1. Trình bày được ý nghĩa và khái niệm Observer:

* Là thể hiện ngModchứa tập 3 call back :
  + Next: Nhận giá trị trả về dưới trạng thái http status là 2xx
  + Error: Nhận giá trị trả về dưới trạng thái http status là 4xx 5xx (lỗi)
  + Complete: được gọi dù có next hay error

1. Trình bày được ý nghĩa và khái niệm Observable

* Observable là đại diện cho khái niệm về 1 tập hợp giá trị hoặc các sự kiện trong tương lai. Khi các đối tượng trên phát sinh thì Observable sẽ điều phối đến Observer

1. Phân biệt sự khác nhau giữa var, let, const:

|  |  |  |
| --- | --- | --- |
| Var | Let | Const |
| * Có thể sử dụng trước khi khai báo theo cơ chế hoisting * Có thể khai báo lại * Phạm vi sử dụng là function Scope | * Phạm vi sử dụng trong 1 khối lệnh * Khai báo xong với đc sử dụng * Không thể khai báo lại | * Khai báo hằng * Không được thay đổi giá trị vùng nhớ mới * Khi khai báo thì phải đi kèm giá trị * Phạm vi sử dụng trong 1 khối lệnh |

1. File angular.json và package.js sử dụng để làm gì?:

* angular.json: Dùng để cấu hình lại Angular.cli
* package.js: Chứa các thư viện cần thiết cho angular, ngoài ra cũng dùng để chứa các thư viện thêm vào

1. Kể tên các directive được hỗ trợ trong angular

* Structure directive: là các directive làm thay đổi cấu trúc của thẻ html, dùng để thực hiện các thao tác CRUD, thường hay có \* ở đầu :
  + \*ngIf, \*ngFor, \*ngSwitch, \*ngCase
* Attribute directives: Thay đổi giao diện, tương tác của các đối tượng hoặc thay đổi directive khác hoặc thêm các thuộc tính động cho element html. ví dụ \*ngStyle, \*ngClass.
* Components directives: Không có nghi ngờ gì khi gọi component là directive cũng được, vì rõ ràng là component cho phép định nghĩa selector và gọi ra như một thẻ html tag (<component-name></component-name>)
* Custom directive: directive do người dùng tự tạo

1. Trình bày được ý nghĩa và khái niệm subcribe
2. Trình bày khái niệm Single page application:

* Là 1 kiểu thiết kế web mà theo đó các thao tác CRUD, thay đổi dữ liệu đều thực hiện trên 1 trang web
* Đem lại trải nghiệm mượt mà cho người dùng (không bị khựng lại khi chuyển trang)

1. Trình bày về Data binding trong component

* Là kỹ thuật hỗ trợ tương tác giứa các component:
* Có 2 loại chính:
  + One way – binding:
    - Interpolation: Binding biến từ file ts sang html với cú pháp{{biến}}
    - Property Binding: Biding thuộc tính cho thẻ bằng cách truyền cho nó 1 giá trị được thể hiện ở trong component. Cú pháp [thuộc tính]
    - Event Binding: Binding dữ liệu từ html về component bằng các sử dụng 1 số sự kiện do người dùng thao tác như click, change, mouse over,.,
  + Two ways – bingding:
    - Dùng để binding dữ liệu song song giữa component và view. Khi thay đổi giá trị trên view thì component cũng thay đổi theo và ngược lại. Sử dụng [(ngModel)]

9 Trình bày sự khác nhau giữa navigateByUrl và navigate

10 TypeScript là gì?

* Là phiên bản nâng cấp (superSet) của JS
* Được phát triển bởi Microsoft

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

* Support được nhiều tính năng trong bản ECMA Script
* Hỗ trợ Debug
* Dễ dàng có gợi ý code trong các tool IDE
* Tạo sự nhất quán trong coding style
* Hỗ trợ ép kiểu
* Có 1 cộng đồng giúp hỗ trợ những vấn đề liên quan
* IDE, library Framework

