

BÀI 05:

DATA BINDING VÀ DIRECTIVES

TRONG ANGULAR

Giảng Viên: ThS. Giang Hào Côn

5.1/ Data Binding là gì ?

- **Data binding** là kỹ thuật đồng bộ dữ liệu giữa component và tầng view (template file html). Ví dụ khi người dùng cập nhật data ở tầng view thì Angular cũng cập nhật giá trị đó ở component.
- **Data binding** trong Angular có thể chia ra làm 2 nhóm. Đó là one way binding (binding 1 chiều) và two way binding (binding 2 chiều).
 - **One way binding** thì dữ liệu được truyền 1 chiều. Có thể từ view sang component hoặc ngược lại từ component sang view.
 - **Two way binding** có nghĩa là chúng ta thay đổi dữ liệu từ component qua view và ngược lại từ view chúng ta thay đổi dữ liệu.

5.2/ One way binding

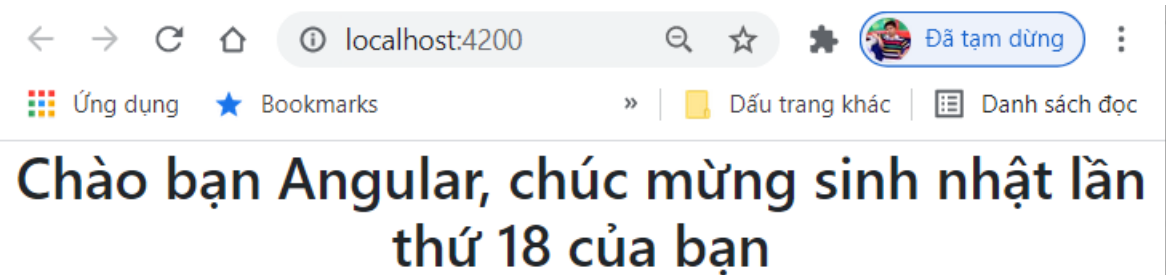
- Truyền dữ liệu Từ component sang view chúng ta sử dụng **Interpolation & Property Binding** để hiển thị dữ liệu như sau:
- Ví dụ ta có **app.component.ts** và **app.component.html** là:

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  hoten = 'Angular';
  tuoi = 18;
}
```

```
<h1 class="text-center">
  Chào bạn {{ hoten }}, chúc mừng sinh nhật lần thứ {{ tuoi }} của bạn
</h1>
```

Kết quả



5.2/ One way binding

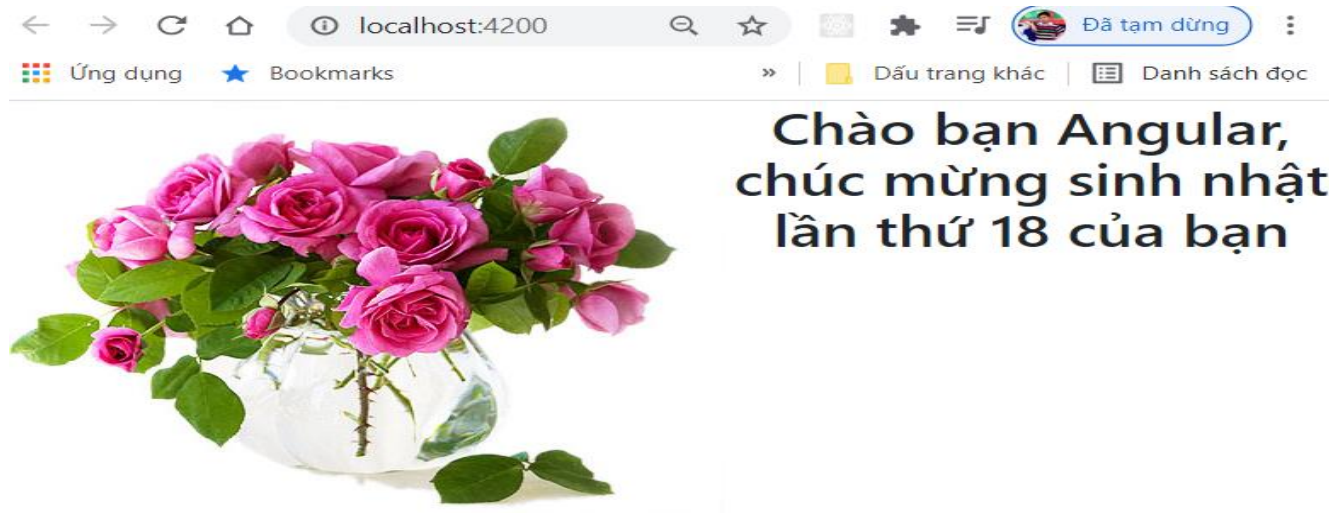
- Ví dụ ta có **app.component.ts** và **app.component.html** là:

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  hoten = 'Angular';
  tuoi = 18;
  imgurl = "assets/images/buoi01/binh-hoa.jpg";
}
```

```
<h1 class="text-center">
  
  Chào bạn {{ hoten }}, chúc mừng sinh nhật lần thứ {{ tuoi }} của bạn
</h1>
```

