



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

Angular Overview

Mục tiêu

- Tìm hiểu được kiến trúc của Angular.
- Cài đặt được môi trường phát triển 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

Cài đặt môi trường phát triển Angular

Node.js

The screenshot shows the official Node.js website. At the top, there's a dark header with the Node.js logo and a navigation bar with links: HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION. The FOUNDATION link is highlighted with a green background. Below the header, a banner for "js Interactive" is displayed, featuring a hexagonal logo and text about an event in Vancouver. A green callout box below the banner says "August 2018 security releases available, upgrade now". Underneath, there are two large green buttons for "macOS (x64)" downloads: one for "8.11.4 LTS" (Recommended For Most Users) and another for "10.9.0 Current" (Latest Features). At the bottom, there are links for "Other Downloads", "Changelog", and "API Docs" for both LTS and Current versions. A note encourages looking at the Long Term Support (LTS) schedule. Finally, a call to action invites users to "Sign up for Node.js Everywhere, the official Node.js Monthly Newsletter."

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Join us at #jsinteractive
October 10-12, 2018
Vancouver, Canada

August 2018 security releases available, upgrade now

Download for macOS (x64)

8.11.4 LTS
Recommended For Most Users

10.9.0 Current
Latest Features

Other Downloads | Changelog | API Docs Other Downloads | Changelog | API Docs

Or have a look at the [Long Term Support \(LTS\) schedule](#).

Sign up for [Node.js Everywhere](#), the official Node.js Monthly Newsletter.

