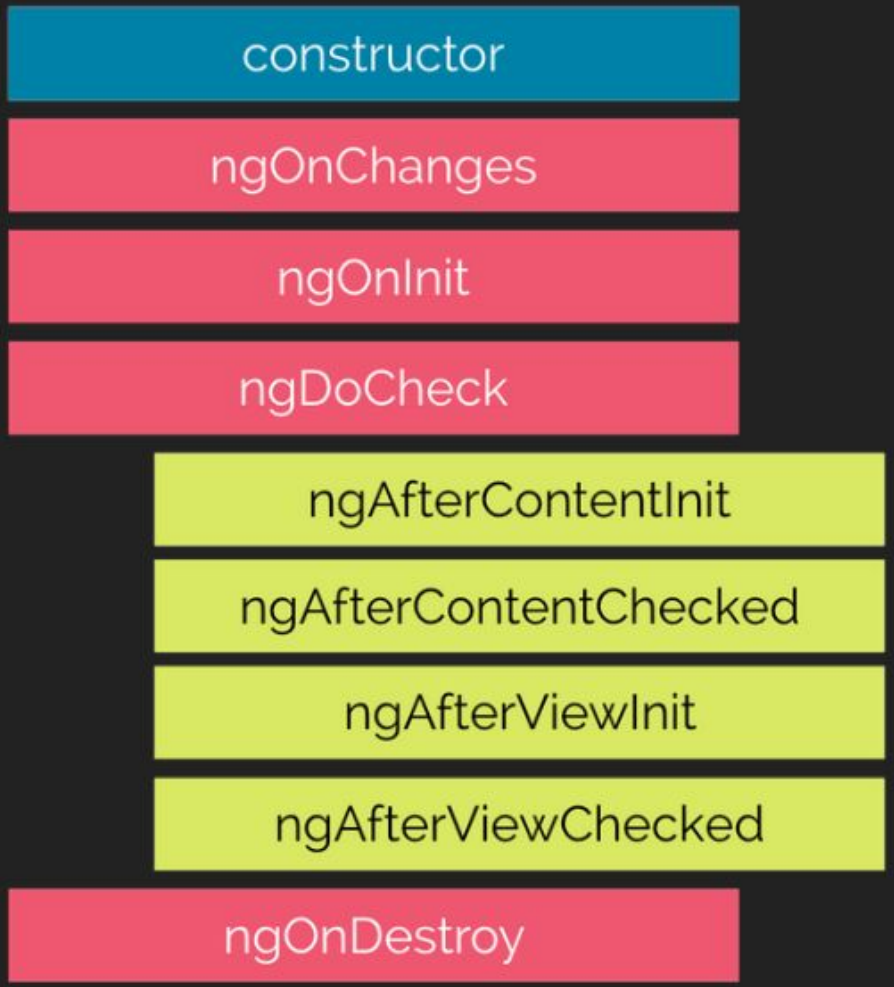
component lifecycle

A component in Angular has a life-cycle, a number of different phases it goes through from birth to death.

We can hook into those different phases to get some pretty fine grained control of our application.

To do this we add some specific methods to our component class which get called during each of these life-cycle phases, we call those methods hooks.


constructor
ngOnChanges
ngOnInit
ngDoCheck
ngAfterContentInit
ngAfterContentChecked
ngAfterViewInit
ngAfterViewChecked
ngOnDestroy

constructor

This is invoked when Angular creates a component or directive by calling `new` on the class.

ngOnChanges

Invoked **every** time there is a change in one of th *input* properties of the component.

ngOnInit

Invoked when given component has been initialized.

This hook is only called **once** after the first `ngOnChanges`

ngDoCheck

Invoked when the change detector of the given component is invoked. It allows us to implement our own change detection algorithm for the given component.

ngOnDestroy

This method will be invoked just before Angular destroys the component.

Use this hook to unsubscribe observables and detach event handlers to avoid memory leaks.

ngAfterContentInit

Invoked *after* Angular performs any content projection into the component's view (see the previous lecture on *Content Projection* for more info).

ngAfterContentChecked

Invoked each time the content of the given component has been checked by the change detection mechanism of Angular.

ngAfterViewInit

Invoked when the component’s view has been fully initialized.

ngAfterViewChecked

Invoked each time the view of the given component has been checked by the change detection mechanism of Angular.