**SỰ KHÁC NHAU GIỮA TYPESCRIPT VÀ JAVASCRIPT**

Về bản chất tất cả code Javascript đều có thể chạy được trong Typescript hay nói cách khác  một phiên bản nâng cao của Javascript.

***JavaScript + more features = Typescript***

Sau đây là một vài điểm khác nhau giữa chúng:

|  |  |
| --- | --- |
| TypeScript | JavaScript |
| Cần cài đặt thông qua npm | Không cần cài đặt |
| Dễ dàng hơn trong phát triển các dự án lớn, được hỗ trợ bởi các Javascript Framework lớn | Tạo trang web động |
| Lỗi có thể được tìm thấy và sửa chữa trong thời gian biên dịch | Lỗi chỉ có thể được tìm thấy lúc run-time |
| Hỗ trợ cả static và dynamic type | Chỉ làm việc với kiểu động (dynamic type) |
| Chuyển đổi thành JavaScript để phù hợp với các trình duyệt | Có thể được sử dụng trực tiếp trong trình duyệt |
| Có hỗ trợ ES3, ES4, ES5 và ES6 | Không hỗ trợ biên dịch ES3, ES4, ES5 hoặc ES6 |
| Kiến thức về script là bắt buộc | Không cần kinh nghiệm về script trước đó. |
| Hỗ trợ modules, generics và interfaces để định nghĩa dữ liệu | Không hỗ trợ modules, generics hay interfaces |
| Hỗ trợ tính năng tạo prototype | Không hỗ trợ prototype |
| Hỗ trợ OOP mạnh: Hầu hết các cú pháp hướng đối tượng đều được hỗ trợ bởi Typescript như kế thừa, đóng gói, constructor, abstract, interface, implement, override... | Có hỗ trợ OOP nhưng khi áp dụng thì lại rất khó khăn bởi cách triển khai code không hề đơn giản so với những ngôn ngữ bậc cao khác |

**CÁC SYNTAX CƠ BẢN CỦA ANGULAR**

- Cài đặt Angular CLI: npm install -g @angular/cli

- Tạo dự án Angular: ng new <name> [options]

- Chạy dự án: ng serve

- Tạo Component: ng g component <name> [options]

- Tạo Module: ng g module <name> [options]

- Tạo Directive: ng g directive <name> [options]

- Tạo Pipe: ng g pipe <name> [options]

- Tạo service: ng g service <name> [options]

**QUY TRÌNH XÂY DỰNG MỘT ỨNG DỤNG ANGULAR**

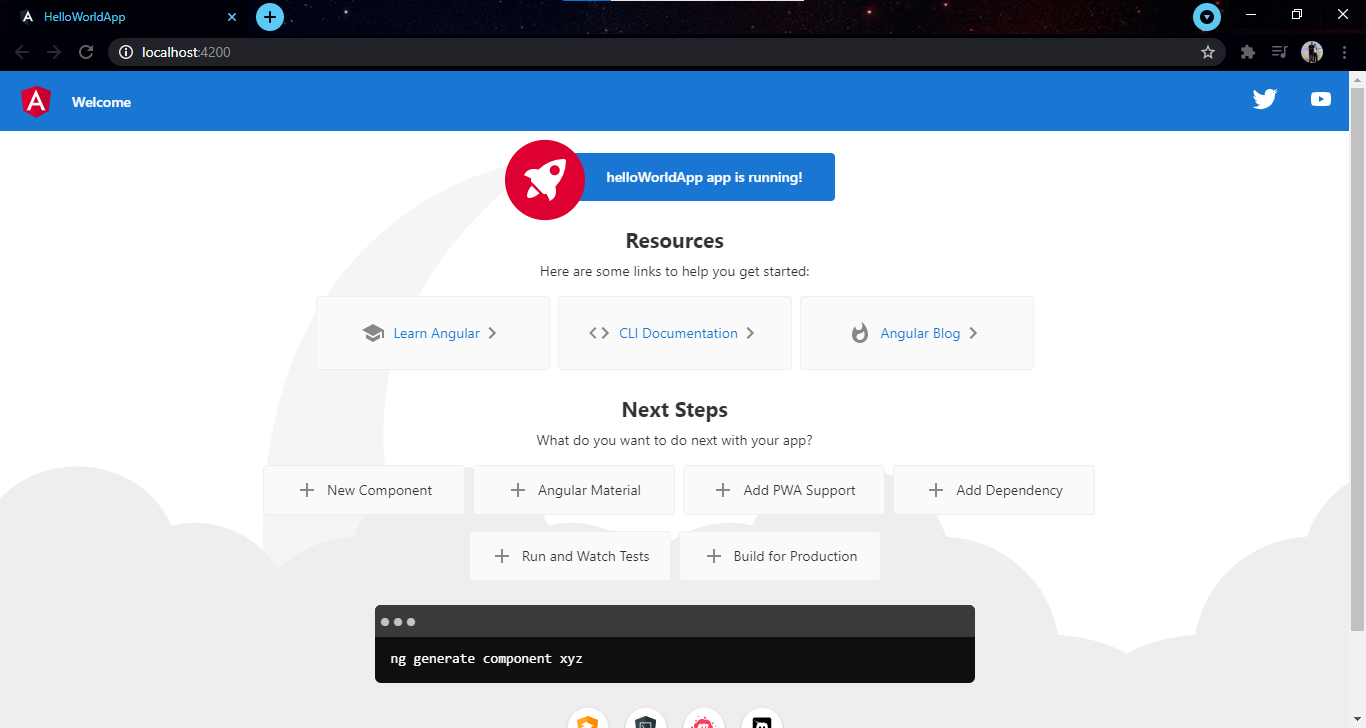
1. Ví dụ chúng ta sẽ làm một ứng dụng đơn giản như Hello World:
   1. Tạo dự án và run dự án Angular:

ng new hello-world-app

cd hello-world-app

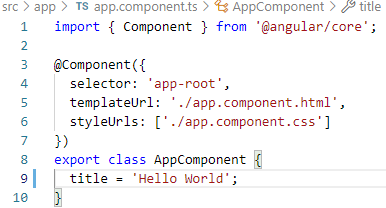
ng serve --open (option *--open* sẽ giúp tự động chạy ứng dụng trên trình duyệt, với cổng mặc định là 4200)

Kết quả như sau:

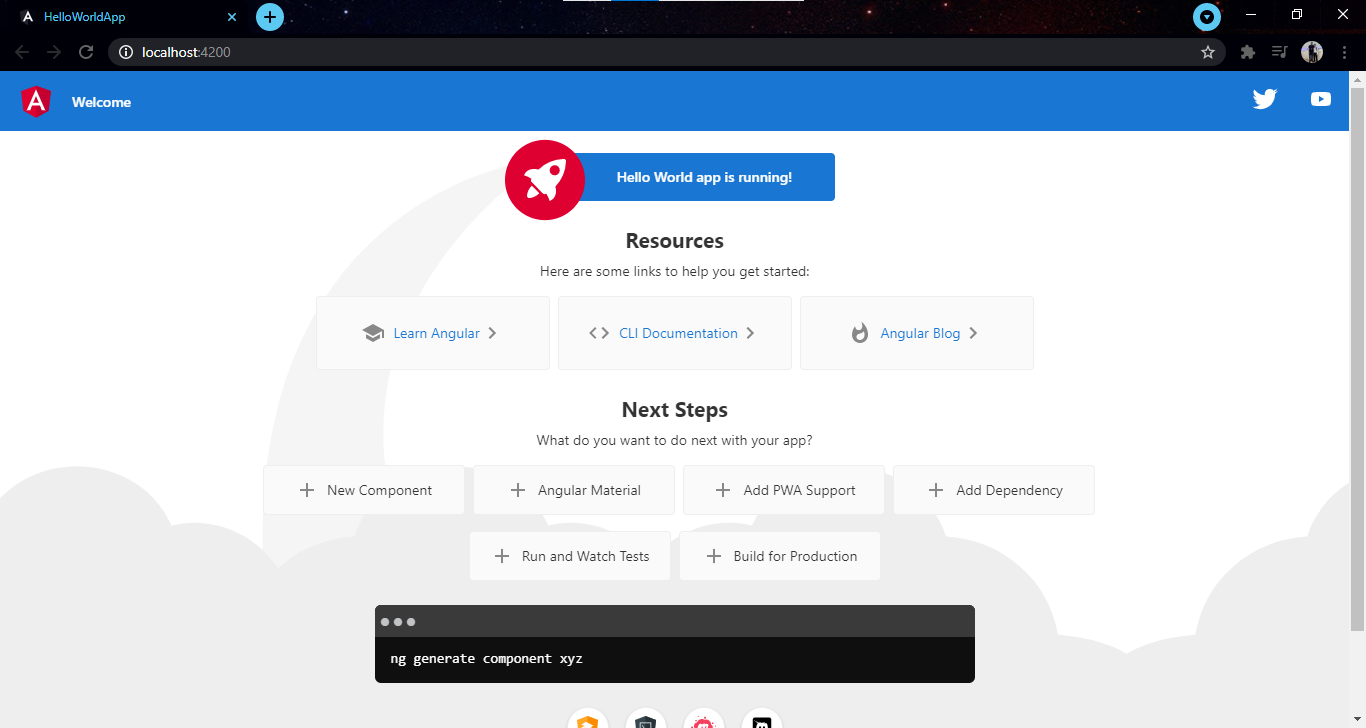


* 1. Chỉnh sửa dự án Angular:

Bây giờ chúng ta sẽ chỉnh sửa ứng dụng thành "Hello World". Quay trở lại file *app.component.ts* và thay đổi giá trị của *title* thành "Hello World", theo đường dẫn *“src/app/app.component.html”*



Và ta đã hoàn thành xong ứng dụng Hello World đầu tiên:



1. Tương tự Hello World, ta tiếp tục xây dựng ToDo App:

Khởi tạo và run dự án:

ng new todo-app

cd todo-app

ng serve –open

* 1. Tạo một component mới:

ng g component todo

