**CONDITIONAL STATEMENTS:**

Conditional statements execute a specific action based on the results of an outcome of [true or false](https://www.digitalocean.com/community/tutorials/understanding-data-types-in-javascript#booleans).

**If Statement**

The most fundamental of the conditional statements is the if statement. An if statement will evaluate whether a statement is true or false, and only run if the statement returns true. The code block will be ignored in the case of a false result, and the program will skip to the next section.

Here is a longer examination of the basic if statement.

if (condition) {

// code that will execute if condition is true

}

As an example, let’s consider a shopping app. Say, for the functionality of this app, a user who has deposited a certain amount of funds into their account would then like to buy an item from the store.

shop.js

// Set balance and price of item

const balance = 500;

const jeans = 40;

// Check if there are enough funds to purchase item

if (jeans <= balance) {

console.log("You have enough money to purchase the item!");

}

Output

You have enough money to purchase the item!

**if-Else Statement**

With if statements, we only execute code when a statement evaluates to true, but often we will want something else to happen if the condition fails.

The else statement is written after the if statement, and it has no condition in parentheses. Here is the syntax for a basic if...else statement.

if (condition) {

// code that will execute if condition is true

} else {

// code that will execute if condition is false

}

Using the same example as above, we can add a message to display if the funds in the account are too low.

shop.js

// Set balance and price of item

const balance = 500;

const phone = 600;

// Check if there is enough funds to purchase item

if (phone <= balance) {

console.log("You have enough money to purchase the item!");

} else {

console.log("You do not have enough money in your account to purchase this item.");

}

Copy

Output

You do not have enough money in your account to purchase this item.

Since the if condition did not succeed, the code moves on to what’s in the else statement.

## if -else if-else-Statement

With if and else, we can run blocks of code depending on whether a condition is true or false. However, sometimes we might have multiple possible conditions and outputs, and need more than simply two options. One way to do this is with the else if statement, which can evaluate more than two possible outcomes.

if (condition a) {

// code that will execute if condition a is true

} else if (condition b) {

// code that will execute if condition b is true

} else if (condition c) {

// code that will execute if condition c is true

} else {

// code that will execute if all above conditions are false

}

JavaScript will attempt to run all the statements in order, and if none of them are successful, it will default to the else block.

You can have as many else if statements as necessary. In the case of many else if statements, the [switch statement](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/switch) might be preferred for readability.

// Set the current grade of the student

let grade = 87;

// Check if grade is an A, B, C, D, or F

if (grade >= 90) {

console.log("A");

} else if (grade >= 80) {

console.log("B");

} else if (grade >= 70) {

console.log("C");

} else if (grade >= 60) {

console.log("D");

} else {

console.log("F");

}

Copy

Output

B

**Ternary Operator**

The **ternary operator**, also known as the conditional operator, is used as shorthand for an if...else statement.

A ternary operator is written with the syntax of a question mark (?) followed by a colon (:), as demonstrated below.

(condition) ? expression on true : expression on false

In the above statement, the condition is written first, followed by a ?. The first expression will execute on true, and the second expression will execute on false. It is very similar to an if...else statement, with more compact syntax.

// Set age of user

let age = 20;

// Place result of ternary operation in a variable

const oldEnough = (age >= 21) ? "You may enter." : "You may not enter.";

// Print output

oldEnough;

Output

'You may not enter.'

**Switch**

The switch statement evaluates an expression and executes code as a result of a matching case. The basic syntax is similar to that of an if statement. It will always be written with switch () {}, with parentheses containing the expression to test, and curly brackets containing the potential code to execute.

Below is an example of a switch statement with two case statements, and a fallback known as default.

switch (expression) {

case x:

// execute case x code block

break;

case y:

// execute case y code block

break;

default:

// execute default code block

}

/ Set the student's grade

const grade = 87;

switch (true) {

// If score is 90 or greater

case grade >= 90:

console.log("A");

break;

// If score is 80 or greater

case grade >= 80:

console.log("B");

break;

// If score is 70 or greater

case grade >= 70:

console.log("C");

break;

// If score is 60 or greater

case grade >= 60:

console.log("D");

break;

// Anything 59 or below is failing

default:

console.log("F");

}

Copy

Output

'B'