**Questions & Assignments List**

1 Introduction to Directives

1. Explain Directives in Angular.

a technique for **adding behavior to DOM elements** and manipulate their appearance, structure dynamically.

2. Structural Directives(\*ngIf)

Also, we can call it control flow

built-in structural directives such as \*ngIf, \*ngFor, and \*ngSwitch, which are commonly used for conditionally rendering elements, iterating over collections, and switching between multiple views.

<div

**\*ngFor="let category of categories"**

>

<a (click)="onCategoryClicked(category)"> {{ category.category }}</a>

</div>

But in version 17, using new control flow

<div \*ngIf="isApple; else other">

This is an apple!

</div>

<ng-template #other>

This is not an apple!

</ng-template>

-- version17

@if (isApple) {

<p>This is an apple</p>

} @else {

<p>This is not an apple</p>

}

<div \*ngFor="let fruit of fruits">

<ul>

<li>{{fruit}}</li>

</ul>

</div>

for (fruit of fruits; track $index) {

<ul>

<li>{{fruit}}</li>

</ul>

} @empty {

Empty list of fruits

}

3. Attribute Directives [ngClass]

Attribute Directives

such as ngClass, ngStyle, and ngModel, which are commonly used for dynamic styling and data binding.

**<div [ngClass]="{'active': isActive, 'error': hasError}">**

This div has dynamic classes applied based on component properties.

</div>

**<div [ngStyle]="{'color': isActive ? 'green' : 'red', 'font-size': fontSize + 'px'}">**

This text has dynamic styles applied based on component properties.

</div>

1. Grouping elements with ng-container

commonly used with other structural directives like \*ngIf, \*ngFor, and \*ngSwitchCase to conditionally render or iterate over elements.