

# Vue.js



# what is vue?



- Front-end, Javascript / Typescript framework
- Used to create dynamic & data-driven single page applications (SPA)
- Create stand-alone widgets
- Create a whole website with multiple pages & components
- All routing is done in the browser & not on the server

# Single Page Application (SPA)



---

- Only a single HTML page sent (initially) to the browser
- Vue intercepts subsequent requests and handles “page” changes in the browser by swapping what components are shown on the page
- Results in a much faster and smoother website experience

# Software Development Environment

---

Activity 1

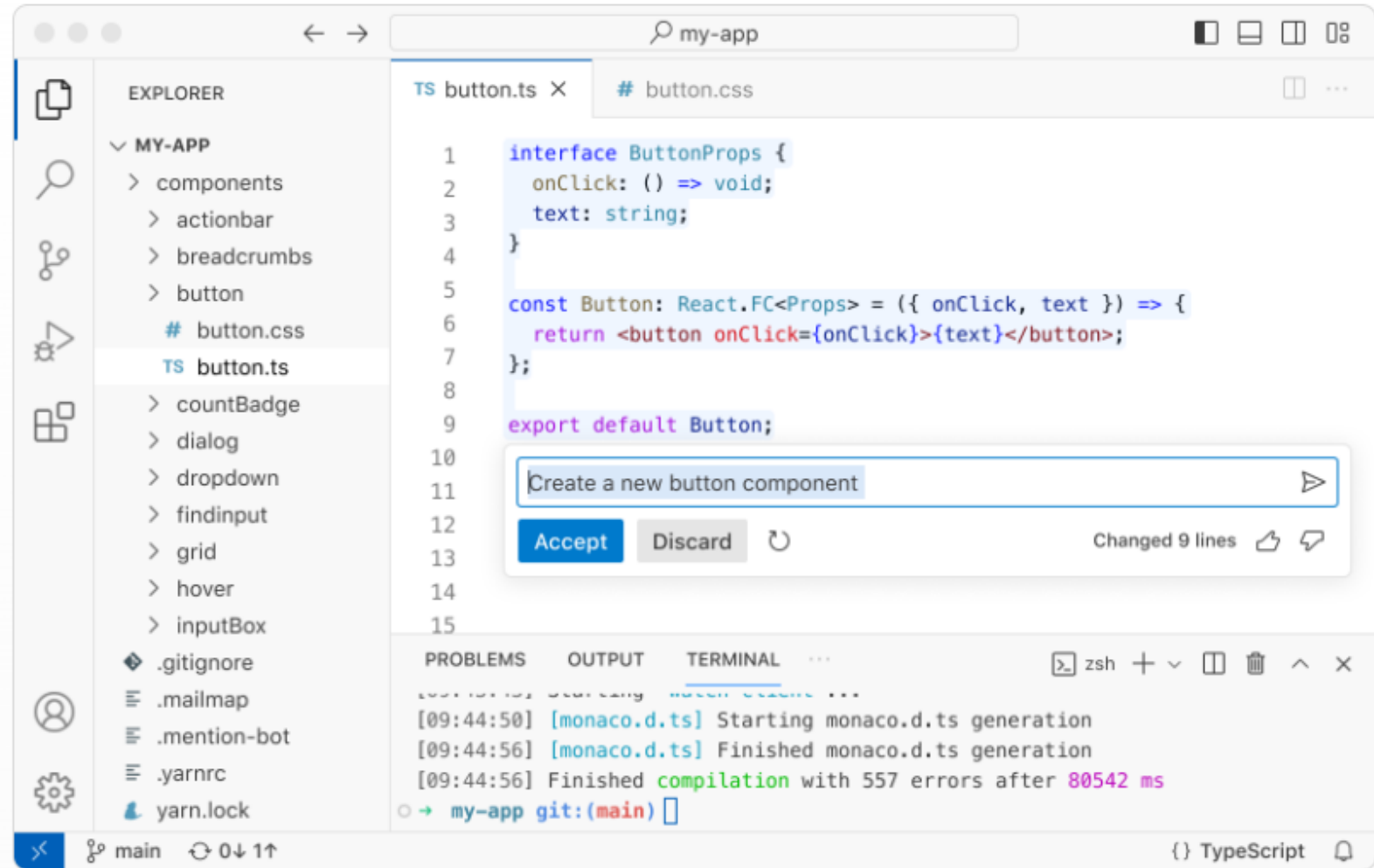
Free. Built on open source. Runs everywhere.