Kết quả



5.2/ One way binding

- Binding giá trị của biến vào thuộc tính
- Ví dụ ta có **app.component.ts** và **app.component.html** là:

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  hoten = 'Angular';
  tuoi = 18;
  imgurl = "assets/images/buoi01/binh-hoa.jpg";
}
```

```
<h1 class="text-center">
  
  <img [src] = "imgurl" class="float-end" alt="bình hoa"/>
  Chào bạn {{ hoten }}, chúc mừng sinh nhật lần thứ {{ tuoi }} của bạn
</h1>
```

Kết quả



5.2/ One way binding

■ Event Binding

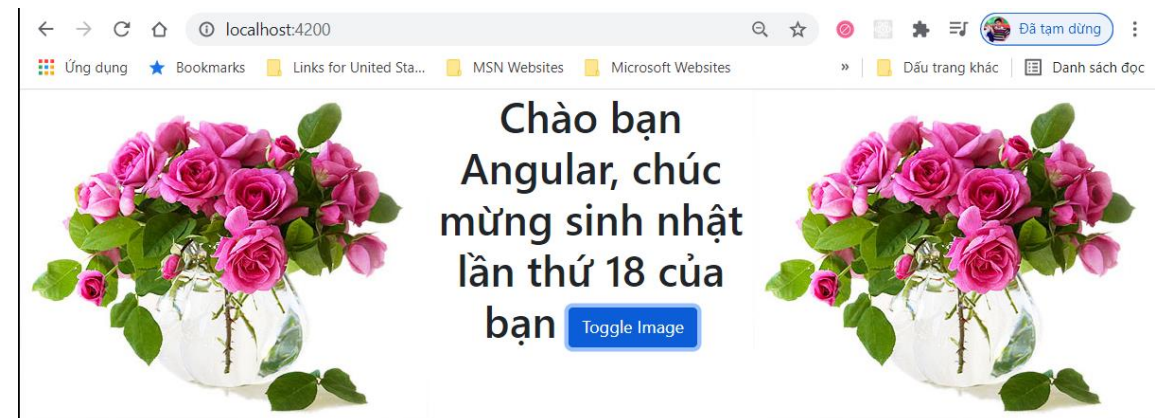
■ Ví dụ ta có **app.component.ts** và **app.component.html** là:

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

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})  
export class AppComponent {  
  hoten = 'Angular';  
  tuoi = 18;  
  imgurl = "assets/images/buoi01/binh-hoa.jpg";  
  trangthai = false;  
  
  toggleImage(){  
    this.trangthai = ! this.trangthai;  
  }  
}
```

```
<h1 class="text-center">  
    
  <img [src] = "imgurl" [hidden]="trangthai" class="float-end" alt="bình hoa"/>  
  Chào bạn {{ hoten }}, chúc mừng sinh nhật lần thứ {{ tuoi }} của bạn  
  <button class="btn btn-primary" (click)="toggleImage()">Toggle Image</button>  
</h1>
```

Kết quả



5.2/ One way binding

- Event Binding – xử lý text input
- Ví dụ ta có **app.component.ts** và **app.component.html** là:

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

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  name = '';
  age = 0;

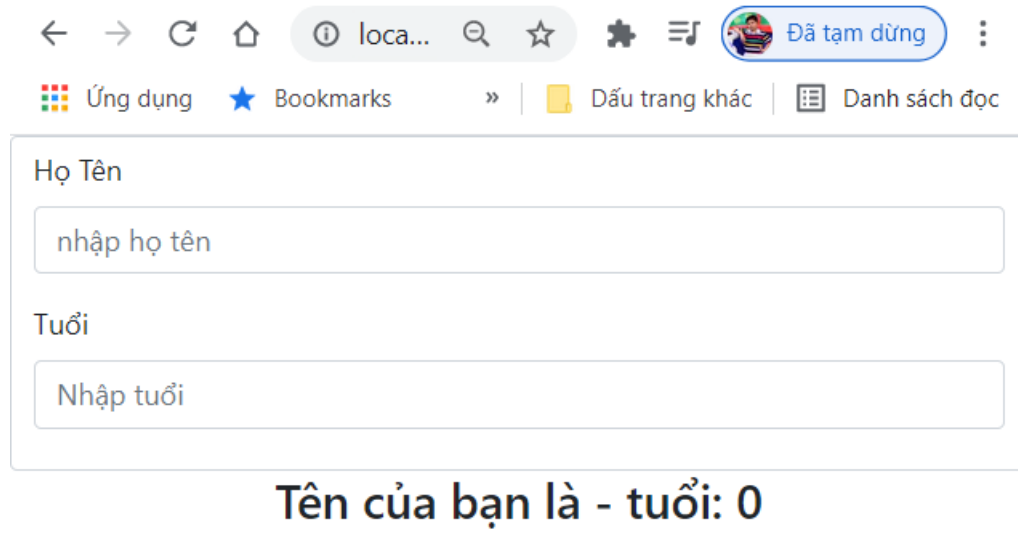
  handle_name(e:any){
    this.name = e.target.value;
  }
  handle_age(e:any){
    this.age = e.target.value;
  }
}
```

