Control Flow:

1. What types of conditional statements are available in your language? (if/else, if/then/else, if/elseif/else). Does allow for statements other than "if" (for example, Perl has an "unless" statement, which does the opposite of "if"!)

    If, else if, and else are available in JavaScript. Any other conditional structures would have to be hand coded into a function.

2. Does your language use short-circuit evaluation? If so, make sure that your code includes an example.

    JavaScript uses short circuit evaluation; this can be illustrated by writing an or statement where the first variable is true and the second is undeclared. There won't be a problem with the undeclared variable because the program stops at the first true statement.

3. How does your programming language deal with the "dangling else" problem?

    The dangling else problem has to do with nested ifs followed by an else. In some cases there's ambiguity about which if the else is connected to. In JavaScript the else can't respond to the nested if unless it is also nested within the same curly brackets. An else is always connected to the last if that isn't separated from it by curly brackets.

4. Does your language include multiple types of loops (while, do/while, for, foreach)? If so, what are they and how do they differ from each other?

    JavaScript includes while, do/while, for, and foreach loops. While loops are controlled by a conditional statement, which is checked before each loop through. Do/while loops are the same, except the conditional statement is checked *after* each loop through. For loops are iterative and can increment the loop control variable without extra lines of code. Foreach loops are more specialized; they step through a list item by item and stop when they reach the end.

5. Can you use break or continue statements (or something similar) to exit loops?

    A break statement will cause the program to exit the loop completely and continue statements will just skip over one round and continue to loop.

6. If your language supports switch or case statements, do you have to use "break" to get out of them? Can you use "continue" to have all of them evaluated?

    If you try to put a continue statement in a switch, there will be errors when you try to run it.

7. Is there anything special in terms of control flow that your language does that isn't addressed in this assignment? If so, what is it and how does it work? Make sure to include an example of it in your code as well.

```
num = 0;
{num = 250;}
console.log("num = "+num);
```

    In JavaScript this statement will print num = 250 because the scope of the change is the same as the scope of when num was first instantiated. In languages like C++ it'd print num = 0 unless you put the print statement in the curly brackets.