Codergirl – Frontend

Unit 2 - Class 1

November 23, 2020

Agenda

- Introductions from your course team.
- Goals
- Getting Started
- The Basics
- Ice breaker
- Studio

Chetna

Bayer since 15 years
Worked as web developer (services, b2b, front end), Tech Lead, ScrumMaster, Agile Coach, Developer
Love to travel, workout, paint, go for walks, watch Netflix

Kumbhalgarh Fort is in Rajasthan, India, was built in the 15th century, and has the second largest wall (38km) after the great wall of China. Constructed in the middle of a forest on the foothills of Aravalli ranges, it is surrounded by thirteen hill peaks and is perched at an elevation of 1,914 m.

Abbey

Abbey Northcutt is a nonbinary software developer for the healthcare company Zelis/Redcard. They come from a VERY nontraditional coding background, having graduated from Launchcode's Codergirl Front End track in 2019. Before that, they were a legal assistant, a resident advisor, a publishing assistant, and the guy who hands out maps at the Science Center. Abbey specializes in Angular and Typescript and has had a lot of experience dealing with interesting codebases - their first coding apprenticeship and job was at Intercard, where they programmed software for planning birthday parties. Abbey is passionate about supporting the queer community in the workplace, and they're super stoked to be helping folks with nontraditional coding backgrounds find reliable, rewarding employment.

Gab

Gab is a fintech Product Manager specializing in software product development and building cross-functional teams. Gab strongly believes diverse perspectives are the key to better, smarter problem solving both in tech and the world at large. In her free time Gab is a video game fanatic and voracious reader - anything that allows her imagination to grow is highly valued. She's pretty sure in another life she would be an actual Viking. Feel free to hit Gab up on LinkedIn or Slack to talk more about games, books or Product Management.

Austin

Austin is a self-taught developer and Udacity nano-degree graduate who's been working professionally as a full-stack software engineer for over three years. He got his start programming at a St. Louis based startup and is currently a contributor to St. Louis University's TraffickCam project. This year, he has also been working on software to facilitate tracking and understanding COVID-19 for the organization US Digital Response. When he's not coding or reading about software, Austin likes to spend time outdoors hiking, and also in the kitchen experimenting with new recipes.

