Angular Course Notes

1. Configure and describe/build dev environment
    a. express
    b. index.html
    c. ts-node - typescript to node transpiling
    d. zone.js - async multi thread
    e. system.js (& config) - module loader
2. HelloWorld (rename as app.ts) - 01-MVC-App
    a. use default directory to start
    b. tsconfig.json
    c. Create app.ts with explanation then save into default for future use
3. Simple Component
    a. Open step3 directory for instruction - deles use default
    b. create search-box directory
    c. create search-box.component.ts
    d. import to app.ts & add to module declarations & <search-box></search-box>
4. Input/Output Data
    a. simple text property displayed in input placeholder="{{text}}"
    b. next make it configurable from the app level @Input()
    c. Simple Events
    d. <input placeholder={{text}} #input>
    e. <button class="btn-clear" (click)="clear(input)">Clear</button>
5. Sharing Events Across Components
    a. Open step5 folder for starting assets
    b. Create **color-picker.ts**
    c. Add to app.ts declarations
    d. Add [ngStyle]="{'color':color}" to colorpicker title in **color-picker.ts**
    e. Add color="red" to selector in app.ts
    f. Add (click)="choose('#0f0')" to divs in **color-picker.ts**
    g. add new EventEmitter @Output in **color-picker.ts**
    h. add method to **color-picker.ts to** emit the event - choose(color:string){ …
    i. add event to **app.ts** (color)="onColor($event)"
    j. create onColor() event handler with console.log test in **app.ts**
    k. create reset button in **app.ts** and then create reset method in **colorpicker.ts**
    l. add #picker to colorpicker in **app.ts**
    m. call picker.reset() to button in **app.ts**
    n. create **colorpreviewer**
    o. add to template and declarations in **app.ts**
6. *ngFor Directive
    a. Open step6 folder
    b. Create heroes.ts component
    c. Add to app.js
7. ngClass Directive
    a. Open step7 folder for instructions - Deles carry on with step6 project
    b. Add properties to array of heroes in **heroes.ts**
    c. Edit **heroes.ts** starting with class then ngClass
8. *ngIf Directive
    a. Open step8
    b. Create demo in **app.ts**
9. Interpolation Extended Demo
    a. Open step9 instructions - deles duplicate default directory
    b. Lead demo on interpolation techniques
10. Zones
    a. Open step 10 instructions - deles duplicate default directory
    b. Demo events and how Zone handles things

    c. When does the view get updated? DOM events, Alerts ,AJAX callbacks, setInterval/setTimeout callbacks etc..

11. Pipes
    a. Open step 11 instructions - deles duplicate default directory
    b. Demo pipes
12. Services
    a. Open step12 instructions - deles **open** step12
    b. explain express server routes
    c. create **lessons-service.ts**
    d. import classes
13. Modules - maybe optional unit
    a. Open step13 instructions - deles **duplicate** step12
    b. create lessons.module.ts
    c. modify app.ts to import module (not providers)
14. Angular-CLI
    a. ng new my-app
    b. ng serve - localhost:4200
15. Creating Components in angular-cli
    a. ng generate component home
    b. add <app-home> to **app.component.html**
    c. ng g component contact
16. Routing
    a. point out <base href='/'> in top level **index.html**
    b. ng g module app-routing
    c. open **app.routing.module.ts** and modify
    d. open **app.module.ts** and add imports for new module
    e. add <router-outlet> to **app.component.html**
    f. test localhost:4200 to see home page working
    g. add contact route to **app-routing.module.ts** and browse to localhost:4200/contact
    h. add <a routerLink="/contact">Contact</a> to **app.component.html**
    i. demo <a **href**="/contact">Contact</a> in **top level index.html**
    j. bootswatch demo
    k. [routerLinkActive]="['active']" on links in **app.component.html**
    l. .navbar-nav a.active in **app.component.css**
17. Routing Extended
    a. Open step17 instructions - deles open step17
    b. <base href="/src/step17/"> in **index.html** (remember conflict with router-introduction base-href!!!)
    c. import router classes and components to **app.ts**
    d. add <router-outlet></router-outlet>s to **app.ts**
    e. create **router-config.ts**
    f. create <button> elements in **dashboard-header.ts** <button (click)="navigate('(section1:section1)')">Graph 1</button>
    g. create navigate() function
    h. uncomment radios and setGraph() function to demo multi outlets
18. Form Builder
    a. import reactiveFormModule into **app.module.ts**
    b. import form classes into **contact.component.ts**
    c. modify class definition in **contact.component.ts**