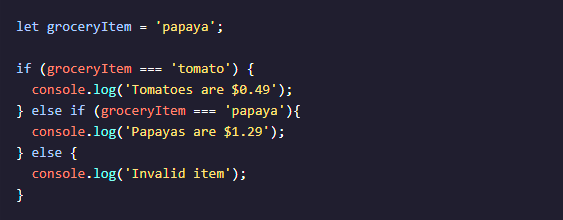
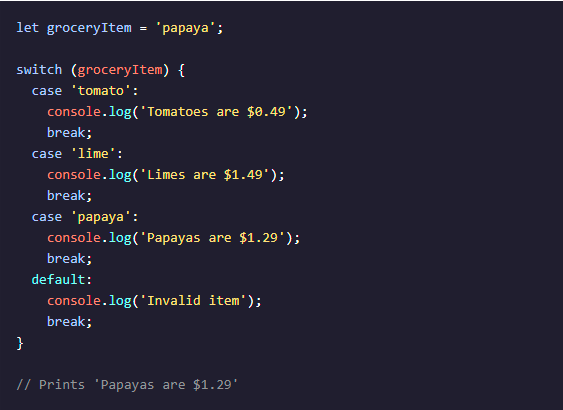
The switch keyword

else if statements are a great tool if we need to check multiple conditions. In programming, we often find ourselves needing to check multiple values and handling each of them differently. For example:



In the code above, we have a series of conditions checking for a value that matches a groceryItem variable. Our code works fine, but imagine if we needed to check 100 different values! Having to write that many else if statements sounds like a pain!

A switch statement provides an alternative syntax that is easier to read and write. A switch statement looks like this:

  
// Prints 'Papayas are $1.29'

* The switch keyword initiates the statement and is followed by ( ... ), which contains the value that each case will compare. In the example, the value or expression of the switch statement is groceryItem.
* Inside the block, { ... }, there are multiple cases. The case keyword checks if the expression matches the specified value that comes after it. The value following the first case is 'tomato'. If the value of groceryItem equalled 'tomato', that case‘s console.log() would run.
* The value of groceryItem is 'papaya', so the third case runs— Papayas are $1.29 is logged to the console.
* The break keyword tells the computer to exit the block and not execute any more code or check any other cases inside the code block. Note: Without break keywords, the first matching case will run, but so will every subsequent case regardless of whether or not it matches—including the default. This behavior is different from if/else conditional statements that execute only one block of code.
* At the end of each switch statement, there is a default statement. If none of the cases are true, then the code in the default statement will run.