```
<form class="form-control">
  <div class="mb-3">
    <label class="form-label">Họ Tên</label>
    <input type="text" class="form-control" name="name" placeholder="nhập họ tên"
      (change)="handle_name($event)">
  </div>
  <div class="mb-3">
    <label class="form-label">Tuổi</label>
    <input type="text" class="form-control" name="age" placeholder="Nhập tuổi"
      (change)="handle_age($event)">
  </div>
</form>
<div class="h2 text-center">Tên của bạn là {{ name }} - tuổi: {{ age }} </div>
```

5.2/ One way binding

- Event Binding – xử lý text input
- Ví dụ ta có **app.component.ts** và **app.component.html** là:

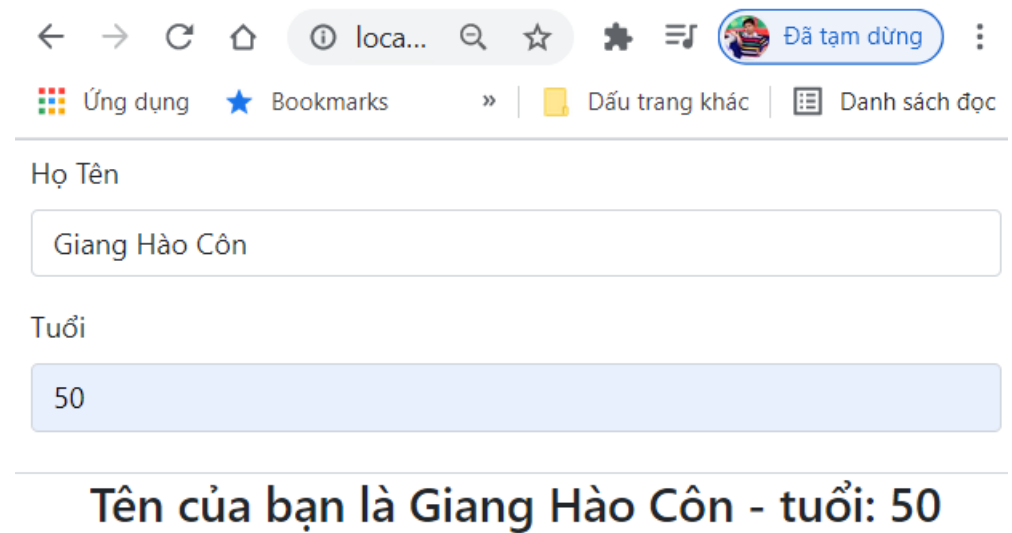
Kết quả



Họ Tên

Tuổi

Tên của bạn là - tuổi: 0



Họ Tên

Tuổi


Tên của bạn là Giang Hào Côn - tuổi: 50

5.3/ Two Way Binding

- Binding 2 chiều có nghĩa là chúng ta thay đổi dữ liệu từ component qua view và ngược lại từ view chúng ta thay đổi dữ liệu. Two way binding thì hữu dụng khi ta làm form. Chúng ta sử dụng **ngModel** để thực hiện việc binding 2 chiều.
- Ta phải import thư viện form trong **app.module.ts** vào bằng lệnh:

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

],
imports: [
  BrowserModule,
  FormsModule
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }
```



A yellow box highlights the `import { FormsModule } from '@angular/forms';` line in the first code block. A yellow arrow points from this box to a corresponding yellow box around `FormsModule` in the `imports` array of the second code block.

