1. TypeScript là gì?

* TS là 1 superset của JS, được hỗ trợ rất mạnh, khắc phục được nhược điểm cua JS kiểu dữ liệu, hướng đối tượng,….

1. Tại sao nên sử dụng Typescript, Ưu điểm của Typescript

* Hỗ trợ nhiều tính năng trong các bản ES mới
* Hệ thống type tốt giúp người dùng biết trước các bug có thể phát sinh
* Gợi ý code
* Hỗ trợ generic
* Coding style tạo sự nhất quán trong các team đông người

1. Làm sao để cài đặt TypeScript

* Cài đặt gói nodejs ( môi trường biên dịch ts)
* Cài đặt ts lệnh : npm install –global typescript

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

Transpile TS ( transpile biên dịch 1 ngôn ngữ cùng họ hàng, khác complie biên dịch từ nn này sang nn khác) :

* Tạo file.ts
* Terminal/cmd vào thư mục chứa file.ts
* Gõ lệnh tsc tên file.ts để transplie ra file.js
* Gõ lệnh node tên file.js chạy file js

1. Kể tên các kiểu dữ liệu trong Typescript

* TS bao gồm các kiểu dữ liệu mà JS có (string, number, undefined,…) tuy nhiên ngoài ra còn có thêm các kiểu dữ liệu mới
* Tuple : khai báo mảng với các giá trị có kiểu dự liệu đã biết
* Interface : khai báo về cấu trúc của 1 đối tượng, mô tả thuộc tính và phương thức của đối tượng đó
* Enum : 1 biến có thể là tập hợp các hằng số đc định sẵn
* Any : có thể ap dụng bất kỳ kiểu dữ liệu nào khi ko biết chắc kiểu dữ liệu

1. Kể tên các vòng lặp được typescript hỗ trợ

* For
* For of : trả lại từng phần tử của tập hợp
* For in : duyệt qua từng phần tử của mảng và trả về vị trí các index của các phần tử trong mảng
* While
* Do-while

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

* Giống nhau : đều dùng để khai báo biến
* Khác nhau :

|  |  |  |
| --- | --- | --- |
| **Var** | **Let** | **Const** |
| Global /Function scope | Block scope {} | Block scope{} |
| Có thể sử dụng trước khi khai báo | Không thể sử dụng trước khi khai báo | Khong thể sử dụng trước khi khai báo |
| Có thể được cập nhật và khai báo lại trong phạm vị tồn tại | Có thể được cập nhật, nhưng không thể khai báo lại | Không thể cập nhật không thể khai báo lại |
| Khai báo có thể ko gán giá trị | Khai báo có thể ko gán giá trị | Phải có giá trị khi khai báo |

1. Trình bày khái niệm Single page application

* Single page application là ứng dụng 1 trang web, các thao tác đều được xử lý trên 1 trang duy nhất

1. lazy-loading là gì?

* Lazy-loading là kỹ thuật trì hoãn các tài nguyên không quan trọng vào thời điểm tải trang. Thay vì tải ngay lập tức các tài nguyên này chỉ tải vào các thời điểm khi cần thiết

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

* Khởi tạo dự án angular dùng câu lệnh : ng new tên-dự-án

1. Để chạy một dự án angular sử dụng câu lệnh nào

* Chạy dự án angular sử dụng câu lệnh : ng s-o hoặc ng s hoặc click vào nút 

1. Tại sao nên sử dụng Angular (điểm mạnh)?

* Custom component : cho phép người dùng xây dựng các component riêng và có thể đóng gói chức năng với logic kết xuất thành các phần có thể tái sử dụng
* Data binding : dễ dàng di chuyển dữ liệu, liên kết dữ liệu từ js sang view và phản hồi sự kiện từ người dùng
* Dependency injection : cho phép viết các service module và đưa vào sử dụng ở bất kỳ nơi nào cần, cải thiện khả năng kiểm tra và tái sử dụng các service có nhiệm vụ giống nhau, giảm sự phụ thuộc
* Testing : dễ dàng thực hiện việc kiểm thử các phần trong ứng dụng
* Browser compatibility : cung cấp đa nền tảng, tương thích với nhiều trình duyệt

1. Nhược điểm của Angular là gì?

* Không an toàn : angular là Framework FE, mà bản chất thì FE không bảo mật bằng BE
* Không áp dụng cho 1 số trình duyệt có tính năng disable javascript
* Phải có kiến thức nền tảng về ts