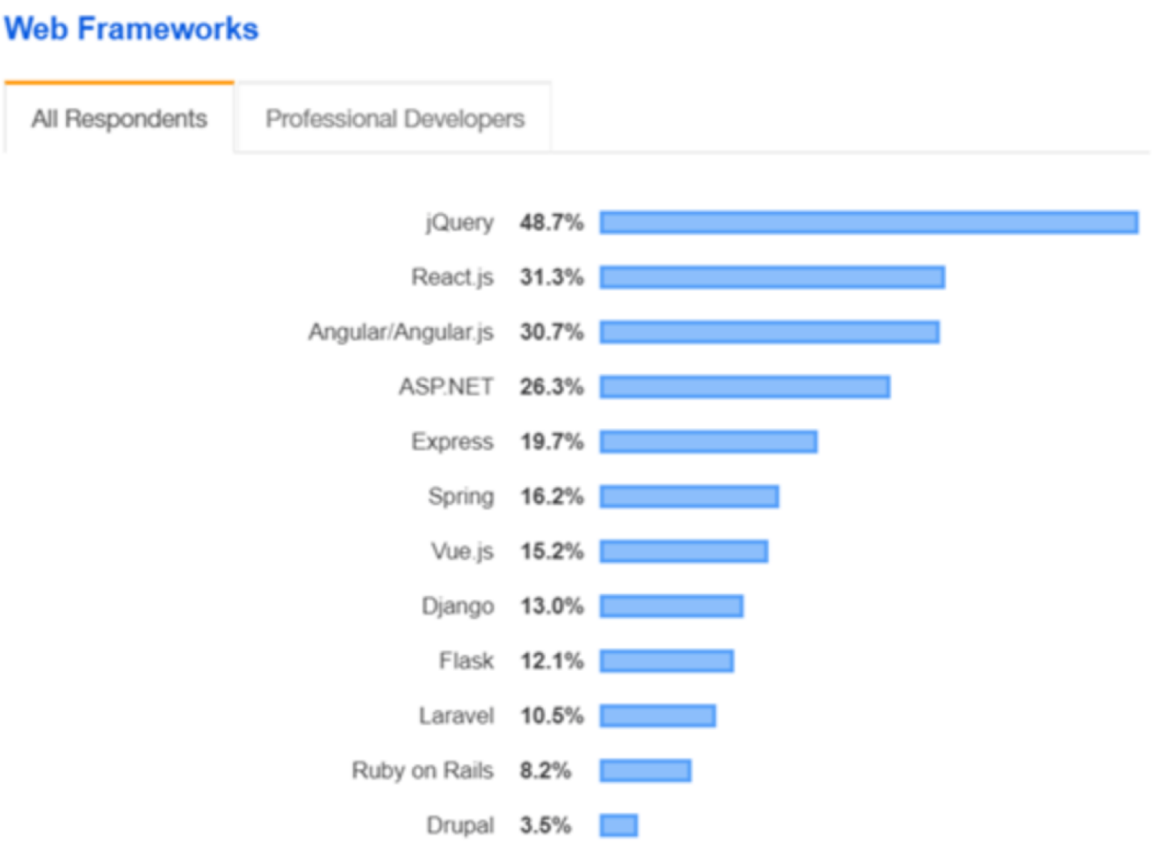
Carrie

Carrie Jones first joined the LaunchCode community in August 2019 as a LC101 student and is currently participating in an apprenticeship at Cognizant Technology Solutions learning MuleSoft. Prior to this career shift, Carrie spent the better part of 20 years wearing multiple hats at local nonprofits and built experience in finance, administration, grant writing/reporting, data management, graphic design, and more — all of which built important skills for a tech career like resourcefulness and creative problem-solving. Carrie loves mixing all things creative and analytical, which is why she's so happy to be mentoring students in the Front End track!

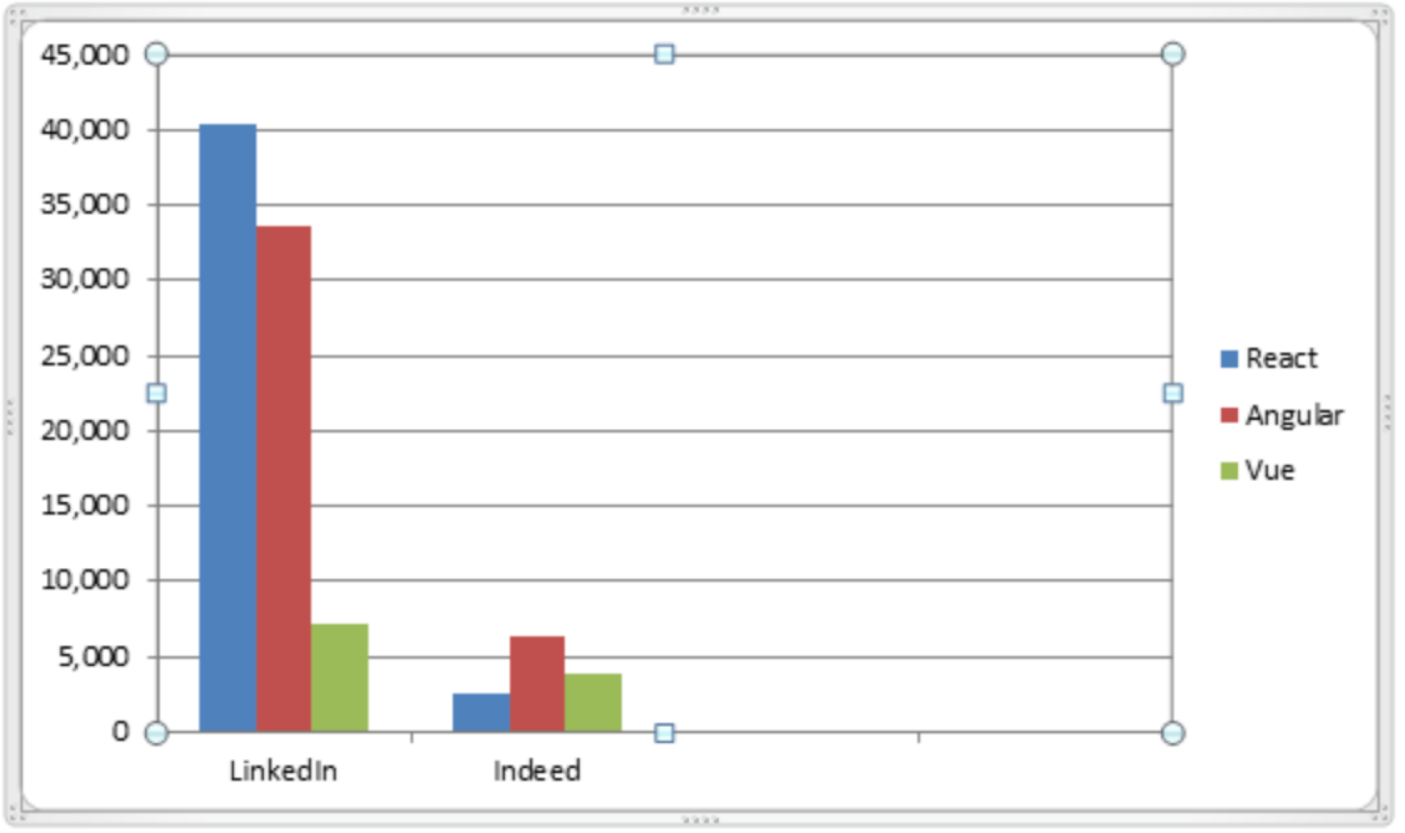
Mandy

Mandy Schultz loves to talk, but not about herself. Thankfully, she gets to meet plenty of awesome new people to talk about as a Candidate Engagement Manager for LaunchCode's CoderGirl program! Past professional experience includes working in the tech consulting/recruiting space in Seattle getting rained on constantly; and has previously served as Director of Events on the Association of Latino Professionals for America (ALPFA) Board of Directors. An empathetic leader, Mandy is an ally-in-training that's been advocating for equity in the workplace for the past several years.

