TypeScript - for Loops

TypeScript supports the following for loops:

1. for loop
2. for..of loop
3. for..in loop
4. while loop

for Loop

- ➔ The for loop is used to execute a block of code a given number of times, which is specified by a condition.

```
for (first expression; second expression; third expression ) {
    // statements to be executed repeatedly
}
```

- ➔ Here, the first expression is executed before the loop starts. The second expression is the condition for the loop to execute. And the third expression is executed after the execution of every code block.

JavaScript

```
for (let i = 0; i < 3; i++) {
 console.log ("Block statement execution no." + i);
}
```

for..of loop

- ➔ The for...of loop is used to iterate and access elements of an array, list, or tuple collection.
- ➔ It returns elements from a collection

JavaScript

```javascript
let arr = [10, 20, 30, 40];

for (var val of arr) {
 console.log(val); // prints values: 10, 20, 30, 40
}
```

for..in loop

- ➔ The for...in loop iterates through a list or collection and returns an index on each iteration.
- ➔ It returns an index on each iteration.

JavaScript

```javascript
let arr = [10, 20, 30, 40];


for (var index in arr) {
 console.log(index); // prints indexes: 0, 1, 2, 3


 console.log(arr[index]); // prints elements: 10, 20, 30, 40
}
```

You can also use `let` instead of `var`. The difference is that the variable declared using `let` will not be accessible out of the for..in loop, as shown below.

JavaScript

```javascript
let arr = [10, 20, 30, 40];

for (var index1 in arr) {

 console.log(index1); // prints indexes: 0, 1, 2, 3

}

console.log(index1); //OK, prints 3

for (let index2 in arr) {

 console.log(index2); // prints elements: 0, 1, 2, 3

}

console.log(index2); //Compiler Error: Cannot find index2
```

while loop

- ➔ The while loop is another type of loop that checks for a specified condition before beginning to execute the block of statements. The loop runs until the condition value is met.

```
while (condition expression) {
    // code block to be executed
}
```

JavaScript

```
let i: number = 2;

while (i < 4) {

  console.log( "Block statement execution no." + i )

  i++;

}
```

do..while loop

- ➔ The do..while loop is similar to the while loop, except that the condition is given at the end of the loop.
- ➔ The do..while loop runs the block of code at least once before checking for the specified condition. For the rest of the iterations, it runs the block of code only if the specified condition is met.

```
do {
// code block to be executed
}
while (condition expression);
```

JavaScript

```javascript
let i: number = 2;

do {

  console.log("Block statement execution no." + i )

  i++;

} while ( i < 4)
```