



R a i s i n g t h e b a r

Angular Overview

Mục tiêu

- Tìm hiểu được kiến trúc của Angular.
- Tạo được ứng dụng Angular sử dụng Angular CLI.
- Tạo được component cơ bản trong Angular.
- Tạo được template cơ bản trong Angular.
- Sử dụng được component cơ bản trong Angular.
- Sử dụng được Data Binding cơ bản trong Angular

Điều kiện tiên quyết

- Nắm được lập trình ứng dụng với JavaScript và TypeScript
- Có kiến thức cơ bản về HTML, CSS

Khởi tạo project

Thực hiện

- Bước 1: Tạo một folder chứa project.
- Bước 2: Di chuyển terminal/CMD vào folder đó.
- Bước 3: Tạo mới project với Angular CLI bằng cách chạy lệnh sau

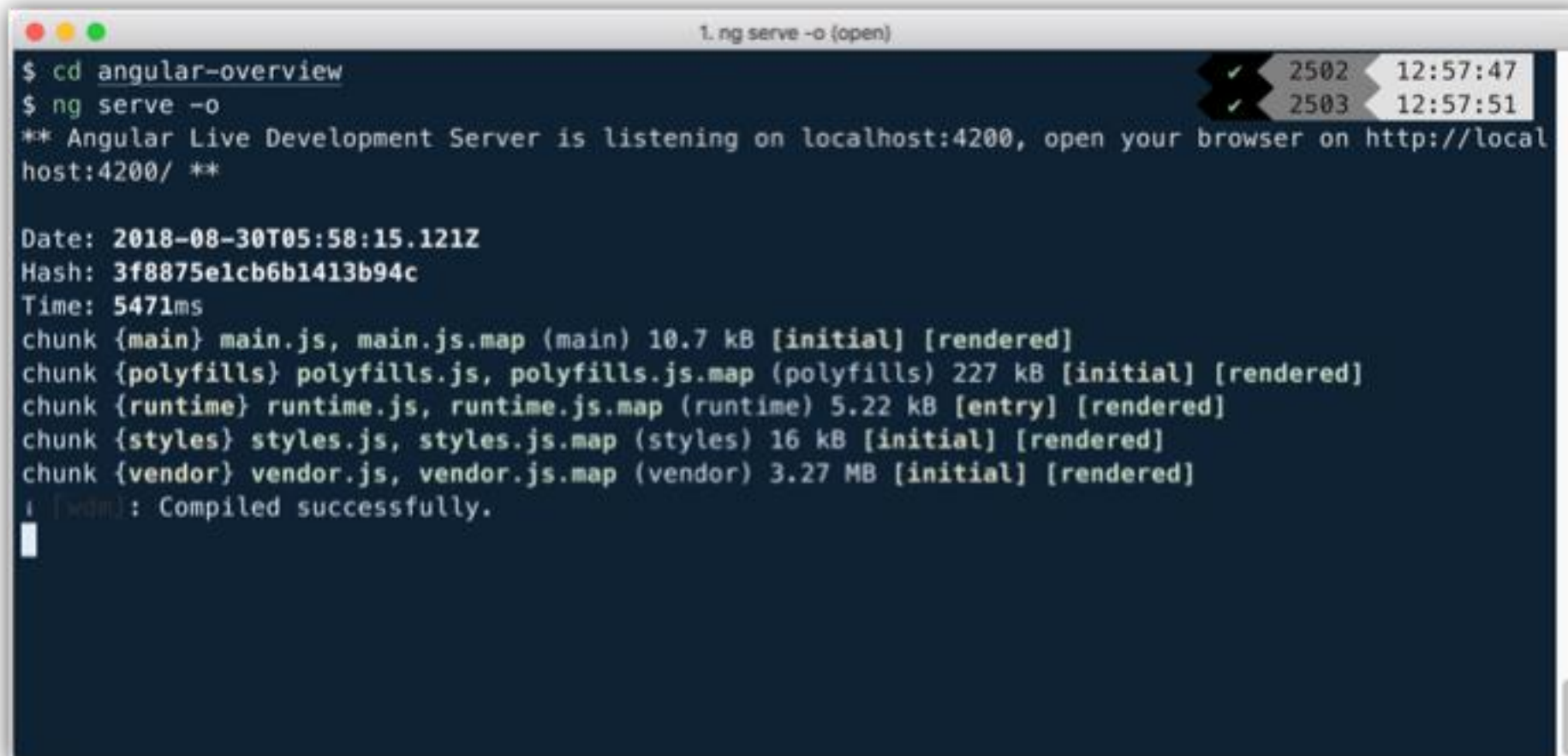
ng new angular-overview --style=scss

Thực hiện

- Bước 4: Sau khi khởi tạo xong project, di chuyển terminal/CMD vào folder **angular-overview** vừa tạo bởi Angular CLI.
- Bước 5: Chạy lệnh sau để khởi chạy project.

