

HTML

- › HTML Introduction
- › HTML Editors
- › HTML Basic
- › HTML Elements
- › HTML Attributes
- › HTML Headings
- › HTML Paragraphs
- › HTML Styles
- › HTML Formatting
- › HTML Quotations
- › HTML Comments
- › HTML Colors
- › HTML CSS
- › HTML Links
- › HTML Images
- › HTML Favicon
- › HTML Tables
- › HTML Lists
- › HTML Block & Inline
- › HTML Classes
- › HTML Id
- › HTML Iframes
- › HTML JavaScript
- › HTML File Paths
- › HTML Head
- › HTML Layout
- › HTML Responsive
- › HTML Semantics
- › HTML Style Guide

CSS

- › CSS Introduction
- › CSS Syntax
- › CSS Selectors
- › CSS How To
- › CSS Comments
- › CSS Colors
- › CSS Backgrounds
- › CSS Borders
- › CSS Margins
- › CSS Padding
- › CSS Height/Width
- › CSS Box Model
- › CSS Outline
- › CSS Text
- › CSS Fonts
- › CSS Icons
- › CSS Links
- › CSS Lists
- › CSS Tables
- › CSS Display
- › CSS Max-width
- › CSS Position
- › CSS Z-index
- › CSS Overflow
- › CSS Float
- › CSS Inline-block
- › CSS Align
- › CSS Combinators
- › CSS Pseudo-class
- › CSS Pseudo-element
- › CSS Opacity

Bootstrap

- BS Get Started
- BS Grid Basic
- BS Typography
- BS Tables
- BS Images
- BS Jumbotron
- BS Wells
- BS Alerts
- BS Buttons
- BS Button Groups
- BS Glyphicons
- BS Badges/Labels
- BS Progress Bars
- BS Pagination
- BS Pager
- BS List Groups
- BS Panels
- BS Dropdowns
- BS Collapse
- BS Tabs/Pills
- BS Navbar
- BS Forms
- BS Inputs
- BS Inputs 2
- BS Input Sizing
- BS Media Objects
- BS Carousel
- BS Modal
- BS Tooltip
- BS Popover
- BS Scrollspy
- BS Affix
- BS Filters

Javascript

- JS Introduction
- JS Where To
- JS Output
- JS Statements
- JS Syntax
- JS Comments
- JS Variables
- JS Let
- JS Const
- JS Operators
- JS Arithmetic
- JS Assignment
- JS Data Types
- JS Functions
- JS Objects
- JS Events
- JS Strings
- JS String Methods
- JS String Search
- JS String Templates
- JS Numbers
- JS Number Methods
- JS Arrays
- JS Array Methods
- JS Array Sort
- JS Array Iteration
- JS Array Const
- JS Dates
- JS Date Formats
- JS Date Get Methods
- JS Date Set Methods
- JS Math
- JS Random
- JS Booleans

- JS Comparisons
- JS If Else
- JS Switch
- JS Loop For
- JS Loop For In
- JS Loop For Of
- JS Loop While
- JS Break
- JS Iterables
- JS Sets
- JS Maps
- JS Typeof
- JS Type Conversion
- JS Bitwise
- JS RegExp
- JS Errors
- JS Scope
- JS Hoisting
- JS Strict Mode
- JS this Keyword
- JS Arrow Function
- JS Classes
- JS Modules
- JS JSON
- JS Debugging
- JS Style Guide
- JS Best Practices
- JS Mistakes
- JS Performance
- JS Reserved Words

JQuery

- jQuery Intro
- jQuery Get Started
- jQuery Syntax
- jQuery Selectors
- jQuery Events
- jQuery Effects
- jQuery Hide/Show
- jQuery Fade
- jQuery Slide
- jQuery Animate
- jQuery stop()
- jQuery Callback
- jQuery Chaining
- jQuery HTML
- jQuery Get
- jQuery Set
- jQuery Add
- jQuery Remove
- jQuery CSS Classes
- jQuery css()
- jQuery Dimensions
- jQuery Traversing
- jQuery Traversing
- jQuery Ancestors
- jQuery Descendants
- jQuery Siblings
- jQuery Filtering
- jQuery AJAX
- jQuery AJAX Intro
- jQuery Load
- jQuery Get/Post