Frameworks


Web Frameworks
All Respondents
Professional Developers
jQuery 48.7%
React.js 31.3%
Angular/Angular.js 30.7%
ASP.NET 26.3%
Express 19.7%
Spring 16.2%
Vue.js 15.2%
Django 13.0%
Flask 12.1%
Laravel 10.5%
Ruby on Rails 8.2%
Drupal 3.5%

Jobs


45,000
40,000
35,000
30,000
25,000
20,000
15,000
10,000
5,000
0
LinkedIn
Indeed
React
Angular
Vue

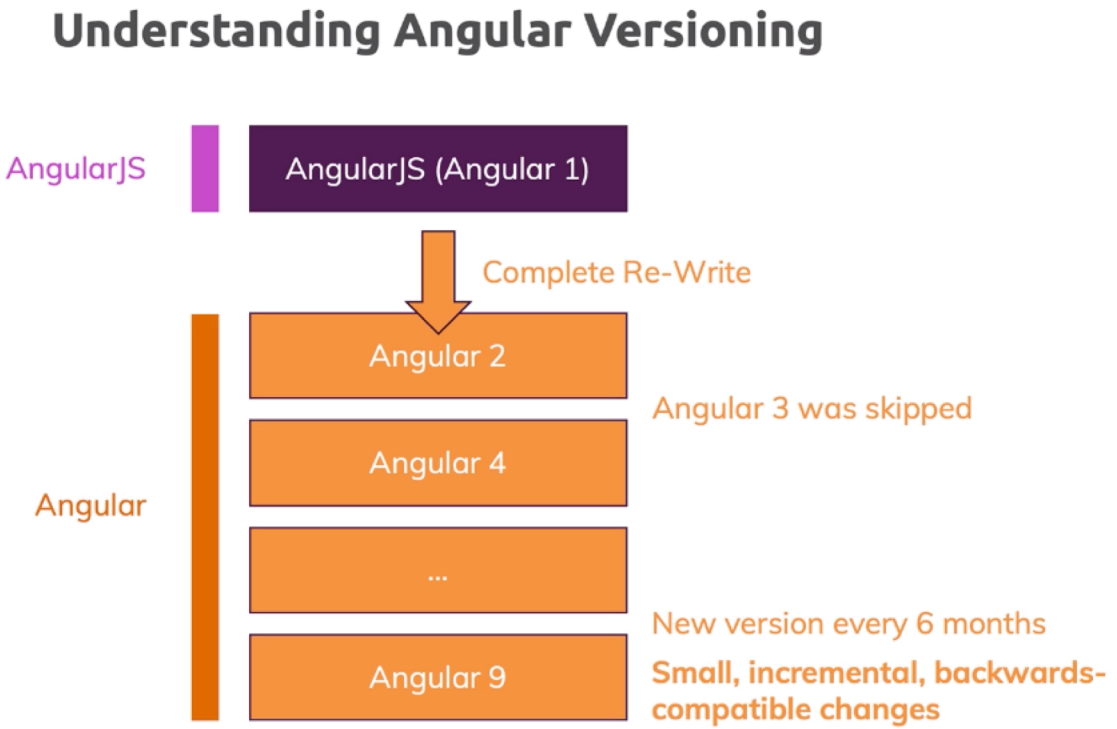
Brief Comparison between Angular vs React vs Vue

	Angular	React	Vue
Type	A Framework	Library to build UI	A library
Why Choose	If you want to use TypeScript	If you want to go for "everything-is-JavaScript" approach	Easy JavaScript and HTML
Founders	Powered by Google	Maintained by Facebook	Created by Former Google Employee
Initial Release	September 2016	March 2013	February 2014
Application Types	If you want to develop Native apps, hybrid apps, and web apps	If you want to develop SPA and mobile apps	Advanced SPA and started supporting Native apps
Ideal for	If you want to focus on large-scale, feature-rich applications	Suitable for modern web development and native-rendered apps for iOS and Android	Ideal for web development and single-page applications
Learning Curve	A steep learning curve	A little bit easier than Angular	A small learning curve
Developer-friendly	If you want to use the structure-based framework	If you want to have flexibility in the development environment	If you want to have separation of concerns
Model	Based on MVC (Model-View-Controller) architecture	Based on Virtual DOM (Document Object Model)	Based on Virtual DOM (Document Object Model)
Written in	TypeScript	JavaScript	JavaScript
Community Support	A large community of developers and supporters	Facebook developers community	Open-source project sponsored through crowd-sourcing
Language Preference	Recommends the use of TypeScript	Recommends the use of JSX – JavaScript XML	HTML templates and JavaScript
Popularity	Widely popular among developers	More than 27,000 stars added over the year	More than 40,000 stars added on GitHub during the year
Companies Using	Used by Google, Forbes, Wix, and weather.com	Used by Facebook, Uber, Netflix, Twitter, Reddit, Paypal, Walmart, and others	Used by Alibaba, Baidu, GitLab, and others