ng serve -o

Thực hiện



```
1. ng serve -o (open)
$ cd angular-overview
$ ng serve -o
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

Date: 2018-08-30T05:58:15.121Z
Hash: 3f8875e1cb6b1413b94c
Time: 5471ms
chunk {main} main.js, main.js.map (main) 10.7 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 227 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 5.22 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 16 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 3.27 MB [initial] [rendered]
i [wdm]: Compiled successfully.
```

Thực hiện

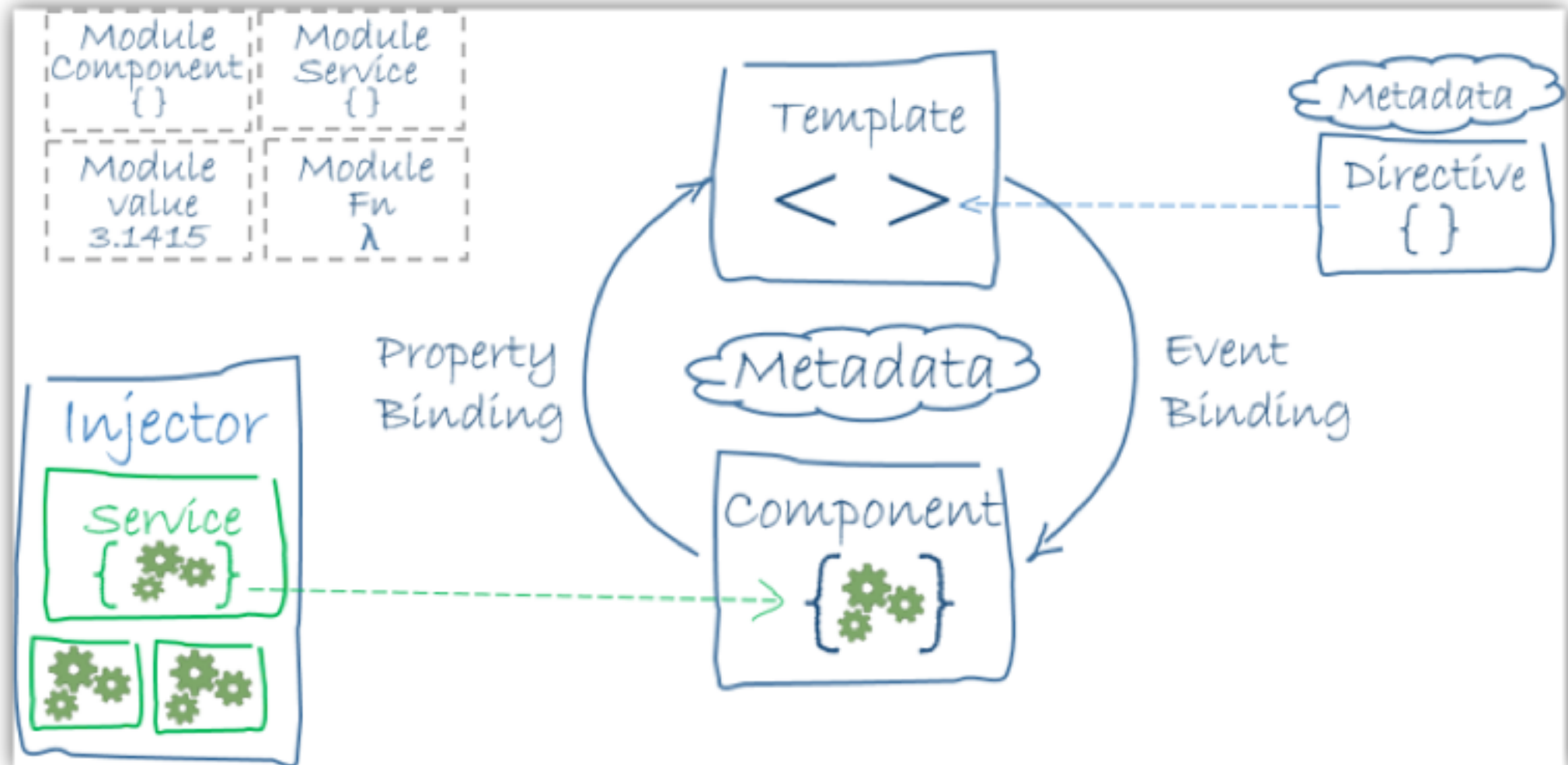
Welcome to angular-overview!