Trong file index.html, thay đổi bộ thẻ <app-root></app-root> thành

<app-todo ></app-todo>

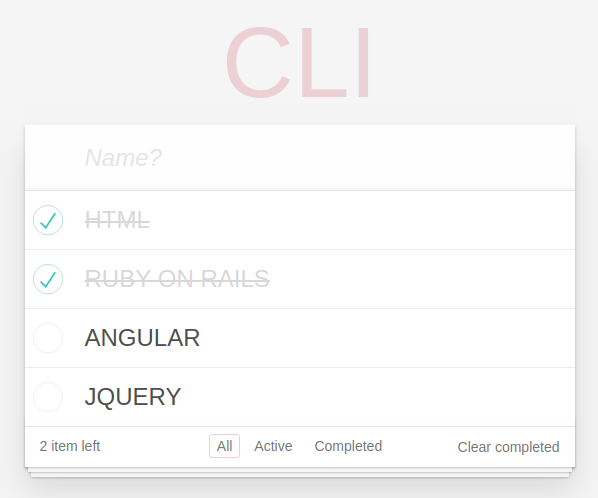
Thay thế *todo.component.html* bằng mã sau:



Cài css mẫu để hiển thị được đẹp hơn:

npm install --save todomvc-app-css

Để sử dụng được file css từ module thì ta cần thêm *"node\_modules/todomvc-app-css/index.css"* vào phần styles của file *angular.json*. Stop và Start lại server để thấy sự khác biệt. Ta sẽ có 1 giao diện trông như sau:



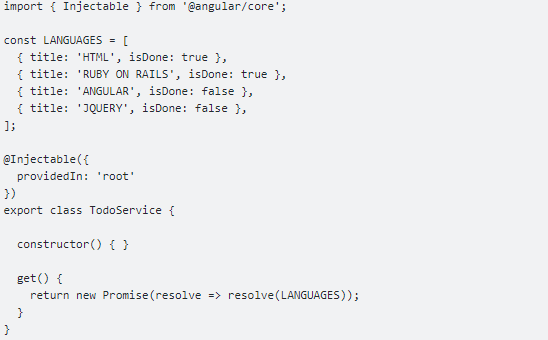
* 1. Service:

Trước tiên hãy bắt đầu bằng cách tạo một service chứa danh sách ban đầu các tác vụ mà chúng ta muốn quản lý. Chúng ta sẽ sử dụng một service để thao tác dữ liệu. Hãy tạo service với CLI bằng cách gõ:

ng g service todo/todo

* 1. Tạo danh sách:

