Next steps: tools and techniques

After you understand the basic Angular building blocks, you can begin to learn more about the features and tools that are available to help you develop and deliver Angular applications. Here are some key features.

Responsive programming

- Lifecycle hooks: Tap into key moments in the lifetime of a component, from its creation to its destruction, by implementing the lifecycle hook interfaces.
- Observables and event processing: How to use observables with components and services to publish and subscribe to messages of any type, such as user-interaction events and asynchronous operation results.

Client-server interaction

- HTTP: Communicate with a server to get data, save data, and invoke server-side actions with an HTTP client.
- Server-side Rendering: Angular Universal generates static application pages on the server through server-side rendering (SSR). This allows you to run your Angular app on the server in order to improve performance and show the first page quickly on mobile and low-powered devices, and also facilitate web crawlers.
- Service Workers: Use a service worker to reduce dependency on the network significantly improving the user experience.

Domain-specific libraries

- Animations: Use Angular's animation library to animate component behavior without deep knowledge of animation techniques or CSS.
- Forms: Support complex data entry scenarios with HTML-based validation and dirty checking.

Support for the development cycle

- Compilation: Angular provides just-in-time (JIT) compilation for the development environment, and ahead-of-time (AOT) compilation for the production environment.
- Testing platform: Run unit tests on your application parts as they interact with the Angular framework.
- Internationalization: Make your app available in multiple languages with Angular's internationalization (i18n) tools.
- Security guidelines: Learn about Angular's built-in protections against common web-app vulnerabilities and attacks such as cross-site scripting attacks.

Setup, build, and deployment configuration

- CLI Command Reference: The Angular CLI is a command-line tool that you use to create projects, generate application and library code, and perform a variety of ongoing development tasks such as testing, bundling, and deployment.
- Workspace and File Structure: Understand the structure of Angular workspace and project folders.
- npm Packages: The Angular Framework, Angular CLI, and components used by Angular applications are packaged as npm packages and distributed via the npm registry. The Angular CLI creates a default `package.json` file, which specifies a starter set of packages that work well together and jointly support many common application scenarios.
- TypeScript configuration: TypeScript is the primary language for Angular application development.
- Browser support: Make your apps compatible across a wide range of browsers.
- Building and Serving: Learn to define different build and proxy server configurations for your project, such as development, staging, and production.
- Deployment: Learn techniques for deploying your Angular application to a remote server.