Here are some links to help you start:

- [Tour of Heroes](#)
- [CLI Documentation](#)
- [Angular blog](#)

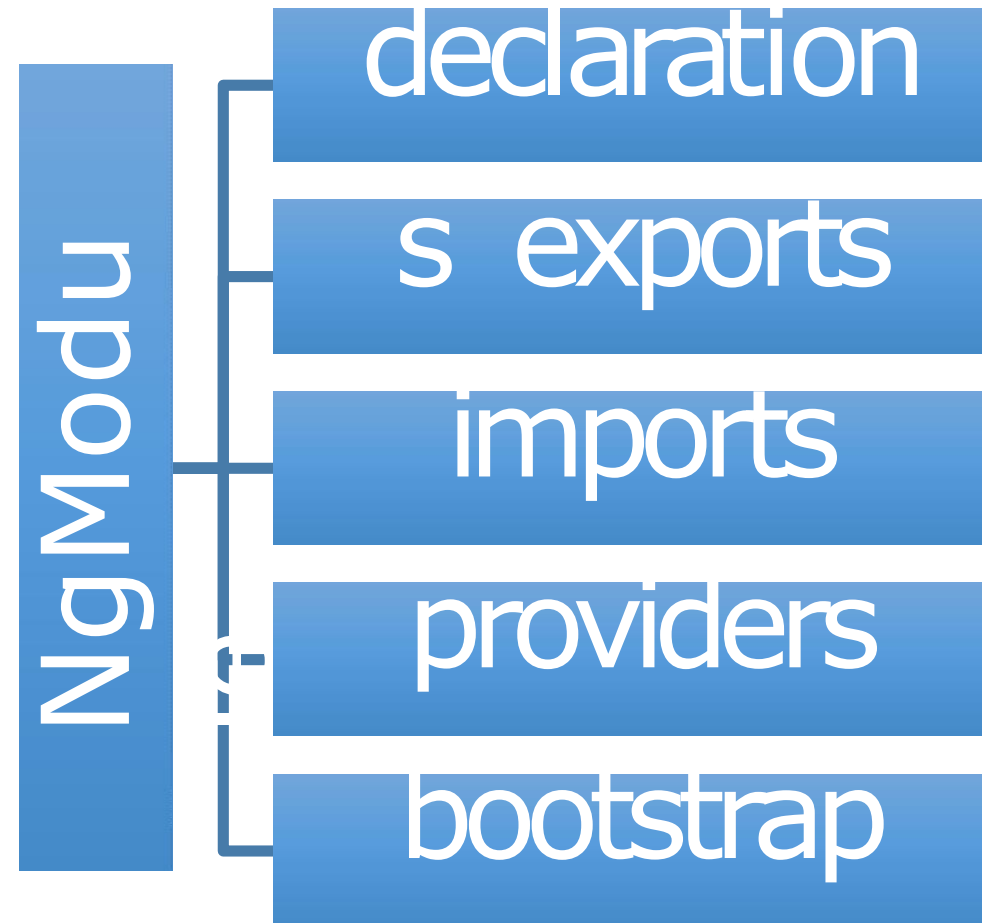
Angular Architecture



NgModule

- NgModules help organize an application into cohesive blocks of functionality.
- Every Angular app has at least one NgModule class, the *root module*, conventionally named AppModule.
- NgModule is a decorator function that takes a single metadata object whose properties describe the module.

NgModule



NgModule

app.module.ts

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

```
import { AppComponent } from './app.component';
```

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

NgModule

app.module.ts

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

```
import { AppComponent } from './app.component';
```

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

NgModule

app.module.ts

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

import { AppComponent } from './app.component';

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

NgModule

app.module.ts

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

```
import { AppComponent } from './app.component';
```

