Templates, directives, and data binding

1. A template combines HTML with Angular markup that can modify HTML elements before they are displayed.
2. Template directives provide program logic, and binding markup connects your application data and the DOM
3. Before a view is displayed, Angular evaluates the directives and resolves the binding syntax in the template to modify the HTML elements and the DOM, according to your program data and logic
4. Your templates can use pipes to improve the user experience by transforming values for display. For example, use pipes to display dates and currency values that are appropriate for a user's locale

## **Services and dependency injection**

For data or logic that isn't associated with a specific view, and that you want to share across components, you create a service class. A service class definition is immediately preceded by the @[Injectable](https://angular.io/api/core/Injectable)() decorator. The decorator provides the metadata that allows other providers to be **injected** as dependencies into your class.