

# 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.

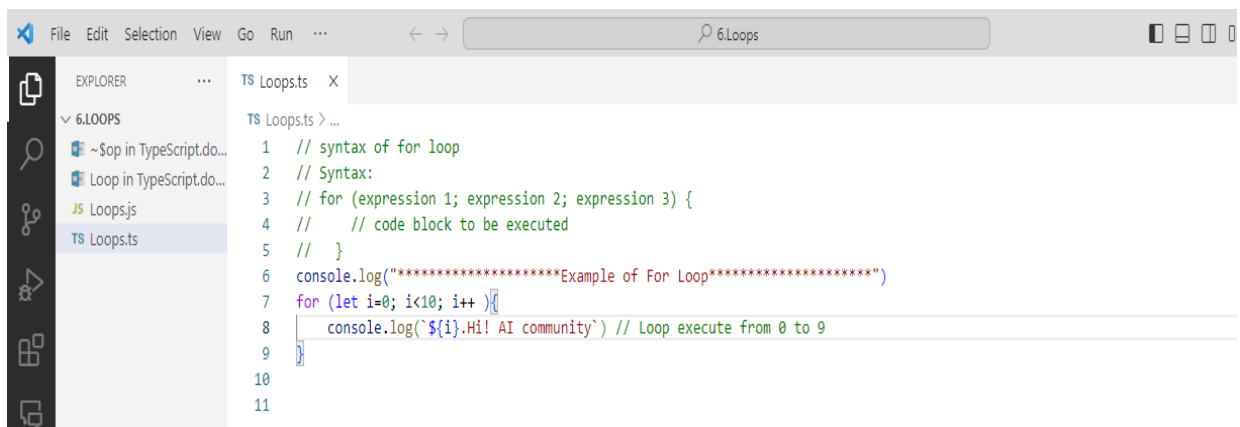
Loops are fundamental to building algorithms and solving problems in programming. They enable you to perform repetitive tasks efficiently and are a key component of many programming languages.

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

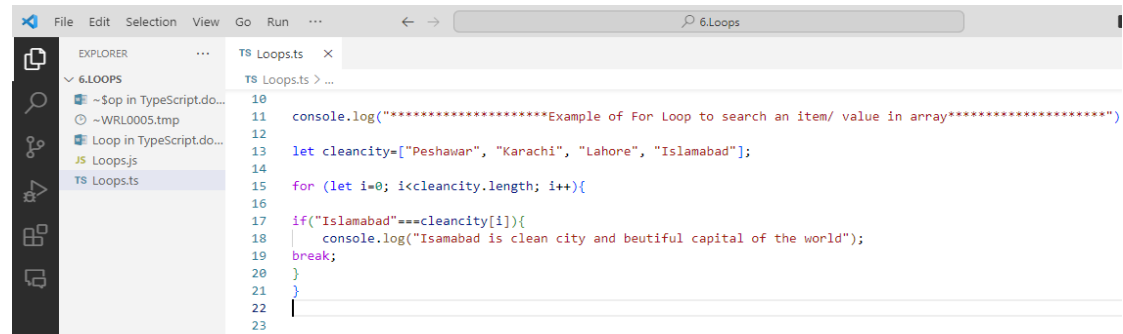
## For Loop:

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



```
TS 6.Loops.ts > ...
1 // syntax of for loop
2 // Syntax:
3 // for (expression 1; expression 2; expression 3) {
4 //     // code block to be executed
5 // }
6 console.log("*****Example of For Loop*****")
7 for (let i=0; i<10; i++) {
8     console.log(`${i}.Hi! AI community`) // Loop execute from 0 to 9
9 }
10
11
```

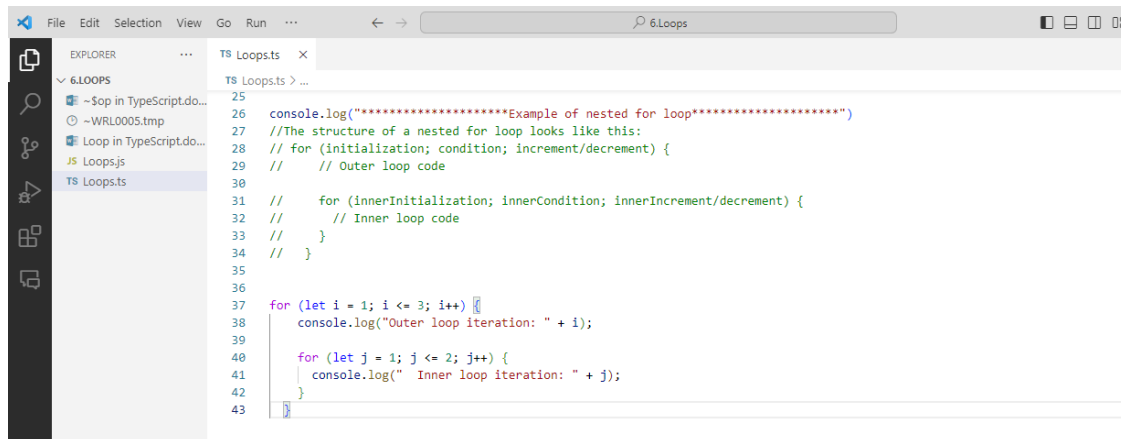
\*\*\*Example of For Loop to search an item/ value in array\*\*\*\*\*



```
10
11 console.log("*****Example of For Loop to search an item/ value in array*****")
12
13 let cleancity=["Peshawar", "Karachi", "Lahore", "Islamabad"];
14
15 for (let i=0; i<cleancity.length; i++){
16
17   if("Islamabad"===cleancity[i]){
18     console.log("Islamabad is clean city and beautiful capital of the world");
19     break;
20   }
21 }
22
23
```

## 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.



```
25
26 console.log("*****Example of nested for loop*****")
27 //The structure of a nested for loop looks like this:
28 // for (initialization; condition; increment/decrement) {
29 //   // Outer loop code
30 //
31 //   for (innerInitialization; innerCondition; innerIncrement/decrement) {
32 //     // Inner loop code
33 //   }
34 // }
35
36
37 for (let i = 1; i <= 3; i++) {
38   console.log("Outer loop iteration: " + i);
39
40   for (let j = 1; j <= 2; j++) {
41     console.log(" Inner loop iteration: " + j);
42   }
43 }
```



```
45 console.log("*****2nd Example of nested for loop*****")
46
47 let firstNames = ["BlueRay ", "Upchuck ", "Lojack ", "Gizmo ", "Do-Rag "];
48 let lastNames = ["Zzz", "Burp", "Dogbone", "Droop"];
49 let fullNames: string[] = [];
50
51 for (let i = 0; i < firstNames.length; i++) {
52   for (let j = 0; j < lastNames.length; j++) {
53     fullNames.push(firstNames[i]+" "+lastNames[j]);
54   }
55 }
56 console.log(fullNames)
```

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.