1. Why were client-side frameworks like Angular introduced?

Client-side frameworks like Angular were introduced to simplify the development of complex and dynamic web applications. These frameworks provide a structured approach to building web applications and allow developers to easily manage application logic and data, while also providing a consistent user experience across different devices and browsers.

2. How does an Angular application work?

An Angular application works by using a combination of HTML templates and TypeScript code to create a dynamic, single-page web application. The application is organized into modules, each of which contains components, services, and other supporting code. Components represent individual parts of the user interface, while services provide application logic and data management. When the application is loaded, Angular bootstraps the root module and begins to render the UI based on the component templates and data provided by the services.

3. What are some of the advantages of Angular over other frameworks?

Some of the advantages of Angular over other frameworks include its powerful data binding system, comprehensive set of built-in features and tools, and its strong community support. Angular also provides excellent support for building large-scale enterprise applications, with features like dependency injection and module loading.

4. What are the advantages of Angular over React?

Some advantages of Angular over React include its more comprehensive feature set and out-of-the-box support for testing, as well as its more opinionated approach to application architecture, which can make it easier for developers to get started and follow best practices. However, React is generally considered to be more lightweight and flexible, with a simpler and more intuitive API.

5. List out differences between AngularJS and Angular?

AngularJS (also known as Angular 1.x) is an older version of the Angular framework, which was rewritten from the ground up in Angular 2+. Some of the key differences between AngularJS and Angular include:

* Angular uses a component-based architecture, while AngularJS uses a controller-based architecture
* Angular uses TypeScript for type checking and provides better tooling support, while AngularJS uses plain JavaScript
* Angular has a more powerful and flexible data binding system, while AngularJS uses a two-way binding system that can be less efficient
* Angular provides better support for server-side rendering and mobile development, while AngularJS is more focused on traditional web applications

6. How are Angular expressions different from JavaScript expressions?

Angular expressions are similar to JavaScript expressions, but they are evaluated within the context of an Angular application and have some additional features. For example, Angular expressions support data binding, which allows values to be automatically updated when they change in the application. Angular expressions also support filters, which can be used to format data or transform it in other ways. Finally, Angular expressions are evaluated within a context that includes additional properties and functions provided by Angular, such as $scope and $http.

7. What are Single Page Applications (SPA)?

Single Page Applications (SPA) are web applications that dynamically update the content of a single web page instead of loading multiple pages from a server. SPAs are built using client-side frameworks like Angular, React, and Vue, and use JavaScript to manipulate the DOM and update the UI. SPAs provide a more seamless and interactive user experience, as users can interact with the application without having to wait for pages to reload.

8. What are templates in Angular?

Templates in Angular are HTML files that define the user interface of a component. Templates are used to render data and display the components in the application. Templates can contain HTML markup, Angular directives, and expressions that allow data to be bound to the UI. Templates are a key part of the Angular framework and help to separate the UI logic from the application logic.

9. What are directives in Angular?

Directives in Angular are markers in the DOM that tell Angular to attach specific behavior to an element or component. Directives can be used to add custom behavior to existing HTML elements, or to create new, reusable components. Angular provides a set of built-in directives that can be used to manipulate the DOM, handle user input, and perform other tasks. Directives can also be created and customized to meet the specific needs of an application.

10. Explain Components, Modules and Services in Angular

1. Components, Modules, and Services are three key concepts in Angular that work together to create a complete web application:

* Components are the building blocks of an Angular application. Each component represents a part of the user interface and can contain a template, styles, and other logic. Components can be nested inside other components to create complex UIs.
* Modules are containers for related components, services, and other code. Modules can be used to organize an application into logical units and can be loaded dynamically to improve performance. Modules can also be shared between different applications or used as libraries.
* Services are used to share data and application logic between different components. Services are typically used to manage data access, handle complex business logic, or communicate with external APIs. Services are often injected into components and can be shared across the entire application.

11. What is the scope?

In AngularJS (also known as Angular 1.x), the scope is an object that refers to the application model. The scope is used to maintain the state of the application and provides a way for different parts of the application to communicate with each other. The scope can be shared between controllers, directives, and other components, and can be used to pass data and events between different parts of the application.

12. What is data binding in Angular?

Data binding in Angular is a mechanism that allows data to be bound to UI elements in a component. Data binding can be one-way or two-way. One-way data binding allows data to be bound from the component to the UI, while two-way data binding allows data to be bound in both directions, so changes in the UI are automatically reflected in the component and vice versa.

13. What is two way data binding in Angular?

Two-way data binding in Angular allows data to be bound in both directions between the component and the UI. When a value in the UI is changed, the change is automatically reflected in the component, and when a value in the component is changed, the change is automatically reflected in the UI. Two-way data binding is achieved using the [(ngModel)] directive in Angular.

14. What are Decorators and their types in Angular?

Decorators in Angular are a way to add metadata to classes, functions, or properties. Decorators provide a way to extend or modify the behavior of a class or component without changing its implementation. There are several types of decorators in Angular, including:

* @Component: Used to decorate a class that represents a component.
* @Injectable: Used to decorate a class that provides a service or other data to the application.
* @Input: Used to decorate a property that is used to pass data into a component.
* @Output: Used to decorate a property that is used to emit events from a component.
* @HostBinding: Used to bind a component property to a host element property.
* @HostListener: Used to listen for events on a host element and trigger a method in the component.

15. What are annotations in Angular ?

Annotations in Angular are a way to provide metadata to classes, functions, or properties. Annotations are similar to decorators, but are used in AngularJS (also known as Angular 1.x) instead of decorators. Annotations provide a way to add additional information to classes and components, and are used to configure different aspects of an AngularJS application, such as routing, dependency injection, and animation. Some examples of annotations in AngularJS include @Inject, @Directive, and @Controller.