JS If and Switch Statements

Lesson Time: 45 Minutes

Simple Logic

Sometimes we want a block of code to only run depending on the situation. The **if statement** allows us to provide a condition, and if the condition is met, the block of code in the if statement will be executed. If the condition is not met, or not “true”, the code block will be skipped.

```
"use strict";

let sky = 'blue';

if(sky=='blue'){
    console.log('The sky is blue during the day.');
}
else if(sky=='black'){
    console.log('The sky is black at night.')
}
else{
    console.log('The sky isn\'t blue or black. Something is very wrong!');
}

```

In this block of code, if the variable sky has a value of blue, the script will print “The sky is blue during the day”. The remaining code would not run.

Else if allows us to specify more conditions to test. If ‘sky’ does not have a value of blue, we will test again for the value of black. If that condition is met, the remaining code will not run.

If there are no remaining conditions to test, and no conditions have been met, the statements in the ***else*** block will be ran.

Ternary condition

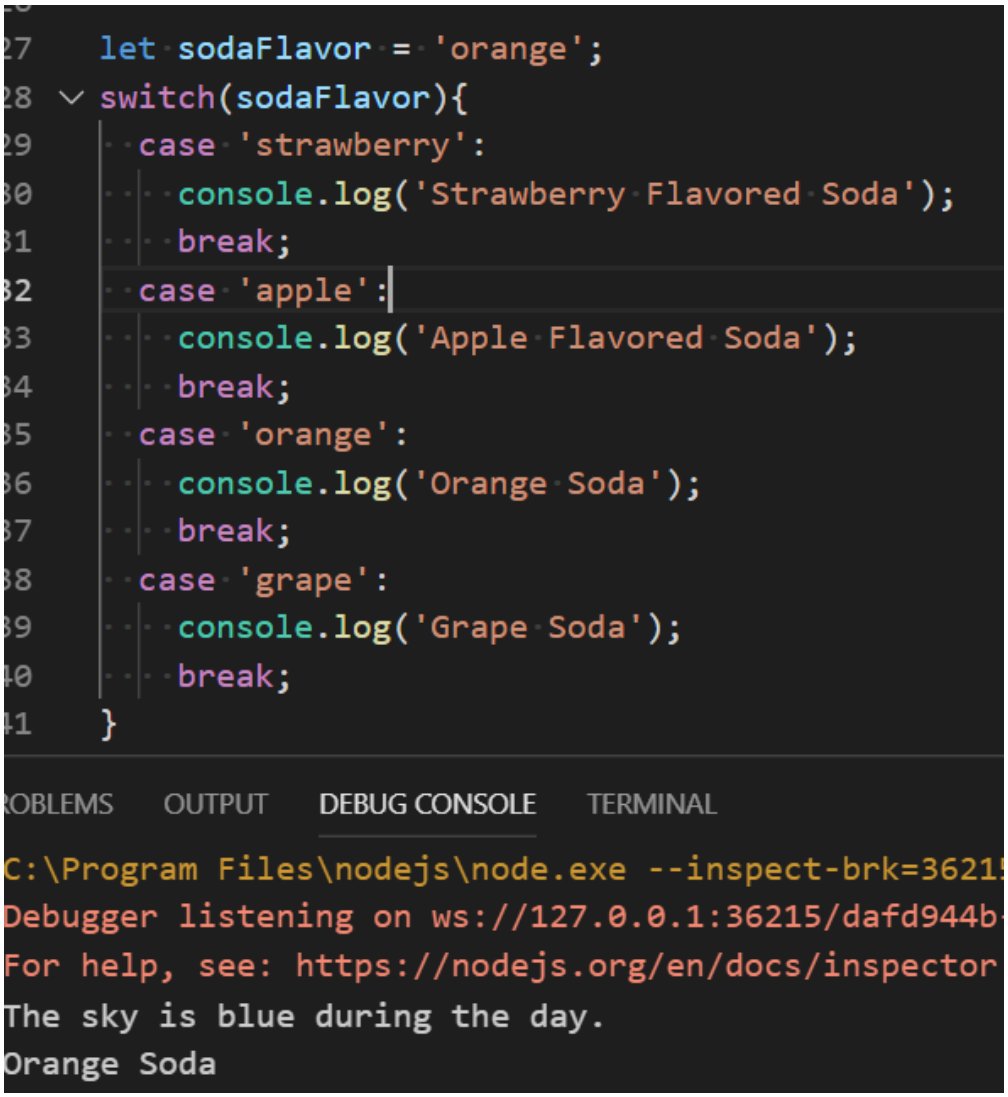
```
let age = 30;
let canDrive = (age > 16) ? true : false;
```

