Angular Interview Prep - Key Concepts

@Input()

@Input() allows a parent component to bind values to a child component.

Example:

```
Child: @Input() userName: string;
Parent: <child-comp [userName]="'Ajay'">
```

@Output()

@Output() allows a child component to send events/data to a parent component using EventEmitter.

Example:

```
Child: @Output() notify = new EventEmitter<string>();
Parent: <child-comp (notify)="onNotify($event)">
```

@for (Angular 17+)

New syntax to iterate lists in Angular 17+. Cleaner and more efficient.

```
@for (item of items; track item.id) {
  <div>{{ item }}</div>
}
```

@if (Angular 17+)

New conditional rendering block in Angular 17+.

```
@if (isLoggedIn) {
  <p>Welcome</p>
}
```

@switch (Angular 17+)

Control flow for switching conditions.

```
@switch (role) {
  @case ('admin') { <p>Admin</p> }
  @default { <p>Guest</p> }
}
```

*ngFor

Used to loop through data in templates.

```
<div *ngFor="let item of items">{{ item }}</div>
```

*ngIf

Conditional rendering in templates.

```
<div *ngIf="isLoggedIn">Welcome</div>
```

Angular Interview Prep - Key Concepts

[ngSwitch] & [ngSwitchCase]

Used for conditionally displaying views.

```
<div [ngSwitch]="role">
  <div *ngSwitchCase="'admin'">Admin Panel</div>
  <div *ngSwitchDefault>Guest</div>
</div>
```

ng-template

Defines a template that is not rendered until used.

```
<ng-template #tpl><p>Hello</p></ng-template>
```

ng-container

Used as a grouping element without rendering extra DOM.

```
<ng-container *ngIf="show"><p>Visible</p></ng-container>
```

[ngClass]

Adds/removes classes conditionally.

```
<div [ngClass]="{'active': isActive}">Box</div>
```

[ngStyle]

Applies inline styles conditionally.

```
<div [ngStyle]="{'color': isError ? 'red' : 'green'}">Status</div>
```