

**Advanced Svelte**

**Course Number:** SVLT-102
**Duration:** 2 days

**Overview**

This Advanced Svelte course teaches attendees how to work with environment variables, advanced bindings, component composition, routing, actions, UI special effects, special elements, hooks, and unit testing. Participants learn how to create complex UI effects, understand the use of special elements, and master advanced routing techniques. This course also covers hooks, server-side and client-side rendering, and unit testing. By the end of this Svelte training, participants will be able to immediately create complex, high-performance web applications with Svelte.

**Prerequisites**

All Svelte training students must have Svelte, JavaScript, and HTML programming experience, either on their own or from another class they attended. Experience with CSS is helpful but not required.

**Materials**

All students receive comprehensive courseware covering all topics in the course. The instructor distributes courseware via GitHub. The courseware includes documentation and extensive code samples.

**Software Needed on Each Student PC**

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

**Objectives**

* Understand the use of environment variables in Svelte
* Work with advanced bindings, including block bindings, media elements, and more.
* Master advanced component composition techniques, including slots, named slots, slot fallbacks, and the context API.
* Use advanced routing techniques including optional parameters, REST parameters, param matchers, and route groups
* Understand actions and element-level lifecycle functions
* Interface with 3rd party libraries
* Create UI special effects using motion, transitions, key blocks, and animations directives
* Understand special elements in Svelte such as Svelte Self, Svelte Component, Svelte Element, and more
* Use hooks in Svelte and handle hook requests and errors
* Master server-side rendering, client-side rendering, pre-rendering, links, and unit testing in Svelte

**Outline**

* Introduction
* Environment Variables
* Advanced Bindings
	+ Content-Editable
	+ Each Block Bindings
	+ Media Elements
	+ Readonly Element Dimensions
	+ This Bind
	+ Bind to Components Props
	+ Bind to Component Instances
* Advanced Component Composition
	+ Slots
	+ Named Slots
	+ Slot Fallbacks
	+ Slot Props
	+ Slot Content
	+ Module Context
	+ Context API
* Advanced Routing
	+ Optional Parameters
	+ REST Parameters
	+ Param Matchers
	+ Route Groups
	+ Organize Layouts and Routes
* Actions
	+ What are Actions?
	+ How are Actions used?
	+ Element-Level Lifecycle Functions
	+ Customizing Reusable Events
	+ Interface with 3rd Party Libraries
* UI Special Effects
	+ Motion: Tweens & Spring
	+ Transitions Directive
	+ Parameters
	+ In and Out
	+ Custom CSS/JS Transitions
	+ Transition Events
	+ Key Blocks
	+ Global & Deferred Transitions
	+ Animations Directive
* Special Elements
	+ Svelte Self
	+ Svelte Component
	+ Svelte Element
	+ Svelte Window & Bindings
	+ Svelte Body
	+ Svelte Document
	+ Svelte Head
	+ Svelte Options
	+ Svelte Fragment
* Hooks
	+ What is Hook?
	+ How are Hooks used?
	+ Handle Hook
	+ Request Event
	+ Handle Fetch & Error
* Pages and Layout
	+ Server-Side Rendering
	+ Client-Side Rendering
	+ Pre-Rendering
	+ Trailing Slash
	+ Universal Loading
	+ Invalidation
* Links
	+ Preloading
	+ Reloading
* Unit Testing
	+ What is Unit Testing?
	+ Why Unit Test Svelte?
	+ Unit Testing Framework
	+ Testing Svelte components
	+ Testing Svelte stores
	+ Testing SvelteKit
* Conclusion