Module 1 – Getting Started

- › Course Introduction
- › What is Angular?
- › Angular vs Angular 2 vs Latest Angular Version
- › CLI Deep Dive & Troubleshooting
- › Project Setup and First App
- › Editing the First App
- › What is TypeScript?
- › A Basic Project Setup using Bootstrap for Styling

Module 2 – The Basics

- › How an Angular App gets Loaded and Started
- › Creating a New Component
- › Understanding the Role of AppModule and Component Declaration
- › Using Custom Components
- › Creating Components with the CLI & Nesting Components
- › Working with Component Templates
- › Working with Component Styles
- › Fully Understanding the Component Selector
- › What is Databinding?
- › String Interpolation
- › Property Binding
- › Property Binding vs String Interpolation
- › Event Binding
- › Bindable Properties and Events
- › Passing and Using Data with Event Binding
- › Two-Way-Databinding

- › Combining all Forms of Databinding
- › Understanding Directives
- › Using ngIf to Output Data Conditionally
- › Enhancing ngIf with an Else Condition
- › Styling Elements Dynamically with ngStyle
- › Applying CSS Classes Dynamically with ngClass
- › Outputting Lists with ngFor

Module 3 - Debugging

- › Understanding Angular Error Messages
- › Debugging Code in the Browser Using Sourcemaps
- › Using Augury to Dive into Angular Apps

Module 4 - Components & Databinding Deep Dive

- › Splitting Apps into Components
- › Property & Event Binding Overview
- › Binding to Custom Properties
- › Binding to Custom Events
- › Understanding View Encapsulation
- › Using Local References in Templates
- › Getting Access to the Template & DOM with @ViewChild
- › Projecting Content into Components with ng-content
- › Understanding the Component Lifecycle
- › Lifecycle Hooks and Template Access
- › Getting Access to ng-content with @ContentChild
- › Recap and Section Assignment

Module 5 - Directives Deep Dive

- › ngFor, ngClass, ngStyle and ngIf Recap
- › Creating a Basic Attribute Directive
- › Using the Renderer to build a Better Attribute Directive
- › Using HostListener to Listen to Host Events
- › Using HostBinding to Bind to Host Properties
- › Binding to Directive Properties
- › What Happens behind the Scenes on Structural Directives
- › Building a Structural Directive
- › Understanding ngSwitch

Module 6 - Using Services & Dependency Injection

- › Why would you Need Services?
- › Creating a Logging Service
- › Injecting the Logging Service into Components
- › Creating a Data Service
- › Understanding the Hierarchical Injector
- › How many Instances of Service Should It Be?
- › Injecting Services into Services
- › Using Services for Cross-Component Communication

Module 7 - Changing Pages with Routing

- › Why do we need a Router?
- › Setting up and Loading Routes
- › Navigating with Router Links
- › Understanding Navigation Paths
- › Styling Active Router Links
- › Navigating Programmatically
- › Using Relative Paths in Programmatic Navigation
- › Passing Parameters to Routes
- › Fetching Route Parameters
- › Fetching Route Parameters Reactively
- › Route Observables

- › Passing Query Parameters and Fragments
- › Retrieving Query Parameters and Fragments
- › Setting up Child (Nested) Routes
- › Using Query Parameters - Practice
- › Configuring the Handling of Query Parameters
- › Redirecting and Wildcard Routes
- › Outsourcing the Route Configuration
- › An Introduction to Guards
- › Protecting Routes with canActivate
- › Protecting Child (Nested) Routes with canActivateChild
- › Using a Fake Auth Service
- › Controlling Navigation with canDeactivate
- › Passing Static Data to a Route
- › Resolving Dynamic Data with the resolve Guard
- › Understanding Location Strategies

Module 8 - Understanding Observables

- › Analyzing Angular Observables
- › Getting Closer to the Core of Observables
- › Building a Custom Observable
- › Errors & Completion
- › Understanding Operators
- › Subjects