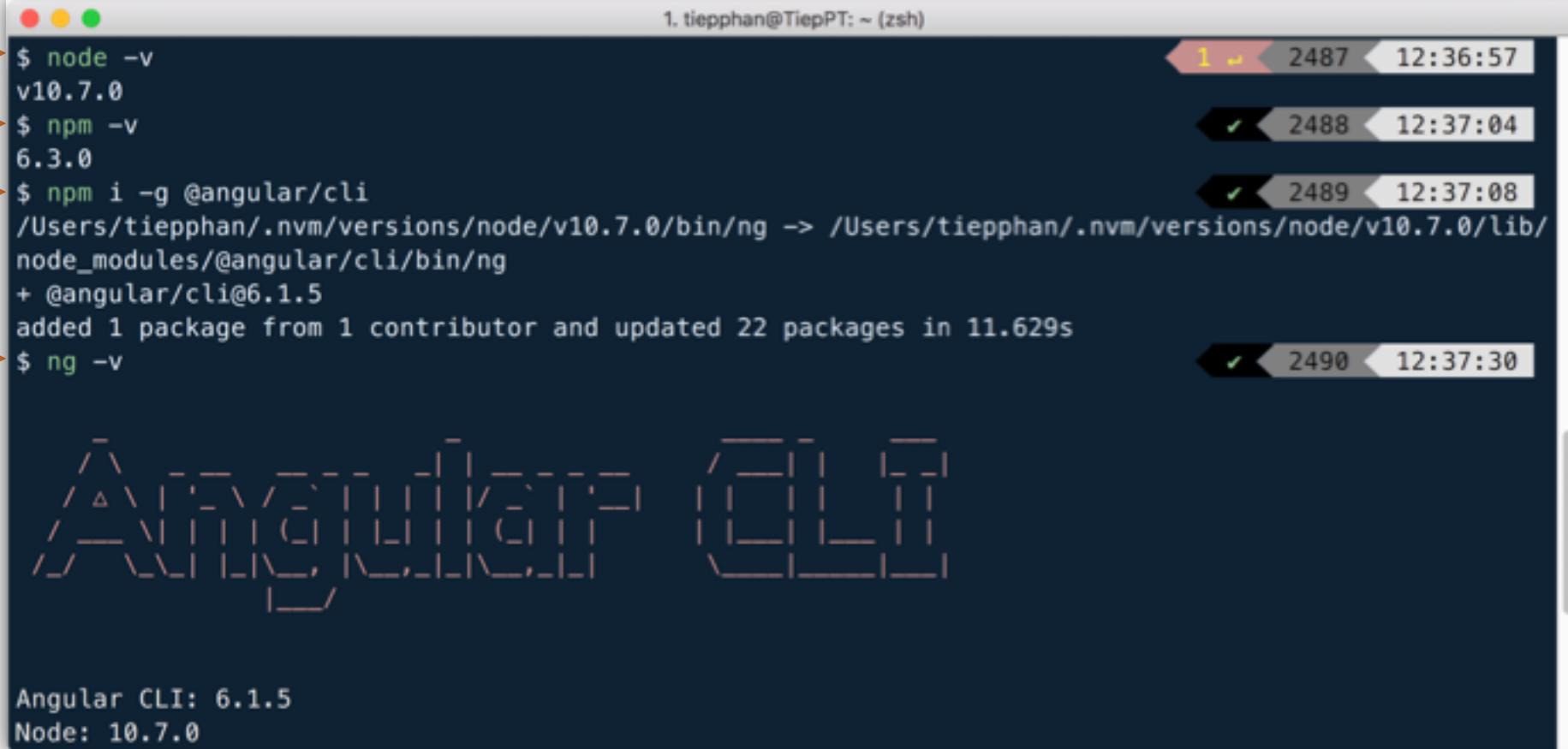
Node.js on Windows

- On Windows? Want to compile native Node modules? Install the build tools with this one-liner.
- Start PowerShell as **Administrator** and run:

```
npm i -g --production windows-build-tools
```

<https://www.npmjs.com/package/windows-build-tools>

Node.js & Angular CLI



The terminal window shows the following command history:

```
1. tiepphan@TiepPT: ~ (zsh)
$ node -v
v10.7.0
$ npm -v
6.3.0
$ npm i -g @angular/cli
/Users/tiepphan/.nvm/versions/node/v10.7.0/bin/ng -> /Users/tiepphan/.nvm/versions/node/v10.7.0/lib/