5.3/ Two Way Binding

- Ví dụ ta có **app.component.ts** và **app.component.html** là:

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  name = '';
  age = 0;
}
```

```
<form class="form-control">
  <div class="mb-3">
    <label class="form-label">Họ Tên</label>
    <input type="text" class="form-control" name="name" placeholder="nhập họ tên"
      [(ngModel)]="name"/>
  </div>
  <div class="mb-3">
    <label class="form-label">Năm sinh</label>
    <input type="text" class="form-control" name="age" placeholder="Nhập tuổi"
      [(ngModel)]="age">
  </div>
</form>
<div class="h2 text-center">Tên của bạn là {{ name }} - tuổi: {{ age }} </div>
```

5.4/ Style Binding

- Ví dụ ta có **app.component.ts** là:

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  name="";
  age = 0;
  mau = '';
  changecolor(){
    this.mau = this.age<=10? 'yellow' : this.age<= 25? 'red' : 'black';
  }
}
```

5.4/ Style Binding

- Ví dụ ta có **app.component.html** là:

```
<form class="form-control">
  <div class="mb-3">
    <label class="form-label">Họ Tên</label>
    <input type="text" class="form-control" name="name" placeholder="nhập họ tên"
      [(ngModel)]="name"/>
  </div>
  <div class="mb-3">
    <label class="form-label">Năm sinh</label>
    <input type="text" class="form-control" name="age" placeholder="Nhập tuổi"
      [(ngModel)]="age" (change)="changeColor()">
  </div>
</form>

<div class="h2 text-center" [style.color]="age>=18 ? 'red' : 'blue'">
  Tên của bạn là {{ name }} - tuổi: {{ age }}
</div>

<div class="h2" [style.color]="mau">Style Binding</div>
```

5.5/ Class Binding

- Ví dụ ta có **app.component.css** và **app.component.html** là:

```
.tron {  
  width: 50px;  
  height: 50px;  
  border-radius: 25px;  
  background-color: ■ blueviolet;  
}
```

```
.vuong {  
  width: 50px;  
  height: 50px;  
  border-radius: 2px;  
  background-color: ■ blue;  
}
```

```
<div [class]='''tron'''></div>  
<div [class]='''vuong'''></div>
```

Hiển thị

Họ Tên

Năm sinh

Tên của bạn là - tuổi: 0

Style Binding



Câu hỏi & Bài tập áp dụng

- 1) Data Binding trong Angular là gì?
- 2) Có mấy kiểu Data Binding và bày và cho ví dụ cụ thể ?
- 3) Style Binding là gì ? cho ví dụ ?
- 4) Class Binding là gì ? cho ví dụ ?
- 5) Thực Hành áp dụng Data Binding

DIRECTIVES TRONG ANGULAR

Giảng Viên: ThS. Giang Hào Côn

5.6./ Directives là gì ?

- **Directives** là một đối tượng giúp chúng ta dễ dàng thay đổi một đối tượng khác và cách áp dụng rất đơn giản và linh hoạt.
- Chúng ta sử dụng Directive để thao tác, cập nhật, thêm, xóa các thành phần giao diện trên web. (*Hay nói cách khác chúng ta thao tác với các element của DOM. Các nút, ảnh, text box, table mà ta thấy trên website được một đối tượng DOM quản lý. DOM có nhiệm vụ vẽ ra các đối tượng này trên website để ta có thể thấy được*).
- Angular Directive được chia là 3 loại gồm: **Component Directive**, **Structural Directive**, **Attribute Directive**.

5.7/ Attribute Directive

- Đây là các chỉ thị có tác thay đổi cách thức hiển thị của các element, thuộc tính... trong HTML. Thông thường chúng cũng được dùng giống như một thuộc tính của một element.
- Có 3 loại chỉ thị thường dùng là:
 - **NgModel**: kết nối dữ liệu 2 chiều.
 - **NgClass**: thêm/bớt các lớp CSS
 - **NgStyle**: thêm/bớt các style

5.7.1/ Sử dụng ngStyle

Đặt vấn đề

html

```
<div [style.color]='red' [style.fontSize]='40px' [style.backgroundColor]='blue'>  
  Ví dụ ngStyle  
</div>
```

