# Requirements & Tools

Languages:

* HTML
* CSS
* JavaScript
* Typescript

Packages:

* [Node.js](https://nodejs.org/en/download/current)
* [Angular CLI](https://angular.io/cli)

Tools:

* [Visual Studio Code](https://code.visualstudio.com/download)

# Installation

Install angular/cli on administrator command prompt

> npm install -g @angular/cli

Create first project:

> ng new [App Name]

Run first project:

[App]> ng serve

Install requirement packages:

[App]> npm install

# Angular project structure

## Main file and folders

1. app folder: Your codes
2. Index.html: Main output of the project
3. main.ts: Project starter file
4. Style.css :
5. package.json: Project dependencies

Parts:

* dependencies: Running requirements
* devDependencies: Developing req.

1. angular.json: Project settings

Important parts:

* projects/[app]/ architect/build/assets
* projects/[app]/ architect/build/styles
* projects/[app]/ architect/build/scripts

## Run connections

main.ts: read bootstrap module

(default: AppModule)

app.module.ts: read bootstrap component

(default: AppComponent)

Show AppComponent

# Components

Each component is identified by the @Components decorator.

Some attributes of @Components are:

* selector: name of the component
* template: Content of HTML
* templateUrl: Url of UI HTML file
* styleUrls: list of this file style sheets

## Make a component

Make a component with angular/cli:

[App]> ng generate component [Component Name]

Make a component manually:

* Create [Component name].ts in the app folder (It’s better to create under your component folder)
* Define a component class

@Component({

  selector: "app-user",

  templateUrl: "./user.component.html",

  styleUrls: ["./user.component.scss"]

})

export class UserComponent{

}

* Add the component to AppModule

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

import { AppComponent } from './app.component';

import { UserComponent } from './user.component';

@NgModule({

  declarations: [AppComponent, UserComponent],

  imports: [BrowserModule],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule {}

## Using component

Add a component selector tag to a custom HTML file

For example, you can add in the AppComponent:

<app-user></app-user>

# Selectors

You can use your component by the selector in another component.

There are 3 types of selectors:

* Element: ‘app-user’ Ex.: <app-user></app-user>
* Class: ‘.app-user’ Ex.: <div class=”app-user”></div>
* Property ‘[app-user]’ Ex.: <div app-user></div>

# Data binding

## TS File -> HTML File

* String Interpolation {{ var }}
* Property Binding [property]=”var”

## HTML File -> TS File

* Event Binding (event)=”function($event)”

## HTML File <-> TS File

* Two-Way binding [(ngModule)]=”var”

// component.ts

export class UserComponent{

name string;

isEnable : Boolean;

public OnInputChanged(e: event){

this.name = (<HTMLInputElement>e.target).value;

};

}

// component.html

<p>{{name}}<p>

<button [disabled]=”isEnable”><button>

<input type=”Lable” (input)=”onInputChanged($event)”><button>

Creator: [Yahya Ebneddin](https://www.linkedin.com/in/yahya-ebneddin/)

Thanks a lot, Er. [Foad](https://www.linkedin.com/in/foadshariat/)

[](https://www.buymeacoffee.com/yebneddin)