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


Bootstrap Angular Application

main.ts

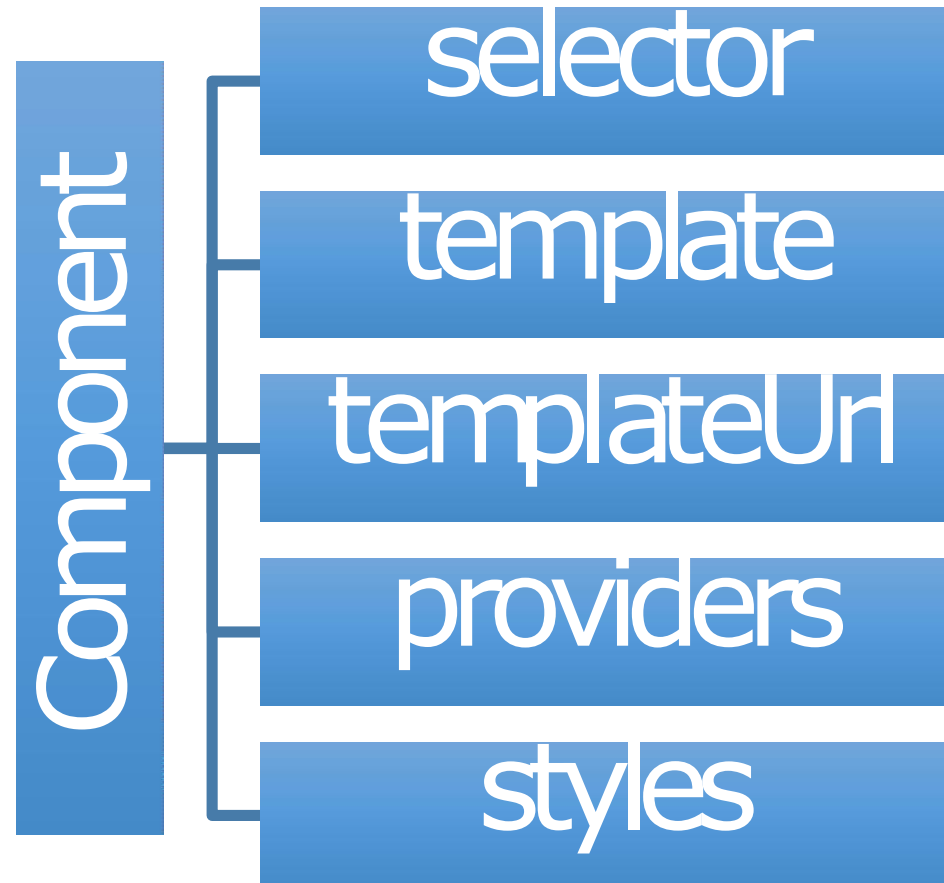
```
import { enableProdMode } from '@angular/core';
import {
  platformBrowserDynamic
} from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

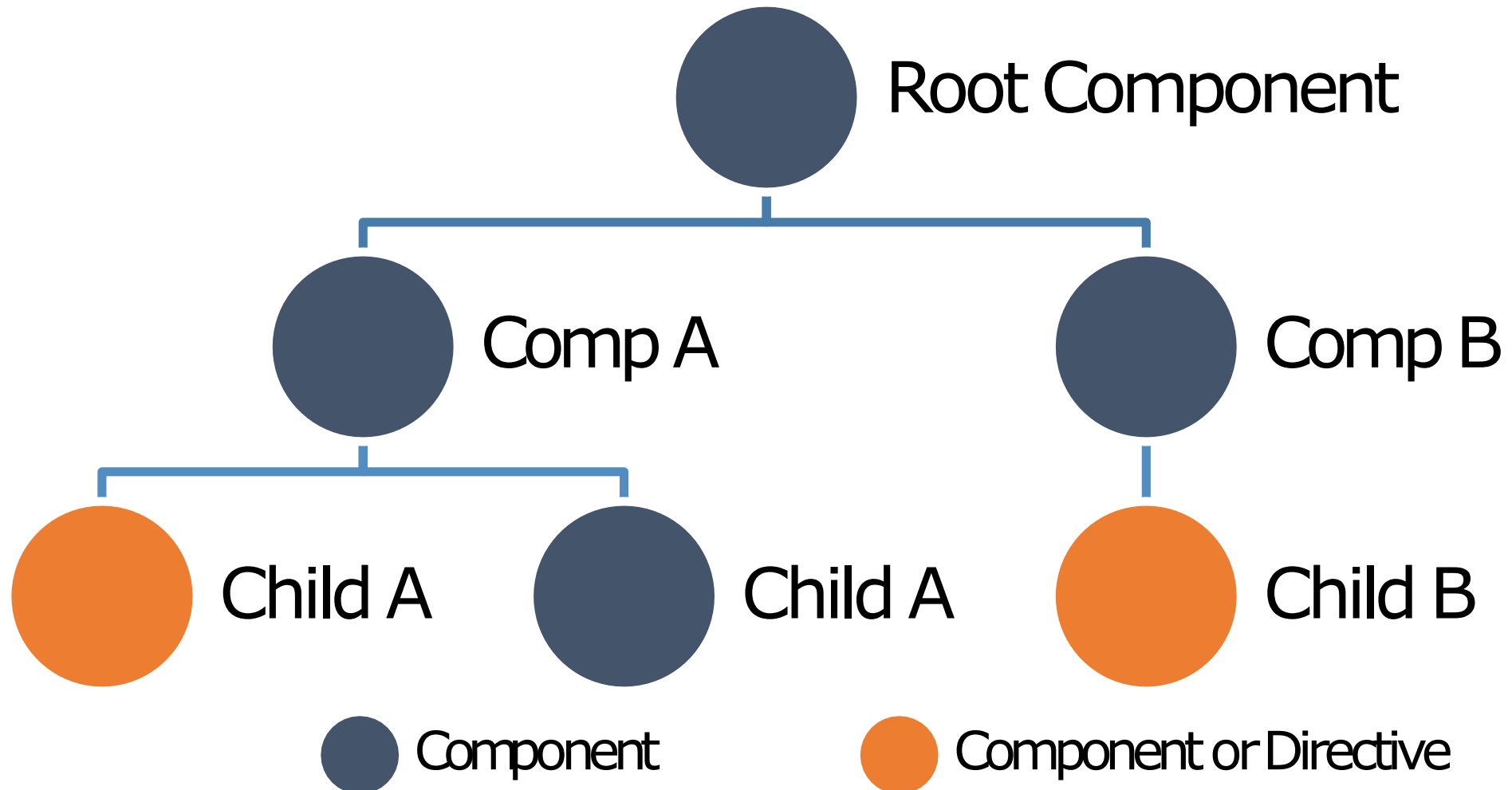
if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.log(err));
```

