Loops in TypeScript

Loops are handy, if you want to run the same code over and over again, each time with a different value.

A loop is a control flow structure that allows a certain block of code to be executed repeatedly, based on a specified condition. Loops are essential for automating repetitive tasks and iterating over collections of data. They help make code more efficient, concise, and easier to maintain.

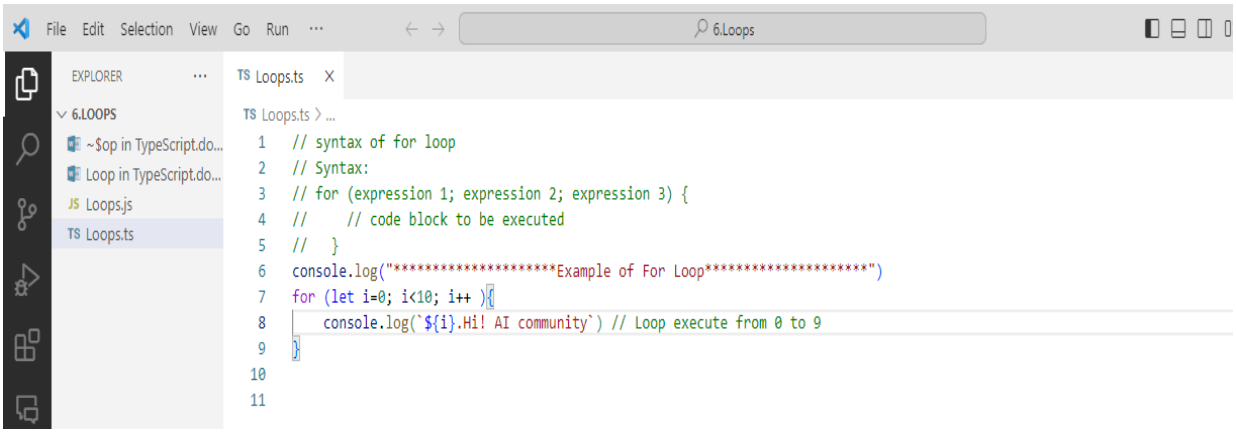
Loops are fundamental to building algorithms and solving problems in programming. They enable you to perform repetitive tasks efficiently and are a key component of many programming languages.

There are various types of loops, and each serves a specific purpose. The most common types of loops are given below:

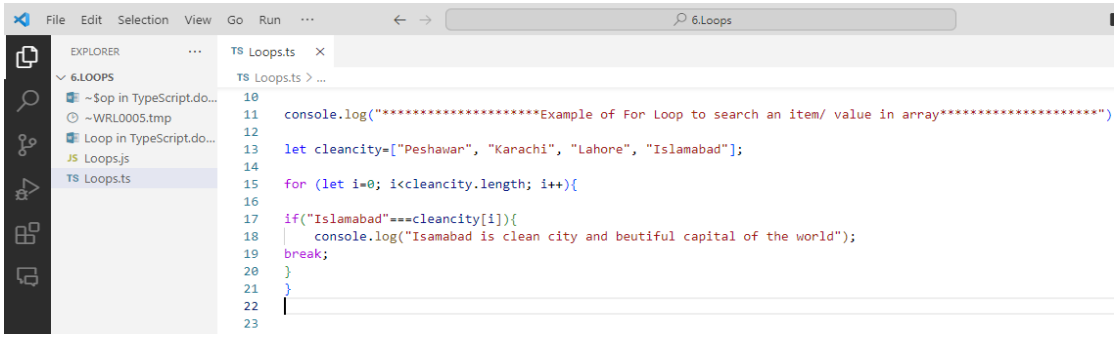
1. For Loop
2. While Loop
3. do While Loop

For Loop:

Executes a block of code a specified number of times. It typically has three components: initialization, condition, and an increment/decrement statement.

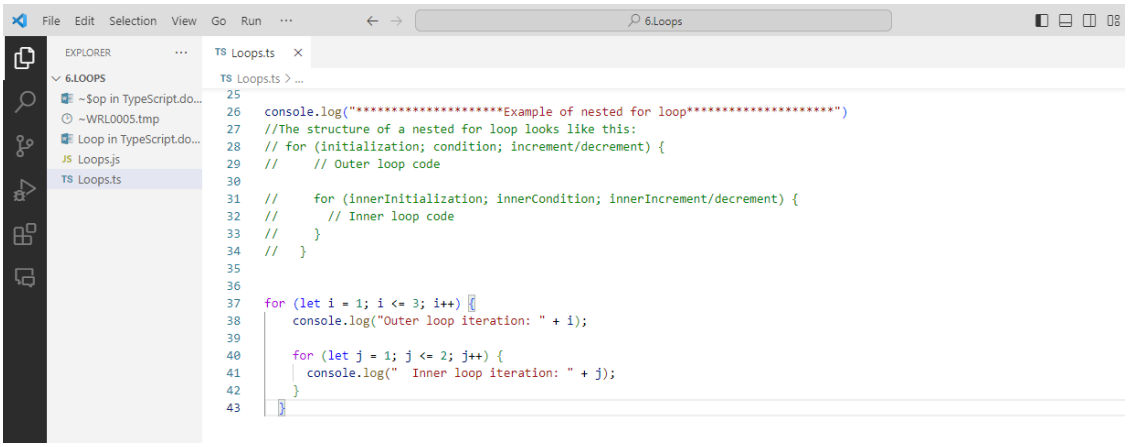


Example of For Loop to search an item/ value in array*****



Nested for loop:

A nested for loop is a loop inside another loop. This means that there is an outer loop, and inside it, there is one or more inner loops. Each iteration of the outer loop triggers the inner loop to run its full course.



Nested for loops are commonly used for tasks that involve working with two-dimensional arrays, matrix operations, or when you need to perform a certain action for each combination of elements from two sets.