# Code Editing. Redefined.

Download for Windows

[Web](#), [Insiders edition](#), or [other platforms](#)


By using VS Code, you agree to its [license](#) and [privacy statement](#).





<https://code.visualstudio.com>


EXTENSIONS: MARKETPLACE


live server


 **Live Server** 53.2M ★ 4.5  
Launch a development local S...  
Ritwick Dey [Install](#)


 **Live Server (Fiv...** 1.2M ★ 4.5  
A better Live Server with insta...  
Yannick [Install](#)

 **Live Server Pre...** 1.1M ★ 3.5  
Preview your HTML file with lo...  
negokaz [Install](#)


 **Live Preview** 6.1M ★ 4.5  
Hosts a local server in your wo...  
Microsoft [Install](#)

 **XVBA - Live Ser...** 156K ★ 4  
Edit VBA files from Excel/Acce...  
Local Smart [Install](#)

 **html-live-server** 78K ★ 4  
Live server to render html web...  
Naveen Vignesh [Install](#)

 **php-live-server** 68K ★ 1  
Teodoro Villanueva [Install](#)

Extension: Live Server

 **Live Server** v5.7.9  
Ritwick Dey | 53,234,087 | ★★★★★ (491)  
Launch a development local Server with live reload feature for static & dynamic pages  
[Install](#) ⚙️

DETAILS

FEATURES

CHANGELOG

[Wanna try **LIVE SERVER++ (BETA)** ? It'll enable live changes without saving file. <https://github.com/ritwickdey/vscode-live-server-plus-plus> ]

## Live Server

Live Server loves ❤️ your multi-root workspace

Live Server for server side pages like PHP. [Check Here](#)

[For 'command not found error' #78]

vscode marketplace

v5.7.9

downloads

81M

rating

4.4/5 (491)

404

badge not found

appveyor branch

passing

license


MIT

Launch a local development server with live reload feature for static & dynamic pages.

Install Live Server >> Preview Code in Browser

EXTENSIONS: MARKETPLACE

vetur


**Vetur**

14.2M ★ 4.5

Vue tooling for VS Code

Pine Wu

Install


**Vetur-wepy**

68K ★ 3

Vue tooling for VS Code 简单...

hu2ren

Install


**Vetur E**

60K ★ 1

Vue tooling for VS Code

qiang014


Install

**Vetur Extended**

9K ★ 5

zardoy

Install


**Vetur TypeScript ...**

17K ★ 5

Workaround for the performa...


Stefan Novakovic

Install


**Vetur Plus**

7K ★ 5

Vue tooling for VS Code

 Weijian Chen

Install


**Vetur-SFC**

6K ★ 5

Custom vetur build with some...

mwilski

Install

**hippy-vetur**


161

hippy-vue 插件

mokywu

Install


Extension: Vetur

**Vetur**

v0.37.3

Pine Wu | 14,209,419 | ★★★★★(153)

Vue tooling for VS Code

Install 

DETAILS

FEATURES

CHANGELOG

## Vetur

### Project status


issue: <https://github.com/vuejs/vetur/issues/3476>

New official vue editor support: Volar

VSCode extension: <https://marketplace.visualstudio.com/items?itemName=Vue.volar>

Project: <https://github.com/johnsoncodehk/volar>

LSP: <https://github.com/johnsoncodehk/volar>

 <https://github.com/badges/shields/issues/8671> **v0.37.3** **14.21M** **4.32/5 (153)**

Vue tooling for VS Code.