We can take a shortcut for writing an if...else statement to assign a value. The code above says “If age is greater than 16, set canDrive to true, else, set it false. It’s the exact same as the longer code below:

```
let age = 30;
let canDrive = false;
if(age > 16){
  canDrive = true;
}
else {
  canDrive = false;
}
```

Switch Statement

We can take another shortcut to writing multiple if...else statements with a switch statement. A Switch statement will compare a single value to many different test cases.



In this code, we test sodaFlavor for many possible values. Each value we test for is case value. When one of our case values is true, we run the code for the case and then we break, or stop, our testing.

Break/Continue

When running loops, functions and switch statements, break and continue are useful commands. **Break** allows us to “break out of” or stop what we are doing. The loop will end, the switch testing will stop, the function will exit. We use break in the switch example above to break out of the switch once a match is found.

Continue is used in a loop stop executing the loop earlier. However, instead of completely ending the loop, continue moves on to the next run of the loop. In the example below, line 50 of our script is never executed when i is equal to 3. The loop skips ahead.

```
43  for (let i = 1; i < 6; i++) {
44
45      if(i == 3){
46          console.log("i is equal to 3. Skipping ahead.")
47          continue;
48      }
49
50      console.log("Loop has ran " + i + " times.");
51
52  }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
C:\Program Files\nodejs\node.exe --inspect-brk=49827 ifelse.js
Debugger listening on ws://127.0.0.1:49827/31b6c163-9543-46cf-808a-bde2f25630ca
For help, see: https://nodejs.org/en/docs/inspector
The sky is blue during the day.
Orange Soda
Loop has ran 1 times.
Loop has ran 2 times.
i is equal to 3. Skipping ahead.
Loop has ran 4 times.
Loop has ran 5 times.
```

.

JS IF/Switch Lab

Lesson Time: 45 Minutes

Begin this project by typing this line of code

```
let randomNum = Math.random();
```

This line of code creates a random number between 0 and 1. Using this function, we'll create a simple coin flipping game.

Here are the tasks for this assignment.

1. Create a function called CoinFlip()
    - CoinFlip() will create a random number.
    - If the number created by coin flip is greater than or equal to 0.5, we will say the coin landed on HEAD. Coinflip should print to the console "HEADS!"
    - If the number is less than 0.5, we will say the coin landed on TAILS. CoinFlip() should print to the console "TAILS!"

2. Use a for loop to run CoinFlip() 10 times. You should see it print heads or tails each time.

3. Use this article on MDN to see a code example on how to randomly create an integer.
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random

4. Copy the getRandomInt function from the MDN article into your script.
5. Create an array called **years**. The array should have these values
    - 1980, 1990, 2000, 2010
6. Create a variable called **someYear** and assign it the results of getRandomInt(0,4)
7. Write a switch statement to evaluate **someYear**.
    - The switch should test each possible value someYear could have (0 to 3).
    - The switch should print to the console the **years[someYear]** when the true value is found.

Question & Hints
The answer can be found by carefully reading the MDN article.

- HINT: someYear can have a value between 0 and 3.
- Question: So why do we write getRandomInt(0,4) when assigning values to someYear?