

Khóa học Angular căn bản 2023 - mới nhất

Yêu cầu trước khi học

- Có kiến thức về [HTML](#) và [CSS](#) căn bản
- Có kiến thức về [JavaScript](#) căn bản
- Cơ bản về [TypeScript](#)
- Link khóa học tại TEDU:
<https://tedu.com.vn/khoa-hoc/khoa-hoc-angular-can-ban-2023-moi-nhat-51.html>

Nội dung khóa học

Bài 1: Giới thiệu tổng quan về Angular

Trang chủ: <https://angular.io/>

Trang tài liệu chính thức: <https://angular.io/docs>

Giáo trình sử dụng trực tiếp từ trang chủ Angular, TEDU sẽ cố gắng truyền tải theo tutorial này để sát với tài liệu official nhất. Vì hiện tại các khóa học Angular căn bản trên mạng custom theo từng giảng viên cũng rất nhiều rồi.

Bài 2: Cài đặt môi trường phát triển

Tài liệu gốc: <https://angular.io/guide/setup-local>

1. Cài đặt Visual Studio Code: <https://code.visualstudio.com/>
2. Cài đặt NodeJS: <https://nodejs.org/en> sau đó kiểm tra phiên bản cài đặt bằng câu lệnh: `node -v`
3. Giới thiệu NPM: <https://docs.npmjs.com/about-npm>
4. Cài đặt Angular CLI: `npm install -g @angular/cli`
5. Để chắc chắn PowerShell trên Windows có quyền run câu lệnh: Hãy chạy lệnh sau trên PowerShell: `Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned`

Bài 3: Cách tạo mới và chạy một project Angular

Để tạo mới một project: `ng new my-app`

Để chạy dự án:

`cd my-app`

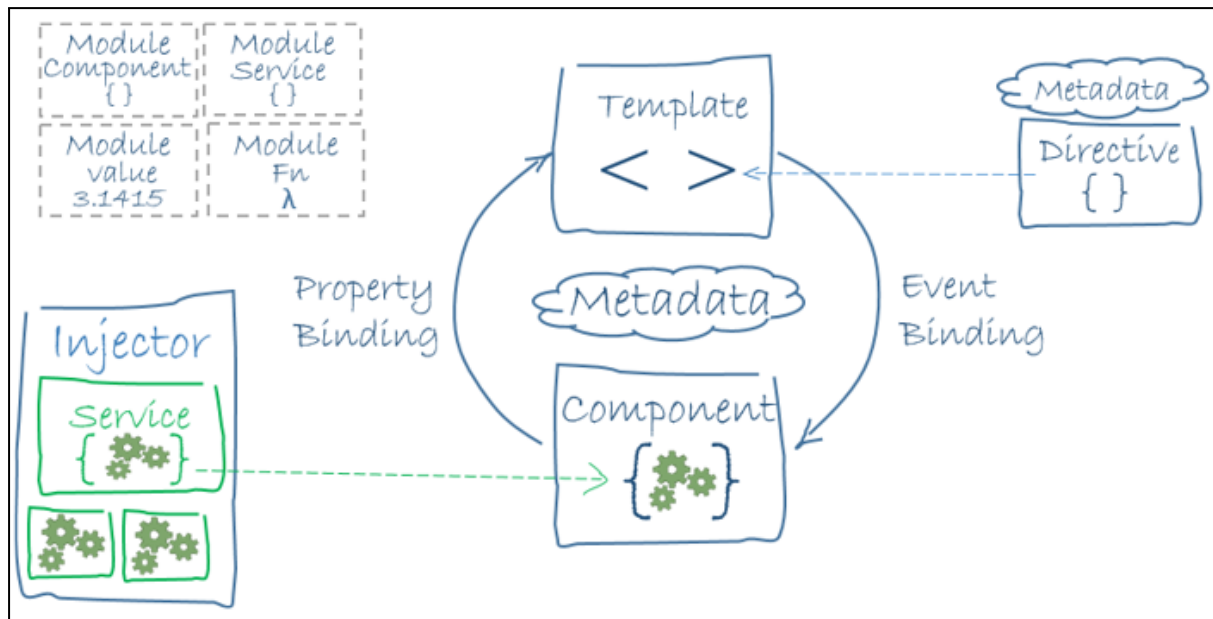
`ng serve --open`

Cấu trúc dự án: <https://angular.io/guide/file-structure>

Bài 4: Kiến trúc tổng quan ứng dụng

Tài liệu gốc: Kiến trúc tổng quan: <https://angular.io/guide/architecture>

DI: <https://angular.io/guide/architecture-services>



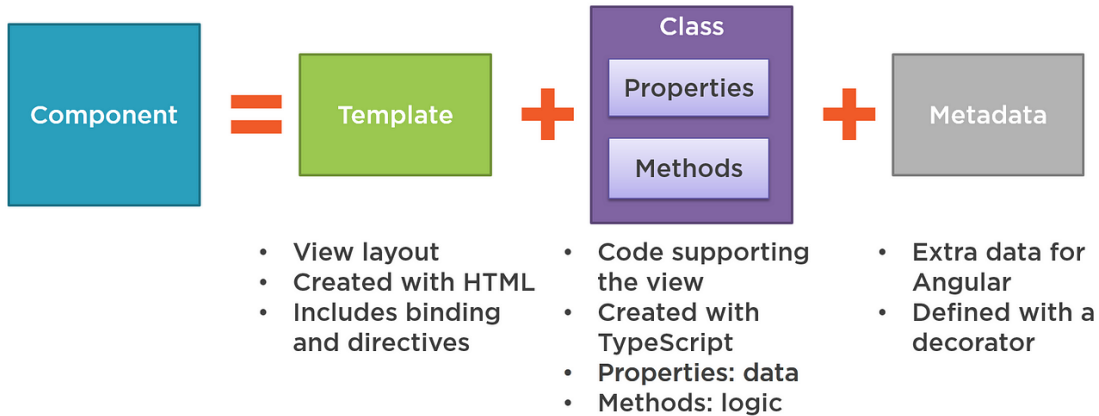
Bài 5: Tổng quan về các thành phần chính trong Angular

- Module: <https://angular.io/guide/architecture-modules>
- Component: <https://angular.io/guide/architecture-components>
- Service: <https://angular.io/guide/architecture-services>
- Routing: <https://angular.io/guide/router>
- Tools: <https://angular.io/guide/architecture-next-steps>

Bài 6: Tổng quan về Components trong Angular

Tài liệu: <https://angular.io/guide/component-overview>

What Is a Component?



Component là một thành phần chính tạo nên ứng dụng Angular, nó bao gồm:

1. Một file HTML Template hoặc đoạn HTML Template
2. Một file TypeScript định nghĩa logic
3. Một file CSS để định nghĩa các quy tắc hiển thị