Angular

- Angular is a framework which allows you to create reactive, single page applications.
- Single page application – we visit diff pages but it’s still just one page and one html file
- JS code – DOM changes, changes HTML code are rendered in browser
- Reactive – web applications that look and feel like mobile applications and fast
- Data is loaded in background

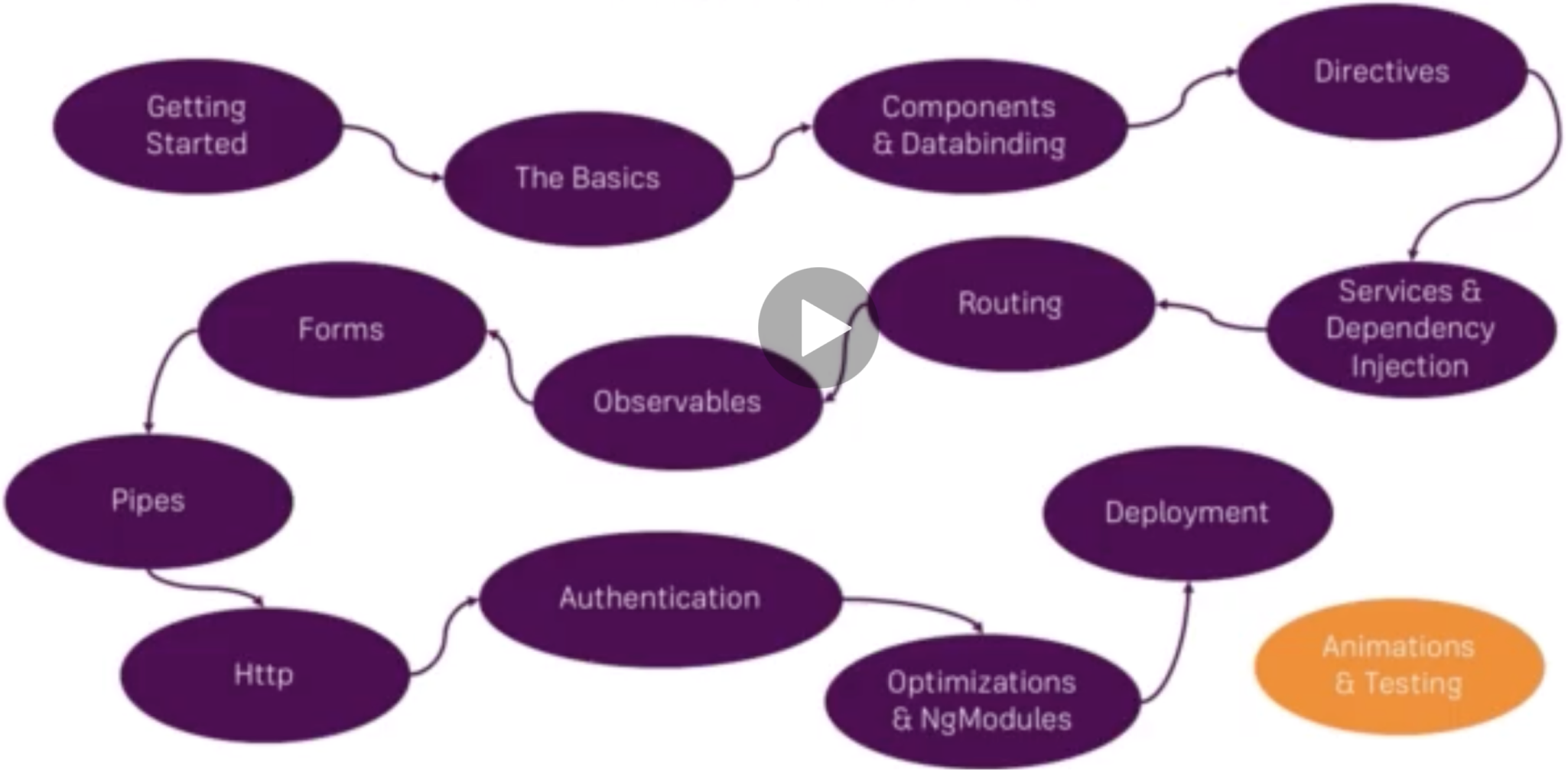
Angular



https://angular.io/docs

https://angular.io/tutorial

Course Structure



How to get the Most out of the Course

Watch the Videos

Do the Assignments

Do the Course Project

Ask in Q&A, but ...

