JavaScript

**Variables & Strings**

Best to use **const** than **let**

Const – restrictions that make code more readable

Let & const are block-scoped (meaning that is we declare a variable with let or const they will not be accessed outside of the block-scope fx. an if-statement)

**// template literal**

Let username = “Jane Doe”;

Let message = `Hi ${username}, how are you?`;

Console.log(message);

Let weight = 150

Let message = `You weigh ${weight \* 0.165}, pound on the moon?`;

Console.log(moonWeight);

`I’m a string`;

\n = newline

OR

A screenshot of a computer

Description automatically generated with medium confidence

Variables that should never change = ALL\_CAPS

Boolean variables = isLoading, isModalVisible

**Conditionally Run JavaScript**

**If else statements**

The if-statement that is evaluated to true first, if run

**Switch statements**

Remember the break statement after each case.

Text

Description automatically generated

**Types and How They Can be Changed**

Use the **typeof** to find out which type the value is.

**Explicit conversion:**

console.log(Boolean(message);

console.log(String(42);

**Truthy and Falsy**

**Rule: everything that is not falsy is truthy**

**Falsy values are:**

false

0

‘’, “”, `` (empty string)

null

undefined (empty variable)

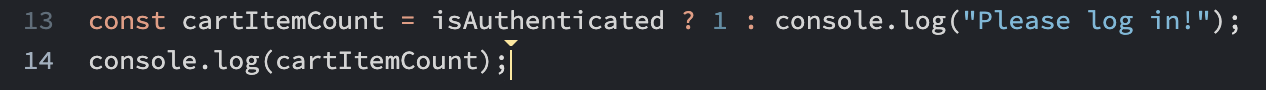
NaN

Graphical user interface, text, application

Description automatically generated

Checks whether the username variable is an empty string. Also works if the username variable is null or undefined.

Ternaries



If we want to console.log or do another thing other than assigning a value, we should use an if/else statement or switch instead

Chaining ternaries

DO NOT DO THIS:



We should avoid chaining ternaries – so for complex conditions use conditionals

Short-circuiting

* If response if an valid username, the username will be the response otherwise it will be “guest”

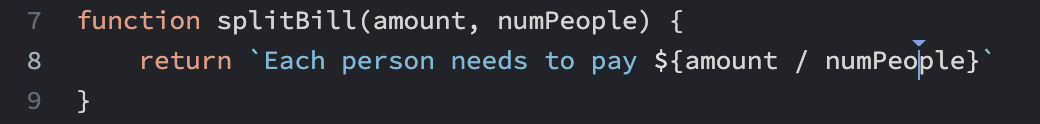
Text

Description automatically generated

When combining

* && operators are run **before** ||
* If we want an OR operator to run before an && operator, we need to put the OR operator in parenthesis ().

**Function**



Closures

* Allow us to remember values
* For example if we have a likeCount function, we want to remember if the count has already been liked before.
* Global variables lives as long as the program is running
* Local variables lives as long the function is running