**Conditional Rendering in Angular**

Conditional rendering allows you to dynamically display or hide parts of your Angular application's UI based on specific conditions. This makes your application more interactive and user-friendly by adapting its appearance and behavior based on the current state or user input.

**Key Techniques:**

1. **\*ngIf Directive:**
   * **Purpose:** Conditionally render an element or a block of HTML based on a boolean expression.
   * **Syntax:**

HTML

<element \*ngIf="condition">...</element>

* + - If condition evaluates to true, the element is rendered in the DOM.
    - If condition evaluates to false, the element is removed from the DOM.
  + **Example:**

HTML

<div \*ngIf="isLoggedIn">

Welcome, user!

</div>

1. **\*ngIf with else Block:**
   * **Purpose:** Render alternative content when the \*ngIf condition is false.

HTML

<div \*ngIf="isLoggedIn; else notLoggedIn">

Welcome, user!

</div>

<ng-template #notLoggedIn>

Please log in.

</ng-template>

1. **\*ngSwitch Directive:**
   * **Purpose:** Display different content based on the value of an expression.

HTML

<div [ngSwitch]="status">

<div \*ngSwitchCase="'loading'">Loading...</div>

<div \*ngSwitchCase="'success'">Success!</div>

<div \*ngSwitchCase="'error'">Error!</div>

<div \*ngSwitchDefault>Unknown status</div>

</div>

**Benefits of Conditional Rendering:**

* **Improved User Experience:** Tailor the UI to the user's current state or actions.
* **Enhanced Performance:** Avoid unnecessary rendering of elements, improving application performance.
* **Code Maintainability:** Organize and simplify your templates by conditionally displaying different parts of the UI.

**Example:**

TypeScript

import { Component } from '@angular/core';

@Component({

selector: 'app-my-component',

template: `

<div \*ngIf="showDetails">

<h2>User Details</h2>

<p>Name: {{ user.name }}</p>

</div>

`,

})

export class MyComponent {

showDetails: boolean = true;

user = { name: 'John Doe' };

}

In this example, the \*ngIf directive controls whether the user details section is displayed based on the value of the showDetails property.

By effectively using conditional rendering techniques, you can create dynamic and user-friendly Angular applications that adapt to various conditions and user interactions.