...also answer in Q&A!

Admin items

- GitHub and submitting assignments.
- Studio will follow completing the assignments in class that follow the course sections. If you have completed all the assignments in class, use the studio for discussion or pairing with someone in your group.
- Since in the past mentors have reported that studio time has been lacking, I will try to wrap up lecture by 6:45 pm and latest 7 pm.
- Your own project – pair up with someone! We will share more on both of those on Dec 2.

Project setup and first app using CLI

Need NodeJS and NPM – used by CL, TS, Use Chrome

- npm install –g @angular/cli@latest (might need sudo)
- ng –version
- In your projects folder run ng new my-dream-app
  - No to stricter type checking and Routing and CSS for styling
- cd my-dream-app
- ng serve, keep this running Ctrl + C to kill
- You can replace the app.component.html
- IDE of choice – VSC/ATOM/IntelliJ/Eclipse

Basic setup using bootstrap

- npm i --save bootstrap@3
- It will be in node_modules and we need to make angular aware of it
- angular.json has

```
"styles": [

"src/styles.css",

"node_modules/bootstrap/dist/css/bootstrap.min.css"

],
```

Angular App loading

- index.html
- App.component.ts
- ng serve injects the scripts in to the index.html. You can inspect and see those.
- main.ts gets executed first
- Bootstrap has [AppComponent] that lists all the components to start in app.module.ts
- Angular parses the app-root, understands it

Components

Components are a key feature in Angular

- template, HTML code, styling, business logic
- split up complex application/webpage into reusable parts
- selectors for other components will be added to the app.component.html file
- Folder name = component name
- name.component.ts
- We use the command to generate all the 4 files.

CLI to create

Emmet plugin to writing html

cli to create a component inside another component

ng generate component servers OR

ng g c servers

It updated the app.module.ts!

You can add the selector to the app.component.html!

Can inspect to see the hierarchy!

Template

- Inline template inside ts file
- Wrap the template with single quotes or backtick
- If the html code isn’t too long inline works
- but if more than a few lines, use external file.

Styling

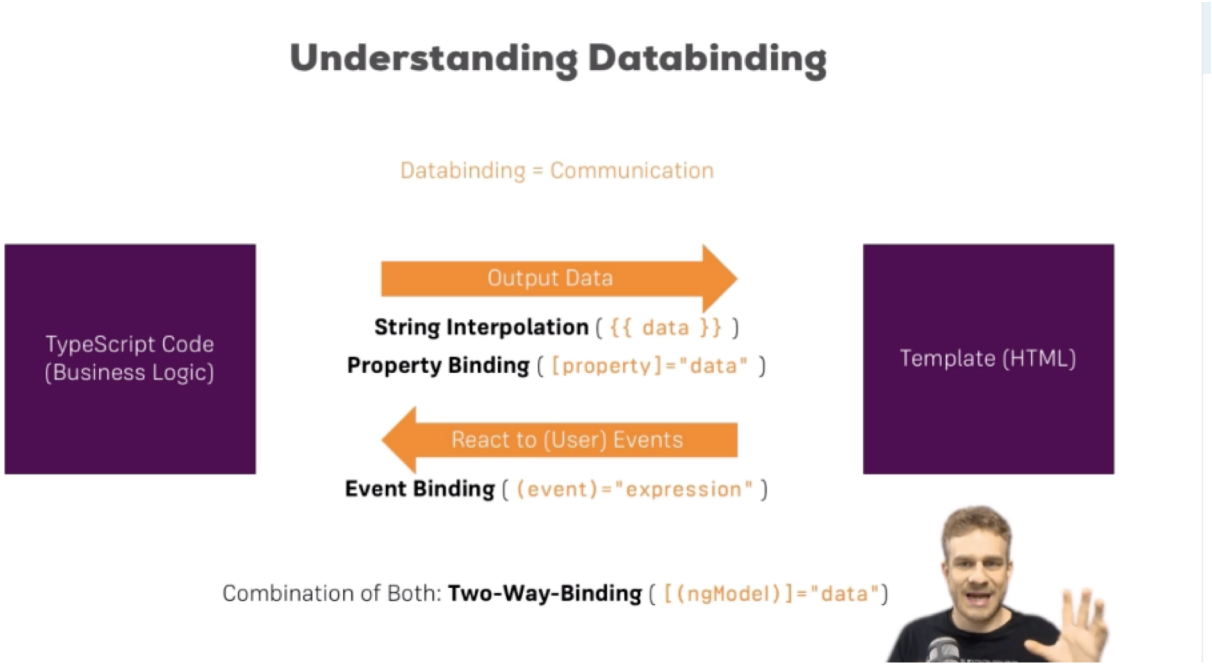
- app.component.css file to write some css code
- Can be inline or external file(s)
- styles[] or styleUrls
- More code = external file
- Short = inline