Tạo một component bằng tay

Định nghĩa template

Định nghĩa thẻ mới (selector) cho Component

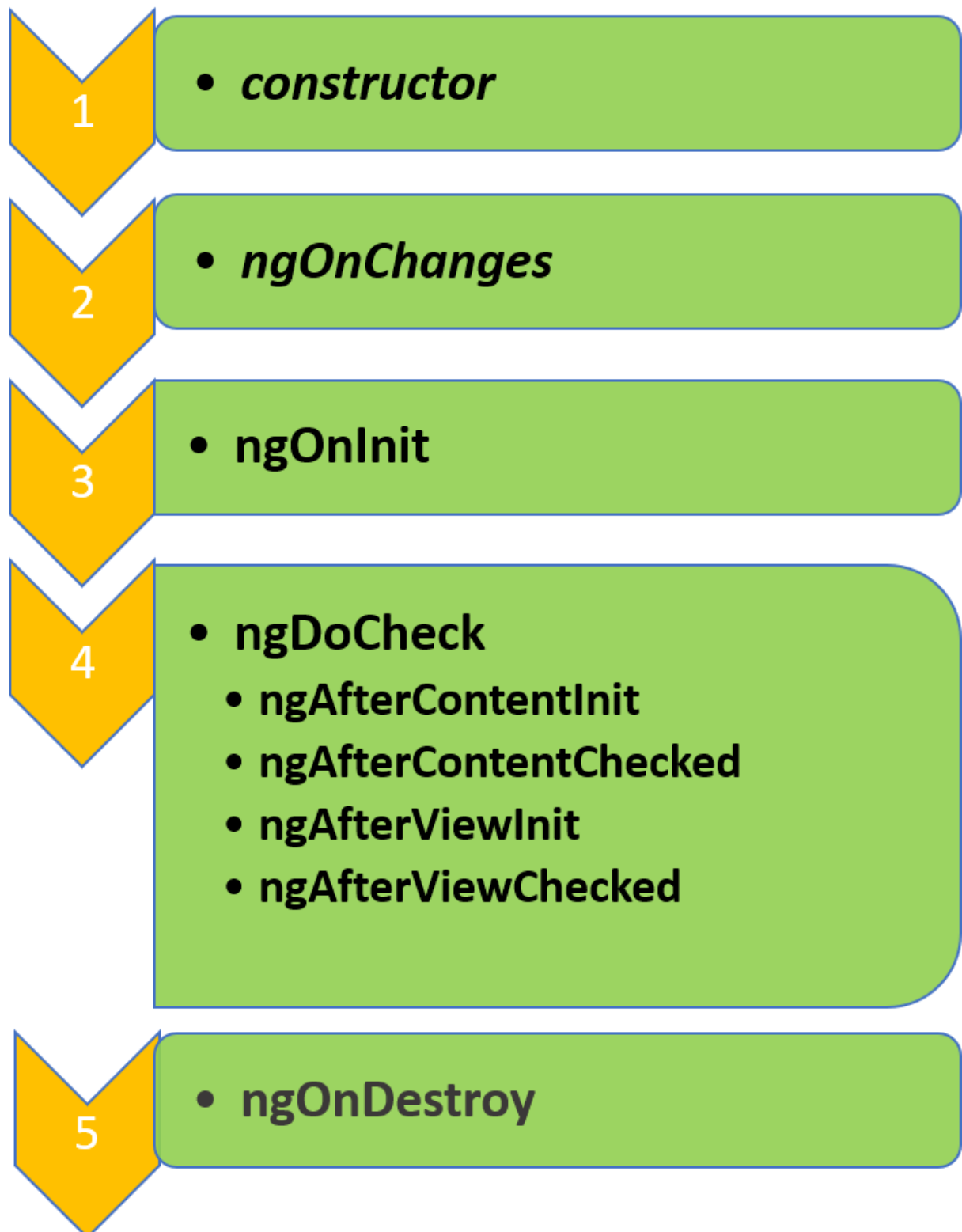
Định nghĩa CSS

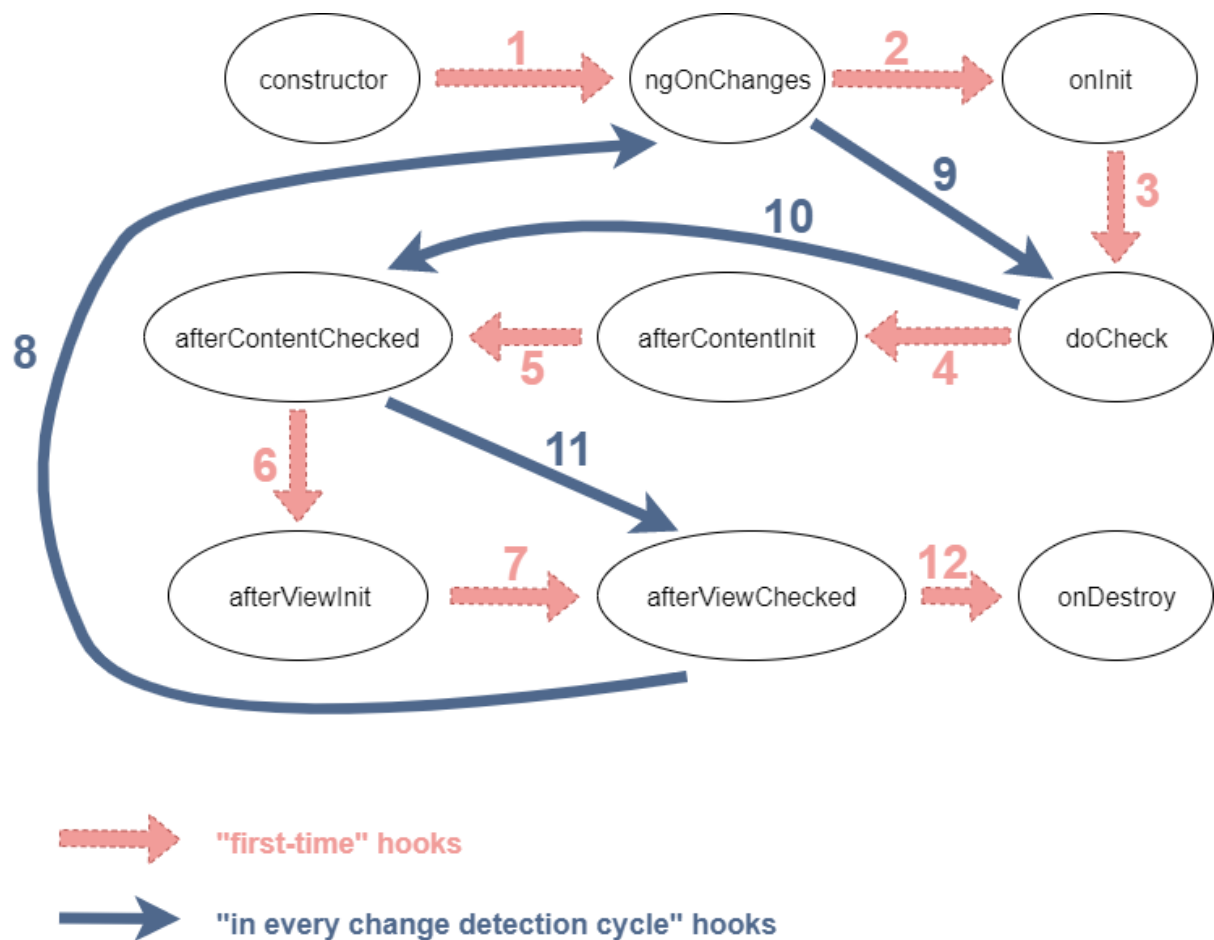
Bài 7: Vòng đời của Components (Component Lifecycle)