Mở file *todo.service.ts* và thêm vào như sau:



Và file *todo.component.ts:*



Khi Khai báo thêm providers vào thì chúng ta có thể sử dụng todoService các function trong này. Tiếp theo là thay đổi 1 chút code html

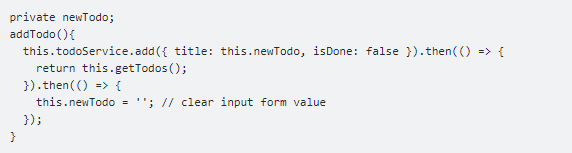


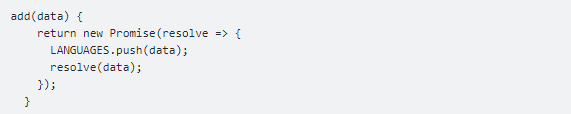
* ngFor: lặp qua mảng mà chúng ta đã xác định trong component.
* [ngClass]: áp dụng class cho language tương ứng.
* [checked]: áp dụng checked thuộc tính cho input checkbox ( todo.isDone).
  1. Create:

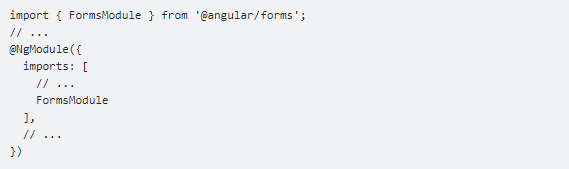
Update *src/app/todo/todo.component.html*

<input class="new-todo" placeholder="Name?" [(ngModel)]="newTodo" (keyup.enter)="addTodo()" autofocus>

*src/app/todo/todo.component.ts*

**

*src/app/todo/todo.service.ts*

*app.module.ts* **

* 1. Update khi click đôi:

Update *src/app/todo/todo.component.html*

<li \*ngFor="let todo of todos" [ngClass]="{completed: todo.isDone, editing: todo.editing}" >

<div class="view">

<input class="toggle" type="checkbox" [checked]="todo.isDone">

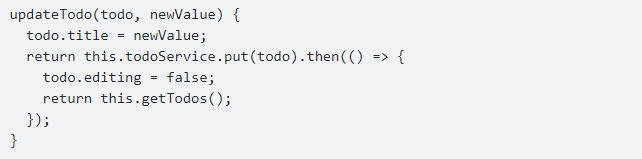
<label (dblclick)="todo.editing = true">{{todo.title}}</label>

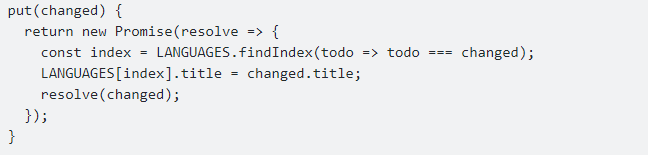
<button class="destroy"></button>

</div>

<input class="edit" #updatedTodo [value]="todo.title" (blur)="updateTodo(todo, updatedTodo.value)" (keyup.escape)="todo.editing = false" (keyup.enter)="updateTodo(todo, updatedTodo.value)">

</li>

Viết thêm function vào *src/app/todo/todo.component.ts*

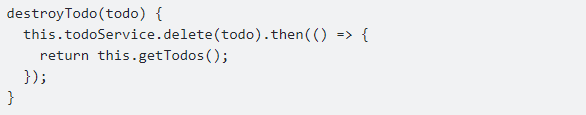
Xử lý *src/app/todo/todo.service.ts*

* 1. Delete:

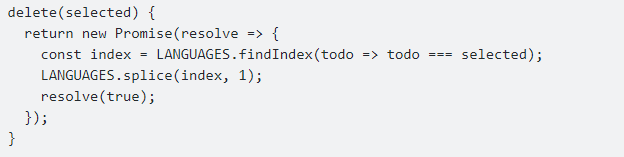
Update *src/app/todo/todo.component.html*



*src/app/todo/todo.component.ts*

**

*src/app/todo/todo.service.ts*

**

Qua phần giới thiệu trên, ta thấy được ứng dụng này mới chỉ là trên client, nó không thể lưu vào database như backend.

- - - Hết - - -