Selector

- Unique in element selector
- Can be an attribute by using []
- So for example it will be used as
- <div app-servers>
- Attribute selector has limitations

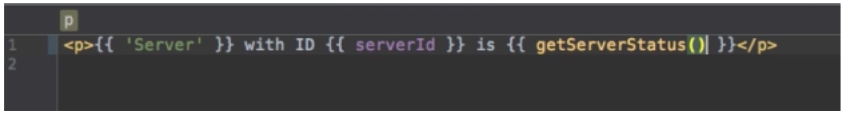
Databinding

- Communication between TS and html



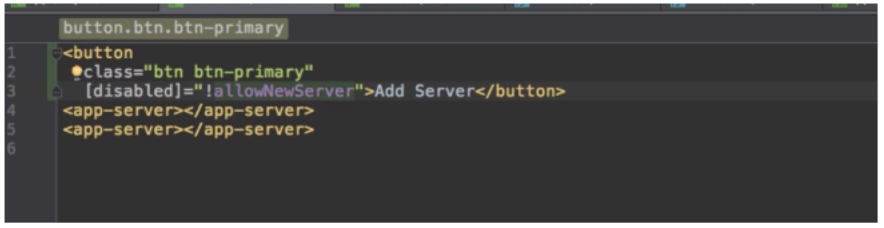
String interpolation

- Derive them in ts code, write a ts expression within {{ }}
- serverId: number = 10;
- serverStatus: string = ‘offline’;
- {{ serverId }} or {{ ‘server’ }} or a {{ myFunction() }}
- {{ serverStatus }}



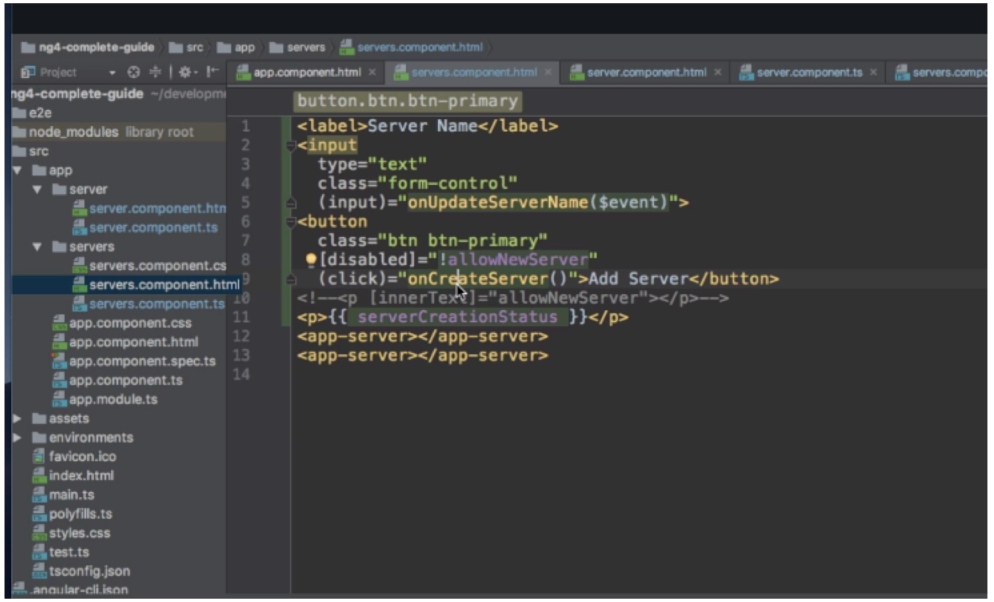
Property binding

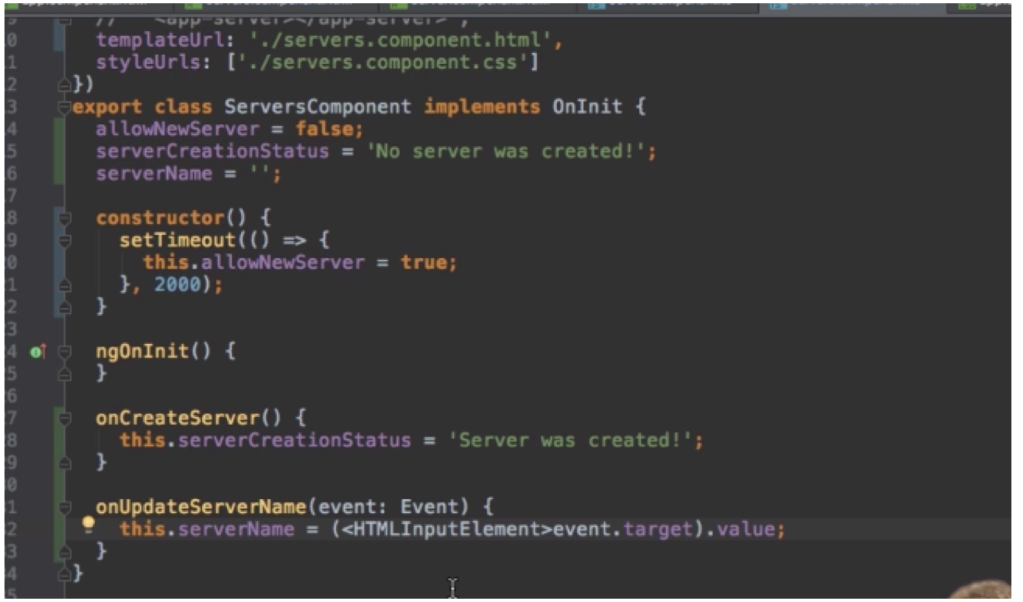
- [disabled]=“allowNewServer” or ts expression.
- Print something = use String Interpolation
- Just modify might be property binding
- Don’t mix use them



