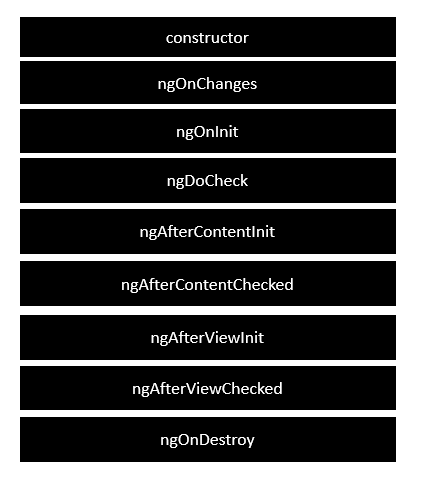
Angular 2 application goes through an entire set of processes or has a lifecycle right from its initiation to the end of the application.

The following diagram shows the entire processes in the lifecycle of the Angular 2 application.



Following is a description of each lifecycle hook.

* **ngOnChanges** − When the value of a data bound property changes, then this method is called.
* **ngOnInit** − This is called whenever the initialization of the directive/component after Angular first displays the data-bound properties happens.
* **ngDoCheck** − This is for the detection and to act on changes that Angular can't or won't detect on its own.
* **ngAfterContentInit** − This is called in response after Angular projects external content into the component's view.
* **ngAfterContentChecked** − This is called in response after Angular checks the content projected into the component.
* **ngAfterViewInit** − This is called in response after Angular initializes the component's views and child views.
* **ngAfterViewChecked** − This is called in response after Angular checks the component's views and child views.
* **ngOnDestroy** − This is the cleanup phase just before Angular destroys the directive/component.

Following is an example of implementing one lifecycle hook. In the **app.component.ts** file, place the following code.

import {

Component

} from '@angular/core';

@Component ({

selector: 'my-app',

template: '<div> {{values}} </div> '

})

export class AppComponent {

values = '';

ngOnInit() {

this.values = "Hello";

}

}

In the above program, we are calling the **ngOnInit** lifecycle hook to specifically mention that the value of the **this.values** parameter should be set to “Hello”.

Once you save all the code changes and refresh the browser, you will get the following output.

