

# Vue.js – Syllabus

**Vue.js** is among the world's most popular language **frameworks** for **applications** and **interface development**. Its relative **lightweight platform**, ease of understanding, **simple integration**, and flexibility have given it huge support from developers and organizations such as Alibaba, Xiaomi and others.

Vue.js can be used to **create single page apps** or **server side rendered apps**. This course will help you learn about **instances**, **single page apps** and **mixins**, **routing** and other concepts needed to build state of the art applications. As front end applications get more and more sophisticated the need for Vue.js experts is on the rise. This is your chance to get started on a great career.

## **What you will learn:**

- Learn the basics of Vue.js, including its use of templates to display data on a page
- Learn to set up Vue projects from scratch, or use vue-cli to set up from a template
- Understand the fundamental concepts of Vue.js and how to use it in applications
- Learn to create instances, learn about single page apps, Custom directives, Mixins, Routing and more
- Discover how Vue.js works with CSS to style your websites and applications
- Use render functions and JSX, rather than templates, to determine what Vue displays
- Be able to control how code is executed and displayed with vue-router
- Learn to effectively use and navigate the Vuex library
- Build a fully functional app from scratch, and unit test your applications.

Sl. no	Topic	Hrs
1	Introduction to Vue.js	

	What is Vue.js?	
	Why Vue.js?	
	Quick tour of Vue-cli	
	Hello World with Vue.js	
	Anatomy of Hello World Vue.js App	
	Create a new Vue App using vue cli, execute and debug	
<b>2</b>	<b>Working with Template and ViewModel</b>	
	What is Vue.js Template	
	What is Vue Model / Vue Instance	
	Template Syntax	
	Interpolation	
	Directive	
	Filters	
	Shorthands	
	Javascript Expressions	
	Computed Properties	
	Watchers	
	Class and Style Bindings	
	HTML Classes	
	Inline Styles	
	Conditional Rendering	
	Interactive Rendering	
	Create a Vue App which explores the Template syntax features like Interpolation, Directives, Expressions, Watchers, Filters,	

	Computed Properties, Iterative, and Conditional Rendering	
<b>3</b>	<b>Vue Instance Revisited</b>	
	Basic Vue Instance Usage	
	Multiple Vue Instances	
	Vue Instance Lifecycle	
	Accessing Data from Outside Vue Instance	
	Data and Methods	
	More about \$el, \$data and \$refs	
	Create a Vue App that creates Multiple Vue Instances, explore through the LifeCycle methods and Code Data Access from Outside Vue Instance	
<b>4</b>	<b>Event Handling</b>	
	Introduction about Event Handling	
	Listening to Events	
	Method Event Handlers	
	Event Modifiers	
	Key Modifiers	
	Create a Vue App that Responds to Events	
<b>5</b>	<b>Working with Forms &amp; Two-Way Binding</b>	
	Introduction to Two-Way Binding via V-Model	
	Handling User Input with HTML Form Controls	
	Data Binding	

	Jazy, .number, .trim Modifiers	
	Submitting a Form	
	Create Forms and apply Two Way Binding to Read and Write Data to Forms	
<b>6</b>	<b>Introduction to Components</b>	
	What are Components?	
	Why Components?	
	Hello World Components?	
	Anatomy of a Component	
	Registering Components Locally and Globally	
	Root Components	
	Thinking in Components	
	Composing Application as Components	
	Convert a Single HTML Web Page into a Series of Components, Register Components and Compose Components into Application	
<b>7</b>	<b>Passing Data to Components</b>	
	Props, Parent to Child Communication	
	Custom Events, Child to Parent	
	Non Parent Child Communication Event Emitter Style using .emit, .on	
	Create Multiple Components and Enable Communication	

	between Independent Component Props for Parent to Child Communication, Events for Child to Parent and Non Parent Child Components using Event Emitter	
<b>8</b>	<b>Content Distribution with Slots</b>	
	Compilation Scope	
	Single Slot	
	Named Slot	
	Scoped Slot	
	Create a Re-usable Component which accepts arbitrary Content using Slots	
<b>9</b>	<b>Dynamic Components</b>	
	Switching Multiple Components with Dynamic Components	
	Dynamic Component Behavior	
	Keeping Dynamic Components Alive	
	Lifecycle Hooks	
	Create a Component with Tabbed Layout that Switches Components between Tabs Dynamically	
<b>10</b>	<b>Custom Directives</b>	
	Introduction about Directives, Custom Directives	
	Directive Hook Functions	
	Developing Custom Directive	
	Passing Values and Arguments to Custom Directives	

	Directive Hook Arguments	
	Create a Custom Directives that Modifies the Background Color of the Element when the Directive is Applied	
<b>11</b>	<b>Mixins</b>	
	Introduction to Mixins	
	Creating and using Mixins	
	Option Merging	
	Global Mixin	
	Create a Mixin and Apply to Components	
<b>12</b>	<b>Filters</b>	
	Introduction about Filters	
	Local Filters	
	Global Filters	
	Chaining Filters	
	Create a Component that uses Default Filters to Format Data in Template, also Chain Multiple Filters	
<b>13</b>	<b>Remote Communication with VueJs</b>	
	Using vue-resource to working with HTTP	
	Fetch Data	
	Post Data to Server	
	Request Response Interception	

	Template URL's	
	Create Components that Fetch Data from Remote API and Post Data to Remote API	
<b>14</b>	<b>Routing</b>	
	Introduction of Routing	
	SettingUp of Routing via vue-router	
	Loading Routes	
	Routing Modes	
	Navigation with Router Links	
	Styling Active Links	
	Imperative Navigation	
	Route Parameters	
	Setting Up Child or Nested Routes	
	Names Routes	
	Query Parameters	
	Named Router Views	
	Wildcard Routes	
	Router Guards	
	Setup and Configure vue-router, Create Navigation Component and Code Declarative Routing using Router Links, Code Imperative Routing from Component Methods, Configure Route Parameters and Read them from Inside Component. Protect Routes using Guards	

<b>15</b>	<b>State Management</b>	
	Introduction to State Management	
	What is need for State Management	
	What is vueX	
	Centralized Store	
	Understanding Getters	
	Using Getters	
	Mapping Getters to Properties	
	Create a Centralized Store, Getter and Setters, Actions	
<b>16</b>	<b>Understanding Mutations</b>	
	Using Mutations	
	Using Actions	
	Mapping Actions to Methods	
	Create Mutations to Change the Data in the Store	
<b>17</b>	<b>Unit Testing</b>	
	What is Unit Testing?	
	Setup and Tooling	
	Testing Components	
	Creating the Unit Test to Test Components	