**1 Trình bày được ý nghĩa và khái niệm Observer**

Phương thức subcribe() sẽ nhận vào một Observer. Một Observer sẽ định nghĩa phương thức callback để xử lý 3 loại trạng thái mà một observable có thể gửi như sau:

* next:  Bắt buộc phải có, được sử dụng để lấy ra dữ liệu từ Observable
* error: Không bắt buộc, được sử dụng để xử lý các lỗi xảy ra trong quá trình xử lý.
* complete: Không bắt buộc, đượ sử dụng để xử lý các nội dung mà chúng ta muốn được xử lý sau khi một Observers hoàn thành.

**2 Trình bày được ý nghĩa và khái niệm Observable**

**6 Trình bày được ý nghĩa và khái niệm subscribe**

Phương thức subcribe() được gọi để khởi động 1 Observable stream. Nếu chúng ta không gọi phương thức subcribe thì stream sẽ không được hoạt động và do đó chúng ta sẽ không lấy được giá trị trả về từ Observable.

3 Phân biệt sự khác nhau giữa var, let, const

|  |  |
| --- | --- |
| Let/const | Var |
| -Không có hosting  -Không khai báo lại  -Block scope | -có  -được khai báo lại  -funtion scope |

**4 File angular.json và package.json sử dụng để làm gì?**

- package.json: file này chứa các thư viện cần thiết cho dự án angular, ngoài ra nếu ta thêm một thư viện bên thứ 3 vào thì khai báo trong này.

- angular.json : dùng để cấu hình lại Angular Cli. giúp chúng ta có thể build ứng dụng Angular

**5 Kể tên các directive được hỗ trợ trong angular**

Angular Directive được chia là 3 loại gồm:

+ Component Directive :Component cũng là 1 directvice (<app-list-customer>)

+ Structural Directive: để thay đổi layout của trang website bằng cách thêm hoặc xóa các thành phần trên web (DOM Elements). NgFor,NgIf,NgSwitch

+ Attribute Directive: Attribute Directive được dùng để thay đổi sự hiển thị hoặc hành vi của một thành phần trên web. NgModel,NgClass,NgStyle

**7 Trình bày khái niệm Single page application**

Single Page Application (SPA) là một ứng dụng web động được thiết kế để cung cấp trải nghiệm người dùng tương tự như một ứng dụng desktop truyền thống.

**8 Trình bày về Data binding trong component**

Databinding là kỹ thuật nơi dữ liệu được đồng bộ giữa component và tầng view

- One way binding: One way binding thì dữ liệu được truyền 1 chiều. Có thể từ view sang component hoặc ngược lại từ component sang view

+ Interpolation, Property Binding, Event Binding

- 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.

**9 Trình bày sự khác nhau giữa navigateByUrl và navigate?**

Cả hai phương thức navigateByUrl và navigate đều được sử dụng trong Angular để điều hướng giữa các trang (route) trong ứng dụng. Tuy nhiên, chúng có những khác biệt nhất định.

Phương thức navigateByUrl cho phép bạn điều hướng đến một đường dẫn cụ thể trong ứng dụng bằng cách cung cấp một chuỗi URL, ví dụ như "/home" hoặc "/users". Phương thức này sẽ thực hiện việc điều hướng bằng cách tạo ra một yêu cầu HTTP GET đến đường dẫn tương ứng và tải lại toàn bộ trang. Điều này có nghĩa là nó sẽ xóa hết dữ liệu và trạng thái của trang hiện tại và tải lại tất cả các tài nguyên liên quan đến trang mới.

Phương thức navigate cho phép bạn điều hướng đến một trang khác trong ứng dụng bằng cách cung cấp một đối tượng Route với các thông tin về đường dẫn, tham số và các tùy chọn khác. Phương thức này sẽ thực hiện việc điều hướng bằng cách chỉ cập nhật các thành phần của trang mới mà không tải lại toàn bộ trang. Điều này có nghĩa là nó sẽ giữ lại dữ liệu và trạng thái của trang hiện tại và chỉ cập nhật các thành phần mới mà không làm ảnh hưởng đến các thành phần khác trên trang.

Vì vậy, phương thức navigateByUrl được sử dụng khi bạn muốn tải lại toàn bộ trang, còn navigate được sử dụng khi bạn muốn giữ lại dữ liệu và trạng thái của trang hiện tại và chỉ cập nhật các thành phần mới của trang mới.