Event Binding

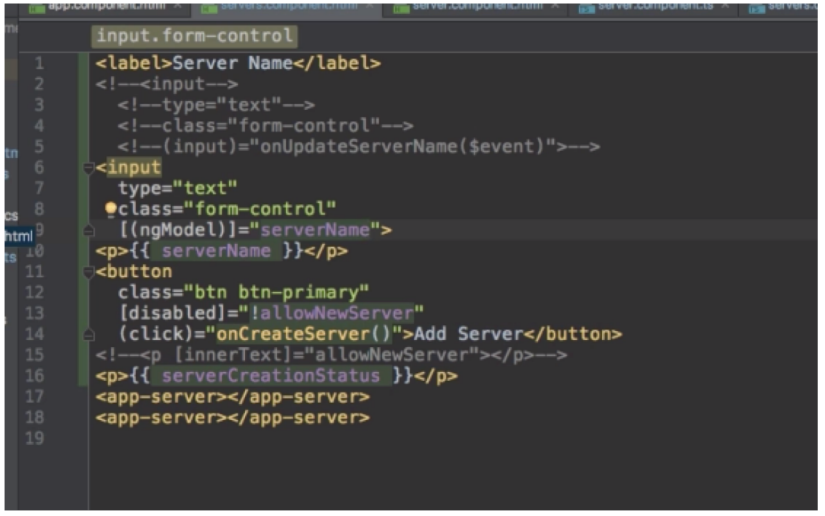
- onCreateServer (symbolizes the event will trigger this method)
- Events are specified using ()
- (click)=”onCreateServer()”
- (input)=“onUpdateServerName($event)”

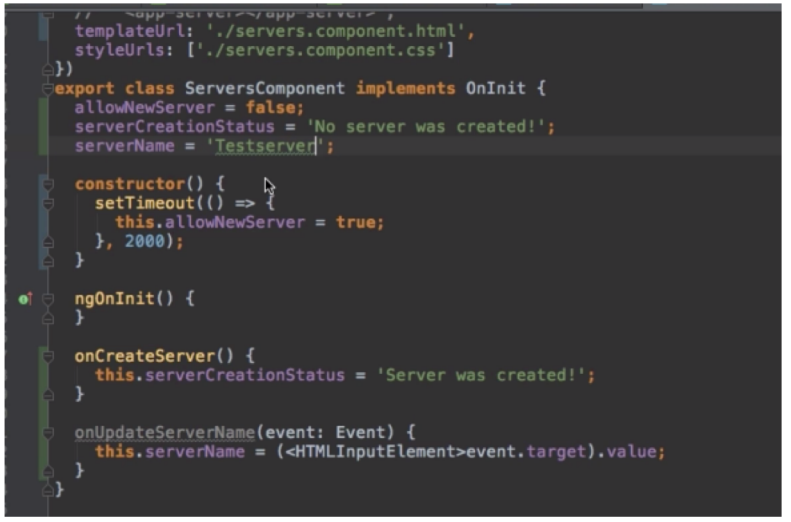




Two-way-Databinding

- [(ngModel)]=“serverName”





Built in Directives

- Instructions in DOM (components are directives)
- *ngIf =“booleanExpression; else noServer” is structural directive(add a tag or not)
- For Else, # is local reference or marker
- <ng-template #noServer>Test</ng-template>

```
p
<label>Server Name</label>
<!--<input-->
  <!--type="text"-->
  <!--class="form-control"-->
  <!--(input)="onUpdateServerName($event)">-->
<input
  type="text"
  class="form-control"
  [(ngModel)]="serverName">
<!--<p>{{ serverName }}</p>-->
<button
  class="btn btn-primary"
  [disabled]="!allowNewServer"
  (click)="onCreateServer()">Add Server</button>
<!--<p [innerText]="allowNewServer"></p>-->
<!--<p>{{ serverCreationStatus }}</p>-->
<p *ngIf="serverCreated">Server was created, server name is {{ serverName }}</p>
<app-server></app-server>
<app-server></app-server>
```

