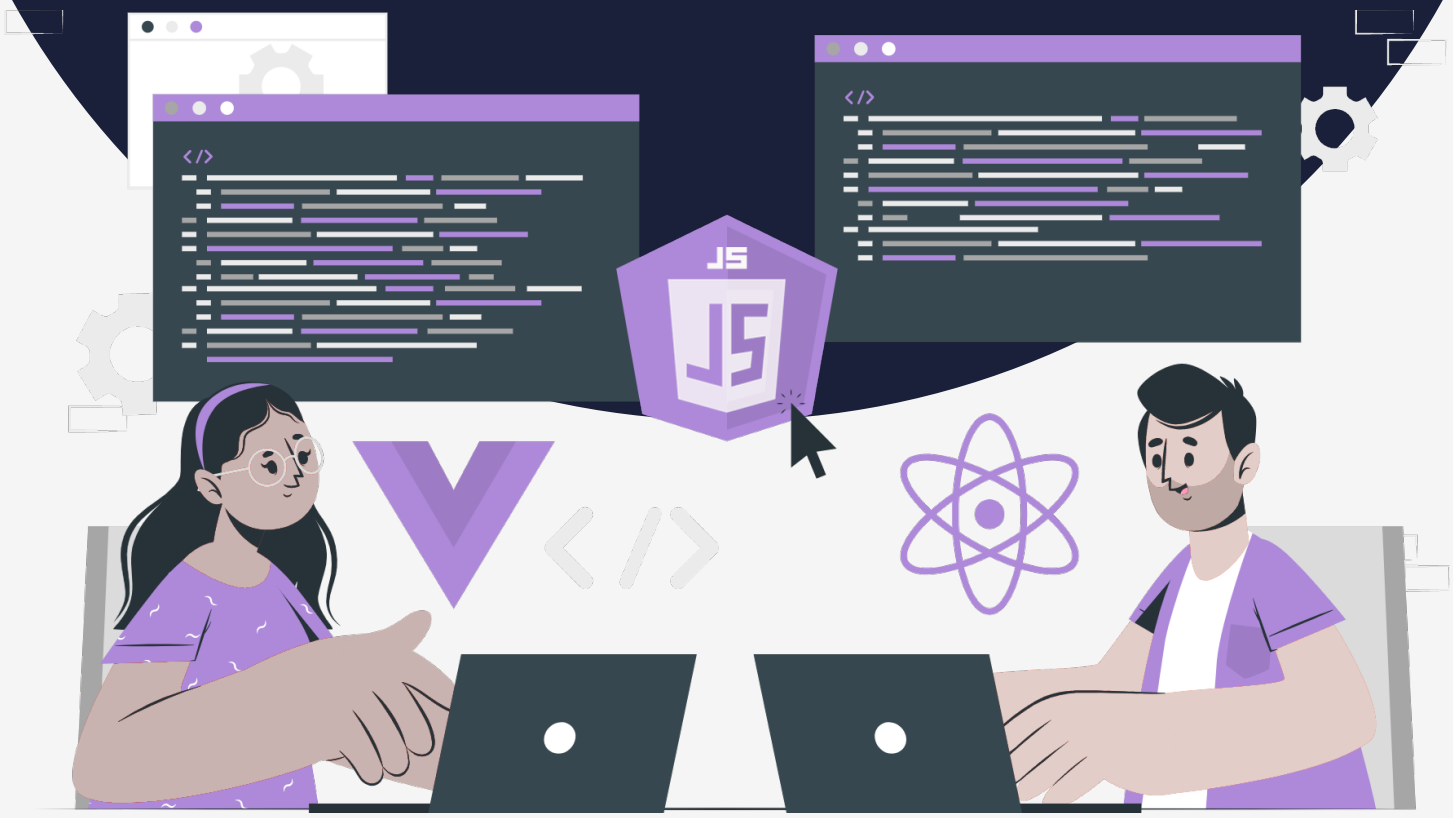
Lesson:

Introduction to loops



PW SKILLS

List of content:

What are loops?

As you may see in the diagram, a loop, is a structure, series, or process, the end of which is connected to the beginning.

In programming, a loop is a set of instructions that are repeatedly carried out until a specific condition is met in computer programming. Typically, a certain action is taken, such as receiving and modifying a piece of data, and then a condition is verified, such as determining whether a counter has reached a predetermined value. If not, the following instruction in the sequence directs the computer to go back to the first instruction in the series and repeat it. A loop is a fundamental concept in programming that is frequently applied when creating programs.

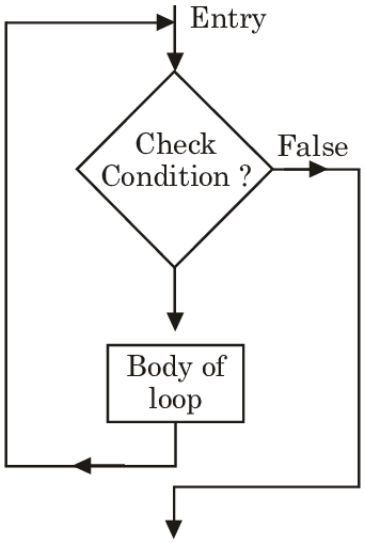
A loop that never ends is known as an infinite loop.
As a result, the loop keeps repeating until the operating system notices it and crashes the program, or until another event happens (such as having the program automatically terminate after a certain duration of time).

Why loops?
The control statement and the body of a loop can be conceptually separated into two components. The prerequisites for the execution of a loop's body are listed in the control statement of the loop. The conditions in the control statement must be true for each loop iteration. The block of code or series of logical assertions that will be executed repeatedly make up the body of a loop.

As a result, you will save time when using a loop in your program by eliminating the need to repeatedly write the body of the loop's code. As long as the conditions in the control statement are true, the code block will be run several times. The loop will end when the conditions in the control statement are no longer true. The loop will continue to run even if the conditions are not explicitly stated in the control statement. We refer to these loops as infinite loops. A loop automatically becomes infinite if there is no termination condition specified in the control statement.

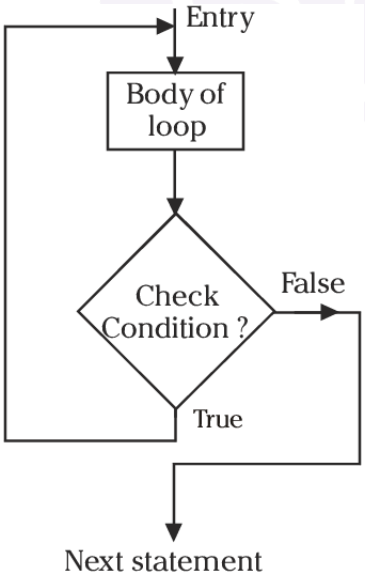
Types of loops:
When you have a basic understanding of the syntax and purpose of the different forms of loops, the concept of "what is Loop" will become evident. In most computer programming languages, loops can be divided into two categories: entry-controlled loops and exit-controlled loops.

1. Entry controlled loop



The control statement is written right at the start of an entry-controlled loop. Pre-checking Loop is another name for this kind of loop. The control statements' conditions are initially verified, and the body of the loop is only run if the conditions are met. The lines of code in the Loop's body won't be run if the condition turns out to be false. For loop is an entry-controlled loop, which means the control statements are written at the very beginning of the loop structure.

2. Exit controlled loop



Here, the control statements are written at the end of the loop structure which means that the loop will at least run once even if the condition of loop is false.
do-while loop is an exit-controlled loop.
Each of these loops would be discussed in the subsequent lectures.