

Typescript syllabus

Overview:

Angular is a framework for building client applications in HTML and either JavaScript or a language like Typescript that compiles to JavaScript. The framework consists of several libraries, some of them core and some optional.

You write Angular applications by composing HTML templates with Angularized markup, writing component classes to manage those templates, adding application logic in services, and boxing components and services in modules.

Then you launch the app by bootstrapping the root module. Angular takes over, presenting your application content in a browser and responding to user interactions according to the instructions you have provided.

Course Objectives:

- ❖ Develop modern, complex, responsive and scalable web applications with Angular 4
- ❖ Use their gained, deep understanding of the Angular 4 fundamentals to quickly establish themselves as frontend developers
- ❖ Fully understand the architecture behind an Angular 4 application and how to use it
- ❖ Create single-page applications with one of the most modern JavaScript frameworks out there

Pre-requisite / Target Audience:

- ❖ JavaScript OOP basics (constructor pattern, inheritance, polymorphism, this object)
- ❖ ES6 - class, extend, arrow function, import
- ❖ Typescript is identical to ES6, but has few more important features such as Interface, Type system & Decorators

Module 1: Introduction

In this module, we will learn what typescript is and what the benefits of typescript are over other scripting languages. We can also learn how to install and setup the environment and how we can create our first example using typescript.

- ❖ What is Typescript?
- ❖ Benefits of Typescript
- ❖ Setup the Environment
- ❖ First Typescript Example

Module 2: Data Types and Variables

In this module, we will learn what the datatypes that are used in typescript are, and we will introduce into a new data type **let**.

- ❖ Basic Data Types
- ❖ Arrays

- ❖ Tuples
- ❖ Enum
- ❖ Any and void
- ❖ null and undefined
- ❖ Type Inference
- ❖ Type Casting
- ❖ Difference between let and var
- ❖ Const declaration

Module 3: Destructuring & Spread

In this module, we will learn how to extract the data from objects and arrays in a convenient way using destructuring and we can learn how to break arrays and objects into components using spreads.

- ❖ Array Destructuring
- ❖ Object Destructuring
- ❖ Mixed Destructuring
- ❖ Property renaming
- ❖ Default Values
- ❖ Spreads

Module 4: Working with Classes

In this module, we will learn how to write classes and how to create objects for our classes like we do in other programming languages like java, C# etc. and we can also learn how we can fulfil OOPS concepts using typescript.

- ❖ Writing and Using Classes
- ❖ Constructor method
- ❖ Inheritance of classes
- ❖ Type casting
- ❖ Type Assertion
- ❖ Static Properties
- ❖ Abstract class

Module 5: Working with Interfaces

In this module, we will learn how to work with interfaces in typescript.

- ❖ Interface Declaration and Initialization with an object
- ❖ Duck Typing
- ❖ Interface Implementation by class
- ❖ Interface having Optional Property
- ❖ Class extending another class and also implementing an Interface
- ❖ Excess Property Checks

- ❖ Indexable Types
- ❖ Extending Interfaces

Module 6: Function Types

In this module, you will learn about typescript functions and one more new operator called as rest

- ❖ Optional and Default Parameters
- ❖ Rest Parameters
- ❖ Function Overloading
- ❖ Function Types

Module 7: Modules

In this module we will learn how to export and import between module. And how can we group the required modules into a namespace.

- ❖ Introduction
- ❖ Exporting and Importing of Modules
- ❖ Re-Export of Modules
- ❖ Default Exports

Module 8: Namespaces

- ❖ Introduction
- ❖ Syntax with Example
- ❖ Compiling multiple TS files as one JS file

Module 9: Ambients

In this module, we will learn how to make use of our javascript functions with typescript.

- ❖ Definition
- ❖ Syntax
- ❖ Examples

Angular 4 Syllabus

Module 1: Introduction

In this module, you will learn what is angular, what are the major differences between Angular 1.X, Angular 2 and Angular 4. We will also learn how to setup in the local environment and a small introduction about typescript.

- ❖ What is Angular?
- ❖ Angular (vs) Angular2 (vs) Angular4
- ❖ Setup for local environment
- ❖ What is Typescript?

Module 2: Angular Architecture

In this module, you will learn how angularjs and angular work and flow of the angular application in depth.

- ❖ Basic Building Blocks of Angular Applications

Module 3: Directives and Templates

In this module, you will learn about directives, directives are the classes that can change the behaviour or appearance of the components by using CSS Classes, CSS Styles and events Here you will find how to use directives in our application.