- [vls](#): Vue Language Server
- [vti](#): Vetur Terminal Interface
- [Docs](#)

Install Vetur >> Syntax highlighting & Formatting

# Create Standalone Widgets

---

Activity 2



# vue.js documentation

https://vuejs.org/guide/introduction.html



Search

Docs ▾

API

Playground

Ecosystem ▾

About ▾

Sponsor

Partners

## API Preference

Options



Composition ?

## Getting Started

Introduction

Quick Start

## Essentials

Creating an Application

Template Syntax

Reactivity Fundamentals

Computed Properties

Class and Style Bindings

Conditional Rendering

List Rendering

Event Handling

Form Input Bindings

Lifecycle Hooks

## What is Vue?

Vue (pronounced /vju:/, like **view**) is a JavaScript framework for building user interfaces. It builds on top of standard HTML, CSS, and JavaScript and provides a declarative, component-based programming model that helps you efficiently develop user interfaces of any complexity.

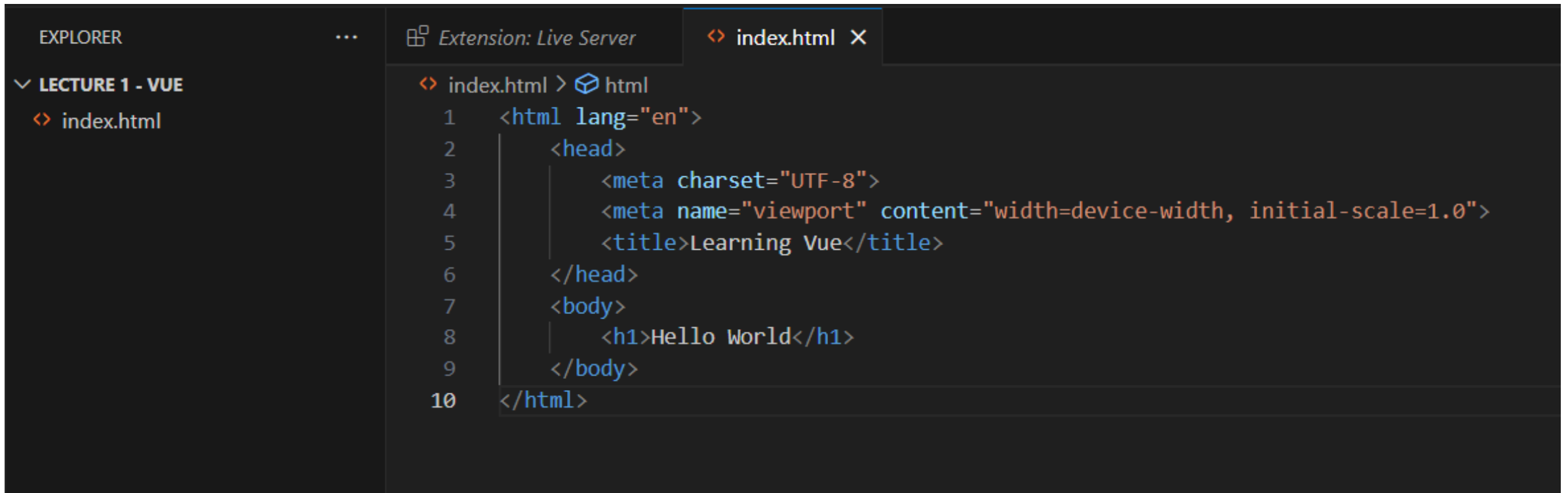
Here is a minimal example:

```
import { createApp, ref } from 'vue'

createApp({
  setup() {
    return {
      count: ref(0)
    }
  }
}).mount('#app')
```

js

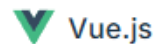
# Create index.html



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar is open, showing a folder named 'LECTURE 1 - VUE' with a file 'index.html' inside. The main editor area has two tabs: 'Extension: Live Server' and 'index.html'. The 'index.html' tab is active, displaying the following HTML code:

```
<?xml version="1.0" encoding="UTF-8" ?>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Learning Vue</title>
  </head>
  <body>
    <h1>Hello World</h1>
  </body>
</html>
```

