/

2

https://www.youtube.com/watch?v=RCLwwWQ8-V0

Angular component life cycle:

Angular component life-cycle.

A component has a life cycle managed by angular

->creates the component

->renders the component

->create and renders the component children

->checks when the component data-bound properties change,and

->Destroy the component before removing it from DOM.

---->To tap into and react when these lifecycle events occur, angular offer several lifecycle hooks.

1. ngOnChanges

2. ngOninit

3. ngDoCheck

    ->ngAfterContentinit

    ->ngAfterContentChecked

    ->ngAfterViewinit

    ->ngAfterViewChecked

4.ngOnDestroy

ngOnChanges:

Executes,every time the value of an input property changes,The hook method receives a simpleChanges object containing current and previous property values. this is called before ngOnInit

ngOninit:

Executes after the constructor and after ngOnChange hook for the first time.It is most commonly used for component initialization and retrive data from database.

ngOnDestroy:

Executes just before angular destroys the component and generally used for performance clean up.

simple example for @input:

```
//app.component.ts

import { Component } from '@angular/core';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  userText:string = "chandra";

}

//app.component.html

Your text: <input type="text" [(ngModel)]="userText" />

//here we just displayed textbox

//now create simple.component.ts

here we will use that text box value.

import { Component, Input } from "@angular/core";

@Component({

  selector:"simple",

  templateUrl:"./simple.component.html"

})

export class SimpleComponent{

  @Input() simpleData:string

}

//simple.component.html
```

you enterd {{simpleData}}

now update simple tag in app.component.html...

```
//app.component.html
Your text: <input type="text" [(ngModel)]="userText" />
<br /><br />
<simple [simpleData]="userText"></simple>
```

now,my moto is to print changes value and previous values of textbox.....

for this we need to import OnChanges interface

this ngOnChanges hook recevies a parameter called simpleChanges so need to import simpleChanges also

```
import { Component, Input, OnChanges, SimpleChanges } from "@angular/core";
@Component({
  selector: "simple",
  templateUrl: "./simple.component.html"
})
export class SimpleComponent implements OnChanges {
  @Input() simpleData: string;
  ngOnChanges(changes: SimpleChanges) {//this method will trigger when ever the changes were happen
    //simpleChanges contains all the properties like current and previous values.
    for(let propertyName in changes){
      let change = changes[propertyName];
      let current  = JSON.stringify(change.currentValue);
      let previous = JSON.stringify(change.previousValue);
      console.log(propertyName + "current value"+ current + " and previous value"+ previous);
    }
  }
}
```

ngOnInit:

```
import { Component, Input, OnInit } from "@angular/core";
@Component({
  selector: "simple",
  templateUrl: "./simple.component.html"
})
export class SimpleComponent implements OnInit {
    constructor(){
    //first constructor will execute this is related to class
    //if we make any service call here..it has to wait until the entire component load, so better way to
  make service is in ngOnInit
     console.log("in constructor");
   }
   ngOnInit(){
     console.log("in component initialized");
   }
  }
```