TypeScript Mini-Course: Loops, Functions & Arrays

---

1. Introduction

This mini-course covers the basics of **loops**, **functions**, and **arrays** in TypeScript with examples and exercises for practice.

---

2. Loops in TypeScript

2.1 `for` Loop

```
for (let i = 1; i <= 5; i++) {
  console.log("Count:", i);
}
```

2.2 `while` Loop

```
let x = 1;
while (x <= 5) {
  console.log("x =", x);
  x++;
}
```

2.3 `do...while` Loop

```
let num = 1;
do {
  console.log("num =", num);
  num++;
} while (num <= 3);
```

2.4 `for...of` Loop (Array)

```
const fruits = ["Apple", "Banana", "Orange"];
for (const fruit of fruits) {
```

```
  console.log(fruit);
}
```

2.5 `for...in` Loop (Object)

```
const person = { name: "Nader", age: 25, country: "Tunisia" };
for (const key in person) {
  console.log(`${key}: ${person[key as keyof typeof person]}`);
}
```

---

3. Functions in TypeScript

3.1 Basic Function

```
function sayHello(): void {
  console.log("Hello TypeScript!");
}
sayHello();
```

3.2 Function with Parameters

```
function greet(name: string): void {
  console.log(`Hello, ${name}!`);
}
greet("Nader");
```

3.3 Function with Return Value

```
function add(a: number, b: number): number {
  return a + b;
}
console.log(add(5, 7));
```

3.4 Arrow Function

```
const multiply = (x: number, y: number): number => x * y;
console.log(multiply(4, 5));
```

---

4. Arrays in TypeScript

4.1 Basic Arrays

```
let numbers: number[] = [1, 2, 3, 4, 5];
let fruits: string[] = ["Apple", "Banana", "Orange"];
```

4.2 Mixed Types

```
let mixed: (string | number)[] = [1, "Two", 3, "Four"];
```

4.3 Accessing Elements

```
let colors: string[] = ["Red", "Green", "Blue"];
console.log(colors[0]); // Red
```

4.4 Looping Through Arrays

```
let nums: number[] = [10, 20, 30, 40];
for (const num of nums) console.log(num);
nums.forEach((num) => console.log(num));
```

4.5 Adding/Removing Elements

```
let animals: string[] = ["Cat", "Dog"];
animals.push("Rabbit");
animals.pop();
animals.unshift("Elephant");
animals.shift();
```

4.6 Array Methods

```
let nums2: number[] = [1,2,3,4,5];
console.log(nums2.map(n => n*2)); // [2,4,6,8,10]
console.log(nums2.filter(n => n%2===0)); // [2,4]
console.log(nums2.reduce((a,b)=>a+b,0)); // 15
console.log(nums2.find(n => n>3)); // 4
```

4.7 Multidimensional Arrays

```
let matrix: number[][] = [[1,2,3],[4,5,6],[7,8,9]];
for (let row of matrix) {
  for (let col of row) process.stdout.write(col+" ");
  console.log();
}
```

4.8 Array of Objects

```
type Person = { name: string; age: number };
let people: Person[] = [{name:"Alice", age:25}, {name:"Bob", age:30}];
for (const p of people) console.log(`${p.name} is ${p.age} years old`);
```

4.9 Combine Arrays

```
let arr1 = [1,2,3];
let arr2 = [4,5,6];
console.log([...arr1,...arr2]);
```

4.10 Practical Example — Average

```
let scores: number[] = [80, 90, 70, 60];
function average(arr: number[]): number {
  return arr.reduce((sum,n)=>sum+n,0)/arr.length;
}
console.log("Average:", average(scores));
```

---

5. Exercises

1. Print numbers from 1 to N using a loop.
2. Sum numbers from 1 to N.
3. Print all even numbers between 1 and 20.
4. Calculate factorial of a number.
5. Sum all elements in an array.
6. Reverse a string.
7. Count vowels in a word.
8. Find the maximum number in an array.
9. Check if a number is prime.
10. Print multiplication table of a number.

---

6. Summary

- Loops: `for`, `while`, `do...while`, `for...of`, `for...in`
- Functions: basic, with parameters, return values, arrow functions
- Arrays: single, multidimensional, objects, array methods, combination
- Exercises for hands-on practice