AngularComponent Lifecycle Hooks

**ngOnChanges()**

the lifecycle hook ngOnChanges() is Invoked every time whenever there is a change in one of the input properties of the component.

**ngOnInit()**

The ngOnInit() hook is the most important lifecycle hook in Angular because it identifies the initialization of the newly created component. For the fact that this hook is called only once during the rendering, Most importantly, this hook is being used for fetching data from external sources like servers and APIs.

**ngDoCheck()**

Angular does not detect changes by itself so for that, the lifecycle hook called ngDoCheck() is used. This is run every time after ngOnInit() and ngOnChanges() and is executed every time the process of change detection takes place.

**ngAfterContentInit()**

This lifecycle hook is invoked whenever there is content projection happened inside the component’s view. This one is invoked just once after ngDoCheck() and once all the bindings of the components have been checked.

**ngAfterContentChecked()**

Invoked right after ngAfterContentInit() and subsequent ngDoCheck(), this lifecycle hook responds after the content of the component has been checked by the content projection process. This hook method is always called in response.

**ngAfterViewInit()**

This one is invoked after all the component bindings have been checked with the use of ngAfterContentInit() and ngAfterContentChecked(). This is also a response hook method that invokes when component views and child views are initialized by Angular. And this applies only to components and not directives.

**ngAfterViewChecked()**

A response hook method is invoked every time Angular has checked the component views and the child view. It takes place even if there has been no change or update. This also applies only to components. This is invoked right after ngAfterViewInit() and every subsequent ngAfterContentChecked().

**ngOnDestroy()**

a lifecycle hook that is called when a directive, pipe, or service is destroyed. Use this for any custom cleanup that needs to occur when the instance is destroyed.