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

* Tạo 1 biến dữ liệu ở component cha
* Đặt @input + “attributes” vào component con để nhận dữ liệu truyền vào
* Trong app-component ta truyền dữ liệu từ attributes của con = attributes khai báo ở component cha

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

* Import thư viện Even Mitter từ Angular Core
* Đặt @output + “attributes” vào component con để nhận dữ liệu truyền vào rồi gán cho = new EvenMitter
* Dùng phương thức .emit để đẩy dữ liệu vừa hứng
* ở component nhận giá trị qua đối tượng $event

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

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

* Angular giúp thiết kế ứng dụng trên 1 trang giúp xử lý người dùng được mượt mà
* Độ phổ biến cao, cộng đồng người dùng lớn => support tốt
* Được phát triển bởi Google, một tập đoàn công nghệ lớn
* Giảm chi phí
* Mã nguồn mở
* Xây dựng một bộ tiêu chuẩn chung
* Document rất chi tiết và cụ thể

16 Nhược điểm của Angular là gì?

* Được viết trên ngôn ngữ JS. Khi người dùng tắt JS thì sẽ không hiển tị đc

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

* Npm install –g@ angular/cli

18 Directive trong angular là gì ?

* Là thành phẩn mở rộng cho các thuộc tính trong thẻ html do Angular định nghĩa thêm. Mục đích là để thực hiện các thao tác CRUD

19 Trình bày được khái niệm template trong angular

* Là một dạng HTML giúp cho Angular hiển thị component cho người dùng

1. Giải thích kiến trúc của Angular:

* Angular chia dự án thành nhiều Module: Dùng để thực hiện 1 chức năng nào đó. Lập trình viên chia module làm 2 hướng:
  + - Common: Những module dùng chung: tính toán, đọc ghi file
    - Theo 1 tiêu chí nào đó
* Module bao gồm
  + Component:
    - Thành phần thấy được trên view mà làm 1 nghiệp vụ nào đó
    - Bao gồm Template, metadata, component
  + Ngoài ra còn có Directive, thay đổi cấu trúc của template
  + Injection dùng để DI service vào class
  + Service&DI:
  + Để component hiển thị được thì phải gọi data từ server
  + Server sẽ thực hiện DI vào component
* Mô hình MVVM

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

* One way – binding:
  + Interpolation: Binding biến từ file ts sang html với cú pháp{{biến}}
  + Property Binding: Biding thuộc tính cho thẻ bằng cách truyền cho nó 1 giá trị được thể hiện ở trong component. Cú pháp [thuộc tính]
  + Event Binding: Binding dữ liệu từ html về component bằng các sử dụng 1 số sự kiện do người dùng thao tác như click, change, mouse over,.,
    - Onclick: hỗ trợ bởi JS
    - Click: hỗ trợ bởi Angular
* Two ways – bingding:
  + Dùng để binding dữ liệu song song giữa component và view. Khi thay đổi giá trị trên view thì component cũng thay đổi theo và ngược lại. Sử dụng [(ngModel)]

22 ngModule là gì?:

* Angular app chia ra các Module => đc gọi là ngModule
* Bao gồm: Component, pipe, direcctive, service, provide và 1 số thành phần khác tự định nghĩa
* Export, Import, provides, bootraps, declaration

23 Component là gì ? các thành phần của component

* Thành phần thấy được trên view mà làm 1 nghiệp vụ nào đó
* Các thành phần :
  + Template : file html dùng để hiển thị
  + Class : file ts bao gồm các thuộc tính và phương thức dùng để xử lý logic, bussiness
  + Meta data :
    - Các thông tin bổ sung cho component
    - Khai báo selector cho view
    - URL đến template hiện tại
    - URL đến file css của component hiện tại

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

* Có thể chuyển mã TypeScript thành mã Javascript bằng cách:
  + Trong terminal sử dụng lệnh tsc
  + Trực tiếp trong Visual Studio hoặc các IDE và trình soạn thảo khác
  + Sử dụng các task runner như Gulp

