**1)component based model.**

Each component know how to render it can be plugin to some other application contains both html and JavaScript code

Example: header component

Every component has a selector means give name to that component and register to angular.

Machine generated alternative text:
«main-sectionx/main-sectiorv 
«footer-sectionx/footer-sectior» 

**NOTE :** component can have sub component

Ex: body section can have sub component

Machine generated alternative text:


Every component in angular have root component and then those component can have child component

Machine generated alternative text:
header 
main 
summary 
Root 
component 
footer 
details 
sidebar 
nav 
updates 

**Earlier:**

Machine generated alternative text:
HTML 
Add a div and paragraph 
Add button 
JS 
Code to get date/time 
Get the paragraph DOM element 
Update value 
Code function to handle 
button click 

**Now:**

Machine generated alternative text:
Root 
component 
date 
click 

Then tell angular to handle root component and all the child component

1. **Setting up Angular js**

Three thing need to install

1)install node js ( check node -v , npm -v)

1. Install visual studio
2. Npm install @angular/cli -g ( will install globally)

Machine generated alternative text:
Angular Basics 
This is called a "scoped" package name, These names are Of the format "roup/ package 

1. **Creating and running a project**

1. Ng new first-project

Machine generated alternative text:
saurakes@SAURAKES-96H02 MINGW64 /c/AFM_DOC/fi rst-project 
angul ar . json 
node_modul es/ package-I ock . json 
src/ 
package . json 
README . md 
tsconfig 
(master) 
tslint.js 
• Json 

It will download all the dependencies in in node\_modules folder and list all the dependencies in package.json

NOTE: in order to start a project use **ng serve .** It will create localhost server

1. **Introducing the app component**

**Index.html**

**<app-root></app-root>** :------------ it is root component of application

It is combination of html,css and ts file present under app folder

Machine generated alternative text:


View defined in html and css and backend logic defined in ts file

NOTE: 3 files involved to rendering the view i.e html , css and ts file

1. **Creating your first component**

**Note :** component consist of html , css ,spec.ts and ts file

**Generate component from cli**:- **ng generate component hello-world**

Note: every component has a selector

Machine generated alternative text:
( Onlnit ; 
Komponent( 
te"'lateuri: • 
styleUrls: ./hello-•orld.cnponent.css') 
export class i•ptemnts Onlnit 
constructor() 
ngOnInit() 
e2e 
app 
Ts wp.mo"e.ts 
'"icon 'co 
12 

Machine generated alternative text:
2 app-he I 
app 
Ts 
corn ponent 
"p. module. ts 

**Note:** We can use multiple times <app-hello-world> and each time we use it we are creating it instances

Machine generated alternative text:
AppComponent 
HelloWorldComponent 

1. **Anatomy of a component**

Root component will load 2 component

Machine generated alternative text:
u) app.cornponent.htrnl vc/app 
• app 
date 
date. component rss 
date.componcnt.spec.ts 
TS date.component.ts 
hello-world 
wp.component.htrN 
app.component.spec.ts 
U) TS date. c ornpone,nr. t s 
capp-hel app-het to-world» 
3 øpp-date*/apø-dat» 

**Query:** how three different file in a component works together

Machine generated alternative text:
Angular Basics 
Every Angular component is mainly a TypeScript class. Think of the HTML and CSS files as just 
extra "attachments" to the main TypeScript file. 

**2 thing need to do:**

First create class for the component that include all the functionality which is essential for component

Second thing register that class as a angular component

**NOTE:** component consist of two parts first part is class and second part is register

@Component :- hey angular when someone using that selector , instantiate that class

NOTE: starting point is ts (typescript) file

1. **Binding data from component class ( one way data binding example)**

Machine generated alternative text:
date.component.html 
Show the 
computed date 
Compute the 
current date 
DateComponent Class 

Machine generated alternative text:
Angular Basics 
' Onen1.S9eCÆS 
Chunk 
in 1 1 nei (inline) 5.79 : entry I 
Data binding in Angular refers to binding the data in the component instance to the HTML 
template. Any changes to the data automatically get updated in the view! 

**Datecomponent.ts**

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

@Component({

selector: 'date',

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

styleUrls: ['./date.component.scss']

})

export class DateComponent implements OnInit {

**message:string= new Date().toDateString();**

constructor() { }

ngOnInit() {

}

}

**Datecomponent.html**

<p>

{{message}}

</p>

**8) Data binding and async**

**Requirement:** every second want to change value of time when page opens

Machine generated alternative text:
Angular Basics 
setlnterval is a JavaScript API that lets you run a function at regular time intervals. 

**setInterval has 2 argument first is callback that function need to execute and second is how often**

**Callback need to call**

Machine generated alternative text:
Angular Basics 
Since I'm using a "fat arrow" function, the this keyword still refers to the component instance. 
To learn about this concept, check out Objects and Prototypes In-Depth course at javabrains,io 

 A **JavaScript Callback Function** is a **function** that is passed as a parameter to another **JavaScript function**, and the **callback function** is run inside of the **function** it was passed into.

**Date.component.ts**

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

@Component({

selector: 'date',

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

styleUrls: ['./date.component.scss']

})

export class DateComponent implements OnInit {

dateMessage:string;

constructor() {

setInterval(()=>{ let currentDate = new Date();

this.dateMessage = currentDate.toDateString() +" "+currentDate.toLocaleTimeString();

},1000);

}

ngOnInit() {

}

}

**Date.component.html**

<p>

{{dateMessage}}

</p>

**Note :-** there is no concept of digest cycle in angular 2+

**9) Template Interpolation**

**We can call function from interpolation**

**Example: date.component.ts**

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

@Component({

selector: 'date',

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

styleUrls: ['./date.component.scss']

})

export class DateComponent implements OnInit {

dateMessage:string;

constructor() {

}

ngOnInit() {

}

addTwoNumber(a:number,b:number)

{

return a+b;

}

}

**Date.component.html**

<p>

{{addTwoNumber(2,3)}}

</p>