Tài liệu: <https://angular.io/guide/lifecycle-hooks>

Tham khảo: <https://indepth.dev/posts/1494/complete-guide-angular-lifecycle-hooks>

Một vòng đời của component (Component Lifecycle) là một quá trình từ khi Component được ứng dụng Angular khởi tạo instance đến khi nó được hủy. Trong toàn bộ quá trình sẽ bao gồm các sự kiện:





Bài 8: Tính đóng gói View trong Component

Tài liệu: <https://angular.io/guide/view-encapsulation>

Shadow DOM:

https://developer.mozilla.org/en-US/docs/Web/API/Web_components/Using_shadow_DOM

Ví dụ: <https://www.tektutorialshub.com/angular/angular-view-encapsulation/>

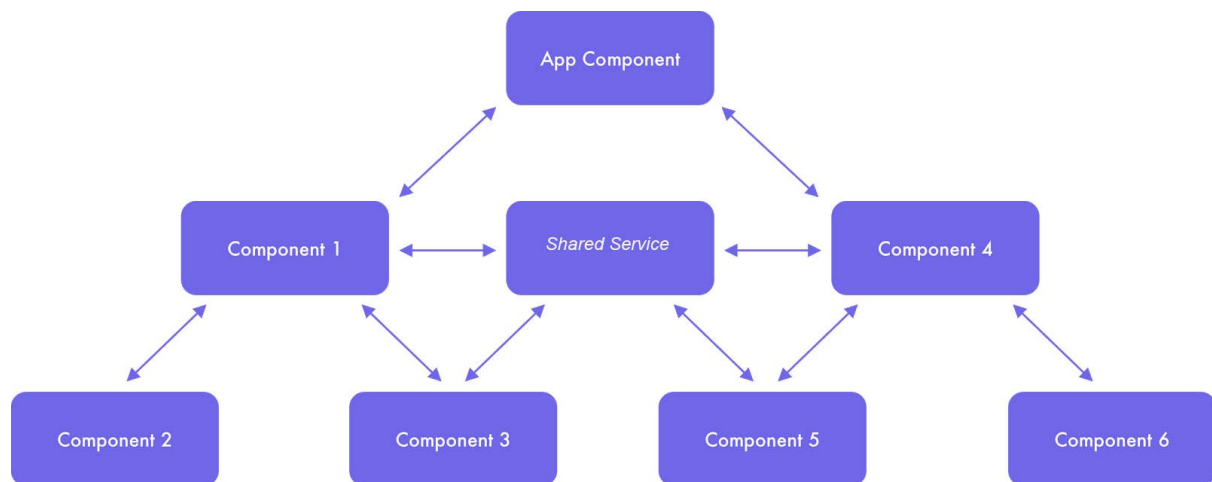
Link hỗ trợ Browser: <https://caniuse.com/shadowdomv1>

| | |
|----------|--|
| Emulated | <ul style="list-style-type: none"> • no Shadow DOM • style encapsulation • value : 0 |
| Native | <ul style="list-style-type: none"> • Shadow DOM • style encapsulation • value : 1 |
| None | <ul style="list-style-type: none"> • no Shadow DOM • no Style encapsulation • value : 2 |

