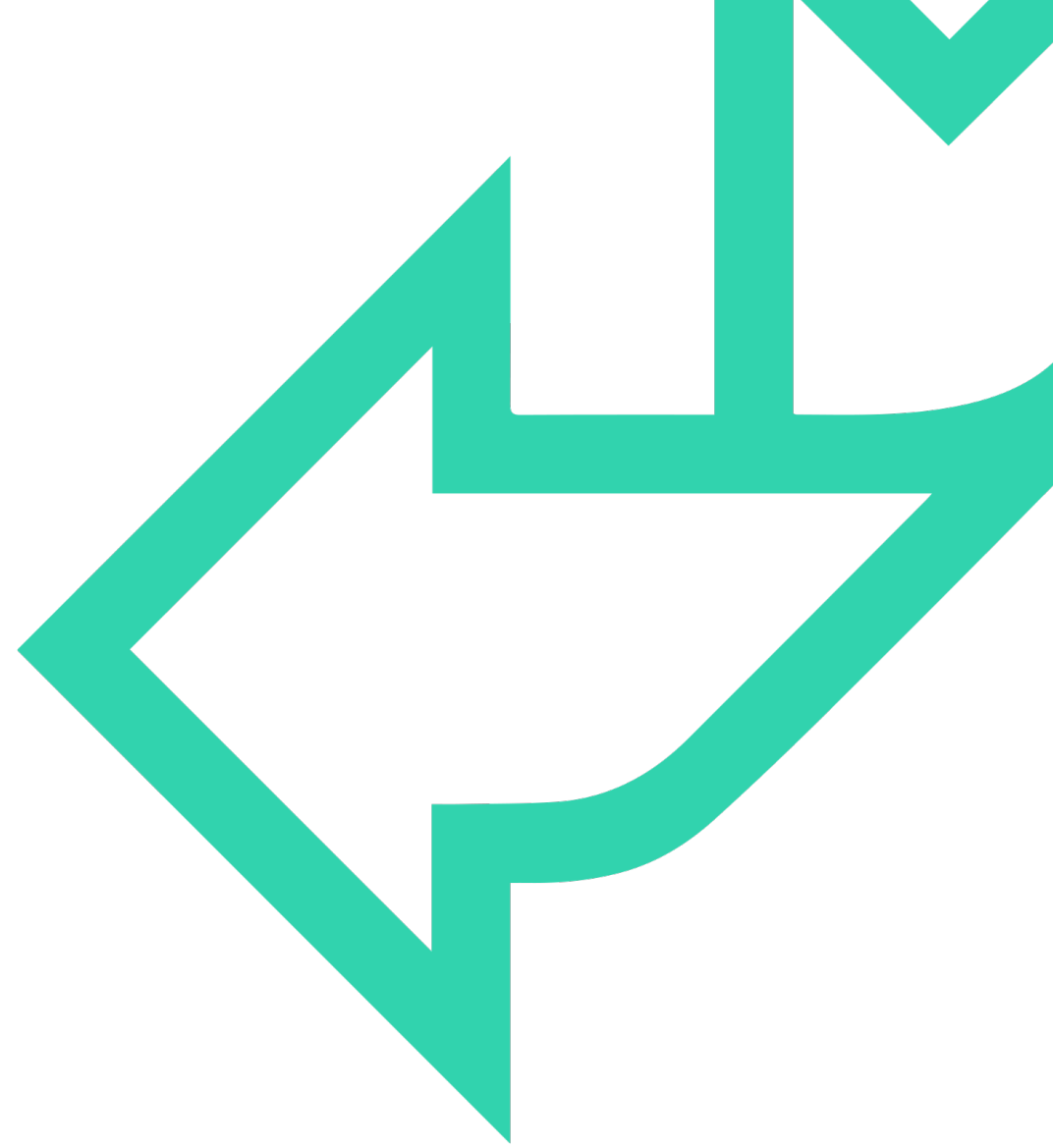




Flow of Control

JavaScript Fundamentals





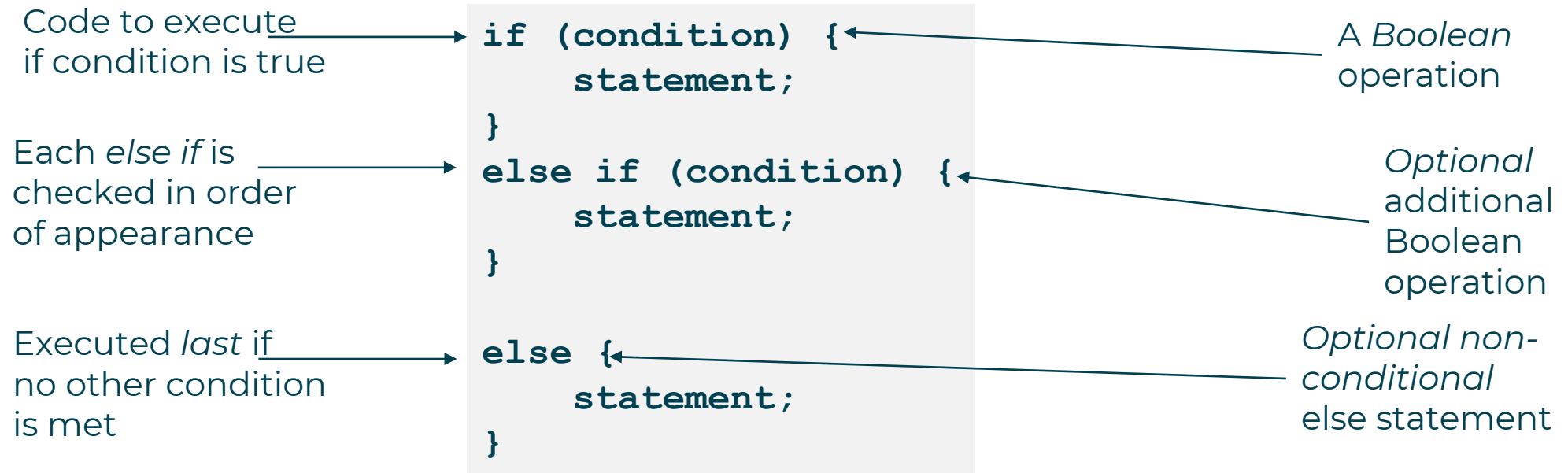
INTRODUCTION

- Understanding conditional statements
 - The if statement
 - The switch statement
- Understanding loops
 - The while and do while loops
 - for loops



QA If statements

The if statement conditionally executes if a Boolean condition is met



The if statement has optional else if and else branches

- Additional Boolean conditions executed in order

QA The ternary if

A common pattern with if statements is to assign one of two values to a variable based on a simple condition

```
let now = new Date();
let greeting = "Good";
if (now.getHours() > 17) {
    greeting += " evening.";
}
else {
    greeting += " day.";
}
```

Use of the ternary operator (?) to create a ternary-if can make this more concise

```
let now = new Date();
let greeting = "Good" + ((now.getHours() > 17) ? " evening." : " day.");
```

QA The switch statement

switch statement

- Control passes to the case label that matches the expression
- Carries on until hits a break statement
- If no case labels match, control passes to the default label (if there is one)

```
switch (expression) {  
    case label:  
        statement;  
        break;  
    case label:  
        statement;  
        break;  
    default:  
        statement;  
        break;  
}
```

QA QuickLab 4a

- Experiment with conditional statements

QA The while loop

Loops allow a set of statements to be run more than once

- Either for a fixed number of iterations or until a condition is met

The while loop has two varieties the while and do while

- The while checks before it executes

```
while (condition){  
    statement;  
}
```

- The do while always runs at least once

```
do {  
    statement;  
} while (condition);
```

← Note the semi-colon

QA The for loop

The for loop utilises a counter until a condition is met

```
for ([initial-expression]; [condition]; [loop-expression]) {  
    statement;  
}
```

In the below example “i” is incremented by 1 after each iteration

- The loop expression can be any arithmetic operation

```
for (let i = 0; i < 10; i++) {  
    i += i;  
    console.log(i);  
}
```


QuickLab 4b

- Exploring looping statements



REVIEW

- Flow of control and loops are the basis of programming
 - Along with operators
- If statements allow conditional logic
- Loops allow reuse of code without repetition

