**Introduction to Angular**

1. What is angular
   1. A UI framework
   2. A component-based framework for building scalable app
   3. Provided collection of libraries to cover features like Routing, Forms, etc.
2. Frameworks Vs libraries
   1. Frameworks:
      1. Generally, contains set of libraries
      2. We can extend the framework
      3. Some examples: Angular, Ionic, .NET, ExpressJS
   2. Libraries:
      1. Can generally perform specific tasks
      2. Can combine multiple libraries to build apps
      3. E.g., React, MomentJS, Lodash
3. Features of Angular
   1. Template
   2. Data binding
   3. Forms
   4. Routing
   5. Observables
   6. PWA

**Introduction to Typescript**

1. Intro to typescript
   1. A strongly typed programing language
   2. Created and maintained by Microsoft
   3. Superset of JavaScript
2. Why typescript
   1. Compiles down to JS
   2. Keeps your code evergreen
   3. Supported by all major libraries and frameworks
3. Type safety
   1. Keeps your application free from type errors
   2. Languages like C#, Java are type-safe languages.
   3. Keeps your js code free from undefined and null values.
   4. In typescript types are stripped when your code is converted to JS
4. Superset to JS
5. Installing and creating first typescript program. (Stopping here: <https://youtu.be/3qBXWUpoPHo?t=1165> )
   1. Installing NodeJS and npm
   2. npm init to create a package.json
   3. npm i typescript
   4. tsc –init
   5. tsc compilation

**Single page applications**

1. what is SPA?
   1. SPA stands for single page applications
   2. Most modern frameworks support SPA
   3. SPA does not make requests to server for every URL requests
2. How angular helps?

**Software installation**

1. Installing NodeJS
2. Installing VS code

**Data types**

[ Stopping here: <https://youtu.be/3qBXWUpoPHo?t=2559> ]

1. Data types [ .filter() function, .find() function, Array.prototype.reduce() ]
   1. String
   2. Number
   3. Boolean
   4. Array
   5. enum
   6. tuple
   7. any
   8. void
   9. never

[paused here: <https://youtu.be/3qBXWUpoPHo?t=3584> ]

1. Type inferences
2. Union and intersection types
3. Generics
4. Literal types