**10 TypeScript là gì?**

TypeScript là một ngôn ngữ lập trình mã nguồn mở dựa trên JavaScript, được phát triển bởi Microsoft. TypeScript cho phép bạn viết các chương trình JavaScript lớn và phức tạp hơn một cách dễ dàng hơn bằng cách cung cấp kiểu dữ liệu tĩnh, lớp, giao diện, mô-đun, và nhiều tính năng khác để tăng tính bảo mật, sửa lỗi và quản lý mã nguồn.

**11 Tại sao nên sử dụng Typescript, Ưu điểm của Typescript**

Một số lý do nên sử dụng TypeScript bao gồm:

Kiểu dữ liệu tĩnh: TypeScript cho phép bạn xác định kiểu dữ liệu cho các biến, tham số và trả về của các hàm. Việc này giúp giảm thiểu lỗi do kiểu dữ liệu không phù hợp, tăng tính bảo mật và hiệu suất của ứng dụng.

Đa hình và kế thừa: TypeScript hỗ trợ đa hình và kế thừa, giúp bạn tổ chức mã của mình một cách rõ ràng và dễ dàng bảo trì.

Tính mô-đun: TypeScript hỗ trợ các module, cho phép bạn chia mã của mình thành các phần nhỏ và dễ dàng tái sử dụng.

Dễ đọc và bảo trì: TypeScript cho phép bạn viết mã của mình một cách rõ ràng và dễ đọc hơn, với các chức năng như tính năng truy xuất, gợi ý và ghi chú trong mã.

Hỗ trợ tốt cho các IDE và trình biên tập mã: TypeScript cung cấp khả năng gợi ý mã và kiểm tra lỗi cho các trình biên tập mã, giúp giảm thời gian phát triển và tăng tính chính xác của mã.

Một số ưu điểm khác của TypeScript bao gồm tính mở rộng, hỗ trợ tốt cho ES6 và ES7, tính năng mã nguồn mở và cộng đồng lớn của các nhà phát triển.

**12 Làm thế nào để lấy được dữ liệu từ component cha**

**13 Làm thế nào để lấy được dữ liệu từ component con**

**14 Sự khác biệt giữa RouterModule.forChild và RouterModule.forRoot là gì?**

RouterModule.forRoot() được sử dụng để cấu hình routing cho ứng dụng chính, trong khi RouterModule.forChild() được sử dụng để cấu hình routing cho các module con trong ứng dụng.

**15 Tại sao nên sử dụng Angular (điểm mạnh)?**

Angular giúp nâng cao năng suất của các lập trình viên.

Cấu trúc phát triển rõ ràng.

Data binding

Hỗ trợ đầy đủ tính năng điều hướng (routing)

Angular giúp giảm tối đa kích thước và tăng tối đa hiệu suất của ứng dụng.

Có cộng đồng lớn mạnh

**16 Nhược điểm của Angular là gì?**

1. Không an toàn: Thông thường, bản chất của Angular là một trong những Framework Front End, mà Front end này thường vốn không thể bảo mật bằng Back-end. Chính vì vậy, khi sử dụng API thì bạn cần xây dựng cho một hệ thống kiểm tra dữ liệu sao cho việc trả về được tốt nhất.

2. Với một số trình duyệt sở hữu tính năng Disable JavaScript nên có nghĩa là website sẽ không hoàn toàn có thể sử dụng được dựa trên những trình duyệt đó nữa

3. Bạn nên viết mã ứng dụng Angular bằng TypeScript, vì vậy bạn phải học TypeScript.

**17 Angular Cli là gì? Làm sao để sử dụng nó ?**

Angular CLI (Command Line Interface) là một công cụ mạnh mẽ giúp tạo, cấu hình và quản lý dự án Angular một cách dễ dàng và nhanh chóng. Với Angular CLI, bạn có thể tạo một ứng dụng Angular mới, thêm các thành phần mới vào ứng dụng, thực hiện kiểm tra và triển khai ứng dụng một cách dễ dàng

**18 Directive trong angular là gì ?**

**19 Trình bày được khái niệm template trong angular**

Trong Angular, template là một phần quan trọng của các thành phần (component) trong ứng dụng. Nó cung cấp một cách để định nghĩa giao diện người dùng của một thành phần bằng cách sử dụng HTML và các định dạng khác như directive, binding, và pipes. Template của một thành phần có thể được định nghĩa trong một file HTML riêng biệt hoặc trực tiếp trong file TypeScript của thành phần đó bằng cách sử dụng template literals.

