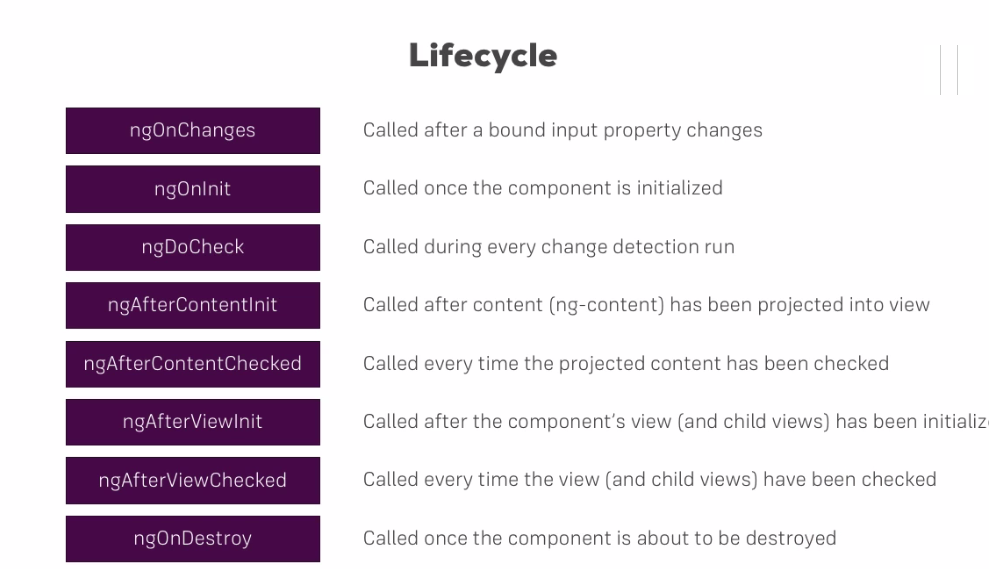
1. **ngOnInit()** inside component is lifecycle hook and angular supports a couple of lifecycle hooks.
2. Angular is responsible to instantiate a new component on finding its selector somewhere in template and responsible to inject into DOM.
3. When a new component is instantiated, Angular goes through a couple of **different phases** in this creation process.
4. Angular gives us a chance to hook into these phases and execute some code.
5. We **can hook into these phases by implementing some methods** Angular will call if they’re present.
6. The kinds of phases and how to hook.
   1. **1st Phase**: ngOnChange: It may be executed multiple times.
      1. When a component is instantiated.
      2. Whenever, our bound input property changes. (**@Input**)
   2. **ngOnInit:** It’s called once the component has been initialized (Would run after the constructor).
   3. **ngDoCheck:** Runs multiple times. Called during each change detection on.
      1. See, it doesn’t mean it would be called whenever any change is made in the template but would be changed even though no change has been made. Such as,
         1. When a button is **clicked**. Even though, nothing has been changed. Who knows. Angular makes sure.
         2. **Timer fires.**
         3. **Observable was resolved.**
   4. **ngAfterContentInit:** Whenever the content which is projected via ng-content has been initialized.
   5. **ngAfterContentChecked:** Called every time the projected content has been checked.
   6. **ngAfterViewInit:** Called after the component’s view (and child views) has been initialized.
   7. **ngAfterViewChecked:** Called every time the view (and the child view) has been checked (Either all the changed detected by the angular have been rendered in the view or no changes were detected).
   8. **ngOnDestroy:** Called once the component is about to be destroyed. Such as \*ngIf on false, removes the element from the DOM.
7. ****