String concatenation in TypeScript refers to the process of combining multiple strings into a single string. TypeScript provides several ways to concatenate strings, including using the + operator, template literals (introduced in ECMAScript 6), and string interpolation. Let's explore each method:

**1. Using the + Operator**

The + operator can be used to concatenate strings in TypeScript:

let firstName: string = "John"; let lastName: string = "Doe";

let fullName: string = firstName + " " + lastName;

console.log(fullName); // Output: "John Doe"

### 2. Using Template Literals

Template literals provide a more flexible and readable way to concatenate strings. They use backticks () instead of quotes and allow for embedded expressions using ${}`:

let firstName: string = "Alice"; let lastName: string = "Smith";

let fullName: string = `${firstName} ${lastName}`;

console.log(fullName); // Output: "Alice Smith"

Template literals support multi-line strings as well, making them useful for constructing more complex string outputs:

### 3. Using String Interpolation

String interpolation is similar to template literals but is specifically used within string literals ("" or ''). It allows variables and expressions to be embedded directly in a string using ${}:

let firstName: string = "Bob"; let lastName: string = "Brown";

let greeting: string = "Hello, " + firstName + " " + lastName + "!";

console.log(greeting); // Output: "Hello, Bob Brown!"