**One-Way Data Binding:**

In one-way data binding, data flows from the component to the template (view). Changes in the component will be reflected in the template, but not the other way around.

Exercise: Create an Angular component that displays a simple counter. Bind a property in the component to the template to show the current counter value, and add buttons to increment and decrement the counter.

**Two-Way Data Binding:**

Two-way data binding combines one-way binding with event binding. Changes in the component or the template will be reflected in both places.

Exercise: Create an Angular component with an input field that allows users to enter their name. Use two-way data binding to display the entered name below the input field in real-time.

**Event Binding:**

Event binding allows you to respond to user actions (e.g., button clicks, keypress) in the template and trigger methods in the component.

Exercise: Build a simple form with two input fields for username and password. Add a "Login" button, and when the user clicks it, display an alert with the entered username and password.

**Property Binding:**

Property binding allows you to set an element's property in the template based on a property value in the component.

Exercise: Create an Angular component that contains an image. Use property binding to change the image's source URL dynamically when the user clicks a button.

**ngFor Directive:**

The ngFor directive is used to display a list of items in the template by iterating over an array in the component.

Exercise: Create an Angular component that contains an array of objects (e.g., books). Use the ngFor directive to display the list of books with their titles and authors in the template.

**ngIf Directive:**

The ngIf directive conditionally shows or hides elements in the template based on a condition in the component.

Exercise: Build a simple Angular component that contains a button and a paragraph. When the button is clicked, toggle the visibility of the paragraph using the ngIf directive.