# Install vue.js from CDN



Search

Docs ▾

API

Playground

Ecosystem ▾

About ▾

Sponsor

Partners



API Preference

Options ☐ Composition

## Getting Started

Introduction

Quick Start

## Essentials

Creating an Application

Template Syntax

Reactivity Fundamentals

Computed Properties

Class and Style Bindings

Conditional Rendering

List Rendering

Event Handling

Form Input Bindings

Lifecycle Hooks

Watchers

Template Refs

## Using Vue from CDN

You can use Vue directly from a CDN via a script tag:

```
<script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
```

html

Here we are using [unpkg](#), but you can also use any CDN that serves npm packages, for example [jsdelivr](#) or [cdnjs](#). Of course, you can also download this file and serve it yourself.

When using Vue from a CDN, there is no "build step" involved. This makes the setup a lot simpler, and is suitable for enhancing static HTML or integrating with a backend framework. However, you won't be able to use the Single-File Component (SFC) syntax.

## Using the Global Build

The above link loads the *global build* of Vue, where all top-level APIs are exposed as properties on the global `Vue` object. Here is a full example using the global build:

```
<script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
```

html

ON

Try

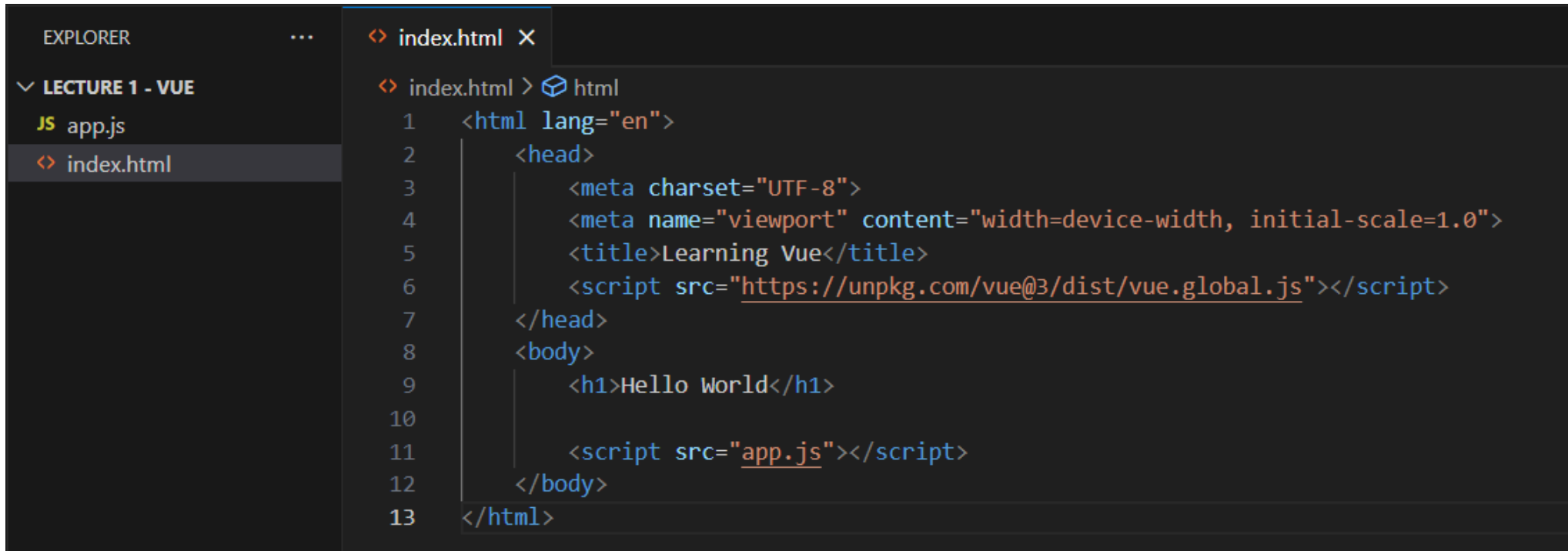
Cre

Usi

Nex

SPC

# Add js CDN to HTML and create app.js

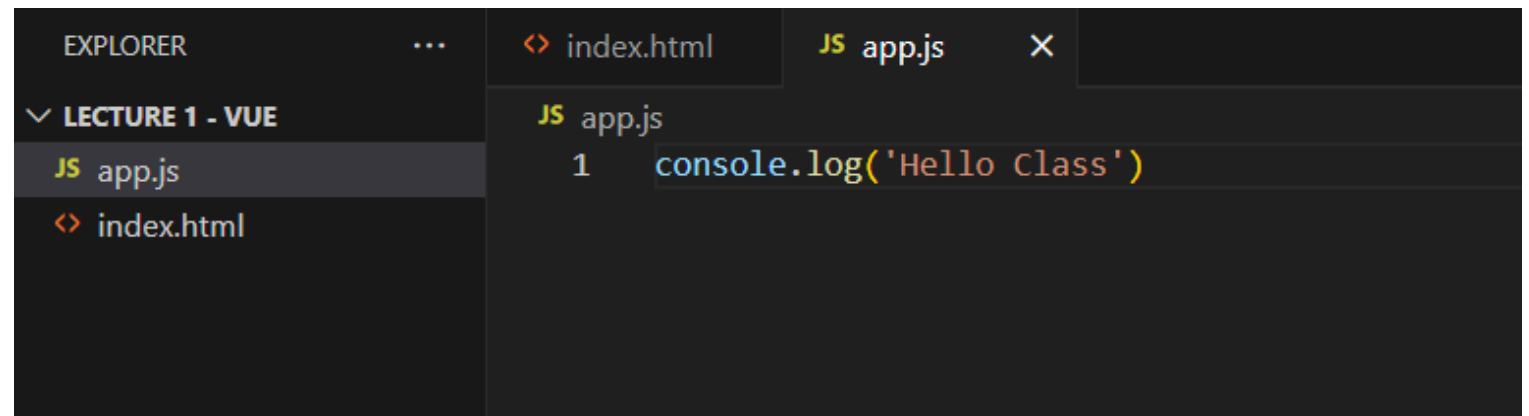


EXPLORER

- LECTURE 1 - VUE
  - JS app.js
  - index.html

index.html

```
1 <html lang="en">
2   <head>
3     <meta charset="UTF-8">
4     <meta name="viewport" content="width=device-width, initial-scale=1.0">
5     <title>Learning Vue</title>
6     <script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