Component



Component



Angular Component

app.component.ts

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

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.scss']  
})  
export class AppComponent {  
  title = 'Awesome App';  
}
```

app.component.html

```
<h1>  
  Welcome to {{title}}!  
</h1>
```

Angular Component

app.component.ts

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

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.scss']  
})  
export class AppComponent {  
  title = 'Awesome App';  
}
```

app.component.html

```
<h1>  
  Welcome to {{title}}!  
</h1>
```

Angular Component

app.component.ts

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

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.scss']  
})  
export class AppComponent {  
  title = 'Awesome App';  
}
```

app.component.html

```
<h1>  
  Welcome to {{title}}!  
</h1>
```

Angular Component

app.component.ts

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

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.scss']  
})  
export class AppComponent {  
  title = 'Awesome App';  
}
```

app.component.html

```
<h1>  
  Welcome to {{title}}!  
</h1>
```

Tạo Angular Component

Manually

- Bước 1: Tạo folder **profile** để chứa các phần tử của một component trong thư mục **src/app**.
- Bước 2: Trong folder **profile**, tạo mới file **profile.component.ts**
- Bước 3: Thực hiện thêm các đoạn code sau vào file vừa tạo.
- Bước 4: Thêm mới Component vừa tạo vào phần **declarations** của **AppModule**.
- Bước 5: Sửa đổi template của AppComponent để hiển thị component vừa tạo.

Manually – profile.component.ts

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

Import decorator
Component từ thư viện
@angular/core

```
@Component({  
  selector: 'app-profile',  
  template: `  
    <h2>Profile Component</h2>  
  `,  
})
```

Metadata của
component

```
export class ProfileComponent {  
}
```

Khởi tạo class trong
TypeScript

Manually – app.module.ts

```
import { ProfileComponent } from './profile/profile.component';
```

```
@NgModule({
```

```
  declarations: [  
    AppComponent,  
    ProfileComponent  
  ],
```

```
  //... other config
```

```
})
```

```
export class AppModule { }
```

Thêm mới component class vào mảng declarations, để Angular có thể quản lý chúng

Manually – app.component.html

```
<div style="text-align:center">  
  <h1>  
    Welcome to {{ title }}!  
  </h1>  
    
</div>
```

```
<app-profile></app-profile>
```



Thêm **selector** của component vừa tạo

Manually – kết quả đạt được



Tạo Angular Component Template

Manually

- Bước 1: Trong folder profile, thêm mới file profile.component.html
- Bước 2: Thêm mới code template cho file vừa tạo.
- Bước 3: Sửa đổi file profile.component.ts để trỏ tới file template vừa tạo.