Kết quả hiển thị



5.7.1/ Sử dụng ngStyle

Giải quyết vấn đề

ts

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

@Component({
  selector: 'app-vidung-style',
  templateUrl: './vidung-style.component.html',
  styleUrls: ['./vidung-style.component.css']
})
export class VidungStyleComponent implements OnInit {
  currentStyle = { color: 'red', fontSize: '40px', backgroundColor: 'blue' };
  constructor() {}

  ngOnInit(): void {}
}
```

html

```
<div [ngStyle]="currentStyle">
  Ví dụ ngStyle
</div>
```

Kết quả hiển thị



5.7.1/ Sử dụng ngStyle

Ví dụ: code file ts

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

@Component({
  selector: 'app-vidung-style',
  templateUrl: './vidung-style.component.html',
  styleUrls: ['./vidung-style.component.css']
})
export class VidungStyleComponent implements OnInit {
  isshow = true;
  currentStyle = { color: 'red', fontSize: '40px', backgroundColor: 'blue' }
  style01 = { backgroundImage: 'url(/assets/hinh/banh1.png)', height: '250px', width: '289px' }
  style02 = { backgroundImage: 'url(/assets/hinh/banh2.jpg)', height: '250px', width: '289px' }
  constructor() { }

  ngOnInit(): void {
  }
  thayhinh(){
    this.isshow = ! this.isshow;
  }
}
```

5.7.1/ Sử dụng ngStyle

Ví dụ: code file html

```
<div [ngStyle]="isshow ? style01 : style02"></div>  
<button type="button" (click)="thayhinh()">Thực hiện</button>
```

Hiển thị



Thực hiện



Thực hiện

5.7.2/ sử dụng ngIf

- **NgIf** rất hữu ích nếu ta muốn hiển thị hoặc ẩn các phần của ứng dụng dựa trên một điều kiện.

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

```
@Component({  
  selector: 'app-vidung-if',  
  templateUrl: './vidung-if.component.html',  
  styleUrls: ['./vidung-if.component.css']  
})
```

```
export class VidungIfComponent implements OnInit {
```

```
  show:boolean = true;
```

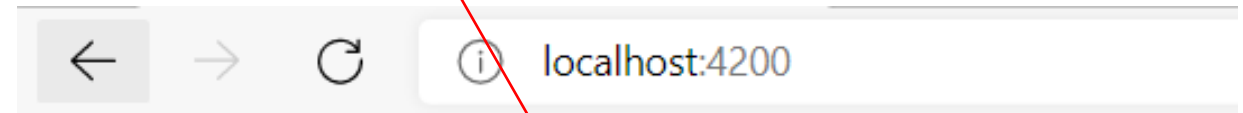
```
  constructor() { }
```

```
  ngOnInit(): void {  
  }
```

```
}
```

```
<p>Show this all the time</p>
```

```
<p *ngIf="show">Show this only if "show" is true</p>
```



Show this all the time

Show this only if "show" is true

5.7.2/ sử dụng ngIf

- **NgIf** rất hữu ích nếu ta muốn hiển thị hoặc ẩn các phần của ứng dụng dựa trên một điều kiện.

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

```
@Component({  
  selector: 'app-vidung-if',  
  templateUrl: './vidung-if.component.html',  
  styleUrls: ['./vidung-if.component.css']  
})
```

```
export class VidungIfComponent implements OnInit {
```

```
  show:boolean = false;
```

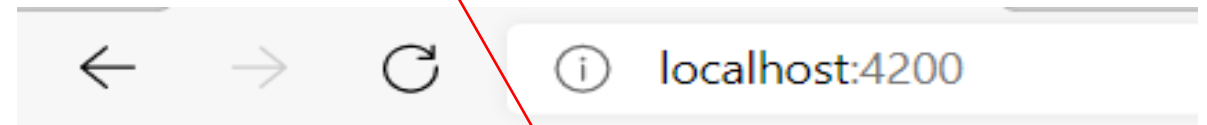
```
  constructor() { }
```

```
  ngOnInit(): void {  
  }
```

```
}
```

```
<p>Show this all the time</p>
```

```
<p *ngIf="show">Show this only if "show" is true</p>
```



Show this all the time

5.7.2/ sử dụng ngIf

- **NgIf** xử lý điều kiện này cũng áp dụng cho tất cả các phần tử con của phần tử có chỉ thị ngIf:.

```
<p>Show this all the time</p>
```

```
<div *ngIf="show">
```

```
  <p>Show this only if "show" is true</p>
```

```
  <p>Show this only if "show" is true</p>
```

```
</div>
```

5.7.2/ sử dụng ngIf

- **NgIf** cũng hỗ trợ các toán tử logic, giống như một câu lệnh if thông thường trong JavaScript.

Logical NOT (!)

```
<p *ngIf="!show">
```

```
  Show this only if "show" is NOT true
```

```
</p>
```

5.7.2/ sử dụng ngIf

- **NgIf** cũng hỗ trợ các toán tử logic, giống như một câu lệnh if thông thường trong JavaScript.

Logical AND (&&)

```
<p *ngIf="a && b">
```

```
  Show this only if a AND b are true
```

```
</p>
```

5.7.2/ sử dụng ngIf

- **NgIf** cũng hỗ trợ các toán tử logic, giống như một câu lệnh if thông thường trong JavaScript.