7   </head>
8   <body>
9     <h1>Hello World</h1>
10
11     <script src="app.js"></script>
12   </body>
13 </html>
```



EXPLORER

- LECTURE 1 - VUE
  - JS app.js
  - index.html

index.html

app.js

```
1 console.log('Hello Class')
```

# Create Vue App

---

Activity 3 - Control parts of  
webpage

# Create Vue App

```
<> index.html X
<> index.html > html
1  <html lang="en">
2    <head>
3      <meta charset="UTF-8">
4      <meta name="viewport" content="width=device-width, initial-scale=1.0">
5      <title>Learning Vue</title>
6      <script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
7    </head>
8    <body>
9      <h1>Hello World</h1>
10
11      <div id="app"></div>
12
13      <script src="app.js"></script>
14    </body>
15  </html>
```

```
JS app.js X
JS app.js > ...
1  // use createApp object to create a vue app
2  const app = Vue.createApp()
3
4  // specify part of web page to control
5  app.mount('#app')
6
```

# Inject dynamic data, methods and templates

```
JS app.js X
JS app.js > ...
1 // use createApp object to create a vue app
2 const app = Vue.createApp({
3
4   //data
5
6   //functions
7
8
9
10  //templates
11  template: '<h2> You are welcome to the class. </h2>'
12
13 })
14
15 // specify part of web page to control
16 app.mount('#app')
17
```

```
<> index.html X
<> index.html > html
1 <html lang="en">
2   <head>
3     <meta charset="UTF-8">
4     <meta name="viewport" content="width=device-width, initial-scale=1.0">
5     <title>Learning Vue</title>
6     <script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
7   </head>
8   <body>
9     <h1>Hello World</h1>
10
11     <div id="app">
12       <h2> You are welcome to the class. </h2>
13     </div>
14
15     <script src="app.js"></script>
16   </body>
17 </html>
```