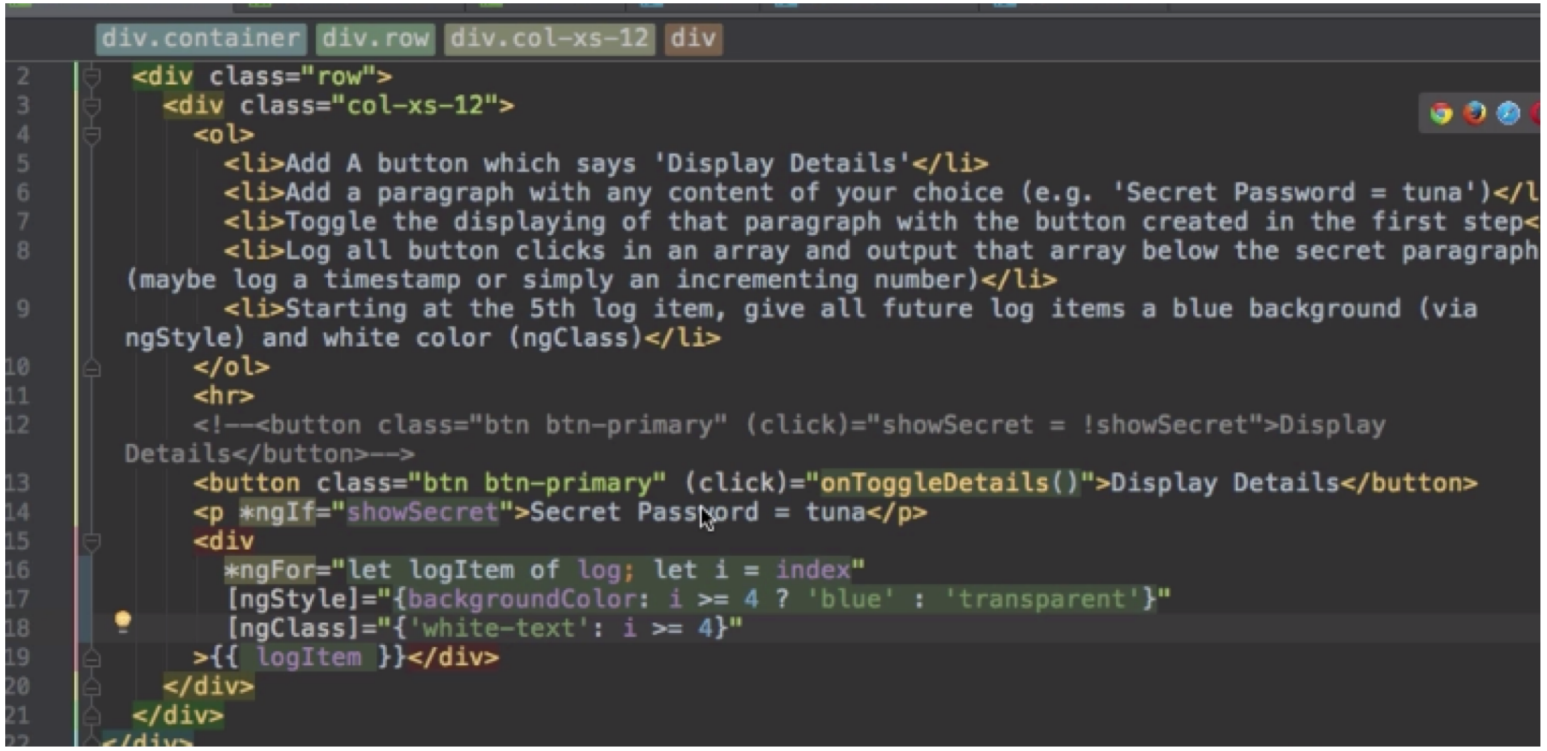
```
app.component.html | servers.component.html | app.component.css | server.component.html | server.component.ts
<label>Server Name</label>
<!--<input-->
  <!--type="text"-->
  <!--class="form-control"-->
  <!--(input)="onUpdateServerName($event)">-->
<input
  type="text"
  class="form-control"
  [(ngModel)]="serverName">
<!--<p>{{ serverName }}</p>-->
<button
  class="btn btn-primary"
  [disabled]="!allowNewServer"
  (click)="onCreateServer()">Add Server</button>
<!--<p [innerText]="allowNewServer"></p>-->
<!--<p>{{ serverCreationStatus }}</p>-->
<p *ngIf="serverCreated; else noServer">Server was created, server name is {{
  serverName }}</p>
<ng-template #noServer>
  <p>No server was created!</p>
</ng-template>
<app-server></app-server>
<app-server></app-server>
```

ngStyle

- ngStyle is an attribute directive with property binding
- <p [ngStyle]="{backgroundColor: getColor()}"
- Dynamically update style!
- ngClass is also attribute directive
- [ngClass]="{online: serverStatus === 'online'}"

ngFor

- *ngFor="let server of servers"



Questions?

Ice Breaker!