



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

Vue Essentials

Class Duration

3 days

Target Audience

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

Description

This Vue Essentials course teaches modern Vue 3 from the ground up. The course centers on the **Composition API** with `<script setup>` — the current standard idiom for Vue development — covering Vue’s reactivity system (`ref`, `computed`, `watch`), single-file components, props, emits, and slots, all with TypeScript end-to-end. Participants build real applications using Pinia for state management and Vue Router 5, which now ships optional file-based routing in the core package. Coverage includes forms with `v-model` and `defineModel`, asynchronous data handling, Vite tooling, and a thorough testing module using Vitest 4 with Vue Testing Library plus Playwright for end-to-end tests. The course also covers the Options API just enough that participants can read older Vue 2 and Vue 3 codebases — but every new pattern taught is the current Composition API one. Participants also get an instructor-led preview of Vapor Mode, the opt-in virtual-DOM-free compilation strategy currently in beta for Vue 3.6. By the end, participants build, test, and run a complete Vue 3 application.

Objectives

- Understand Vue 3’s reactivity model and how it compares to React, Svelte, and Angular.
- Set up a development environment with `create-vue`, Vite, TypeScript, and the official Vue extension for Visual Studio Code.
- Master the Composition API with `<script setup>` — `ref`, `reactive`, `computed`, `watch`, and `watchEffect` — and know when each is the right tool.
- Build components with typed props (`defineProps`), events (`defineEmits`), two-way binding (`defineModel`), and slots.



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

- Share state across components with provide/inject and manage application state with Pinia setup stores.
- Build single-page applications with Vue Router 5 — routes, params, nested routes, navigation guards, and the new built-in file-based routing.
- Build and validate forms using v-model and modifiers.
- Fetch and render asynchronous data with loading and error states.
- Test components with Vitest 4, Vue Testing Library, and Playwright.
- Read legacy Options API code (data, methods, computed options) and migrate selectively to the Composition API.

Training 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. 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. Students 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. We will provide a cloud-based environment if students cannot configure a local environment.

Training Topics

Introduction to Vue 3

- What is Vue? What problem does it solve?
- Vue as a progressive framework
- Composition API vs. Options API: what changed and why
- Vue vs. React, Svelte, and Angular reactivity models
- Vapor Mode (Vue 3.6 beta): virtual-DOM-free compilation preview

Development Environment

- Requirements
- Project scaffolding with create-vue (`npm create vue@latest`)
- Vite dev server and HMR
- Official Vue extension for Visual Studio Code



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

- TypeScript out of the box
- Vue DevTools

Single-File Components

- Anatomy of a .vue file: `<template>`, `<script setup>`, `<style>`
- Scoped CSS and CSS modules
- How SFCs are compiled
- `<script setup>` vs. the `setup()` function

Template Syntax

- Text interpolation and expressions
- Attribute binding with `v-bind (:)`
- Conditional rendering (`v-if`, `v-else-if`, `v-else`, `v-show`)
- List rendering with `v-for` and `key`
- Dynamic components with `<component :is>`

Reactivity Fundamentals

- `ref()` for reactive values
- `reactive()` for objects — and when to prefer `ref`
- `computed()` for derived values
- `watch()` and `watchEffect()` for side effects
- Deep vs. shallow reactivity (`shallowRef`, `shallowReactive`)
- Reactivity gotchas: destructuring, `toRefs`, losing reactivity

Component Basics

- Component file structure and registration
- Props with `defineProps()` and TypeScript types
- Prop defaults and validation
- Events with `defineEmits()`
- Two-way component binding with `defineModel()`
- Reading legacy Options API components

Component Composition

- Nested components
- Slots: default, named, and scoped
- Fallthrough attributes
- Provide/inject for cross-tree data
- Composables: extracting reusable logic



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

- Component tree best practices

Event Handling

- Listening with `v-on` (`@`)
- Event modifiers (`.prevent`, `.stop`, `.once`)
- Key and mouse modifiers
- Emitting and handling component events

Data Binding and Forms

- Two-way binding with `v-model`
- `v-model` on inputs, selects, checkboxes, and radios
- `v-model` modifiers (`.lazy`, `.number`, `.trim`)
- `v-model` on components with `defineModel()`
- Form validation and submission patterns

Lifecycle and Effects

- `onMounted`, `onUpdated`, `onUnmounted`
- Watcher cleanup functions
- `nextTick()` for awaiting DOM updates
- Template refs for DOM access

State Management with Pinia

- Why Pinia? (the official store, replacing Vuex)
- Defining stores: setup stores vs. option stores
- State, getters, and actions
- `storeToRefs()` and preserving reactivity
- Sharing stores across components
- Pinia and Vue DevTools

Routing with Vue Router 5

- Installing and configuring the router
- Routes, dynamic params, and nested routes
- `<RouterLink>` and `<RouterView>`
- Programmatic navigation with `useRouter` and `useRoute`
- Navigation guards
- Lazy-loading route components
- File-based routing — now built into Vue Router 5



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

Asynchronous Data

- Fetching data in setup with composables
- Loading and error states
- Async components and `defineAsyncComponent`
- `<Suspense>` (experimental) overview

TypeScript with Vue

- Typing props, emits, refs, and computed values
- Generic components
- Typed Pinia stores and typed routes

Testing

- Unit testing with Vitest 4
- Component testing with Vue Testing Library
- `@vue/test-utils` and when to drop down to it
- Mocking Pinia stores and the router
- End-to-end testing with Playwright

Conclusion