Manually – profile.component.html

<h2>Profile Component Template</h2>

Manually – profile.component.ts

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

@Component({
  selector: 'app-profile',
  templateUrl: './profile.component.html'
})
export class ProfileComponent {
}
```

Manually – kết quả đạt được

Welcome to angular-overview!



Profile Component Template

Tạo Angular Component với Angular CLI

Thực hiện

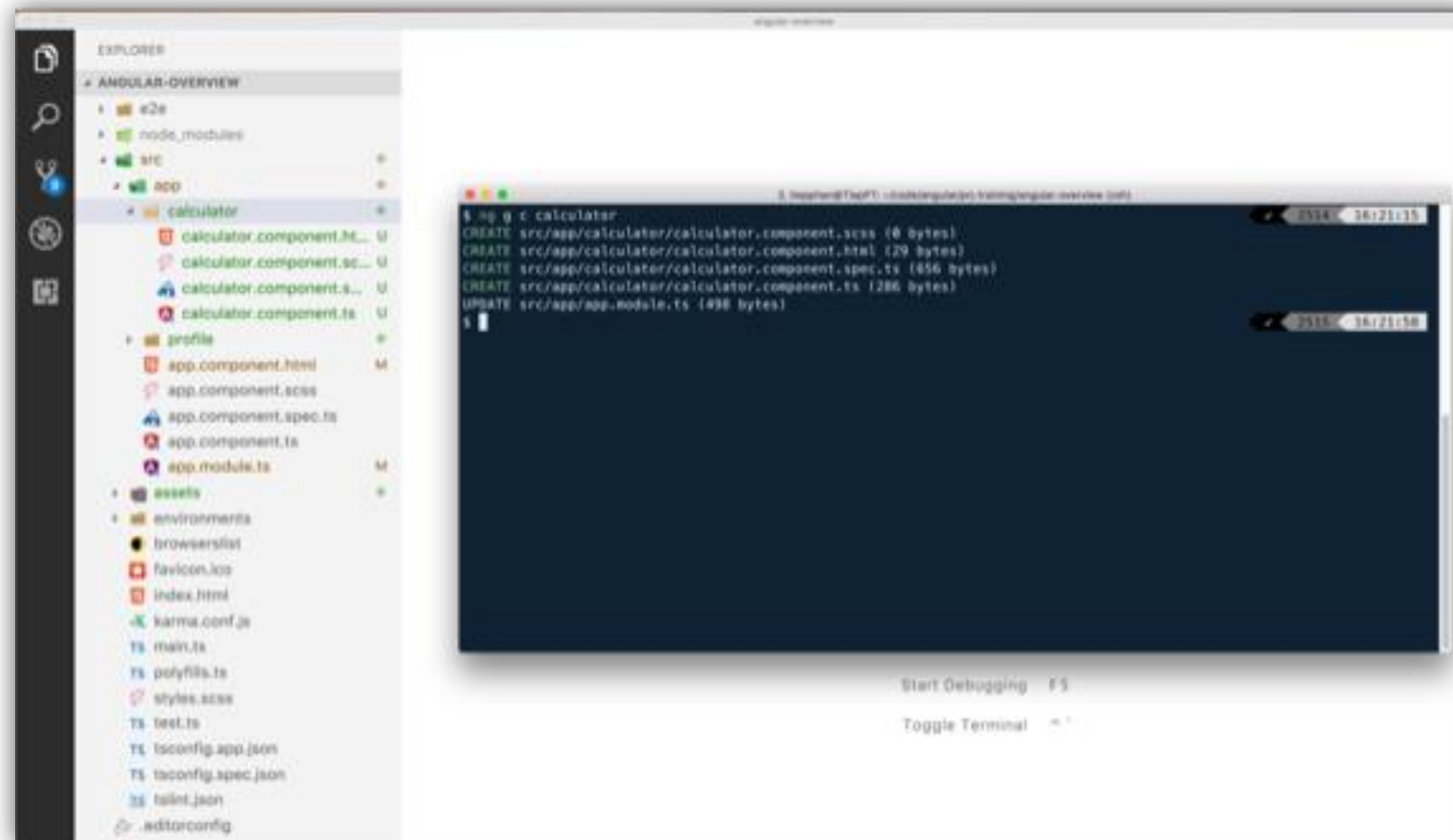
- Bước 1: Di chuyển terminal/CMD vào thư mục gốc (hoặc thư mục con mà bạn mong muốn đặt component ở đó) của project.
