Type inference in TypeScript is a feature where the compiler automatically deduces the types of variables, function return values, and other expressions based on the values assigned to them or the context in which they are used. This allows developers to omit explicit type annotations in many cases, making the code more concise while still benefiting from TypeScript's type checking.

**Variables**

When you initialize a variable without explicitly specifying its type, TypeScript infers the type from the value assigned to the variable.

let num = 42; // TypeScript infers the type as number

let message = "Hello, world!"; // TypeScript infers the type as string

let isCompleted = true; // TypeScript infers the type as Boolean

In the examples above, TypeScript infers the types number, string, and boolean for num, message, and isCompleted, respectively.

#### Functions

Type inference also applies to function return types and parameters. If a function's return type is not explicitly annotated, TypeScript infers it from the function's return statements.

function add(a: number, b: number) { return a + b; // TypeScript infers the return type as number } let result = add(5, 10); // TypeScript infers the type of result as number

#### Object Properties

TypeScript can infer the types of object properties based on their values.

let person = { name: "Alice", age: 30 }; // TypeScript infers the type of person as { name: string, age: number }

#### Contextual Typing

TypeScript uses contextual typing to infer types based on the surrounding context, such as function parameters and return types when passed as arguments or assigned to variables.

window.addEventListener("click", (event) => { console.log(event.clientX); // TypeScript infers the type of event as MouseEvent });

### Benefits of Type Inference

1. **Conciseness**: Reduces the need for explicit type annotations, making the code shorter and more readable.
2. **Type Safety**: Even without explicit annotations, TypeScript provides type checking and catches errors at compile time.
3. **Intelligent Suggestions**: IDEs can provide better code completion and suggestions based on inferred types.