

**Comprehensive Vue**

**Course Number:** SCRPT-164  
**Duration:** 5 days

**Overview**

This Comprehensive Vue training course teaches attendees the basics of Vue as well as advanced skills needed to build robust, modern applications.

**Prerequisites**

All attendees must be experienced JavaScript developers with a fairly advanced understanding of JavaScript, including prototypes and functions as first-class citizens.  If your group doesn’t yet have this experience, we could readily add one or two days to the beginning of your course to appropriate prepare them.

**Materials**

All attendees receive comprehensive courseware.

**Software Needed on Each Student PC**

* Google Chrome
* Other modern browsers as desired
* IDE/development environment of your choice
* Node
* Other free software and lab files that Accelebrate would specify

**Objectives**

* Understand the difference between single-page web applications and other JavaScript uses
* Articulate what Vue.js is and why it is useful in both scenarios
* Learn Vue.js coding and architecture best practices, including project layout and component management
* Set up and manage projects with the Vue.js CLI
* Add, remove, and manage features with the Vue.js CLI
* Use Vue’s reactivity features to build fresh, snappy user interfaces
* Wield the power of dynamic CSS thanks to Vue’s CSS manipulation capabilities
* Comprehend the differences among Vue’s data, computed, and method properties
* Leverage Vue’s template language and directives in the HTML portion of components
* Manage forms and respond to user input
* Communicate with a back-end server to retrieve data for Vue components
* Write effective and efficient unit and integration tests for Vue.js applications
* Build fun, modern user experiences thanks to Vue’s transitions and animations
* Configure the Vue router
* Manage dynamic routes
* Extract data from route information
* Configure Vue.js to use Vuex, the state management library
* Test Vuex to ensure the integrity of state manipulation
* Plug Vuex into a back-end REST server

**Outline**

* Introduction and Setup
  + Introductions
  + Verify setup
    - Node.js and npm
    - IDE
    - Git
    - Class repo
  + What is Vue.js?
  + What does Vue do for me?
* Core concepts in Vue.js
  + Including Vue into a page
  + Reactive rendering
  + The Vue.js instance
  + Instances vs components
  + The Vue CLI
  + The Vue devtools
* Components and Instances
  + The Vue instance
    - Instance data with the data property
    - Actions with the methods property
    - Attaching methods to events
    - Lifecycle methods
  + Vue components
    - Differences with a Vue instance
    - Changes to the data property
    - Changes to methods
    - When should I use components, when should I use the instance?
    - The single file component
  + Testing in Vue.js
    - Testing instances
    - Testing against components
* Templates
  + Rendering a variable in a template
  + Binding properties
  + Binding events
  + Testing templates
  + Optimizing with computed properties
  + When to use a computed property vs data
  + Interactions between computed, data, and methods
  + Watching for changes
  + Testing watches
* Working with Components
  + Registering components
    - Global vs local registration
    - Module systems
    - Registering common components
  + Properties
    - Passing properties to a component
    - Property types
    - Validating properties
    - Testing properties
    - Attribute inheritance
* Event Handling
  + In-line vs method handlers
  + Testing event handling
  + Event modifiers
  + Picking up keyboard events
* Custom Events
  + Custom event concepts
  + Naming custom events
  + Binding to native events over custom events
  + Using .sync
* Control Structures
  + Conditionals with v-if
    - v-else
    - v-else-if
  + Hiding and showing with v-show
    - v-show vs v-if
  + Testing conditionals
  + Loops with v-for
    - Basic v-for
    - State management and watching for changes
    - Sorting and filtering
    - Testing loops
* CSS Bindings
  + Binding CSS classes
    - Object syntax
    - Array syntax
    - Checking CSS with tests
  + Binding inline CSS styles
    - Object syntax
    - Array syntax
* Working with Forms
  + Using v-model
  + Specific form controls
    - Text fields
    - Checkboxes and radio buttons
    - Selects / Drop-downs
    - Multi-line text fields
  + Testing form controls
  + Modifiers for v-model
  + Forms and custom events
* Slots
  + What is a slot and what does it do?
  + The v-slot attribute
  + Sharing data between slots
    - Scoped slots
  + Managing content with slots
    - Named slots
    - Dynamic names
  + Testing slot behavior
* Transitions
  + Animating changes in Vue.js
  + List transitions
    - Entering and leaving
    - When list items move (e.g. in sorting)
    - Reusing transitions
    - Making transitions dynamic
  + State transitions
    - What to listen for
    - Watching the watchers
    - Dynamic state transitions
    - Turning transitions into components
* Introduction to Routing
  + What is routing and what does it do for me?
  + Adding routing to a project
    - Via the CLI
    - Other methods
    - HTML updates
    - JS updates
  + Matching routes
  + Programmatic navigation
  + Route transitions
* Route matching
  + Matching dynamic routes
    - Updating on param changes
    - Catch-alls and 404s
  + Nested routes
  + Named routes
  + Redirects and aliases
  + Passing properties
* Advanced routing
  + Navigation guards
  + Fetching data and routing
  + Lazy loading routes
  + Scroll behavior
* State management with Vuex
  + What is Vuex and what does it do for me?
  + Installing Vuex
    - Via the CLI
    - Other methods
    - Vuex and dev tools
  + Core concepts
    - Flux-style state management
    - State
    - Getters
    - Mutations
    - Actions
    - Modules
* Working with Vuex
  + Structuring your application
  + Handing form data and interactions
  + Vuex transitions
  + Testing Vuex
* Conclusion