node_modules/@angular/cli/bin/ng
+ @angular/cli@6.1.5
added 1 package from 1 contributor and updated 22 packages in 11.629s
$ ng -v
```

Four orange arrows point to the first four commands in the history.

At the bottom of the terminal window, there is a decorative footer graphic consisting of a grid of geometric shapes (triangles, squares, and rectangles) arranged in a stylized pattern.

Angular CLI: 6.1.5
Node: 10.7.0

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

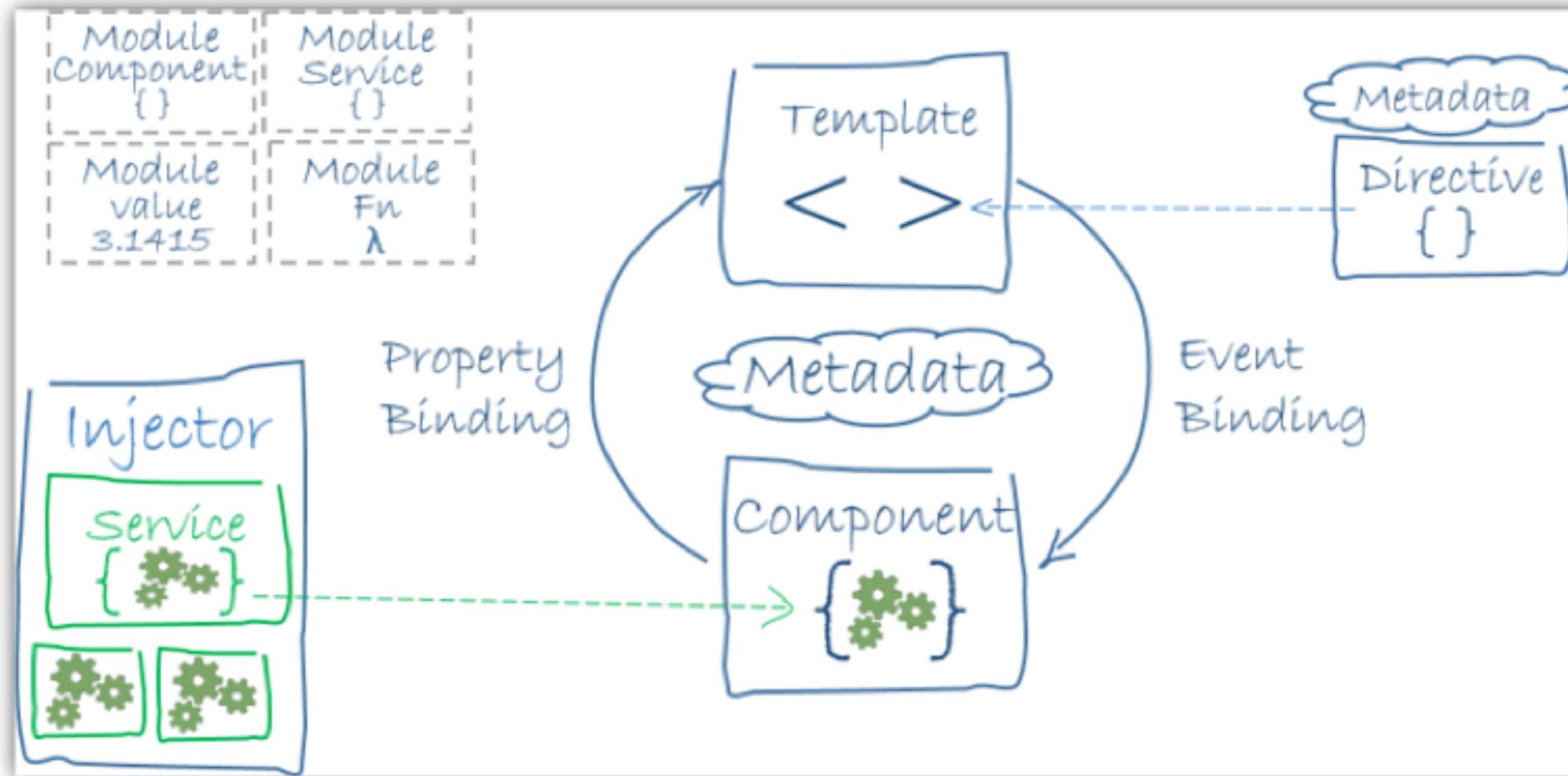
Welcome to angular-overview!



