**Type Annotations With Arrays**

Type annotations with arrays allow you to specify the expected data types of the array elements, ensuring type safety in your code.

// Annotating an array of strings

let fruits: string[] = ["apple", "banana", "cherry"];

// Using a loop to process each element in the array

for (let i = 0; i < fruits.length; i++) {

console.log(`Fruit: ${fruits[i].toUpperCase()}`);

}

1. We annotate the fruits array as an array of strings.
2. We use a for loop to iterate over the fruits array and display each fruit in uppercase letters.

**Type Annotations With Multidimensional Arrays**

let arrayOne: number[] = [1, 2, 3, 4, 5];

let arrayTwo: string[] = ["A", "B", "C"];

let arrayThree: (string | number)[] = [1, 2, 3, 4, "A", "B", "C"];

In this code:

1. arrayOne is declared as an array of numbers, annotated with the type number[]. This means it can only contain numeric values.
2. arrayTwo is declared as an array of strings, annotated with the type string[]. It should only contain string values.
3. arrayThree is declared as an array that can hold elements of either type string or number, annotated with (string | number)[]. This array allows for a mixture of numeric and string values within the same array.