Logical OR (||)

```
<p *ngIf="a || b">
```

```
    Show this only if a AND b are true
```

```
</p>
```

5.7.2/ sử dụng ngIf

- **NgIf** cũng hỗ trợ các toán tử logic, giống như một câu lệnh if thông thường trong JavaScript. **Working with "else"**

```
<div *ngIf="show; else notShow">
  <p>
    Show this only if "show" is true
  </p>
</div>

<ng-template #notShow>
  <p>
    Show this only if "show" is not true
  </p>
</ng-template>
```


5.7.2/ sử dụng ngIf

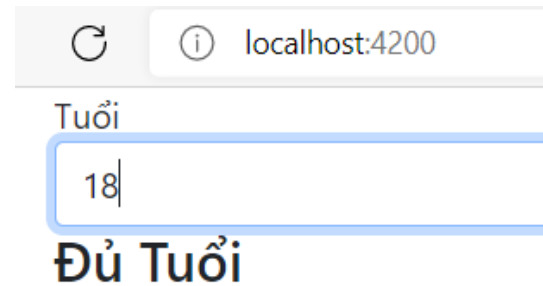
- **NgIf** cũng hỗ trợ các toán tử logic, giống như một câu lệnh if thông thường trong JavaScript.

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

@Component({
  selector: 'app-vidung-if',
  templateUrl: './vidung-if.component.html',
  styleUrls: ['./vidung-if.component.css']
})
export class VidungIfComponent implements OnInit {
  age:number = 0;

  constructor() { }
  ngOnInit(): void {
  }
}
```

```
<div class="form-group">
  <label>Tuổi</label>
  <input type="text" class="form-control" [(ngModel)]="age"/>
</div>
<h3 *ngIf="age>=18; else elseBlock"> Đủ Tuổi</h3>
<ng-template #elseBlock>
  Chưa đủ tuổi
</ng-template>
```

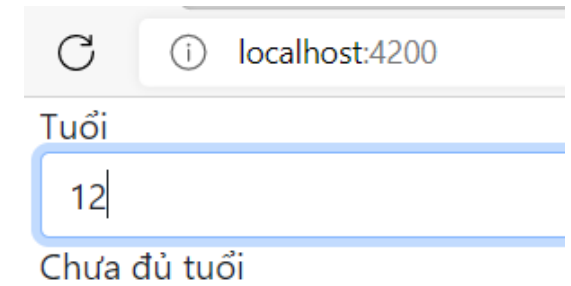


localhost:4200

Tuổi

18

Đủ Tuổi



localhost:4200

Tuổi

12

Chưa đủ tuổi

5.7.2/ sử dụng ngIf

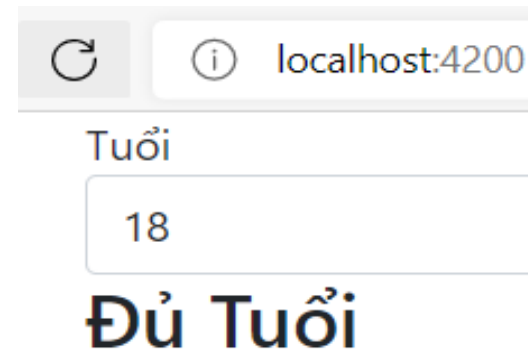
- **NgIf** cũng hỗ trợ các toán tử logic, giống như một câu lệnh if thông thường trong JavaScript. **NgIf also has a "then"**

```
<ng-template
  *ngIf="show;then showBlock; else notShowBlock">
</ng-template>
```

```
<ng-template #showBlock>
  <p>
    Show this only if "show" is true
  </p>
</ng-template>
```

```
<ng-template #notShowBlock>
  <p>
    Show this only if "show" is not true
  </p>
</ng-template>
```

