@Output() toggleEvent = new EventEmitter<boolean>(); 설명해줘

Angular 컴포넌트에서 **자식 컴포넌트의 데이터를 부모 컴포넌트로 전달**하기 위해 사용하는 코드입니다.

이것은 Angular의 **커스텀 이벤트(Custom Event)** 메커니즘을 구현하는 핵심적인 부분입니다.

**1. @Output() 데코레이터의 역할**

@Output()은 해당 속성(toggleEvent)이 **부모 컴포넌트가 구독(Listen)할 수 있는 이벤트**임을 Angular에게 알려줍니다.

* **방향:** 데이터의 흐름은 **자식 컴포넌트 $\rightarrow$ 부모 컴포넌트**입니다.
* **용도:** 자식 컴포넌트에서 특정 이벤트(예: 버튼 클릭, 값 변경 등)가 발생했을 때, 그 사실과 데이터를 부모 컴포넌트에 알립니다.

**2. EventEmitter<boolean>()의 역할**

EventEmitter 클래스는 Angular의 이벤트 시스템을 구현하는 핵심 도구입니다.

* **역할:** 컴포넌트 내에서 이벤트를 발생시키는(Emit) 기능을 수행합니다.
* **제네릭 타입 (<boolean>):** 이 이벤트가 부모 컴포넌트에 전달할 데이터의 타입을 정의합니다. 여기서는 이벤트가 발생할 때 **boolean 타입의 값**을 함께 보낸다는 의미입니다. (예: true 또는 false)

**3. 사용 예시 (Toggle 버튼 구현)**

이 코드는 일반적으로 자식 컴포넌트에서 내부 상태(예: 토글 상태)가 변경되었음을 외부에 알릴 때 사용됩니다.

**A. 자식 컴포넌트 (ChildComponent) 내부**

자식 컴포넌트에서는 emit() 메서드를 사용하여 이벤트를 발생시키고 데이터를 보냅니다.

TypeScript

// child.component.ts

import { Component, Output, EventEmitter } from '@angular/core';

@Component({...})

export class ChildComponent {

// ⭐️ @Output 선언: 부모가 구독할 수 있는 이벤트를 정의합니다.

@Output() toggleEvent = new EventEmitter<boolean>();

private isToggled = false;

onToggleClick() {

this.isToggled = !this.isToggled;

// ⭐️ 이벤트 발생: 부모 컴포넌트로 현재 상태(boolean)를 전달합니다.

this.toggleEvent.emit(this.isToggled);

}

}

**B. 부모 컴포넌트 (ParentComponent) 내부**

부모 컴포넌트는 자식 컴포넌트의 셀렉터에서 () 괄호를 사용하여 이 이벤트를 구독합니다.

HTML

<app-child (toggleEvent)="handleToggle($event)"></app-child>

TypeScript

// parent.component.ts

import { Component } from '@angular/core';

@Component({...})

export class ParentComponent {

public toggleState: boolean | undefined;

handleToggle(isToggled: boolean) {

// 자식 컴포넌트에서 보낸 boolean 값을 받아 처리합니다.

this.toggleState = isToggled;

console.log('자식 컴포넌트에서 새로운 상태를 받았습니다:', isToggled);

}

}