**20 Giải thích kiến trúc của Angular**

1. Mã nguồn mở: Angular là một framework mã nguồn mở, cho phép các nhà phát triển tạo ra các ứng dụng web đáng tin cậy, linh hoạt và dễ bảo trì.
2. TypeScript: Angular được viết bằng TypeScript, một ngôn ngữ lập trình mã nguồn mở có thể biên dịch thành JavaScript. TypeScript cho phép các nhà phát triển viết mã hiệu quả hơn và giúp giảm thiểu lỗi trong khi phát triển.
3. Dễ sử dụng: Angular cung cấp các công cụ, thư viện và khung thời gian mạnh mẽ giúp các nhà phát triển tạo ra các ứng dụng web phức tạp một cách dễ dàng.
4. Mô hình MVC: Angular sử dụng mô hình MVC (Model-View-Controller), giúp tách biệt các phần khác nhau của ứng dụng và làm cho mã của bạn dễ bảo trì hơn.
5. Binding dữ liệu hai chiều: Angular hỗ trợ binding dữ liệu hai chiều, cho phép các nhà phát triển liên kết dữ liệu từ model của mình đến giao diện người dùng, giúp tạo ra các ứng dụng động và phản hồi nhanh chóng.
6. Tích hợp nhiều tính năng: Angular cung cấp nhiều tính năng như routing, dependency injection, form validation và testing, giúp giảm thời gian phát triển và cải thiện chất lượng sản phẩm.
7. Hỗ trợ đa nền tảng: Angular hỗ trợ phát triển ứng dụng web đa nền tảng, bao gồm cả web và di động, giúp các nhà phát tri

**21Hãy mô tả khái niệm Databinding One Way & two way phân biệt sự khác nhau?**

**22 ngModel là gì?**

Ng-model là một directive dùng để liên kết dữ liệu với client, một số người gọi nó là ngModel. Người ta thường dùng nó trong các form html để cho user nhập liệu.

**23 Component là gì ? các thành phần của component**

Trong Angular, một Component là một thành phần cơ bản của ứng dụng, được sử dụng để hiển thị dữ liệu trên giao diện người dùng. Mỗi Component bao gồm một class TypeScript, một template HTML và một stylesheet CSS hoặc SCSS. Các thành phần của một Component bao gồm:

Class: Là một lớp TypeScript, chứa các thuộc tính và phương thức để xử lý dữ liệu và logic của Component.

Template: Là một đoạn mã HTML, chứa các thẻ HTML, các directive Angular và các biểu thức để hiển thị dữ liệu và xử lý sự kiện trên giao diện người dùng. Stylesheet: Là một tập tin CSS hoặc SCSS, chứa các quy tắc CSS để tạo kiểu cho các thẻ HTML trong template.

Metadata: Là một đối tượng trong phần decorators của class, chứa các thông tin về Component như tên, selector, templateUrl, styleUrls, providers,...

**24 Có thể chuyển mã TypeScript thành mã Javascript không, bằng cách nào?**

**25 Mô tả vòng đời component?**

Phương thức ngOnChanges được gọi khi component phát hiện có giá trị được binding vào component bằng phương pháp Input properties

Phương thức NgOninit được gọi, khi component được tạo lần đầu tiên. Chúng ta được chạy sau khi hàm constructor và hàm ngOnchange được thực hiện.

Phương thức ngOnDoCheck được gọi mỗi khi nó phát hiện ra có sự thay đổi dữ liệu ở component.

Phương thức ngOnDestroy được gọi trước khi component hoặc directive bị phá huỷ bởi Angular.

**26 Pipe là gì?**

Pipe được dùng để chuyển đổi dữ liệu mà bạn hiển thị lên template cho người dùng có thể hiểu được.

