Angular is a popular open-source JavaScript framework developed by Google. It is designed to build robust and scalable web applications. Here's an overview of how Angular works:

1. **Component-based architecture:** Angular follows a component-based architecture, where the application is divided into reusable components. Each component represents a part of the user interface and contains its own logic, template, and styles.
2. **Templates and data binding:** Angular uses HTML templates to define the structure and layout of the user interface. It provides powerful data binding mechanisms that allow you to bind component properties to elements in the template, enabling automatic synchronization of data between the component and the view.
3. **Dependency injection:** Angular's dependency injection system helps manage the dependencies between different parts of an application. Dependencies, such as services or other components, can be injected into a component, making it easier to maintain and test the application.
4. **Directives:** Angular provides various built-in directives that allow you to extend HTML with additional functionality. Directives can modify the behavior or appearance of elements in the template. For example, the **ngFor** directive is used for repeating elements, and the **ngIf** directive is used for conditional rendering.
5. **Services:** Services are used to encapsulate reusable business logic, data retrieval, or communication with external APIs. Services are typically injected into components and can be shared across multiple components, providing a central place for managing application state and functionality.
6. **Routing:** Angular's router enables navigation between different views or pages within the application. You can define routes, associate them with components, and navigate between them programmatically or through user interaction.
7. **Lifecycle hooks:** As mentioned earlier, Angular provides lifecycle hooks that allow you to respond to different stages of a component's life. These hooks enable you to perform initialization, cleanup, or custom logic when certain events occur.
8. **Change detection:** Angular employs a change detection mechanism to track changes in the component's data and update the user interface accordingly. It uses a hierarchical change detection strategy that efficiently determines which components need to be updated, minimizing unnecessary DOM manipulations.
9. **Ahead-of-Time (AOT) compilation:** Angular supports both Just-in-Time (JIT) compilation and Ahead-of-Time (AOT) compilation. AOT compilation converts the Angular application into efficient JavaScript code during the build process, resulting in faster startup time and improved performance.
10. **Testing:** Angular provides robust testing support with tools like Karma and Jasmine. You can write unit tests, integration tests, and end-to-end tests to ensure the correctness and reliability of your application.

Overall, Angular simplifies the development of complex web applications by providing a structured framework, declarative templates, powerful data binding, and a rich set of features for building scalable and maintainable applications.

Top of Form