

nqt - Data Binding trong Angular

 chatgpt.com/c/67d82626-e268-8001-bd69-57bd7474c129

ChatGPT

Bài 2: Data Binding trong Angular

1. Data Binding là gì?

Data Binding trong Angular là cơ chế giúp kết nối dữ liệu giữa **component** (logic) và **template** (giao diện). Nó giúp đồng bộ dữ liệu giữa UI và class TypeScript một cách tự động mà không cần thao tác thủ công.

2. Các loại Data Binding trong Angular

Angular cung cấp 4 loại data binding chính:

1. **Interpolation** (`{{ }}`) – Một chiều từ component đến template.
2. **Property Binding** (`[property]`) – Một chiều từ component đến template.
3. **Event Binding** (`(event)`) – Một chiều từ template đến component.
4. **Two-way Binding** (`[(ngModel)]`) – Hai chiều giữa component và template.

3. Chi tiết từng loại Data Binding

3.1. Interpolation (`{{ }}`)

Interpolation dùng cặp dấu `{{ }}` để hiển thị dữ liệu từ component vào HTML.

 Ví dụ

typescript

```
// app.component.ts
export class AppComponent {
  title = 'Học Angular';
}
```

html

```
<!-- app.component.html -->
<h1>{{ title }}</h1>
```

 **Kết quả hiển thị:**

 *Học Angular*

✓ **Lưu ý:** Chỉ dùng được với kiểu dữ liệu **string**, **number**, **boolean**, không dùng được với thuộc tính HTML.

3.2. Property Binding ([property])

Dùng dấu `[]` để bind dữ liệu từ component vào thuộc tính của thẻ HTML.

 **Ví dụ**

typescript

```
export class AppComponent {  
  isDisabled = true;  
}
```

html

```
<!-- app.component.html -->  
<button [disabled]="isDisabled">Click me</button>
```

📌 **Kết quả:** Nút bị vô hiệu hóa.

✓ **Lưu ý:** Chỉ dùng được với **thuộc tính (property)** của thẻ **HTML**, không dùng với **attribute** (ví dụ: **class**, **id**).

3.3. Event Binding ((event))

Dùng dấu **()** để bắt sự kiện từ UI và truyền vào component.

📌 **Ví dụ**

typescript

```
export class AppComponent {  
  count = 0;  
  
  increase() {  
    this.count++;  
  }  
}
```

html

```
<!-- app.component.html -->  
<button (click)="increase()">Tăng</button>  
<p>Giá trị: {{ count }}</p>
```

📌 **Kết quả:** Khi bấm nút "Tăng", giá trị `count` sẽ tăng lên.

✓ **Lưu ý:** Các sự kiện phổ biến gồm `click`, `keyup`, `change`, `mouseover`...

3.4. Two-way Binding [(ngModel)]

Dùng `[]` để kết nối hai chiều giữa component và UI.

📌 **Ví dụ**

typescript

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css'],
})
export class AppComponent {
  username = 'Angular User';
}
```

html

```
<!-- app.component.html -->
<input [(ngModel)]="username" placeholder="Nhập tên">
<p>Xin chào, {{ username }}!</p>
```

📌 **Kết quả:** Khi nhập vào input, giá trị trong `username` cũng thay đổi.

✓ **Lưu ý:**

Cần import `FormsModule` vào `app.module.ts` để sử dụng `ngModel`.

typescript

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule } from '@angular/forms';

@NgModule({
  declarations: [AppComponent],
  imports: [BrowserModule, FormsModule],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

4. Tổng kết

Loại Binding	Ký hiệu	Hướng dữ liệu	Ví dụ
Interpolation	{{ }}	Component → Template	<p>{{ title }}</p>
Property Binding	[property]	Component → Template	<button [disabled]="isDisabled">
Event Binding	(event)	Template → Component	<button (click)="increase()">
Two-way Binding	[(ngModel)]	Cả hai chiều	<input [(ngModel)]="username">

Bài tập thực hành

Hãy tạo một component **CounterComponent** hiển thị một số và có hai nút: **Tăng** và **Giảm** số đó.

 **Gợi ý:**

1. Dùng Property Binding để hiển thị số.
 2. Dùng Event Binding để bắt sự kiện khi bấm nút.
 3. Dùng Interpolation để hiển thị giá trị số.
-

 Sau khi hoàn thành bài tập, bạn hãy gửi lại code của mình để mình kiểm tra và góp ý nhé!