- Bước 2: Chạy lệnh sau để generate component với Angular CLI.
- Bước 3: Thêm mới **selector** của component vừa tạo vào file **app.component.ts**

ng generate component calculator

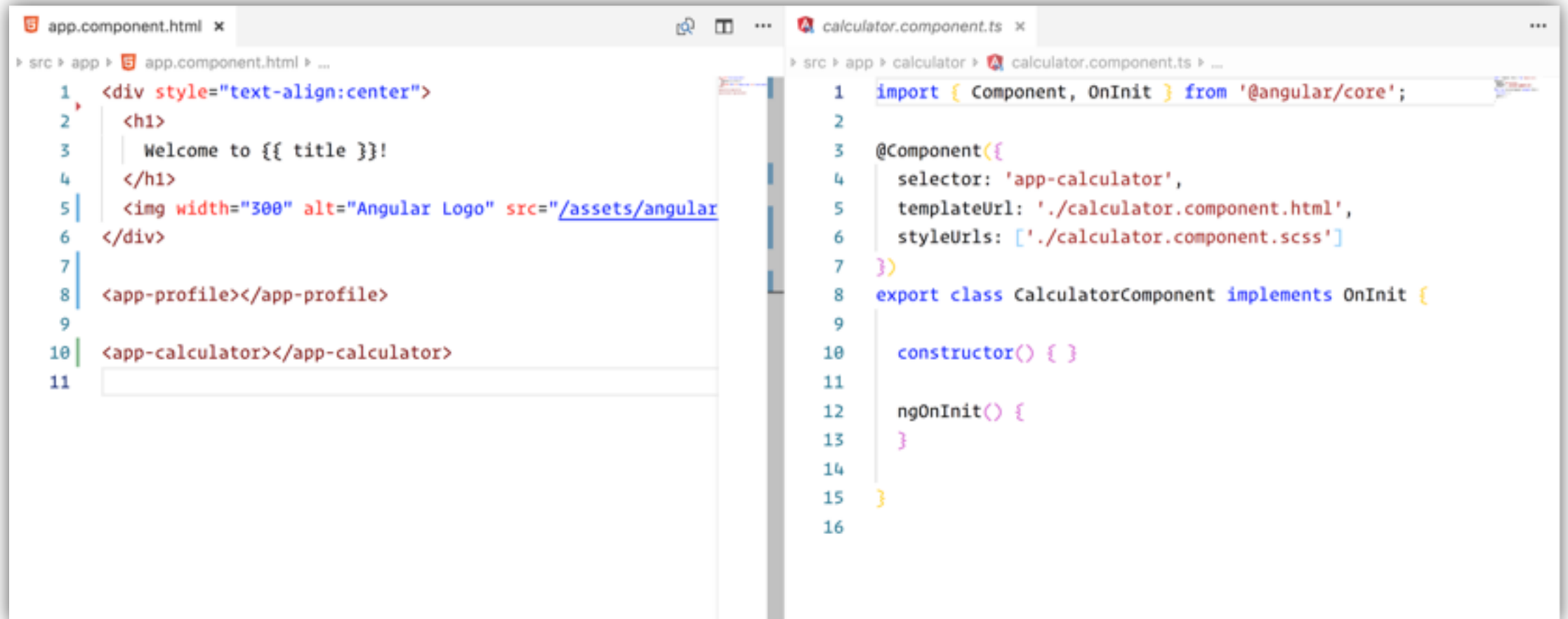
// hoặc

ng g c calculator

Thực hiện



Thực hiện



The image shows a code editor with two files open. The left file, `app.component.html`, contains HTML code for a component. The right file, `calculator.component.ts`, contains TypeScript code for the same component.