```
<ng-template *ngIf="age >= 18;then showblock; else notshow"></ng-template>
<ng-template #showblock>
  <h2>Đủ Tuổi</h2>
</ng-template>
<ng-template #notshow>
  <h2>Chưa đủ tuổi</h2>
</ng-template>
```

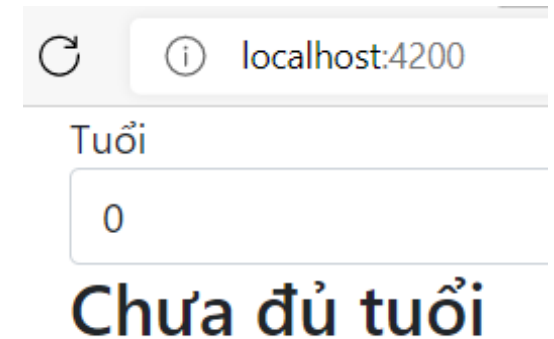


localhost:4200

Tuổi

18

Đủ Tuổi



localhost:4200

Tuổi

0

Chưa đủ tuổi

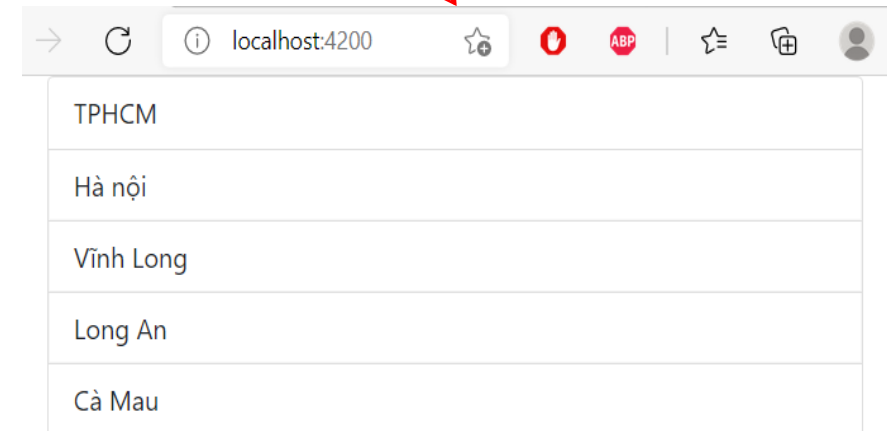
5.7.2/ sử dụng ngFor

■ NgFor hiển thị nhiều phần tử trong html. Duyệt mảng

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

```
@Component({  
  selector: 'app-vidungfor',  
  templateUrl: './vidungfor.component.html',  
  styleUrls: ['./vidungfor.component.css']  
})  
export class VidungforComponent implements OnInit {  
  diadanh: string[] = ['TPHCM', 'Hà nội', 'Vĩnh Long', 'Long An', 'Cà Mau'];  
  constructor() { }  
  
  ngOnInit(): void {  
  }  
}
```

```
<ul class="list-group">  
  <li class="list-group-item" *ngFor="let item of diadanh">  
    {{ item }}  
  </li>  
</ul>
```



5.7.2/ sử dụng ngFor

- **NgFor** hiển thị nhiều phần tử trong html. **Duyệt mảng đối tượng**

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

@Component({
  selector: 'app-vidungfor',
  templateUrl: './vidungfor.component.html',
  styleUrls: ['./vidungfor.component.css']
})
export class VidungforComponent implements OnInit {
  diadanh: string[] = ['TPHCM', 'Hà nội', 'Vĩnh Long', 'Long An', 'Cà Mau'];
  sanpham: any[] = [
    {id:1, Tensp:'Bánh 01', hinhsp:'banh1.png'},
    {id:2, Tensp:'Bánh 02', hinhsp:'banh2.jpg'},
    {id:3, Tensp:'Bánh 03', hinhsp:'banh3.jpg'},
    {id:4, Tensp:'Bánh 04', hinhsp:'banh4.jpg'}];
  constructor() { }
  ngOnInit(): void {
  }
}
```

5.7.2/ sử dụng ngFor

- **NgFor** hiển thị nhiều phần tử trong html. **Duyệt mảng đối tượng**

```
<table class="table table-bordered table-hover">
```

```
  <thead>
```

```
    <tr>
```

```
      <td>Mã Sản Phẩm</td>
```

```
      <td>Tên Sản Phẩm</td>
```

```
      <td>Hình sản phẩm</td>
```

```
    </tr>
```

```
  </thead>
```

```
  <tbody>
```

```
    <tr *ngFor="let sp of sanpham">
```

```
      <td>{{ sp.id }}</td>
```

```
      <td>{{ sp.tensp }}</td>
```

```
      <td>{{ sp.hinhsp }}</td>
```

```
    </tr>
```

```
  </tbody>
```

```
</table>
```

Hiển thị

| Mã Sản Phẩm | Tên Sản Phẩm | Hình sản phẩm |
|-------------|--------------|---------------|
| 1 | Bánh 01 | banh1.png |
| 2 | Bánh 02 | banh2.jpg |
| 3 | Bánh 03 | banh3.jpg |
| 4 | Bánh 04 | banh4.jpg |

