# Getting Started with Angular:

## 1.Intro:

## 2.What Version?

## 3.Why Angular is considered as a complex? [too many Jorgen keywords ]

## 4.Tools to get started;

-Code editor

-NodeJS

--Node -v

--Npm -v

-NG [Angular is framework n there r some rules we have to follow it.]

-https://angular.io/docs

-setup-> npm install -g @angular/cli

--Ng version/ng v

## 5.NG Command Line (CLI Overview ):

-https://angular.io/cli

--version

--new

--generate

--serve

--build

….

## 6.Create Your First Angular App:

Create folder:

-ng new Hello-Angular

Code extensions For VS code:

**Prettier**

Ng serve (4200)

Project Structure:

-ts Files(bottom 4 files) are type script configuration

-ReadME file

-Package.json file (imp)

-package.lock.json

-angular.json (ng config)

-editorconfig file (for ur code format extension setting and all)

-node modules(never give to production) npm install

-e2e

-src (Main)s

**https://angular.io/guide/file-structure**

->Print Hello Angular

Online Editors: (concept learning, short demo etc)

-Play ground (<https://www.typescriptlang.org/play>)

-codesandbox (<https://codesandbox.io/>)

-stackbliz (<https://stackblitz.com/>)

# Type Script That you need for ANGULAR

## Types in Typescript

-number

-string e.g let fanme:string = “ok”;

-boolean

-undefined

-null

-any

Conditionals in typescript

let a:number=20,b=30;

if( a > b) {

    console.log("a is bigger...")

}else

console.log("b is bigger...")

//with ternary

a > b  ? console.log("its a "): console.log(" its b");

## Looping through Array

for(let i=0;i<5;i++){

    console.log(i);

}

let myArr = [10,22,34,55];

for(var val of myArr){

    console.log(val);

}

for(var index1 in myArr){

    console.log(index1);

    //console.log(myArr[index1]);

}

while(condition){

    //block of code

}

## Functions and parameters in Typescript

function Meet(greeting:string,name:string="sam"):string{ //default n for

//optional use ?

    return greeting+" meet me ...I'm "+name

}

Meet("Good Morning!");

let data=Meet("Good Morning!","King");

console.log(data);

## Interface In Typescript (remember keyword any)

Interface is guideline. Just don’t over think.

We can use interface with interface and classes.

interface User{

        userId:number;

        readonly userName?:string; //readonly.we canot change it.

}

interface catUser extends User{

    caTname:string;

    catType:string;

}

class Animal implements User{

}

## Classes, constructor and Interfaces

interface person{

    fName:string;

    hideMe():void;

}

interface MyEmp{

    empCode:string;

}

class Employee implements person,MyEmp{

    fName :string;

    constructor(name:string){

        this.fName=name;

    }

    play(){

        console.log("playing....")

    }

}

## Decorators

Decorators can be used on classes ,interfaces,methods,properties etc.

@CanFly

class Person{

    private fName:string;

    constructor(fName:string){

        console.log("calling constructor...");

        this.fName=fName;

    }

}

  console.log("creating 1st Object");

  let john:Person = new Person("John");

    console.log(john);

    console.log(`Can I flay:${john['fly']}`);

     console.log("creating 2st Object");

  let kabir:Person = new Person("Kabir");

console.log(kabir);

     console.log(`Can i fly:${kabir['fly']}`);

     function CanFly(constructorFunction:Function){

        console.log("Can fly Invoked");

        constructorFunction.prototype.fly=true;

     }

# Lest Build Counter

-Basic CSS and HTML

-counter Logic

-Event binding and interpolation

# Angular From Scratch

Bootstrapping-is self starting process.

-create index.html,main.ts,style.css,polyfills.ts inside src

Create App folder

Create module

Create component

Tsconfig.json->

For Decorator support:

"experimentalDecorators": true,

File separation for component

Create separate app.componet.html,app.component.css

Create second component and inject it

# Word Generator Assignment:

For bootstrap

Npm install bootstrap

Import In styels.css

@import url("~bootstrap/dist/css/bootstrap.min.css");

\*NgIf

Interpolation

Add extra assignment here

# Game: Tic-tac-toe

Required Configuration:

Install ngX toaster library (<https://www.npmjs.com/package/ngx-toastr>)

npm i ngx-toastr

npm i @fortawesome/angular-fontawesome

npm install @fortawesome/free-solid-svg-icons

npm install @fortawesome/fontawesome-svg-core

npm install @fortawesome/free-regular-svg-icons

import { FontAwesomeModule } from '@fortawesome/angular-fontawesome';

FontAwesomeModule

BrowserAnimationsModule, // required animations module

ToastrModule.forRoot(), // ToastrModule added

NgIf,ngFor,ngSwitcgCase,Parrent to chuld communication,event binding,input binding,interpolation n typescrit

RxJx Fundamentals

Intro

Use StackBlitz for demo

For visualization refer[ https://rxviz.com/examples ]

StackBlitz…

document.addEventListener('click', () => {

  console.log('clicked...');

});

import { Observable } from 'rxjs';

const obs = new Observable(oberver => {

  oberver.next(1);

  oberver.next(2);

  oberver.next(3);

  setTimeout(() => {

    oberver.next(4);

    oberver.complete();

  }, 3000);

});

obs.subscribe({

  next(d) {

    console.log(d);

  }

});

ToDo App

Npm I bootstrap

npm i @fortawesome/angular-fontawesome

npm install @fortawesome/free-solid-svg-icons

npm install @fortawesome/fontawesome-svg-core

npm install @fortawesome/free-regular-svg-icons