```
app.component.html
1 <div style="text-align:center">
2   <h1>
3     Welcome to {{ title }}!
4   </h1>
5   <img width="300" alt="Angular Logo" src="/assets/angular
6 </div>
7
8 <app-profile></app-profile>
9
10 <app-calculator></app-calculator>
11
```

```
calculator.component.ts
1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-calculator',
5   templateUrl: './calculator.component.html',
6   styleUrls: ['./calculator.component.scss']
7 })
8 export class CalculatorComponent implements OnInit {
9
10   constructor() { }
11
12   ngOnInit() {
13   }
14
15 }
16
```

Thực hiện

Welcome to angular-overview!

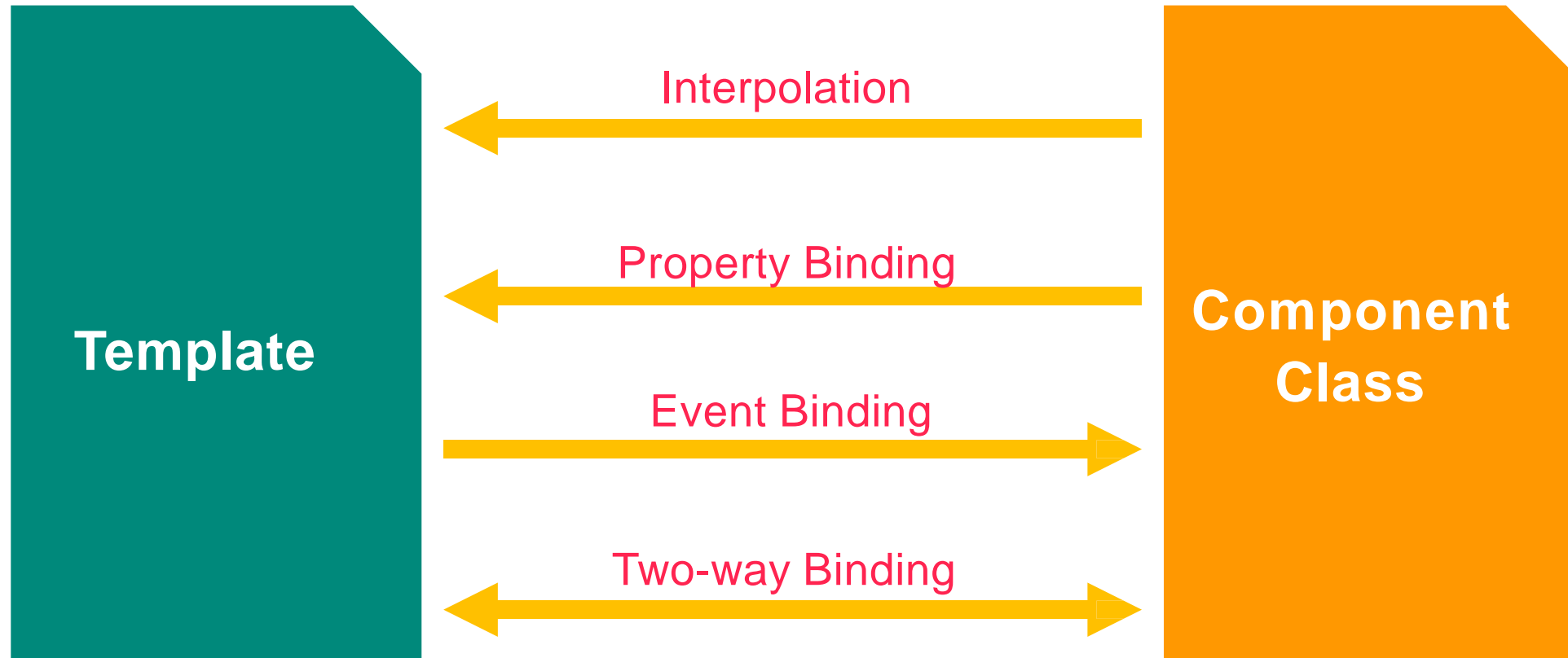


Profile Component Template

calculator works!

Databinding trong Angular

Angular Databinding



Ứng dụng Calculator

```
export class CalculatorComponent {
```

```
  output: number;  
  first: number;  
  second: number;  
  operator = '+';
```

Các properties của component

```
  onFirstChange(value) {}  
  onSecondChange(value) {}  
  onSelectChange(value) {}  
  calculate() {}
```

Các methods của component

```
}
```

Ứng dụng Calculator – calculator.component.ts

```
export class CalculatorComponent {  
  onFirstChange(value) {  
    this.first = Number(value);  
  }  
  onSecondChange(value) {  
    this.second = Number(value);  
  }  
  onSelectChange(value) {  
    this.operator = value;  
  }  
}
```

Ứng dụng Calculator – calculator.component.ts

```
export class CalculatorComponent {  
  calculate(){  
    switch (this.operator) {  
      case '+': this.output = this.first + this.second;  
        break;  
      case '-': this.output = this.first - this.second;  
        break;  
      case '*': this.output = this.first * this.second;  
        break;  
      case '/': this.output = this.first / this.second;  
        break;  
    }  
  }  
}
```

Ứng dụng Calculator – calculator.component.html

```
<h2>Awesome Calculator</h2>
```

```
<div>
```

```
  <input type="number" class="form-input"
```

```
    [value]="first"
```

```
    (change)="onFirstChange($event.target.value)"
```

```
<select
```

```
  class="form-input"
```

```
  [value]="operator"
```

```
  (change)="onSelectChange($event.target.value)">
```

```
    <option value="+">+</option>
```

```
    <option value="-">-</option>
```

```
    <option value="*">*</option>
```

```
    <option value="/">/</option>
```

```
</select>
```

```
<input type="number" class="form-input"
```

```
  [value]="second"
```

```
  (change)="onSecondChange($event.target.value)">
```

```
</div>
```

Property binding

Event binding

Ứng dụng Calculator – calculator.component.ts

```
<div>
```

```
  <button class="form-input" (click)="calculate()">
```

Thực thi

```
  </button>
```

```
</div>
```

```
<div class="output">
```

```
  {{output}}
```

```
]
```

Interpolation

```
</div>
```

Ứng dụng Calculator

Welcome to angular-overview!



Profile Component Template

Awesome Calculator

4 * 5
Thực thi
20

Two-way binding

```
import { FormsModule } from '@angular/forms';
```

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


Two-way binding – calculator.component.html

```
<input type="number" class="form-input"  
      [(ngModel)]="second">
```

Two-way binding – calculator.component.ts

- Trong file class của component chúng ta có thể bỏ đi method **onSecondChange**, vì nó không cần sử dụng đến nữa.
- Ứng dụng sẽ chạy như trước đó khi sử dụng property binding kết hợp với event binding.



R a i s i n g t h e b a r