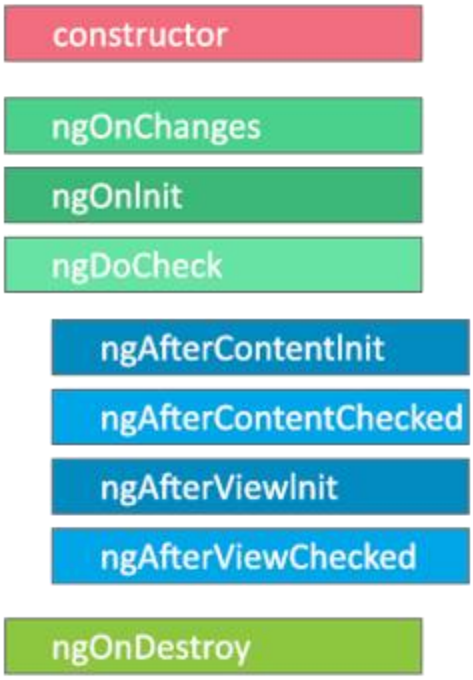
What is Angular Component Lifecycle Hooks?

Lifecycle hooks are Angular methods that are executed at certain points during a component's lifecycle. These methods allow you to tap into the Angular component lifecycle and apply custom logic or operations at specified points in time.



Eight Lifecycle Hooks in Angular

Angular has many lifecycle hooks that allow you to inject code at precise times in a component's or directive's life cycle. These hooks function similarly to checkpoints, allowing you to conduct specified tasks based on the current state of the component. In Angular, there are eight lifecycle hooks:

1. ngOnChanges
2. ngOnInit
3. ngDoCheck
4. ngAfterContentInit
5. ngAfterContentChecked
6. ngAfterViewInit
7. ngAfterViewChecked
8. ngOnDestroy

- *ngOnChanges*
  This event executes every time when a value of an input control within the component has been changed. Actually, this event fires first when a value of a bound property has been changed. It always receives a changed data map, containing the current and previous value of the bound property wrapped in a SimpleChange.

- *ngOnInit*
  This event initializes after angular first displays the data-bound properties or when the component has been initialized. This event basically is called only once just after the ngOnChanges() events. This event is mainly used for initializing data in a component.

- *ngDoCheck*
  This event is triggered every time when the input properties of a component are checked. We can use this hook method to implement the check with our own logic check. Basically, this method allows us to implement our own custom change detection logic or algorithm for any component.

- *ngAfterContentInit*
  This lifecycle method is executed when Angular performs any content projection within the component views. This method executes only for the first time when all the bindings of the component need to be checked for the first time. This event executes just after the ngDoCheck() method. This method is basically linked with the child component initializations.

- *ngAfterContentChecked*
  This lifecycle hook method executes every time when the content of the component has been checked by the change detection mechanism of the Angular. This method called after ngAfterContentInit() method. This method is also called on every subsequent execution of ngDoCheck(). This method is also mainly linked with the child component initializations.

- *ngAfterViewInit*
  This lifecycle hook method executes when the component’s view has been fully initialized. This method is initialized after Angular initializes the component’s view and child views. It is called only the first time after ngAfterContentChecked(). This lifecycle hook method only applies to components.

- *ngAfterViewChecked*
  This method is just called after the ngAterViewInit() method. It is executed every time the when the view of the given component has been checked by the change detection algorithm of Angular. This method executes every subsequence execution of the ngAfterContentChecked(). This method also executes when any binding of the children directives has been changed. So this method is very useful when the component waits for some value which is coming from its child components.

- *ngOnDestroy*
  This method will be executed just before Angular destroys the components. This method is very useful for unsubscribing the observables and detaching the event handlers to avoid memory leaks. Actually, it is called just before the instance of the component is finally destroyed. This method called only once just before the component is removed from the DOM.