1. Angular Cli là gì? Làm sao để sử dụng nó ?

* Angular Cli là là công cụ dòng lệnh giúp tạo ra các ứng dựng angular 1 cách dễ dàng
* Cách sử dụng :
* Cài bằng lệnh : npm install -g @angular/cli@latest
* Tạo dự án angular bằng ng new
* Sử dụng ng g để tạo component, module,… trong cửa sổ terminal

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

* Angular.json : cấu hình và quản lý các thiết lập cho angular
* Package.js : tệp quản lý gói phụ thuộc cho angular

1. Giải thích kiến trúc của Angular

* Components : được xây dựng từ HTML, CSS, TS
* Services : là các lớp chuyên mon độc lập, được sử dụng để thực hiện các chức năng cụ thể trong ứng dụng
* Modules : là các gói code đc tổ chức thành các đơn vị logic và chức năng riêng biệt

1. Angular đi theo mô hình nào? Phân biệt mô hình MVVM và MVC

* Lúc đầu angularjs đi theo mô hình MVC. Sau này angularjs không triên khai MVC nữa mà gắn liền với mô hình Model-View-ViewModel
* MVC :
* Model : biểu diễn dữ liệu, xử lý nghiệp vụ, kết nối db
* View : hiển thị các thành phần giao diện, dữ liệu và tương tác với người dùng
* Controller : điều hướng và xử lý request, là cầu nối giữa View và Model
* MVVM :
* View : giống MVC
* Model : giống MVC
* ViewModel :cung cấp dữ liệu và xử lý các sự kiện cho View. Là lớp trung gian giữa View và Model đc xem là tp thay thế cho Controller trong MVC.

1. Component là gì ? các thành phần của component

* Component là thành phần đẻ xử lý và hiển thị dữ liệu đến người dùng
* Gồm có 3 tp chính :
* Template : hiển thị giao diện đến người dùng
* Class (properties + methods) : giúp liên kết với view
* Metadata : cung cấp thông tin bổ sung và định nghĩa class này angular component

1. Mô tả vòng đời component?

* Contructor ( không nằm trong vòng đời component ) :
* ngOnChanges : được gọi khi thuộc tính đầu vào có sự thay đổi
* ngOnInit : gọi khi khởi tạo component
* ngDoCheck : gọi khi component có sự thay đổi
* ngOnDestroy : gọi khi huỷ component

1. Trình bày được khái niệm template trong angular

* Trong angular, template là 1 phần của component, được sử dụng để hiển thị nội dung
* Template định nghĩa trong file html

1. Biên dịch AOT là gì? Những lợi thế của AOT ?

* Ahead-of-Time (AOT) :là kỹ thuật biên dịch toàn bộ ứng dụng angular thành js trước khi chạy ứng dụng
* Lợi thế :
* Tốc độ nhanh hơn : code đc biên dịch sẵn nên ko cần biên dịch lại
* Phát hiện lỗi tốt hơn : ktra lỗi trong quá trình biên dịch giảm thời gian fix
* Kích thước ctrinhf nhỏ hơn : nhg tính năng ko sd có thể ko đc biên dịch

1. Trình bày về Data binding trong component

* One way binding :
* Interpolation {{}} com -> dom
* Property binding [] com -> dom
* Event binding () dom -> com
* Two way binding [(ngModel)]

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

* One way binding là dữ liệu đc truyền theo 1 chiều. có thể từ view sang component hoặc ngược lại
* Two way binding là dữ liệu được truyền theo 2 chiều từ component sang view và ngược lại từ view

1. Directive trong angular là gì ?

* Directivelaf 1 thành phần trong angular được dùng tăng sức mạnh và mở rộng tính năng cho HTML
* Giúp người dùng có thể thay đổi cấu trúc, thuộc tính các thẻ html

1. Kể tên các directive được hỗ trợ trong angular

Có 3 loại directive

* Attribute : ngClass, ngStyle
* Stuctural : \*ngIf, \*ngFor, \*ngSwitch
* Custom : highliter, dropdown

1. Làm thế nào để lấy được dữ liệu từ component cha

* B1 : đặt @Input cho biến x cần nhận dữ liệu của component con
* B2 : sử dụng directive [x] = “value” trong selector của component con để binding dữ liệu

1. Làm thế nào để lấy được dữ liệu từ component con

* B1 : đặt @Output và Emitter ở con
* B2 : Emit giá trị từ con sang cha
* B3 : cha lấy giá trị từ con thông qua $event

