

syntax for inheritance in typescript

In TypeScript, class inheritance is achieved using the `extends` keyword, allowing a derived class (child class) to inherit properties and methods from a base class (parent class). The `super` keyword is used within the derived class to access the constructor and methods of the base class.

Here is the general syntax for inheritance in TypeScript:

TypeScript

```
// Base Class (Parent/Superclass)
class BaseClass {
  // Properties and methods of the base class
  property1: string;

  constructor(value: string) {
    this.property1 = value;
  }

  method1(): void {
    console.log(`BaseClass method1: ${this.property1}`);
  }
}

// Derived Class (Child/Subclass)
class DerivedClass extends BaseClass {
  // Properties and methods specific to the derived class
  property2: number;

  constructor(baseValue: string, derivedValue: number) {
    // Call the constructor of the base class using super()
    super(baseValue);
    this.property2 = derivedValue;
  }

  method2(): void {
    console.log(`DerivedClass method2: ${this.property2}`);
    // Call a method of the base class using super.methodName()
    super.method1();
  }
}
```

```
// Example usage
const myDerivedObject = new DerivedClass("Hello from Base", 123);
myDerivedObject.method1(); // Inherited from BaseClass
myDerivedObject.method2(); // Specific to DerivedClass
```

Key elements of the syntax:

Used in the class declaration of the derived class to specify which base class it inherits from (e.g., `class DerivedClass extends BaseClass`).

Used within the constructor of the derived class to call the constructor of the base class. This is necessary to initialize any properties defined in the base class's constructor.

Used within the derived class to call a method of the base class. This allows the derived class to extend or override base class functionality while still being able to invoke the original implementation.

Revision #2

Created 29 October 2025 02:43:38 by AI API

Updated 5 December 2025 09:22:08 by AI Channel