

# Vue.js Basics Notes for Beginners

## 1. Introduction to Vue.js

- Vue.js is a progressive JavaScript framework for building user interfaces.
- It is reactive and component-based.
- Can be integrated into existing projects or used for full-fledged SPAs (Single Page Applications).

## 2. Setting Up Vue.js

- Include Vue via CDN:

```
<script src="https://cdn.jsdelivr.net/npm/vue@3/dist/vue.global.js"></script>
```

- Install via npm:

```
npm install vue@next
```

## 3. Vue.js Template Syntax

- Uses double curly braces `{{ }}` for data binding.
- Example:

```
<div id="app">
  <p>{{ message }}</p>
</div>
<script>
  const app = Vue.createApp({
    data() {
      return { message: "Hello Vue!" };
    }
  });
  app.mount('#app');
</script>
```

## 4. Vue Directives

- v-bind: Binds attributes dynamically.
- v-model: Two-way data binding for forms.
- v-if, v-else, v-show: Conditional rendering.
- v-for: Looping through lists.

## 5. Vue Methods

- Methods are functions inside the Vue instance.

- Example:

```
<button @click="sayHello">Click me</button>
```

```
methods: {
```

```
  sayHello() {
```

```
    alert("Hello from Vue!");
```

```
  }
```

```
}
```

## 6. Vue Components

- Reusable and modular parts of the UI.

- Example:

```
app.component('my-component', {
```

```
  template: '<p>This is a component!</p>'
```

```
});
```

## 7. Vue Lifecycle Hooks

- created(), mounted(), updated(), destroyed().
- Used to run code at different stages of a component's life.

## 8. Vue Router (For Navigation)

- Allows single-page applications (SPA) with multiple views.

- Install using:

```
npm install vue-router@next
```

## 9. Vue State Management

- Vuex or Pinia can be used for managing application-wide state.

## 10. Deploying Vue Apps

- Can be deployed on Netlify, Vercel, or Firebase.
- Use npm run build for production builds.

## Next Steps:

- Explore Vue CLI and Vue 3 Composition API.
- Build a small project to practice.

Happy Coding with Vue.js!