

**Day 1**

- **Key Aspects of JavaScript required for Angular**
  - Object Concepts: Prototype-Based Inheritance
  - Function Concepts: Scope, Closure, Context, Call, Bind, Callbacks
  - Event Loop and concept of Next Tick
  - ES5 vs ES6 vs Typescript
  - Typescript Fundamentals: String Typing, Classes, Inheritance, Interface, Access Modifiers
- **Overview**
  - Issues with Server-Side Web Technologies and what Client-Side web technology offer
  - Component Driven Programming of Angular
  - Model-View-ViewModel [MVVM] Design Pattern and how components implement it in Angular
  - Modules in Angular
  - How Single Page Application concept is implemented in Angular
- **Development Environment Setup**
  - Node Installation
  - Creating Angular Project using Angular-CLI
  - Understanding various configurations
  - Understanding the bootstrapping process
- **Simple Component**
  - Creating a Component Class using TypeScript Class
  - Decorating it using @Component Decorator
  - Defining the structure [html] of the component using template OR templateUrl attributes
  - Defining the styling [CSS] of the component using styles OR styleUrls attributes [Static Styles]
  - Determining the correct View Encapsulation Mode

- **Data Binding**
  - Attribute Binding
  - Event Binding
  - Interpolation
  - Two Way Binding Using all the above bindings within a component
- **Directives**
  - Attribute Directives
    - ngModel
  - Structural Directives
    - \*ngFor
    - \*ngIf
  - Components are Directives!

## **Day 2**

- **Nested Components (Intercomponent Communication)**
  - Hosting a component within another component
  - Understanding the Component Tree
  - Passing data from Parent to Child Component using @Input Decorator
  - How can Parent act on Child Events?
  - Concept of Containers
  - Content Projection
- **Client-Server Communication**
  - Using the Http Service
  - Working with Observables [RxJS]
  - Issuing GET and POST requests
- **Services**
  - Purpose of a Service
  - Defining a Service
  - Injecting a Service into a Component [Dependency Injection]
  - Service for a Module or for a Component Tree

- **Custom Directives (Customizations)**
  - Defining our own directives for scenarios which cannot be achieved with inbuilt directives
  - Concept of Content Projection
  - Achieving Content Projection using ng-content
  - Multiple ng-content

### **Day 3**

- **Inbuilt Pipes**
  - What are pipes?
  - Different types of inbuilt pipes
- **Custom Pipes (Customizations)**
  - Defining our own pipes for scenarios which cannot be achieved with inbuilt pipes
  - Working Samples
- **Using RxJS (Customizations)**
  - Understanding use of RxJS Observables
  - Creating a Notification Component using Angular and Observables
- **Routing**
  - Using the Router Service
  - Defining Top Level [Free] Routes
  - Defining Nested Routes
  - Configuration of “Router Link” and “Router Outlet”
  - Creating Menu Navigation
  - Programmatically changing Routes
- **Testing**
  - Unit Testing your code with Karma and Jasmine