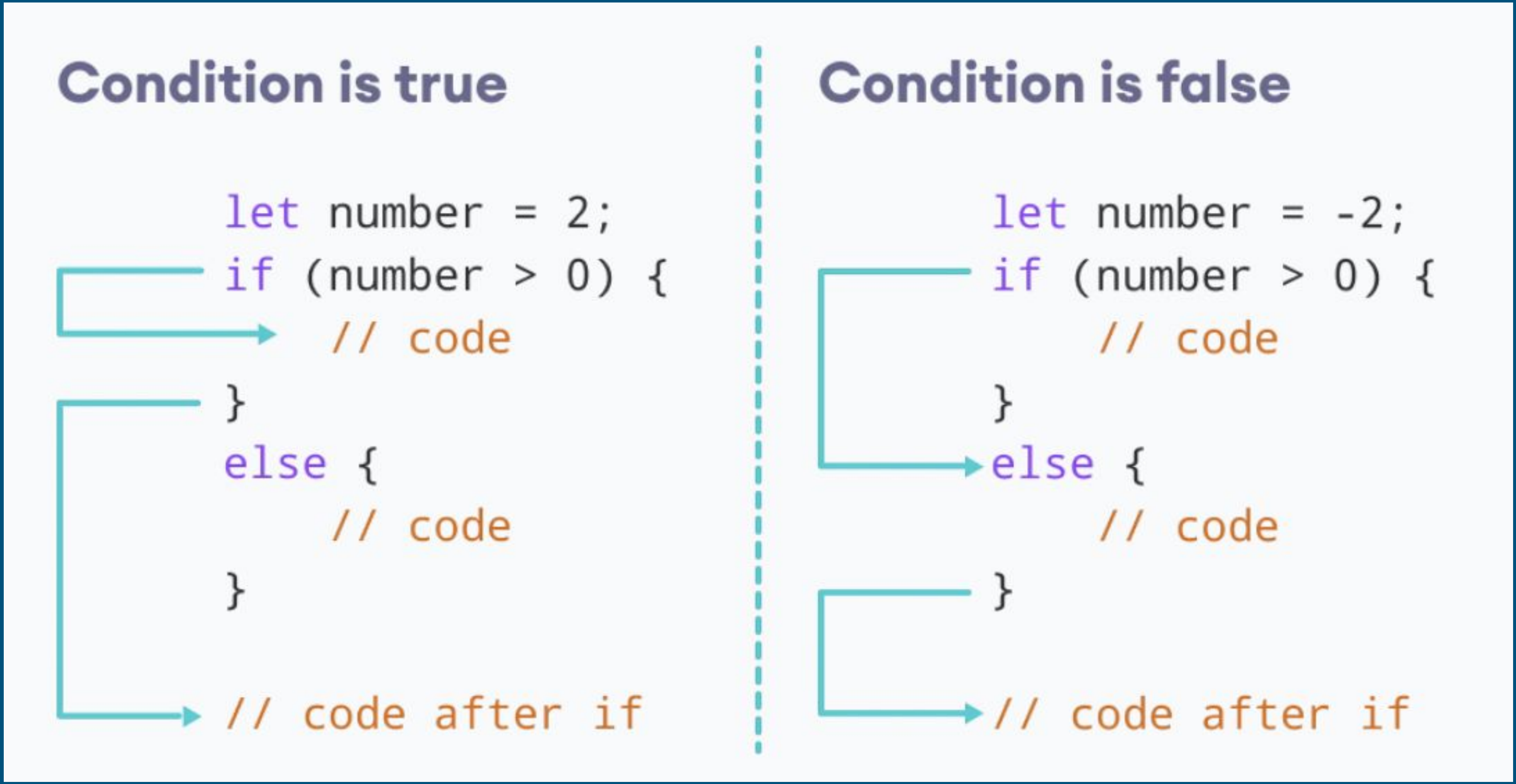
False branch

This is executed if the condition is false



Flow Chart

Example with both the condition flow



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
```

Assignment 02: Only use if else statement and commit the code, File→06_ifelse_grade.js

1. Write a FE and store into variable voteEligibe with one arg—> age to check whether he or she is eligible for voting or not, Then accordingly display message on console, don’t return the value please.

   Ex. 45, 17, 8, 20, -10, 200, 0, undefined, null

   Note:

   - If age value is 0 or negative or greater than 130 then display message → ‘In valid data’
   - If value is less than 18 then it should be → not eligible for vote
   - If value is greater than equal 18 then it should be → eligible for vote

Assignment 03: Only use if else statement and commit the code, File→06_grade.js

Design a grade system with function name as → gradeCalculation with argument marks and no return value

a. If marks is greater than equal to 90 then log the message→ Funtastic marks : ${marks}, Your grade is A+

b. If marks is greater than equal to 75 and less than 90 then log the msg → Excellent marks ${marks}, your grade is A

c. If marks is greater than equal to 50 and less than 75 then log the msg → Good marks ${marks}, your grade is B

d. If marks is greater than equal to 35 and less than 50 then log the msg → Marks is ${marks}, your grade is C, Need improvement

e. If marks is 0 or less than 0 or greater than 100 or not in valid number format ex. “Seventy” then log the msg→ Please provide the valid marks

f. Invoke function for values as shown → gradeCalculation(98), gradeCalculation(80), gradeCalculation(90), gradeCalculation(0), gradeCalculation(150), gradeCalculation(-7) , gradeCalculation(35), gradeCalculation(29), gradeCalculation(64), gradeCalculation(49), gradeCalculation(“91”), gradeCalculation(“Eighty”), gradeCalculation(undefined), gradeCalculation(null)