Bài 9: Tương tác giữa các Components

1. Truyền dữ liệu từ Component cha xuống con qua Input Binding:
<https://angular.io/guide/component-interaction#pass-data-from-parent-to-child-with-input-binding>
2. Truyền dữ liệu từ Component cha xuống con qua Setter:
<https://angular.io/guide/component-interaction#intercept-input-property-changes-with-a-setter>
3. Truyền dữ liệu từ Component cha xuống con qua ngOnChanges event:
<https://angular.io/guide/component-interaction#intercept-input-property-changes-with-ngonchanges>
4. Component cha lắng nghe sự kiện từ component con:
<https://angular.io/guide/component-interaction#parent-listens-for-child-event>
5. Tương tác thông qua biến nội bộ:
<https://angular.io/guide/component-interaction#parent-interacts-with-child-using-local-variable>
6. Sử dụng @ViewChild:
<https://angular.io/guide/component-interaction#parent-calls-an-viewchild>
7. Tương tác thông qua Services:
<https://angular.io/guide/component-interaction#parent-and-children-communicate-using-a-service>

Ví dụ: <https://stackblitz.com/run?file=src%2Fapp%2Fapp.component.html>



Bài 10: Component Styles

Tài liệu: <https://angular.io/guide/component-styles>

SASS: <https://sass-lang.com/>

Bài 11: Truyền dữ liệu giữa component cha và con

Link gốc: <https://angular.io/guide/inputs-outputs>

Bài 12: Trình bày dữ liệu trên view

Link gốc: <https://angular.io/guide/content-projection>

Bài 13: Hiện thị Component động

Link gốc: <https://angular.io/guide/dynamic-component-loader>

Bài 14: Đóng gói custom elements

Link gốc: <https://angular.io/guide/elements>

Bài 15: Tổng quan về Templates

Link gốc: <https://angular.io/guide/template-overview>

Bài 16: Hiện thị nội dung dạng Interpolation

Tài liệu: <https://angular.io/guide/interpolation>

Bài 17: Sử dụng lệnh trong Template

<https://angular.io/guide/template-statements>

Bài 18: Tổng quan về Data Binding

<https://angular.io/guide/binding-overview>

Bài 19: Property Binding

<https://angular.io/guide/property-binding>

Bài 20: Attribute Binding

<https://angular.io/guide/attribute-binding>

Bài 21: Class và Style Binding

<https://angular.io/guide/class-binding>

Bài 22: Event Binding

<https://angular.io/guide/event-binding>

Bài 23: Two-way Binding

<https://angular.io/guide/two-way-binding>

Bài 24: Sử dụng Pipe

<https://angular.io/guide/pipes-overview>

Bài 25: Template reference variable

<https://angular.io/guide/template-reference-variables>

Bài 26: Tổng quan về Directive

Bài 27: Built-in directive

<https://angular.io/guide/built-in-directives>

Bài 28: Attribute Directive

<https://angular.io/guide/attribute-directives>

Bài 29: Structural Directive

<https://angular.io/guide/structural-directives>

Bài 30: Dependency Injection

<https://angular.io/guide/dependency-injection-overview>

Bài 31: Tạo và sử dụng Routing trong Angular

Tạo ứng dụng TodoList sử dụng routing

BE Demo: <https://github.com/teduinternational/todo-list-crud-in-memory-db-ef-core>

Tham khảo: <https://angular.io/guide/routing-overview>

Bài 32: Làm việc với Reactive Forms

Tạo form tạo mới task trong Todolist sử dụng Reactive form

Tham khảo:

<https://angular.io/guide/forms-overview>

<https://angular.io/guide/dynamic-form>

Bài 33: Validate form input

<https://angular.io/guide/form-validation>

<https://angular.io/guide/typed-forms>

Bài 34: Làm việc với HTTP Client

CRUD với Backend sử dụng HTTP Client

<https://angular.io/guide/http>

Bài 35: Upload file với Angular

Upload file đính kèm sử dụng Angular + Backend .NET

Bài 36: Giới thiệu thêm các tính năng quan trọng trong Angular

1. Testing: <https://angular.io/guide/testing>
2. Animation: <https://angular.io/guide/animations>
3. PWA: <https://angular.io/guide/service-worker-intro>
4. Server Rendering: <https://angular.io/guide/universal>
5. Deployment: <https://angular.io/guide/deployment>
6. Security: <https://angular.io/guide/security>
7. Lazy load: <https://angular.io/guide/lazy-loading-ngmodules>
8. Template driven form: <https://angular.io/guide/forms>