5.7.2/ sử dụng ngFor

- **NgFor** hiển thị nhiều phần tử trong html. **Duyệt mảng đối tượng**

```
<table class="table table-bordered table-hover">
```

```
<thead>
```

```
<tr>
```

```
<td>Mã Sản Phẩm</td>
```

```
<td>Tên Sản Phẩm</td>
```

```
<td>Hình sản phẩm</td>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr *ngFor="let sp of sanpham">
```

```
<td>{{ sp.id }}</td>
```

```
<td>{{ sp.tensp }}</td>
```





```
<td align="center"><img [src]="'/assets/hinh/'+sp.hinhsp" width="70px" height="70px"/></td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

Hiển thị

| Mã Sản Phẩm | Tên Sản Phẩm | Hình sản phẩm |
|-------------|--------------|--|
| 1 | Bánh 01 |  |
| 2 | Bánh 02 |  |
| 3 | Bánh 03 |  |
| 4 | Bánh 04 |  |

5.7.2/ sử dụng ngFor

- **NgFor** hiển thị nhiều phần tử trong html. Sử dụng biến local first.

Ví dụ ta muốn dòng đầu tô màu xanh dương

```
<ul class="list-group">
  <li class="list-group-item" *ngFor="let item of diadanh; let f=first"
    [style.background-color]="f === true ? 'blue' : ''">
    {{ item }}
  </li>
</ul>
```

Kết quả hiển thị

| TPHCM |
|-----------|
| Hà nội |
| Vĩnh Long |
| Long An |
| Cà Mau |

5.7.2/ sử dụng ngFor

- **NgFor** hiển thị nhiều phần tử trong html. Sử dụng biến local last. Ví dụ ta muốn dòng cuối tô màu vàng

```
<ul class="list-group">
  <li class="list-group-item" *ngFor="let item of diadanh; let l=last"
    [style.background-color]="l === true ? 'yellow' : ''">
    {{ item }}
  </li>
</ul>
```

Kết quả hiển thị

| |
|-----------|
| TPHCM |
| Hà nội |
| Vĩnh Long |
| Long An |
| Cà Mau |

5.7.2/ sử dụng ngFor

- **NgFor** hiển thị nhiều phần tử trong html. Sử dụng biến local even và odd. Ví dụ ta muốn dòng lẻ tô màu đỏ và chẵn tô màu xanh dương

```
<ul class="list-group">
  <li class="list-group-item" *ngFor="let item of diadanh; let e=even"
    [style.background-color]="e === true ? 'red' : 'blue'">
    {{ item }}
  </li>
</ul>
```

Kết quả hiển thị

| |
|-----------|
| TPHCM |
| Hà nội |
| Vĩnh Long |
| Long An |
| Cà Mau |

5.7.3/ kết hợp ngFor và ngIf

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

@Component({
  selector: 'app-vidungfor',
  templateUrl: './vidungfor.component.html',
  styleUrls: ['./vidungfor.component.css']
})
export class VidungforComponent implements OnInit {

  khachhang:any[] = [ {name: 'Nguyễn Thị Bạch Liễu', country: 'VN'},
                      {name: 'Heneiken Long', country: 'USA'},
                      {name: 'Jacky chan', country: 'HK'}];

  constructor() { }
  ngOnInit(): void {
  }
}
```

5.7.3/ kết hợp ngFor và ngIf

```
<ul class="list-group">
  <li class="list-group-item" *ngFor="let kh of khachhang">
    <span class="text-danger" *ngIf="kh.country === 'VN'">
      {{ kh.name }} - {{ kh.country }}
    </span>
    <span class="text-primary" *ngIf="kh.country === 'USA'">
      {{ kh.name }} - {{ kh.country }}
    </span>
    <span *ngIf="kh.country !== 'USA' && kh.country !== 'VN'">
      {{ kh.name }} - {{ kh.country }}
    </span>
  </li>
</ul>
```

Kết quả hiển thị

Nguyễn Thị Bạch Liễu - VN

Heneiken Long - USA

Jacky chan - HK

5.7.3/ kết hợp ngFor và ngSwitch

```
<ul class="list-group">
  <li class="list-group-item" *ngFor="let kh of khachhang" [ngSwitch]="kh.country">
    <span class="text-danger" *ngSwitchCase="'VN'">
      {{ kh.name }} - {{ kh.country }}
    </span>
    <span class="text-primary" *ngSwitchCase="'USA'">
      {{ kh.name }} - {{ kh.country }}
    </span>
    <span *ngSwitchDefault>
      {{ kh.name }} - {{ kh.country }}
    </span>
  </li>
</ul>
```

Kết quả hiển thị

Nguyễn Thị Bạch Liễu - VN

Heneiken Long - USA

Jacky chan - HK

Bài tập

1. Directive trong Angular là gì ?
2. Directive Angular được chia làm mấy loại ? Hãy trình bày các loại đó và cho ví dụ cách sử dụng của các loại đó ?
3. Thực hành Áp dụng ngIf directive và ngFor directive