- ❖ Adding Properties to Components and Interpolation
- ❖ Creating a class for data (Model object)
- ❖ Template Expressions
- ❖ Working with Arrays and Build-in Directives
- ❖ Repeating Directive (NgFor) o Conditional Directives (NgIf, NgSwitch)
- ❖ *vs <template>
- ❖ External HTML Template File

Module 4: Data Binding

In this module, you will learn how many kind of databinding and we will discuss each of them in this chapter

- ❖ Binding properties and Interpolation
- ❖ One-way Binding / Property Binding
- ❖ Event Binding
- ❖ Two-way Binding
- ❖ Two-way binding with NgModel
- ❖ Attribute Binding

Module 5: Styles Binding In Components

In this module, Angular applications are styled with regular CSS. That means we can apply everything we know about CSS stylesheets, selectors, rules, and media queries to our Angular applications directly, this module will help you to how to apply css to our page

- ❖ Style and Class Bindings
- ❖ Built-In Directives - NgStyle & NgClass
- ❖ Using Component Styles
- ❖ Special selectors
- ❖ Loading Styles into Components

Module 6: Advanced Components Features

In this module we will learn one of the major concept in angular4 i.e., components and we can learn how to create dynamic component using **ngComponentOutlet**.

- ❖ What are Components?
- ❖ Understanding Components life cycle hooks
- ❖ Creating and using components
- ❖ Dynamic components using ngComponentOutlet

Module 7: Template Driven Forms

In this module, you will learn in most of the frontend applications we use HTML forms for grouping HTML elements and we provide validations using HTML5 attributes like required, minlength, maxlength, pattern etc.

- ❖ Two-way data binding with ngModel
- ❖ Add Custom CSS for Visual Feedback.
- ❖ Show and Hide Validation Error messages
- ❖ Resetting the form.
- ❖ Resetting form state
- ❖ Submit the form with ngSubmit

Module 8: Model Driven Forms

Model driven forms are more powerful and easy to do functionalities, which are complex when using template driven forms.

- ❖ Model Driven / Reactive Forms
- ❖ Form with @ViewChild
- ❖ Validations
- ❖ Custom Validators
- ❖ Submitting and Resetting form

Module 9: Working with Pipes

In this module, you will learn about pipe, a pipe takes in data as input and transforms it to a desired output

- ❖ Built-in Pipes
- ❖ Using parameters
- ❖ Chaining Pipes
- ❖ Custom Pipes
- ❖ Pipes and Change Detection
- ❖ Pure and Impure pipes

Module 10: Custom Attribute and Validators

In this module, you will learn about attribute and validators, Using custom attribute directive, we can change the color, back-ground, font-size etc., of the HTML host element by using ElementRef

- ❖ Custom Attribute Directives
- ❖ Using HostListener
- ❖ Using HostBinding
- ❖ Example2: Custom Validators

Module 11: Dependency Injection

In this module, we will learn Dependency Injection (DI) is the software design pattern in which a class receives its dependencies from external sources rather than creating them itself. In addition, we will learn about very interesting topic Dependency Injection in angular.

- ❖ Understanding Dependency Injection
- ❖ Understanding DI in Angular Framework
- ❖ Reflective Injector
- ❖ Exploring Provider
- ❖ Types of Tokens
- ❖ Types of Dependencies
- ❖ Configuring DI using Providers
- ❖ Implementing DI in Angular

Module 12: Services

In this module, you will learn about services, In angular services are reusable classes which can be injected in components when it's needed. Using a separate service keeps components lean and focused on supporting the view, and makes it easy to unit test components with a mock service

- ❖ Services
- ❖ Service Using a Service
- ❖ \$http Services

- ❖ Get Request

Module 13: Routing

In this module we will learn the introduction for routing in angular and how to navigate between views, how to do parameterized routing.

- ❖ Introduction
- ❖ Configuring and Navigating
- ❖ Parameterized routes
- ❖ Nested (or) Child Routes

Module 14: Module

In this module, you will learn module, An NgModule class is adorned with @NgModule decorator function this will tell the angular application how to compile and run the module code.

- ❖ Root App module
- ❖ Ahead-Of-Time(AOT) Compilation
- ❖ Feature modules
- ❖ Lazy Loading a Module
- ❖ Shared Module

Module 15: Crud Operations Using Http Service

In this module, we will create an application with end-to-end start from server to client, getting response and requests using HTTP service.

- ❖ About Http Services
- ❖ Step 1: Setup Web Application project
- ❖ Step 2: Configuring Angular Module to use HTTP and Json Services.
- ❖ Step 3: Fetch data with http.get
- ❖ Step 4: Sending data to Server / Adding Employee details using http.post

Real-time Project involving most of the above concepts with following will be provided

- ❖ Product Abstract Document
- ❖ Requirement Specification Document
- ❖ Step-by-Step procedure for building the project from ground up
- ❖ Complete Source Code
- ❖ Database Script with Sample data
- ❖ Instructions to Setup the Project on a Development box
- ❖ Instruction to Deploy the project on Production Box / Microsoft Azure

At the end of the course participants will be able to

1. Build native mobile apps for Android, iOS and using Angular4
2. Understand the fundamentals of Angular Forms and its architecture
3. Present data in beautiful, interactive lists
4. Build forms and setting pages
5. Implement Single page application(SPA)