**Angular Online Session References**

Git link : <https://github.com/synergy2411/ng-cg>

* Node -v
* Npm -v

Node installer comes with :

* Node Runtime Environment (NRE)
* Node Package Manager (NPM)
* Node Core Module/ Packages

**Angular - v8 : Components, template, Two data binding, MV\*, SPA, Remote Server Calls etc**

Jquery : DOM manipulation, Remote Calls (XHR)

React : Virtual DOM. Great for application where data changes very frequently over a period of time.

Polymer : Creates custom JS components

Backbone : Client side MVC

Knockout : two way data binding. Implements MVVM.

\* Ember : client side framework

Node : platform, runs javascript code.

Express : web application framework, runs on node runtime.

Angular Compilation :

JIT : app size + compiler size (heavy)

AOT : Ahead-Of-Time -> minified, uglify, renderable build.

“**Tree-shaking**” – minimalistic code. Kick-start angular applications in very less time.

Angular can create our own elements : <product-details></product-details>

Transpilers in market : (to convert Typescript into Javascript)

* + Babel
  + Traceur
  + Typescript compiler (> npm install -g typescript)

Angular CLI Tool :

* npm install -g @angular/cli
* ng -v
* ng new users-app
* cd users-app
* ng serve -o
* ng generate component <component-name> (ng g c <comp-name>)
* ng generate directive <directive-name>
* ng generate pipe <pipe-name>
* ng generate module <module-name>
* ng generate service <service-name>

To install Bootstrap :

**npm install bootstrap --save**

Index.html 🡪 main.ts 🡪 AppModule 🡪 AppComponent 🡪 app.component.html

Webpack development Server : Static server > 4200

Ng serve

**Data Bindings :**

*Property Binding* – bind the model with the element attribute/property

[ ] – Property binding Syntax

Supplies the model from component class to view/template

Event Binding – binds the UI model with the component class

( ) – Event Binding Syntax

Two-way data binding 🡪

1. [ Property Binding ] + ( Event Binding )
2. Banana in the box : [ ( ngModel ) ]

**Component Communication :**

Parent to Child : [ Property Binding ] + @Input( )

Child to Parent : ( Event Binding ) + @Output( )

**Directives** : Types of Directives –