Control Statements

Control statements in JavaScript are programming constructs that allow you to control the flow of your code's execution. They determine which blocks of code should be executed under specific conditions or how many times certain codes should be repeated. Control statements are essential for creating dynamic and responsive programs by enabling you to make decisions and loop through code.

There are four main types of control statements in JavaScript:

1. **Conditional Statements:** These statements allow you to execute different blocks of code based on certain conditions. The most common conditional statements are:
    - **if**: Executes a block of code if a specified condition is true.
    - **if else**: Executes a block of code if a specified condition is true and another block is executed if the specified condition is false.
    - **if else if ladder**: Used when we need to execute a set of the statement(s) from among multiple sets of statement(s) according to multiple conditions.
    - **nested if else:** it is similar to **if else if ladder** but have different syntax.
2. **Selection Statements:** The selection statement allow you to select and execute one of many code blocks based on the value of an expression. We have only a single select statement called as **switch case default.**
3. **Looping Statements (Iteration Statements):** Looping statements allow you to execute a block of code repeatedly. They are used when you need to perform a specific action multiple times. The main looping statements are:
    - **for**: Executes a block of code a specific number of times, based on a condition.
    - **while**: Executes a block of code as long as a specified condition is true.
    - **do...while**: Similar to **while**, but the block of code is executed at least once before the condition is checked.
4. **Jump Statements:** Jump statements are used to alter the normal flow of execution by transferring control to a different part of the code. The primary jump statements in JavaScript are:
    - **break**: Terminates the current loop, switch, or label statement.
    - **continue**: Skips the current iteration of a loop and proceeds to the next iteration.
    - **return**: Exits the current function and optionally returns a value to the caller.