AsyncPipe

CurrencyPipe

DatePipe

DecimalPipe

JsonPipe

LowerCasePipe

PercentPipe

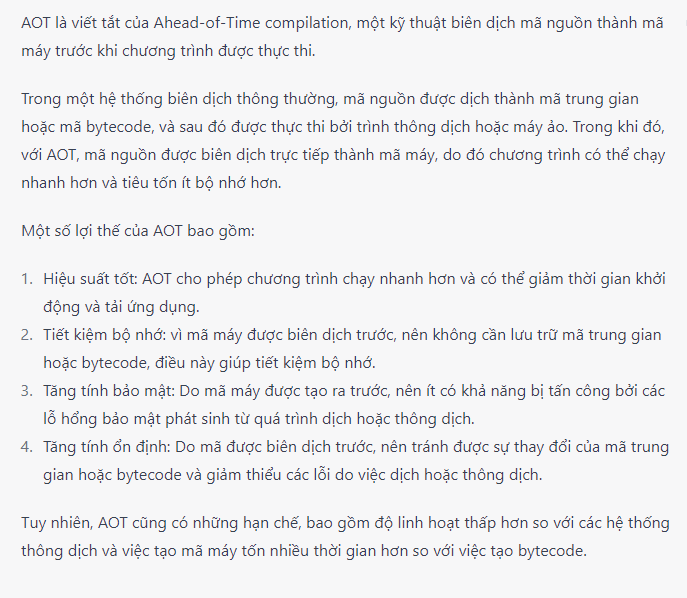
SlicePipe

TitleCasePipe

UpperCasePipe

**27 Phân biệt sự khác nhau giữa Promise và Observable**

**28 Biên dịch AOT là gì? Những lợi thế của AOT?**

****

**29 Trình bày khái niệm Front-end & backend?**

Front-end (hay còn gọi là client-side) là phần của ứng dụng mà người dùng cuối sử dụng để tương tác với ứng dụng. Nó bao gồm giao diện người dùng, các chức năng tương tác với người dùng và các dữ liệu cần thiết để hiển thị trên trình duyệt của người dùng. Front-end thường được phát triển bằng các ngôn ngữ lập trình web như HTML, CSS và JavaScript, và các framework như React, Angular hoặc Vue.js để giúp quản lý mã lập trình và tạo giao diện người dùng. Back-end (hay còn gọi là server-side) là phần của ứng dụng chạy trên máy chủ. Nó quản lý việc lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu, xử lý các yêu cầu từ phía client-side, và trả về các kết quả được yêu cầu.

Back-end thường được phát triển bằng các ngôn ngữ lập trình như Java, Python, Node.js, Ruby on Rails, và các framework như Spring, Django, Express, Ruby on Rails để giúp quản lý mã lập trình và quản lý cơ sở dữ liệu. Khi phát triển một ứng dụng web, front-end và back-end cùng là các phần quan trọng và phải được đồng bộ hóa với nhau. Các yêu cầu từ phía người dùng được gửi đến back-end để xử lý, và các kết quả được trả về cho front-end để hiển thị. Việc phát triển hai phần này đồng thời đòi hỏi kiến thức về nhiều ngôn ngữ lập trình, các framework, cơ sở dữ liệu và các công nghệ khác liên quan đến phát triển web.

**30 Làm sao để cài đặt TypeScript?**

npm install -g typescript

**31 Để khởi tạo một dự án angular sử dụng câu lệnh nào**

**32 Để chạy một dự án angular sử dụng câu lệnh nào**

**33 Kể tên các kiểu dữ liệu trong Typescript**

**34 Kể tên các vòng lặp được typescript hỗ trợ**

**35 ngFor là gì? Cách sử dụng**

**ngFor** là một directive trong Angular framework được sử dụng để lặp lại một danh sách các phần tử và tạo ra các thành phần của giao diện người dùng tương ứng với từng phần tử trong danh sách.

**36 ngIf là gì? Cách sử dụng**

**ngIf** là một directive trong Angular framework được sử dụng để hiển thị hoặc ẩn một phần tử HTML trong giao diện người dùng dựa trên một điều kiện logic.

**37 Các cách tạo form trong Angular**

**38 router trong angular dùng để làm gì?**

**39 Cách để lấy tham số trên URL trong angular**

**40 làm sao để kiểm tra người dùng có quyền truy cập vào 1 route?**

Trong Angular, có thể kiểm tra quyền truy cập vào một route bằng cách sử dụng Route Guards. Route Guards là các lớp được sử dụng để bảo vệ và kiểm tra tính hợp lệ của các route.

Có ba loại Route Guards trong Angular:

CanActivate: Kiểm tra xem người dùng có quyền truy cập vào route hay không. CanActivateChild: Kiểm tra xem người dùng có quyền truy cập vào các route con của một route cha hay không.

CanLoad: Kiểm tra xem người dùng có quyền tải các module lazy-loaded hay không.

**41 lazy-loading là gì?**

Lazy Loading được hiểu nôm na chính là việc load dữ liệu khi bạn cần sử dụng đến chúng. Ví dụ như khi nhiều người cùng vào 1 page sẽ không kéo xuống hết đến cuối trang để đọc nội dung thì điều bạn cần làm là load nội dung trước

Việc nên làm đó chính là khi người dùng scroll đến đâu bạn sẽ load dữ liệu đến đó. lazy loading có thể áp dụng cho bất cứ resource nào trên 1 page, thậm chí là cả file JavaScript.