1)-**ngOnChanges**:

This hook is called whenever a component's input properties (bindings) change,

before any other lifecycle hooks are called.

2)-**ngOnInit**:

After the initial ngOnChanges call, the ngOnInit hook is called.

It is executed once after the component's inputs are set for the first time.

This is a good place for initialization and setup tasks.

3)-**ngDoCheck**:

This hook is called during every change detection cycle, which can be quite frequent.

It allows for custom change detection logic to be implemented. Be cautious with this hook,

as improper use can impact performance.

4)-**ngAfterContentInit**:

This hook is called after Angular projects external content into the component's view,

typically through <ng-content>.

5)-**ngAfterContentChecked**:

After ngAfterContentInit, ngAfterContentChecked is called on every subsequent change

detection cycle related to projected content.

6)-**ngAfterViewInit**:

This hook is called after the component's view and child views are initialized.

It's suitable for interacting with the view's DOM elements.

7)-**ngAfterViewChecked**:

After ngAfterViewInit, ngAfterViewChecked is called on every subsequent change detection

cycle related to the component's view.

8)-**ngOnDestroy**:

This hook is called just before a component is destroyed. It's a good place for cleanup

tasks like unsubscribing from observables or releasing resources.

It's important to note that these hooks are not guaranteed to be called in every

instance of component creation or destruction.

For example, if a component is created but its parent is never attached to the view,

some hooks like ngOnInit, ngAfterViewInit, and ngAfterViewChecked might not be called.

Additionally, certain scenarios, such as changes to a component's input properties

after initialization, can trigger multiple calls to some lifecycle hooks like

ngOnChanges and ngDoCheck.

Understanding the order in which these hooks are called and their purpose

helps you manage your component's behavior effectively and perform necessary

tasks at the appropriate stages of the component's lifecycle.

Regenerate