Module 9 - Handling Forms in Angular Apps

- › Template-Driven (TD) vs Reactive Approach
- › TD: Creating the Form and Registering the Controls
- › TD: Submitting and Using the Form
- › TD: Understanding Form State
- › TD: Accessing the Form with @ViewChild
- › TD: Adding Validation to check User Input
- › Built-in Validators & Using HTML5 Validation
- › TD: Using the Form State
- › TD: Outputting Validation Error Messages
- › TD: Set Default Values with ngModel Property Binding
- › TD: Using ngModel with Two-Way-Binding
- › TD: Grouping Form Controls
- › TD: Handling Radio Buttons
- › TD: Setting and Patching Form Values
- › TD: Using Form Data
- › TD: Resetting Forms
- › Introduction to the Reactive Approach
- › Reactive: Setup
- › Reactive: Creating a Form in Code
- › Reactive: Syncing HTML and Form
- › Reactive: Submitting the Form
- › Reactive: Adding Validation
- › Reactive: Getting Access to Controls
- › Reactive: Grouping Controls
- › Reactive: Arrays of Form Controls (FormArray)
- › Reactive: Creating Custom Validators
- › Reactive: Using Error Codes
- › Reactive: Creating a Custom Async Validator
- › Reactive: Reacting to Status or Value Changes
- › Reactive: Setting and Patching Values

Module 10 - Using Pipes to Transform Output

- Introduction & Why Pipes are Useful
- Using Pipes
- Parametrizing Pipes
- Where to learn more about Pipes
- Chaining Multiple Pipes
- Creating a Custom Pipe
- Parametrizing a Custom Pipe
- Example: Creating a Filter Pipe
- Pure and Impure Pipes (or: How to "fix" the Filter Pipe)
- Understanding the "async" Pipe

Module 11 - Making Http Requests

- How Does Angular Interact With Backends?
- The Anatomy of a Http Request
- Backend (Firebase) Setup
- Sending a POST Request
- GETting Data
- Using RxJS Operators to Transform Response Data
- Using Types with the HttpClient
- Outputting Posts
- Showing a Loading Indicator
- Using a Service for Http Requests
- Services & Components Working Together
- Sending a DELETE Request
- Handling Errors
- Using Subjects for Error Handling
- Using the catchError Operator
- Error Handling & UX
- Setting Headers
- Adding Query Params
- Observing Different Types of Responses
- Changing the Response Body Type
- Introducing Interceptors
- Manipulating Request Objects
- Response Interceptors
- Multiple Interceptors

Module 12 - Authentication & Route Protection in Angular

- How Authentication Works
- Adding the Auth Page
- Switching Between Auth Modes
- Handling Form Input
- Preparing the Backend
- Make sure you got Recipes in your backend!
- Preparing the Signup Request
- Sending the Signup Request
- Adding a Loading Spinner & Error Handling Logic
- Improving Error Handling
- Sending Login Requests
- Login Error Handling
- Creating & Storing the User Data
- Reflecting the Auth State in the UI
- Adding the Token to Outgoing Requests
- Attaching the Token with an Interceptor
- Adding Logout
- Adding Auto-Login
- Adding Auto-Logout
- Adding an Auth Guard

Module 13 - Dynamic Components

- Adding an Alert Modal Component
- Understanding the Different Approaches
- Using ngIf
- Preparing Programmatic Creation
- Creating a Component Programmatically
- Understanding entryComponents
- Data Binding & Event Binding

Module 14 - Angular Modules & Optimizing Angular Apps

- › What are Modules?
- › Analyzing the AppModuleGetting Started with Feature Modules
- › Splitting Modules Correctly
- › Adding Routes to Feature Modules
- › Component Declarations
- › Understanding Shared Modules
- › Understanding the Core Module
- › Adding an Auth Feature Module
- › Understanding Lazy Loading
- › Implementing Lazy Loading
- › Alternative Lazy Loading Syntax
- › More Lazy Loading
- › Preloading Lazy-Loaded Code
- › Modules & Services
- › Loading Services Differently
- › Ahead-of-Time Compilation

Module 15 - Deploying an Angular App

- › Deployment Preparation & Steps
- › Using Environment Variables
- › Deployment Example: Firebase Hosting
- › Server Routing vs Browser Routing

Module 16 - Angular Changes & New Features

- › What's New with Angular Current Version?
- › A First Look At Angular Elements
- › What's New & How to Update