Here are some links to help you start:

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

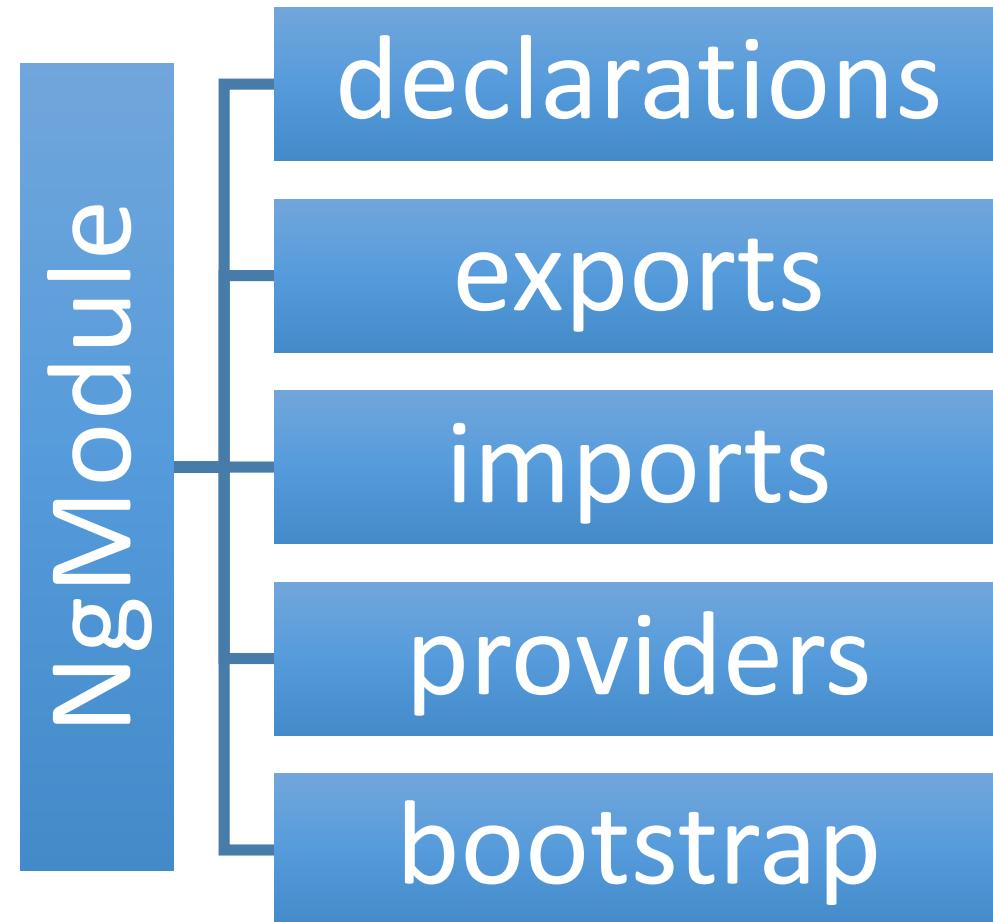
Angular Architecture



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`

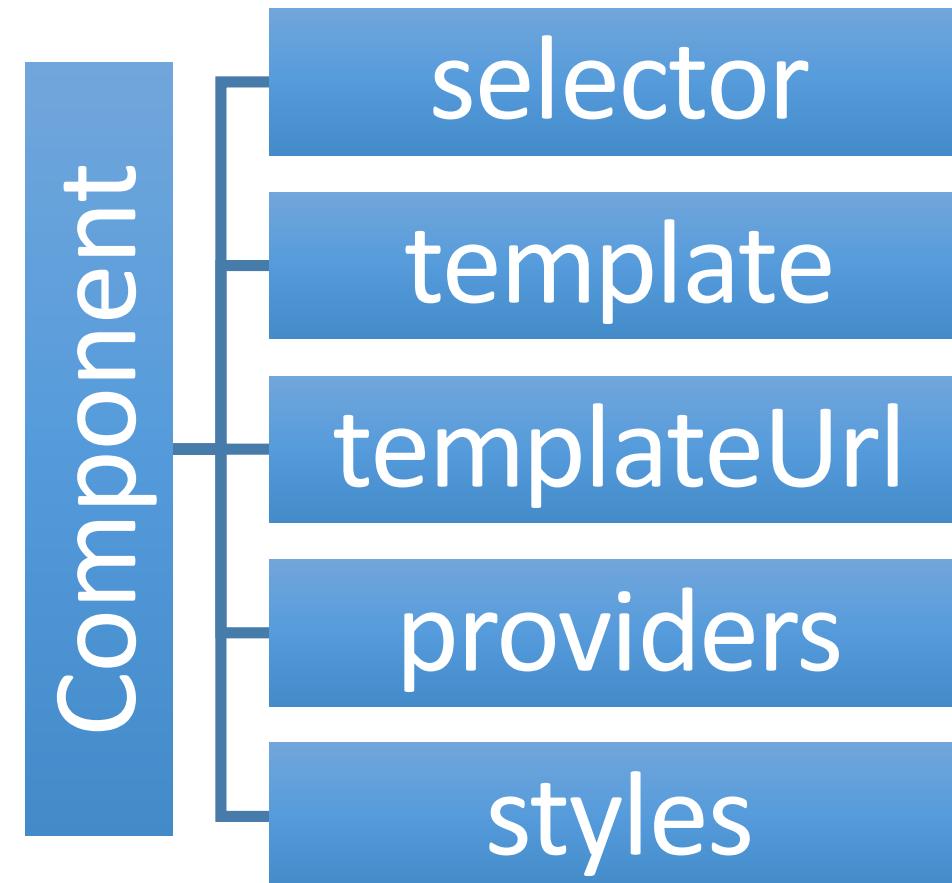
```
import { enableProdMode } from '@angular/core';
import {
  platformBrowserDynamic
} from '@angular/platform-browser-dynamic';

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

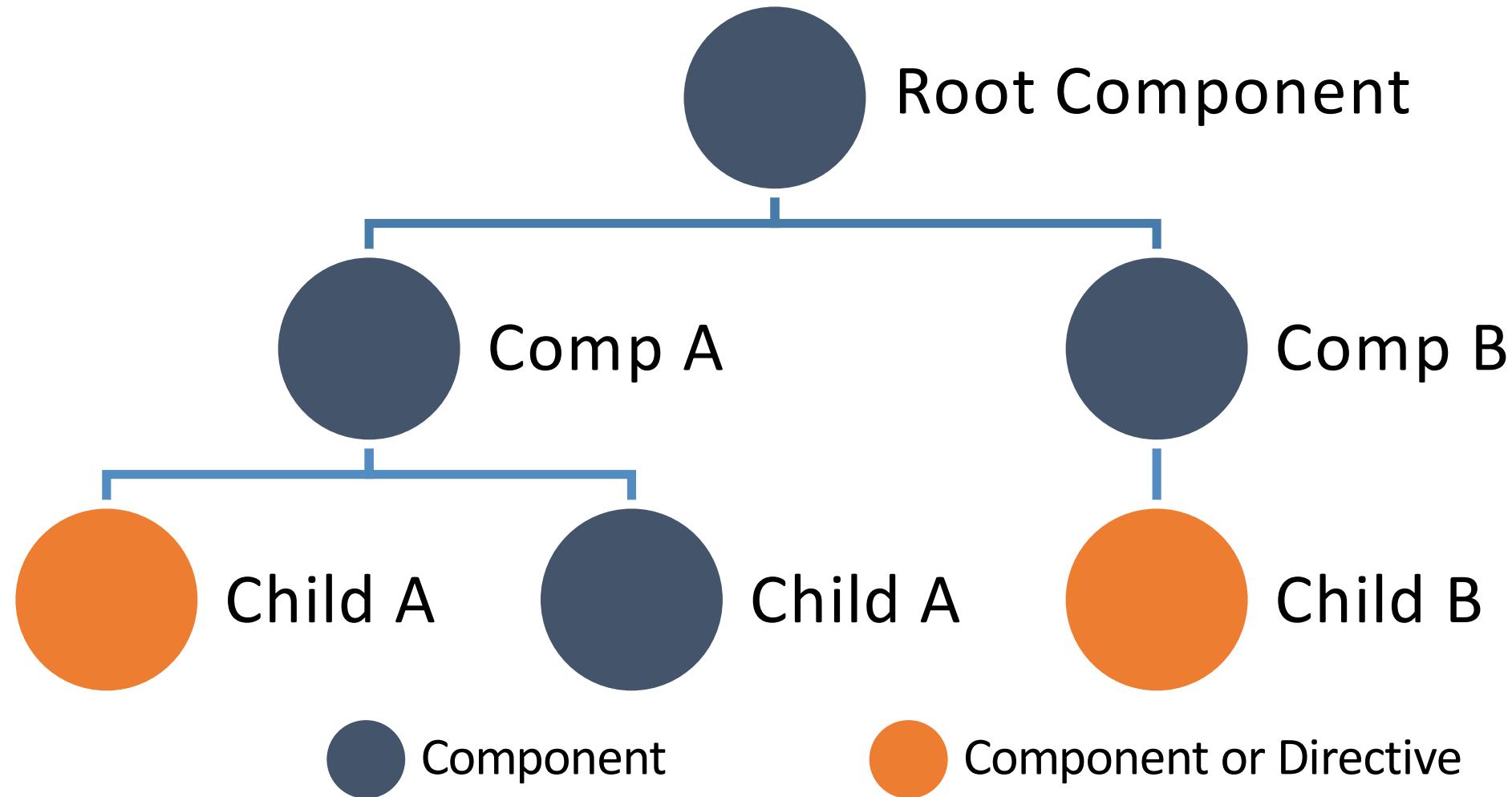
if (environment.production) {
  enableProdMode();
}

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

The diagram illustrates the structure of the `profile.component.ts` file with three callout boxes:

