Angular Interview Preparation Guide

1. Component, Template & Module Basics

A component in Angular is a TypeScript class decorated with @Component, which controls a part of the UI. It consists of:

- HTML (template)

- CSS (styling)

- TypeScript (logic)

Example:

```
@Component({
  selector: 'app-user',
  templateUrl: './user.component.html'
})
export class UserComponent {
  name = 'Sonali';
}
```

2. ViewChild & ViewChildren

@ViewChild: Access a child component or DOM element.

Example:

```
@ViewChild('inputRef') inputElement!: ElementRef;
ngAfterViewInit() {
  this.inputElement.nativeElement.focus();
}
```

@ViewChildren: Query multiple elements.

```
@ViewChildren('item') items!: QueryList<ElementRef>;
ngAfterViewInit() {
  this.items.forEach(item => console.log(item.nativeElement));
}
```

3. ng-content

Used for content projection (like slots). Allows parent components to pass HTML/content to child components.

Example:

```
<!-- Parent -->
<app-card><p>Content here</p></app-card>

<!-- Child -->
<div class="box"><ng-content></ng-content></div>
```

4. Reactive vs Template-driven Forms

Reactive Forms: Code-based using FormGroup and FormControl.

Template-driven: Uses ngModel.

Example (Reactive):

```
form = new FormGroup({
  name: new FormControl('', Validators.required)
});
```

Angular Interview Preparation Guide

5. Subject vs BehaviorSubject

Subject: Emits values to subscribers.

BehaviorSubject: Same, but holds initial value and emits latest value to new subscribers.

Example:

```
const subject = new Subject<string>();
const behavior = new BehaviorSubject<string>('default');
```

6. Observable vs Subject

Observable: Unicast, doesn't act as observer.

Subject: Multicast, acts as observer and observable.

Use Subject for event sharing.

7. Routing & Navigation

Example:

```
const routes: Routes = [
  { path: 'home', component: HomeComponent },
  { path: 'profile/:id', component: ProfileComponent }
];

this.router.navigate(['/profile', userId]);
```