Cloud Contraptions
TRAINING & CONSULTING

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Svelte for Angular Programmers

Class Duration

3 days

Target Audience

All students must have Angular, JavaScript, and HTML programming experience. Experience with CSS is helpful, but not required.

Description

The Svelte for Angular Programmers course offers a deep dive into Svelte, a modern JavaScript framework for building user interfaces. It covers everything from setting up a development environment, understanding the principles of template reactivity, creating static and dynamic pages, to handling forms and lifecycle events. The course also explores more advanced topics such as state management, routing, error handling, and asynchronous data. Throughout the course, comparisons are made to Angular to help students understand the differences and advantages of Svelte. This course is ideal for professionals with a background in Angular who are looking to expand their skill set and learn a new framework.

Objectives

- Understand the fundamentals of Svelte and how it compares to Angular.
- Set up a development environment for Svelte and understand how it differs from Angular's development environment.
- Learn about SvelteKit and its features, including routing, server-side rendering, and unit testing.
- Create static and dynamic pages using Svelte, including understanding their structure and how to handle images, CSS, and JavaScript content.
- Understand the principles of template reactivity in Svelte and how it compares to Angular's Zone.js and Signals.
- Learn about component basics and composition in Svelte, including how to handle events and data.
- Understand how to handle forms and lifecycle events in Svelte.

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Learn about state management, routing, error handling, and asynchronous data in Svelte.

Training Materials

All students receive comprehensive courseware covering all topics in the course. Courseware is distributed via GitHub in the form of documentation and extensive code samples. Students practice the topics covered through challenging hands-on lab exercises.

Software Requirements

Students will need a free, personal GitHub account to access the courseware. Student will need permission to install Node.js and Visual Studio Code on their computers. Also, students will need permission to install NPM Packages and Visual Studio Extensions. If students are unable to configure a local environment, a cloud-based environment can be provided.

Training Topics

Introduction

- What is Svelte?
- What problem does Svelte solve?
- Svelte vs. Angular
- Svelte vs. Angular Compiler
- Svelte vs. Angular Directives
- Moving from Angular to Svelte

Development Environment

- Requirements
- SvelteKit vs. Angular CLI
- Svelte Files vs JSX Files
- Svelte Extension for Visual Studio Code
- Run/Debug Svelte App in Visual Studio Code
- Svelte Extension for WebStorm
- Run/Debug Svelte App in WebStorm

SvelteKit 2 Overview

- Vite Tooling
- Development Server

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Routing
- Deployment
- Server-side rendering
- Unit Testing

Getting Started

- Exploring the REPL
- Svelte Layout
- Svelte Page
- Svelte Component
- Svelte Architecture
- Svelte Element Directives
- Compiling Svelte Files

Static Pages

- What is a Static Page?
- What problem do Static Pages solve?
- Static Page File Structure
- Setting Head Content
- HTML Content
- CSS Content
- Comments
- Scoped CSS
- Handling Images
- Hot Module Reloading
- Server Pre-rendering
- Page Routing

Dynamic Pages

- What is a Dynamic Page?
- What problem do Dynamic Pages solve?
- Client-Side Rendering
- Dynamic Page File Structure
- JavaScript Content
- Using Variables
- Using Expressions
- Data Binding

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Class and Style Directive
- Event Binding
- Logic Blocks
- Debug Tag

Reactivity with Runes (Svelte 5)

- The four runes: `$state`, `$derived`, `$effect`, `$props`
- `$state` for mutable reactive state
- `$derived` for computed values (replacing `$:` reactive statements)
- `$effect` for side effects and DOM interaction
- Updating arrays and objects with deep reactivity
- Reading legacy `$:` and migrating
- Compared to Angular's Zone.js and Signals

Component Basics

- What is a Component?
- What problem do Components solve?
- Svelte Components vs Angular Components
- Calling Components vs HTML Elements
- Compared to Template Rules for Component Selectors
- Component File Structure
- Component Props
- Component Events
- Compared to Angular Inputs and Outputs

Component Composition

- Nested components
- Snippets (`{#snippet}` / `{@render}`) — the modern slot replacement
- Passing data and callbacks to children
- Component tree best practices

Event Handling (Svelte 5)

- Standard DOM event attributes (`onclick`, `oninput`)
- Event modifiers via wrapper functions (legacy `on:click|preventDefault` deprecated)
- Callback props (replacing `createEventDispatcher`)
- Forwarding behavior with spread props

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Data binding

- Top-down data binding by default
- Communication with props and events
- Using two-way data binding

Forms

- HTML Form Element
- Named Form Actions
- Form Validation
- Form Submission
- Progressive Enhancement

Lifecycle and Effects

- `$effect` and `$effect.pre` (the modern approach)
- `onMount` and `onDestroy` (still available)
- `tick()` for awaiting DOM updates
- Effect cleanup functions
- Compared to Angular useEffect Hook

State Management (Svelte 5)

- Reactive primitives with `$state` (the modern default)
- Sharing reactive state via context
- Stores in Svelte 5 (`writable`, `readable`, `derived`) — when they still make sense
- Migrating store-heavy code to runes
- Page, navigation, and updated stores in SvelteKit 2

Routing

- What is Routing?
- What problem does Routing solve?
- Compared to Angular's Router
- Pages
- Layout
- Route Parameters
- API Routes

Errors and Redirects

- Handling Errors and Redirects

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Error Pages
- Fallback Errors
- Redirects
- Compared to Angular Router’s Redirects

Asynchronous Data

- Promises compared to RxJS
- Promises & async/await
- Fetching data from a REST API
- Subscriptions
- Stores

Conclusion