- A callout box labeled "Import decorator Component từ thư viện @angular/core" points to the `import { Component } from '@angular/core';` line.
- A callout box labeled "Metadata của component" points to the `@Component({ ... })` annotation.
- A callout box labeled "Khởi tạo class trong TypeScript" points to the `export class ProfileComponent {}` line.

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



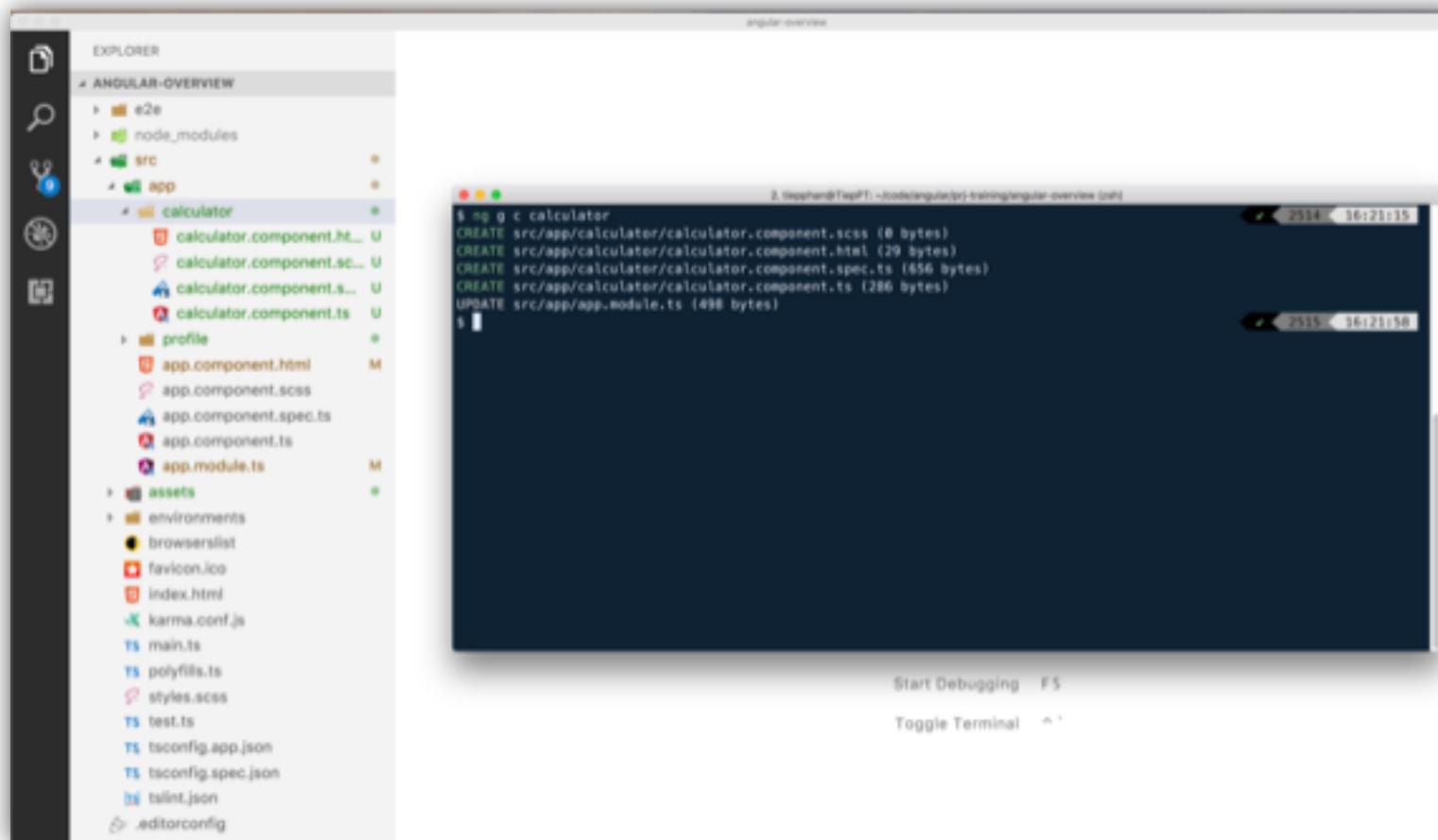
Tạo Angular Component với Angular CLI

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 interface with two tabs open. On the left is the file `app.component.html`, and on the right is the file `calculator.component.ts`. Both files are part of an Angular application structure.

File: app.component.html

```
1 <div style="text-align:center">
2   <h1>
3     | Welcome to {{ title }}!
4   </h1>
5   
  <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}}}
</div>
```

The code snippet shows a portion of an Angular component template. It includes a button labeled "Thực thi" and a div with a placeholder value. A callout box labeled "Interpolation" points to the double curly braces `{{output}}` in the div's template.

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.

CODEGYM

CODEGYM

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