In TypeScript, parameters and arguments are fundamental concepts in the context of functions. Here’s a breakdown of what they are and how they differ:

**Parameters**

Parameters are the variables listed as part of a function's definition. They act as placeholders for the values that will be passed to the function when it is called.

function functionName(parameter1: type1, parameter2: type2): returnType {

// function body}

Example:

function greet(name: string, age: number): string {

return `Hello, my name is ${name} and I am ${age} years old.`;}

In this example, name and age are parameters of the greet function.

**Arguments**

Arguments are the actual values that are passed to the function when it is called. They correspond to the parameters defined in the function.

**Syntax:**

**functionName(argument1, argument2);**

Example:

const greeting = greet("Alice", 30);

console.log(greeting);

In this example, "Alice" and 30 are arguments passed to the greet function.

### Key Points

1. **Default Parameters**: You can provide default values for parameters. If an argument is not provided, the parameter will take on the default value.

function greet(name: string = "Guest"): string {

return `Hello, ${name}!`;}

Here, if name is not provided when greet is called, it will default to "Guest".

1. **Optional Parameters**: Parameters can be marked as optional using the ? operator.

function greet(name?: string): string {

if (name) {

return `Hello, ${name}!`;

} else {

return "Hello!"; } }

1. **Rest Parameters**: You can use rest parameters to handle an indefinite number of arguments as an array.

**function sum(...numbers: number[]): number {**

**return numbers.reduce((acc, num) => acc + num, 0);}**

1. **Type Annotations: TypeScript allows you to specify types for parameters to ensure that the function is called with the correct argument types.**

**function add(a: number, b: number): number { return a + b;}**

1. **Arrow Functions**: Parameters and arguments work similarly with arrow functions.

**const multiply = (a: number, b: number): number => a \* b;**

**const result = multiply(3, 4); // result is 12**

### Summary

* **Parameters** are variables defined in the function signature and used within the function.
* **Arguments** are the actual values passed to the function when it is invoked.
* TypeScript enhances functions with type annotations, default parameters, optional parameters, and rest parameters to create robust and type-safe functions.