Node Installer : (https://nodejs.org)

1. Node Runtime Environment
2. Node Package Manager (NPM)
3. Core Module

JavaScript can be run on :

1. Browser Environment
2. NRE

Library : to achieve specific tasks.

Framework : provides the skeleton to create your application.

Other Libraries vs. Angular Framework: - (Superheroic framework)

React :

React uses *Virtual DOM.*

Only minimal changes will be affected on the browser.

Uni-directional flow of data. (flux/ Redux)

\*EmberJS : Ember kept their API changes very frequently.

PolymerJS : for creating Custom Components

jQuery : DOM manipulation.

Backbone : for achieving MVC pattern at client side.

Knockout : Great for Two-way data binding, also implement MVVM.

VueJS : state management : VueX, XHR (Remote )Call : Vue-resources; MVVM

NodeJS : Platform, to run JS on Server. (Server-side scripting)

ExpressJS : Web application framework for NodeJS

Transpilation : The process of converting TS code in JS Code.

Transpilers : Babel / Typescript compiler / Traceur etc

npm install typescript -g

**Prototypical Hierarchy in JS**

Angular CLI :

* npm install @angular/cli@6 -g
* npm install @angular/cli -g

**CLI Commands :**

ng new <project\_name>

cd users-app

ng serve

ng generate component <component\_name>

ng g c <component\_name>

**Angular App Flow :**

Webpack dev server loads HTML (index.html)

Main.ts loaded on index.html – loads the Root Module (AppModule) on the browser.

AppModule (Root Module) : loads the component (AppComponent) on the Index, cause of selector <app-root>

AppComponent (Root Component): once get the selector, load the associated template on UI.

Loading Component :

1. create component with selector and template
2. declare the component in the module
3. load the selector

**Bootstrap Grid System :** 1 Row = 12 equal columns

XS – mobile : col-xs-12, col-xs-6

SM – Tab : col-sm-12

MD – Laptop/Desktop

LG - Projector

Parent to Child Component Communication :

1. @Input() + Property Binding (when only model being passed)
2. Data Projection - <ng-content></ng-content> (when formatted template being passed)

Child to Parent Component Communication :

1. @Output() + Event Binding

**Component Types :**

* 1. Smart component / Container Component : maintain the domain data, contains the logic how to flow the data in application ( UsersComponent )
  2. Dump component / Presentational Components : Receive data from its parent and also access the parent business logic (UserImage, UserInfo)

ng-container : solves the problem of having two structural directives on one element

Form States / Form Control Classes :

* ng-pristine / ng-dirty
* ng-touched / ng-untouched
* ng-valid / ng-invalid

**VS Code Extensions for Angular :**

Angular Language Service -by Angular

Angular v7 Snippets -by John Papa

Auto Close Tag -by Jun Han

Auto Imports -by Steoates

HTML5 Snippets -by Mohamed Abusaid

Services : implements two principles :

1. Single Responsibility Principle (SRP)
2. Dependency Injection Principle (DIP)

Observables : 2 Category

1. Hot Observables : e.g. Going to hear a live music band
2. Cold Observables : e.g. Watching a movie on Netflix