



# Vue.js Fundamentals

**Course #:** VU-100

**Duration:** 4 days

## Prerequisites

Students should have a good understanding of HTML and CSS and be experienced JavaScript developers, with an advanced understanding of JavaScript objects and functions as first class citizens.

## Details

Vue.js is the progressive JavaScript framework that is rapidly gaining in popularity in the web community. It can be incrementally adopted, scaling easily from being used as a library to add small functionality to web pages to being used as a framework for a complete single-page-application. Vue.js is also easily integrated with other JavaScript libraries and frameworks.

In this course you will start out working with Vue.js to add small bits of functionality to a page. You will then build a complete front-end web application using Vue.js, with routing and animation, interacting with a web API and implementing the Redux state management pattern using Vuex.

## Software Needed

Text Editor, Node.js and a current web browser

## Outline

### Vue.js Fundamentals

- **Introduction**
  - What is Vue.js?
  - How to get Vue.js
  - Hello Vue.js World
  - Data and Methods
- **Interacting with the DOM**
  - Vue.js Virtual DOM
  - Interpolation
  - Methods
  - Interpolation and HTML Encoding
  - Event Binding, Arguments, and Modifiers
- **Data and Event Binding**
  - JavaScript Code in the Template

- 2-way Data Binding
  - Computed Properties
  - Watchers
  - Binding HTML Classes
  - Binding Style Properties
- **Conditionals and Iterations**
  - Built-in Conditional Directives
  - Built-in Looping Directives
- **The Vue.js Instance**
  - Configuration Options
  - Selector Limitations
  - Using Components
  - Template Limitations
  - Lifecycle Hooks
- **Using the Vue CLI**
  - The Development Environment
  - The Development Workflow
  - Creating Projects Using Vue CLI
  - Webpack Template Overview
  - Vue Files and render
  - Using Vue CLI to Build for Production
- **Creating Components**
  - Component Syntax
  - Data and Template Syntax
  - Registering a Component Locally
  - Root Component
  - File Naming Conventions (and Folder Conventions)
  - Selector Naming (Hyphenated vs. Camel-Case)
  - Component Styles and Scoping
- **Communicating Between Components**
  - props
  - Emitting Events
  - Unidirectional Data Flow
  - Callback Functions
  - Vue.js Instance as Event Bus
- **Advanced Component Usage**
  - Data Sharing Review
  - Content Sharing with Slots
  - Multiple, Named Slots
  - Default Slots and Default Content
  - Dynamic Components
- **Working with Forms**
  - Input Form Binding
  - Input Modifiers
  - Checkboxes and Arrays
  - Radio Buttons
  - Dropdown Lists
  - Custom Controls
  - Submitting a Form
- **Directives**
  - Built-in Directives
  - Creating Custom Directives
  - Directive Hook Functions
  - Passing Values to Custom Directives
  - Passing Arguments to Custom Directives
  - Creating Modifiers for Custom Directives
- **Filters and Mixins**
  - Custom Filter
  - Computed Property as a Filter Alternative
  - Mixins

- How Mixins Get Merged
- The Global Mixin Antipattern
- Mixins and Scope
- **Transitions and Animations**
  - Transition Element
  - CSS CLasses Applied
  - Animating on Load
  - Using Animation Libraries
  - TRansitioning Between Elements
  - Transition JavaScript Hooks
  - Animating Dynamic Components
  - Animating Multiple Elements
- **Working with HTTP APIs**
  - Installing Axios
  - HTTP Verb Methods
  - GETting Data
  - Configuring Axios Globally
  - Using Interceptors
  - Custom Axios Instances
- **Vue Routing**
  - Setting up routes
  - Displaying Components with router-view
  - Router Modes
  - Creating Router Links
  - Navigating from Code
  - Using Route Parameters
  - Child Routes
  - Using Named Routes
  - Using Multiple RouterViews
  - Redirecting and Catch-all Routes
  - Animating Router Transitions
  - Route Guards
  - Lazy-Loading Route Components
- **Managing State with Vuex**
  - Installing vuex
  - Central Store for State
  - Using Getters for State
  - Committing Mutations
  - Creating Actions
  - Passing Payloads
  - Vuex and 2-way Data Binding
  - Modularize State
  - Best Practices
- **Deployment**
  - Preparing for Deployment
  - Creating Deployment Build
  - Deploying an App