25 Mô tả vòng đời component?

* ngOnchanges : Gọi khi comp thiết lập giá trị đầu vào bằng phương pháp input properties, điển hình khi binding dữ liệu từ cha => con
* ngOninit : gọi sau ngOnchanges
* ngOnDoCheck : gọi ra khi component có sự thay đổi bao 4 bước
  + ngAfterContentInit : gọi khi component đc khởi tạo
  + ngAfterContentCheck : Sau khi component được khởi tạo thành công
  + ngAferViewInit: Gọi khi khởi tạo view của component
  + ngAferViewChecked: gọi khi view đc khởi tạo thành công
* ngOnDisplay:
  + Được gọi trước khi component bị phá hủy

26 Pipe là gì?

* Dùng để định dạng kiểu hiển thị trên html
* Một vài pipes thường dùng :
  + UpperCase, LowerCase,..

27 Phân biệt sự khác nhau giữa Promise và Observable :

|  |  |  |
| --- | --- | --- |
|  | Promise | Oberserver |
| Hủy request | x | o |
| Retry | x | o |
| Xử lý kết quả khi trả về | x | o |
| Trả về nhiều kết quá | x | o |

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

29 Trình bày khái niệm Front-end & backend

* Phần front-end của một trang web là phần tương tác với người dùng. Tất cả mọi thứ bạn nhìn thấy khi điều hướng trên Internet, từ các font chữ, màu sắc cho tới các menu xổ xuống và các thanh trượt, là một sự kết hợp của HTML, CSS, và JavaScript được điều khiển bởi trình duyệt máy tính của bạn.
* Phần back end của một trang web bao gồm một máy chủ, một ứng dụng, và một cơ sở dữ liệu. Một lập trình viên back-end xây dựng và duy trì công nghệ mà sức mạnh của những thành phần đó, cho phép phần giao diện người dùng của trang web có thể tồn tại được.

30 Làm sao để cài đặt TypeScript:

* Cài Nodejs -.> NPM -> npm install typeScript

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

* Ng g new tên\_dự\_án

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

* Ng s --o

33 Kể tên các kiểu dữ liệu trong Typescript

|  |  |
| --- | --- |
| Kiểu nguyên thủy | Kiểu đối tượng |
| * Number * String * Boolean * Undefined * Any | * Object * Mảng * Enum: value-code * Tuple * Interface * Tuple: kiểu dữ liệu giống như mảng nhưng có ràng buốc số phần tử và kiểu dữ liệu |

34 Kể tên các vòng lặp được typescript hỗ trợ

* Ngoài các vòng lặp được JS hỗ trợ, TS còn hỗ trợ:
  + For index, while, do-while, for of

35 ngFor là gì? Cách sử dụng

* Chúng ta sử dụng ngFor để duyệt qua các tập hợp dữ liệu như array, list. Sau đó ta tạo các thành phần web tương ứng với mỗi vòng lặp.
* Dùng \*ngFor = “let obj of tên\_mảng” trong thẻ html

36 ngIf là gì? Cách sử dụng

* Chúng ta sử dụng ngIf để xóa hoặc thêm một phần tử trên web dựa vào điều kiện có thoả mãn hay không.
* \*ngif=”điều kiện”

37 Các cách tạo form trong Angular

* Template form: Nên sử dụng khi ít validate, đăng nhập
  + Import Form Module
  + ngForm: Dùng để khai báo form
  + ngModel: Khai bác 1 control trong form
  + [(ngModel)]: để binding dữ liệu
* Reactive Form:
  + Import reactive Form module
  + Khai báo form-group ở file ts
  + Khai báo các form control trong form group (các thuộc tính của đối tượng)
  + Khai báo và sử dụng form ở HTML

38 router trong angular dùng để làm gì?

* Quy định component nào sẽ đc hiển thị trong router outlet

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 ?

41 lazy-loading là gì?

* Lazy loading là 1 kĩ thuật tối ưu khi làm web, thay vì tải toàn bộ trang web và render ngay từ đầu, kỹ thuật này cho phép tải ngay các thành phần cần thiết để hiển thị tới người dùng và trì hoãn các tài nguyên còn lại cho đến khi cần.