1. **What is Angular?**

It is a TypeScript-based open-source front-end web application framework.

1. **What is the Angular CLI?**

Angular cli is a command line interface to scaffold and build angular apps using nodejs style (commonJs) modules.

1. **Where is the app folder located? What files does it contain if the project was created by the Angular CLI?**

The app folder is located within the project's src folder. It contains files like: app.module.ts, app.component.css, app.component.html, app.component.ts

1. **What is a decorator?**

It is a special kind of declaration that can be attached to a class declaration, method, accessor, property, or parameter. It allows us to decorate classes and functions, similar to annotations in java and decorators in python.

1. **What is a module?**

In Angular, a module is a mechanism to group components, directives, pipes and services that are related, in such a way that can be combined with other modules to create an application. An Angular application can be thought of as a puzzle where each piece (or each module) is needed to be able to see the full picture.

1. **What is a component?**

Components are the most basic UI building block of an Angular app.

1. **What is @component? What are some attributes you can find inside the decorator?**

It is a decorator that is added to the top of the component's class definition inside its component.ts file. Some attributes are the selector, templateUrl and stylesUrls.

1. **How would you add a new component without the Angular CLI?**

Create the component.ts file inside the src folder.

Add the "@component" decorator

Import the component into the app.module.ts file.

Add a reference to the imported component into the "@NgModule" declarations.

1. **What is a directive? What are the different types?**

They are custom markers on a DOM element that tell Angular to attach behavior to the element.

The types are structural directives, attribute directives, and component directives.

1. **What is data binding? What are some different types?**

Data binding is a core concept in Angular and allows to define communication between a component and the DOM, making it very easy to define interactive applications without worrying about pushing and pulling data. There are four forms of data binding: interpolation {{value}}, property binding [property]="value", event binding (event)="function", two-way data binding [(ngModel)]="value".

1. **What is a pipe in Angular? How do you create a custom pipe?**

It is a way to write display-value transformations that you can declare in your HTML. You create a custom pipe by using the @pipe() on a class, implementing the PipeTransform. To you use it, you must also add it to a module.

1. **What is a SPA?**

Single-Page Applications (SPAs) are Web apps that load a single HTML page and dynamically update that page as the user interacts with the app. SPAs use AJAX and HTML5 to create a fluid and responsive Web apps, without constant page reloads.

1. **What is a service in Angular?**

Angular services are singleton objects which get instantiated only once during the lifetime of an application. They contain methods that maintain data throughout the life of an application, i.e. data does not get refreshed and is available all the time. The main objective of a service is to organize and share business logic, models, or data and functions with different components of an Angular application.

1. **What is routing? What is it used for?**

Angular implements its own routing module in order to allow for single-page web applications. It is used to build SPAs.

1. **How do you perform HTTP requests in Angular?**

Import the HttpClient module and include it in the ngModule imports.

Inject an object of type HttpClient inside of the constructor of the class you want to make an HttpRequest in.

Use the HttpClient property in your object and call one of the HttpRequest functions, such as get() or post().

1. **What is the difference between a promise and an observable?**

Observables do not execute until subscription, while promises execute upon creation. Observables are cancellable, while Promises are not. Promises return only one value and observables return many.