1. Pipe là gì? Kể tên 1 số pipe thường sử dụng

* Pipe là định dạng dữ liệu, chuyển đổi dữ liệu hiển thị lên template cho người dùng cso thể hiểu được
* Vd: titleCase (hoa chữ cái đầu), currency (đơn vị tiền tệ), upperCase(in hoa toàn bộ)

1. ngFor là gì? Cách sử dụng

* ngFor là directive dùng để lặp dựa trên 1 mảng dùng để hiển thị dữ liệu
* let item of items, index as i

1. ngIf là gì? Cách sử dụng

* ngIf là directive dùng để hiển thị hoặc ẩn 1 phần tử html dựa trên điều kiện

1. Các cách tạo form trong Angular

* Có 2 cách tạo form trong angular là Template Driven Form và Reactive Form (Model Driven Form)

1. Phân biệt Template driven form và Reactive form

|  |  |
| --- | --- |
| **Template** | **Reacitve** |
| Tạo form từ template truyền dữ liệu cho file ts | Tạo form từ ts sau đó binding dữ liệu với template |
| Import module là FormModule | Import là ReactiveFormModule |
| Async | Sync |

1. Trình bày 1 số validate thường dùng. Cách custom validate như thế nào?

* Dùng các validators có sẵn như required, maxLength, minLength,pattern,…
* Dùng custom validate
* Tạo custom validator function
* Validate trong form ở component
* Hiển thị lỗi ở template

1. router trong angular dùng để làm gì?

* Router là đối tượng cho phép xác định component nào được hiển thị

1. Cách để lấy tham số trên URL trong angular

* Dùng ActivetedRote để lấy tham số ở trên URL

1. làm sao để kiểm tra người dùng có quyền truy cập vào 1 route ?
2. Trình bày sự khác nhau giữa navigateByUrl và navigate

|  |  |
| --- | --- |
| navigateByUrl | navigate |
| Được sử dụng để điều hướng tuyệt đối, thay đổi trực tiếp thanh vị trí, cung cấp 1 URL mới “hoàn toàn” | Được sử dụng để điều hướng tương đối, tạo ra 1 URL mới bằng cách áp dụng các lệnh được truyền, thêm vào url gốc |
| Ví dụ: muốn điều hướng từ abc.com/user đến abc.com/assets | Ví dụ: nếu đường dẫn hiện tại là abc.com/user , muốn điều hướng đến abc.com/user/10 |

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

* forRoot và forChild đều đc dùng để config routes
* forRoot chỉ được gọi 1 lần duy nhất trong AppRoutingModule
* forChild được gọi trong các module khác

1. Trình bày được ý nghĩa và khái niệm Observable

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

1. Trình bày được ý nghĩa và khái niệm Observer

* Observer là 1 tập hợp các callbacks tương ứng cho việc lắng nghe các giá trị ( next, error hay complete) được gửi đến từ Observable

1. Phân biệt sự khác nhau giữa Promise và Observable

|  |  |
| --- | --- |
| Observable | Promise |
| Có thể trả về nhiều giá trị đồng thời | Chỉ trả về 1 giá trị duy nhất |
| Có thể huỷ request | Không thể huỷ request |
| Có thể retry request | Không thể retry request |
| Có thể xử lý kết quả trước khi trả về | Không thể xử lý kết quả trước khi trả về |

1. Trình bày được ý nghĩa và khái niệm subcribe

* Phương thức subscribe theo dõi các thay đổi của dữ liệu hoặc nhận về kết quả của một công việc nào đó
* Observable quả lý các đối tượng subscribes, khi có sự thay đổi thì observable sẽ thông báo đến các subscribers của mình

1. ngModule là gì?

* Angular app đc chia thành các module, được gọi là ngmodule
* Ngmodule có thể chứa: pipe, component, service, directive
* Ngmodule có thể import các chức năng khác từ ngmodule khác và có thể export 1 số chức năng mà nó quản lý để ngmodule khác sử dụng
* Mỗi 1 angular app đều có ít nhất 1 ngmodule nó gọi là rootmodule và thông thường được đặt tên AppModule trong file app.module.ts
* Bản thân angular cũng chia các chức năng thành các module nhỏ như : ReactiveFormsModule, RouterModule

1. RxJS

* RxJS là 1 library cung cấp xủ lý đa luồng hoặc bất đồng bộ
* Được tích hợp sẵn
* Có nhiều tính năng hữu ích như event emiiters, animation, forms, http request