Full Stack JavaScript Developer

Glossary - Angular Fundamentals

Data binding	A process that allows apps to display data values to a user and respond to user actions (such as clicks, touches, and keystrokes). In data binding, you declare the relationship between an HTML widget and a data source and let the framework handle the details.
Angular CLI	The Angular CLI is a command-line tool for managing the Angular development cycle. Use it to create the initial filesystem scaffolding for a workspace or project, and to run schematics that add and modify code for initial generic versions of various elements. The CLI supports all stages of the development cycle, including building, testing, bundling, and deployment.
Attribute directive	Attribute directives modify behavior and appearance of page elements.
Component	A class with the @Component() decorator that associates it with a companion template. Together, the component class and template define a view. A component is a special type of directive. The @Component() decorator extends the @Directive() decorator with template-oriented features.
Component hierarchy	An abstraction of component relationships in an Angular application. An Angular application contains at least one root "parent" component. Nested below may be one or more "child" components, each possibly with their own child components.
Component lifecycle	Every Angular component goes through a cycle: from the moment it is created and mounted to the document object model (DOM), all the way to the moment it is unmounted and ultimately destroyed. This is what we refer to as the component lifecycle.
Decorator	A function that modifies a class or property definition. Decorators (also called annotations) are an experimental (stage 2) JavaScript language feature. TypeScript adds support for decorators.
Dependency	In Angular, dependencies are typically services, but they also can be values, such as strings or functions.
Dependency injection	A design pattern and mechanism for creating and delivering some parts of an application (dependencies) to other parts of an application that require them.

Directive	A class that can modify the structure of the DOM or modify attributes in the DOM and component data model. A directive class definition is immediately preceded by a @Directive() decorator that supplies metadata.
Event binding	Event binding allows you to listen for and respond to user actions such as keystrokes, mouse movements, clicks, and touches.
Form validation	A check that runs when form values change and reports whether the given values are correct and complete, according to the defined constraints. Reactive forms apply validator functions. Template-driven forms use validator directives.
HTTP client	Performs HTTP requests. This service is available as an injectable class, with methods to perform HTTP requests.
Input	When defining a directive, the @Input() decorator on a directive property makes that property available as a target of a property binding. Data values flow into an input property from the data source identified in the template expression to the right of the equal sign.
Interpolation	A form of property data binding in which a template expression between double-curly braces renders as text. That text can be concatenated with neighboring text before it is assigned to an element property or displayed between element tags
Lifecycle method	An interface that allows you to tap into the lifecycle of directives and components as they are created, updated, and destroyed. Each interface has a single hook method whose name is the interface name prefixed with ng. For example, the OnInit interface has a hook method named ngOnInit.
Module	In general, a module collects a block of code dedicated to a single purpose. Angular uses standard JavaScript modules and also defines an Angular module, NgModule.
Module	In general, a module collects a block of code dedicated to a single purpose. Angular uses standard JavaScript modules and also defines an Angular module, NgModule.
Observable	A producer of multiple values, which it pushes to subscribers. Used for asynchronous event handling throughout Angular. You execute an observable by subscribing to it with its subscribe() method, passing callbacks for notifications of new values, errors, or completion.
Ouput	When defining a directive, the @Output{} decorator on a directive property makes that property available as a target of event binding. Events stream out of this property to the receiver identified in the template expression to the right of the equal sign.

Property binding	Property binding in Angular helps you set values for properties of HTML elements or directives. With property binding, you can do things such as toggle button functionality, set paths programmatically, and share values between components.
Router	A tool that configures and implements navigation among states and views within an Angular app.
Router outlet	A directive that acts as a placeholder in a routing component's template. Angular dynamically renders the template based on the current router state.
Service	In Angular, a class with the @Injectable() decorator that encapsulates non-UI logic and code that can be reused across an application. Angular distinguishes components from services to increase modularity and reusability. The @Injectable() metadata allows the service class to be used with the dependency injection mechanism.
Structural directive	Structural directives modify the structure of the DOM.
Target event name	A portion of the event binding syntax. The target event name (e.g., a click) is noted within parentheses to the left of the equal sign.
Template	Code that defines how to render a component's view. A template combines straight HTML with Angular data-binding syntax, directives, and template expressions (logical constructs). The Angular elements insert or calculate values that modify the HTML elements before the page is displayed. Learn more about Angular template language in the Template Syntax guide.
Template statement	A portion of the event binding syntax. The template statement (i.e., a custom event handler method) is noted to the right of the equal sign.
Template-driven form	A format for building Angular forms using HTML forms and input elements in the view.
Two-way binding	The `NgModel` directive allows you to display a data property and update that property when the user makes changes.
View	The smallest grouping of display elements that can be created and destroyed together. Angular renders a view under the control of one or more directives. A component class and its associated template define a view.