# **NG course - things to read/do after hours**

# (highlighted - need to do/install on work computer)

# (highlighted – ng cli commands)

My additions (pre course):

* npm install -g @angular/cli
* Ng new <projname>

To create a new (cli) project

* Ng serve

**DAY 1**

* Install nvm (node version manager) 🗸🗸

Nvm for windows: <https://github.com/hakobera/nvmw>

* Read angular style guide

<https://angular.io/guide/styleguide>

* learn typescript

<https://www.typescriptlang.org/play/index.html>

* read attributes vs properties
* ng g c <componentName>

auto create component with cli

defaults for not creating files in .angular-cli.json: 🗸

"defaults": {

"styleExt": "css",

"component": {

"inlineTemplate": true, //dont cerate html file

"inlineStyle": true, //dont create css file

"spec": false, //dont create testing file

"flat": true //dont create in folder

}

* Ctrl-shift-P in VS code to have line on the top for quick commands
* Check how to auto add imports in VS code? Ctrl+.
* Check VS code cheat sheet (auto change attribute to binded attribute)
* Checkout stackblitz – online IDE

<https://stackblitz.com>

* Install Augury in chrome 🗸🗸

for debugging/testing components in the browser

* Read about Shadow DOM

Chrome->settings->show user agent shadow DOM

* Read about 4 API of web components:

<https://www.webcomponents.org/>

* check gitub user for pushing the exercise project – create repo and push project 🗸

<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

* download the resources from github that Nir sends 🗸🗸
* create project todos for exercise 🗸
* install VS code extensions from list 🗸🗸

**DAY 2**

* ng g m <module name>

create new module

* ng g c (optional <modulePath>/)<componentName>
* optional - give a path to the component to create it inside a specific module
* <ng-content></ng-content>

Component out of the box of NG that is included in CommonComponent, allow to add content in certain sections like a master page.

* VS code – ctrl + .

Auto add import to component

* \*ngfor

Out of the box directive to create a for loop on an items collection. It is written on the element that we want to reuse. Syntax:

\*ngFor="let item of items"

* In es6 var is deprecated and now we use let and const
  + Const is used for variables which are not changing
  + Let for all others. Also - let is in the scope of curly braces {} and not in the scope of a function.
* (keypress.enter) or (keypress.escape)

In order to catch enter or escape characters press. This is part of NG, can be customizable to other keys.

* ng g s (optional <modulePath>/)<ServiceName>

creates a service under the specific path. Adds a decorator of @Injectable to the service class (this is convention), but it doesn’t add the service to the providers array in the module, so need to add it manually.

* ng g p (optional <modulePath>/)<PipeName> --export

creates a new pipe class under the specific path with @Pipe decorator. The flag for –export is meant for adding it to the exports array in the module.

* `my name is: ${name} and my age is ${age}`

A new way to concatenate strings with parameters, similar to string format.

**DAY 3**

* [ngClass]="{completed: item.completed}"

Option to add class to an element with ngClass directive. Completed is the class name, the second parameter is a Boolean if it should be added or not.

* ng g d (optional <modulePath>/)<directiveName> --export
* ng g class (optional <modulePath>/)<className>
* <template>

Tag that allows to have html on the page which is not displayed on the page, it can be catched in JS and be manipulated, displayed, added to DOM (a copy of it), add events to it, etc.

* <ng-container #entry></ng-container>

Placeholder for inserting elements to the DOM, this tag is meaningless, NG gives it for our use.

* <https://vsavkin.com/>
* Jsonplaceholder
* RxJS

Library that supply functions to filter and manipulate the events that are returned from the subscribe API of the HttpClientModule.

* Superagent, Axios

APIs that are supplements for the NG Http module, because it is not perfect. These libraries are using promises.

* + Extra reading

<https://www.ng-newsletter.com/>

<https://www.youtube.com/user/ngconfvideos>

<https://www.angularconnect.com/>

<https://developers.google.com/experts/>

<https://toddmotto.com/>

**DAY 4**

* Read about Zone.js

NG change detector library

* this.\_items = [ **...this.\_items**, new Item(title)];

another way to create a new array and add an item – use of spread operator to include all the items that exist in the array, then add the new item.

* Npm i @angular-redux/store redux --save

Two packages that we install when using redux. Use –save flag to save it to packages file.

* <http://ramdajs.com/>

js library for util functions for data structure manipulations, it always return new structure which is good for working with redux.

* Angular IL
* Ng girls
* Angular nights