Day 5 about Angular

Basically, directives are used to extend the power of the HTML attributes and to change the appearance or behavior of a DOM element

3 Types:

1. Component
2. Structural
3. Attribute

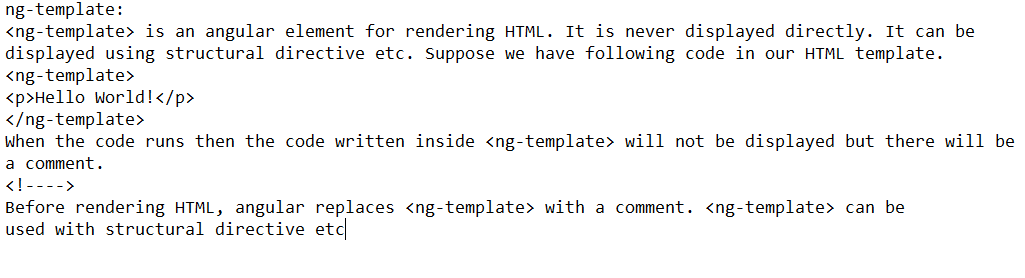
Even if you have never written a structural directive yourself, you have probably been using \*ngIf and \*ngFor in your templates pretty often. The asterisk (\*) states it is a structural directive

1. Structural directive

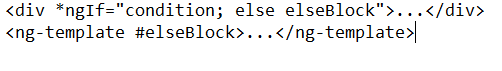
NgIf (\*ngIf ) : alse or non

NgFor (\*ngFor)

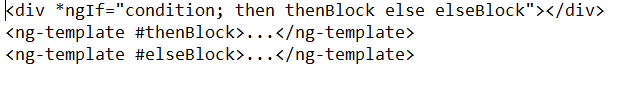
NgSwitch (\*ngSwitch)

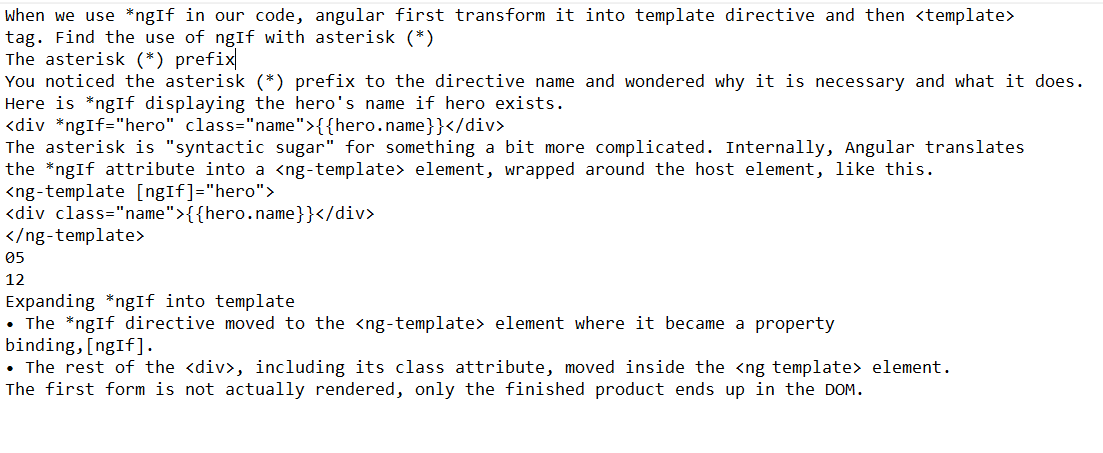


NgIf and Else



ngIf then else





Angular work flow

App.json-> Main:-> boostrapmodule-> appmodule->app,component.ts-> data to it with the help of selector, templateurl-> index.html->app-root->approot.component.ts

**Ngif with Component Elements**

NgIf with Component Elements Here we will use NgIf with component elements. For the demo we are creating a child component as given below.

msg.component.ts

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

@Component({

selector: 'my-msg',

template: 'Hello {{pname}} '

})

export class MsgComponent {

@Input() pname : string;

}

Now find the parent HTML template code snippet with component element from person.component.html. Here we are performing component property binding and using ngIf NgIf with Async Pipe Short Introduction Pipe: AngularJS 1.x has filters which are used for many common uses, like formatting dates, string display in upper or lower case etc. These filters are known as "Pipes" in Angular 2 and higher. Pipes allow us to change the data inside the template. Normally, a pipe takes the data and transforms this input to the desired output. There are many built-in pipes in Angular 2 and higher. Like Date Pipes, Uppercase Pipe, Lowercase Pipe, Percentages/Number Pipe, Currency Pipe, Slice Pipe, Async Pipe etc NgIf with Async Pipe Short Introduction Async Pipe: Angular 2 and higher provides a new special pipe "Async". It allows us to bind the value of a variable directly with template which arrives asynchronously. It provides great ability for working with promises and observables. If we have Observable or Promise instance then we use it directly with AsyncPipe using directive such as NgFor, NgIf and NgSwitch. AsyncPipe belongs to angular common module. AsyncPipe subscribes to Observable or Promise and returns the latest data. NgIf with Async Pipe we will use NgIf with AsyncPipe. We can store conditional result in a variable. This approach is useful when initially the value is null or undefined and we want to avoid exception. As we know that to avoid exception we can access value of an object using safe- nevigation operator?. and these types of scenarios arise when we fetch value over HTTP. Before getting the HTTP response the object may be undefined or null. In case of NgIf angular provides better approach to handle exception without using safe- nevigation operator ?. Now simply to store conditional result in a variable, we can do as follows without using async