Angular 8 ngIf Directive

The ngIf Directives is used to add or remove HTML Elements according to the expression. The expression must return a Boolean value. If the expression is false then the element is removed, otherwise element is inserted. It is similar to the ng-if directive of AngularJS.

ngIf Syntax

1. <p \*ngIf="condition">
2. condition is **true** and ngIf is **true**.
3. </p>
4. <p \*ngIf="!condition">
5. condition is **false** and ngIf is **false**.
6. </p>

The \*ngIf directive form with an "else" block

1. <div \*ngIf="condition; else elseBlock">
2. Content to render when condition is **true**.
3. </div>
4. <ng-template #elseBlock>
5. Content to render when condition is **false**.
6. </ng-template>

The ngIf directive does not hide the DOM element. It removes the entire element along with its subtree from the DOM. It also removes the corresponding state freeing up the resources attached to the element.

The \*ngIf directive is most commonly used to conditionally show an inline template. See the following example:

1. @Component({
2. selector: 'ng-if-simple',
3. template: `
4. <button (click)="show = !show">{{show ? 'hide' : 'show'}}</button>
5. show = {{show}}
6. <br>
7. <div \*ngIf="show">Text to show</div>
8. `
9. })
10. export **class** NgIfSimple {
11. show: **boolean** = **true**;
12. }

Same template example with else block

1. @Component({
2. selector: 'ng-if-else',
3. template: `
4. <button (click)="show = !show">{{show ? 'hide' : 'show'}}</button>
5. show = {{show}}
6. <br>
7. <div \*ngIf="show; else elseBlock">Text to show</div>
8. <ng-template #elseBlock>Alternate text **while** primary text is hidden</ng-template>
9. `
10. })
11. export **class** NgIfElse {
12. show: **boolean** = **true**;