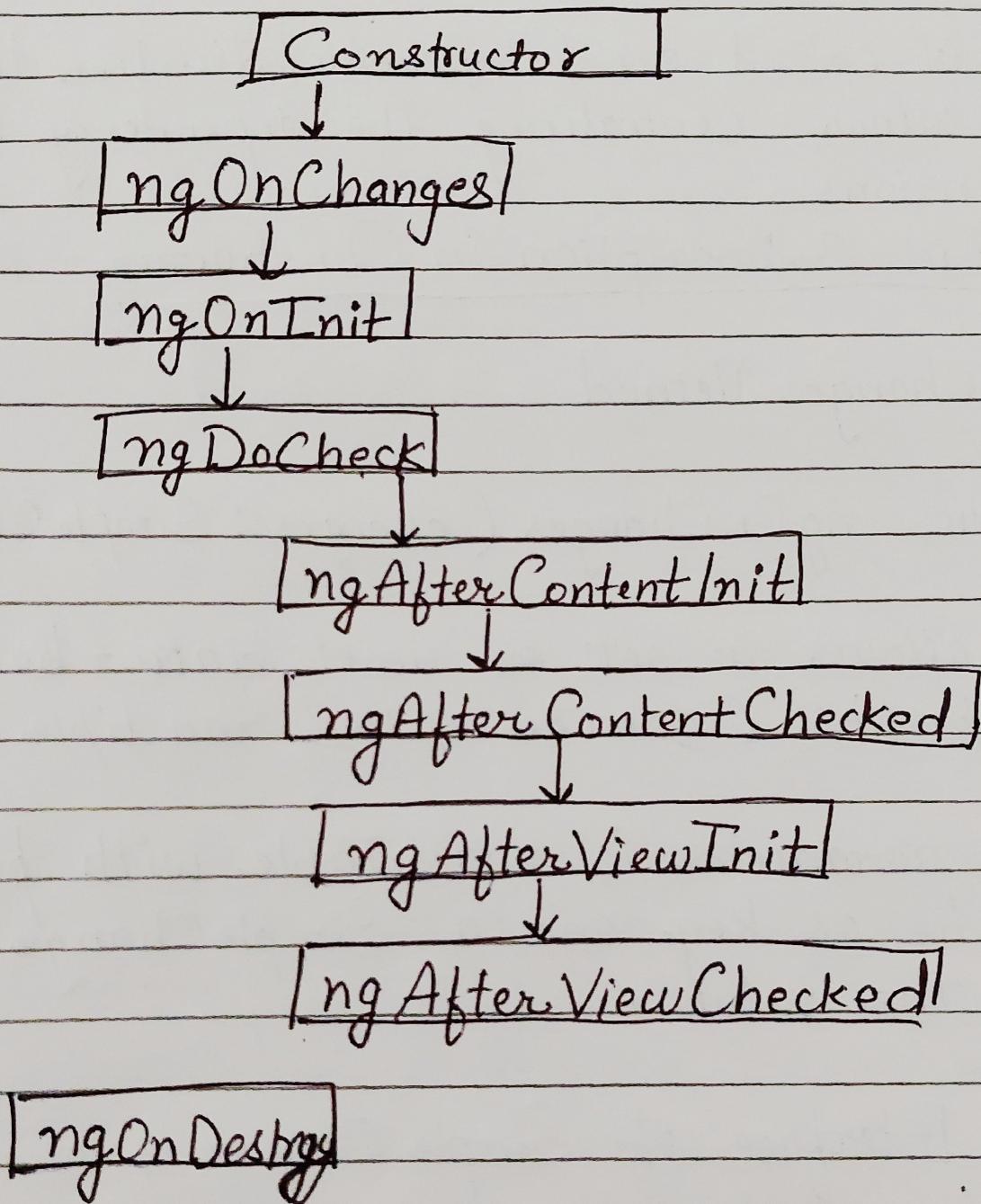


Life Cycle of Angular Components



Instantiation



Method : Constructor()

- It is called on Component/directive instantiation
- It allows centralizing the dependency ~~Instantiation~~ Injections.
- Avoid Subscription in Constructor

Star OnChanges Method

⇒ public ngOnChanges (changes: SimpleChanges): void;

It allows to set or reset data-bound input properties (can be called multiple times).

The parameter is HashTable with properties' name as key and a SimpleChange object as the value.

The Interface of Simple Change is

```
export interface SimpleChange {  
    previousValue: any;  
    currentValue: any;  
    firstValue: boolean;  
}
```

Use Case → To perform light operations depending on data-bound Input properties.

★ OnInit

⇒ `public ngOnInit () : void;`

Use-case → It's called once after the first `ngOnChanges()`.

Data-bound Input properties Initialized

It allows initializing the component/directive.

⇒ Here is a good place for Subscriptions.

★ DoCheck Method

⇒ `public ngDoCheck () : void;`

Use Case → It's called after every `ngOnChanges` and just after `ngOnInit()` first run.

In Some cases, Angular can't detect changes on it's own.

For-e.g - If an Input property is object and it's reference does not change, but one property does, It will not be detected by Angular.

Be careful while using this

★ AfterContentInit Method

⇒ public ngAfterContentInit(): void;

Use-case → It's called once after ngDoCheck()

If you use one of @ContentChild to get contentChild, they are initialized here

★ AfterContentChecked Method

⇒ public ngAfterContentChecked(): void;

Use-Case → Same as ngDoCheck() but for ContentChild properties

★ AfterViewInit Method

⇒ public ngAfterViewInit(): void;

Use - Case → It's called only once after first ngAfterViewInit

★ AfterViewChecked Method

⇒ public ngAfterViewChecked(): void;

Same as ngDOCheck() but for ViewChildren

★ OnDestroy Method

called immediately before Angular destroys the component/directives.