**🔰 Very Basic (Prerequisites & Setup)**

These are foundational concepts and setup requirements before diving into Angular:

* What is Angular?
* Angular vs AngularJS vs React vs Vue
* Single Page Application (SPA) basics
* Installing Node.js and npm
* Installing Angular CLI
* Creating a new Angular project (ng new)
* Angular project structure explained
* Running Angular project (ng serve)

**✅ Basic**

These are essential topics that help you build simple Angular apps:

* Modules and AppModule
* Components
  + Creating components (ng generate component)
  + Component metadata and decorator
  + Component lifecycle hooks (OnInit, OnDestroy, etc.)
* Templates and Template Syntax
  + Interpolation ({{ }})
  + Property binding ([property])
  + Event binding ((event))
  + Two-way binding ([(ngModel)])
* Directives
  + Structural directives (\*ngIf, \*ngFor)
  + Attribute directives ([ngClass], [ngStyle])
* Pipes
  + Built-in pipes (date, currency, uppercase, lowercase, etc.)
  + Custom pipes
* Services and Dependency Injection (DI)
  + Creating services
  + Injecting services into components
* Routing
  + Basic routing
  + Route parameters
  + RouterLink and RouterOutlet
* Forms
  + Template-driven forms
  + Form validation
* Angular CLI commands (ng generate, ng build, ng test, etc.)

**🔁 Medium**

These topics cover more dynamic and scalable application needs:

* Reactive Forms
  + FormBuilder
  + FormGroup, FormControl
  + Form validation (sync and async)
* HttpClient
  + Making API calls
  + Handling responses and errors
  + Interceptors
* Observables & RxJS
  + Observable, Subject, BehaviorSubject
  + Operators (map, filter, switchMap, mergeMap, concatMap)
* Routing (Advanced)
  + Lazy loading modules
  + Route guards (CanActivate, CanDeactivate)
  + Child routes
* Component communication
  + @Input and @Output
  + EventEmitter
  + Service-based communication
* Custom Directives
* Custom Pipes with parameters
* Environment configuration
* Working with external libraries (e.g., Bootstrap, Angular Material)

**🚀 Advanced**

These are for production-level applications and performance optimization:

* State Management
  + NgRx (Store, Effects, Reducers, Actions)
  + Akita or NGXS (alternatives to NgRx)
* Dynamic Components
* Content Projection (ng-content)
* ViewChild and ViewChildren
* Advanced Dependency Injection
  + Multi-providers, Injection tokens
* Angular Universal (Server-Side Rendering)
* Angular Animations
* Progressive Web Apps (PWA)
* Performance Optimization
  + ChangeDetectionStrategy
  + TrackBy in \*ngFor
  + Lazy loading & preloading strategies
* Unit Testing
  + Jasmine & Karma
  + Component and service tests
* End-to-End Testing
  + Cypress or Protractor
* Internationalization (i18n)
* Security
  + Sanitizing HTML
  + XSS protection
  + CORS
* Monorepos using Nx