# Data Injection

---

Activity 4



# Inject dynamic data

```
// use createApp object to create a vue app
const app = Vue.createApp({

  //data
  data() {
    return {
      name: 'Trends in Technology',
      time: '14:00pm EST'
    }
  }

  //functions

  //templates
  //template: '<h2> You are welcome to the class. </h2>'
})

// specify part of web page to control
app.mount('#app')
```

```
<body>
  <h1>Hello World</h1>

  <div id="app">
    <h2> You are welcome to the class: {{ name }} </h2>
    <h3> Class starts at {{ time }}</h3>
  </div>

  <script src="app.js"></script>
</body>
```

# Event Handling

---

Activity 5

# use directive 'v-on' – how to react to events

```
<body>
  <h1>Hello World</h1>

  <div id="app">
    <h2> You are welcome to the class: {{ name }} </h2>
    <h3> Class starts at {{ time }}</h3>
    <p> Number of students: {{students}} </p>

    <button v-on:click="students++">Add Student</button>
    <button v-on:click="removeStudent()">Remove Student</button>
    <div @click="changeTime()">Change Time</div>
  </div>

  <script src="app.js"></script>
</body>
```

```
// use createApp object to create a vue app
const app = Vue.createApp({

  //data
  data() {
    return {
      name: 'Trends in Technology',
      time: '14:00pm EST',
      students: 1
    }
  },

  //functions
  methods: {
    changeTime() {
      this.time = '16:00pm EST'
    },

    removeStudent() {
      this.students = this.students - 1
    }
  }
})
```

# Conditional Rendering

---

Activity 6

# use directive 'v-if'

```
<body>
  <h1>Hello World</h1>

  <div id="app">
    <h2> You are welcome to the class: {{ name }} </h2>

    <div v-if="showTime">
      <h3> Class starts at {{ time }}</h3>
    </div>
    <p> Number of students: {{students}} </p>

    <button v-on:click="students++">Add Student</button>
    <button v-on:click="removeStudent()">Remove Student</button>
    <div @click="changeTime()" >Change Time</div>

    <button v-on:click="showTime=!showTime">Hide Time</button>

  </div>

  <script src="app.js"></script>
</body>
```

```
const app = Vue.createApp({

  //data
  data() {
    return {
      name: 'Trends in Technology',
      time: '14:00pm EST',
      students: 1,
      showTime: true
    }
  },

  //functions
  methods: {
    changeTime() {
      this.time = '16:00pm EST'
    },

    removeStudent() {
      this.students = this.students - 1
    },

    showTimes(){
      this.showTime = !this.showTime
    }
  },

  //templates
  //template: '<h2> You are welcome to the class. </h2>'

})
```

# use directive 'v-if'

```
<body>
  <h1>Hello World</h1>

  <div id="app">
    <h2> You are welcome to the class: {{ name }} </h2>

    <div v-if="showTime">
      <h3> Class starts at {{ time }}</h3>
    </div>
    <p> Number of students: {{students}} </p>

    <button v-on:click="students++">Add Student</button>
    <button v-on:click="removeStudent()">Remove Student</button>
    <div @click="changeTime()" >Change Time</div>

    <button v-on:click="toggleShowTimes()">
      <span v-if="showTime">Hide Time</span>
      <span v-if="!showTime">Show Time</span>
      <span v-else>Show Time</span>
    </button>

  </div>

  <script src="app.js"></script>
</body>
```

```
const app = Vue.createApp({

  //data
  data() {
    return {
      name: 'Trends in Technology',
      time: '14:00pm EST',
      students: 1,
      showTime: true
    }
  },

  //functions
  methods: {
    changeTime() {
      this.time = '16:00pm EST'
    },

    removeStudent() {
      this.students = this.students - 1
    },

    showTimes(){
      this.showTime = !this.showTime
    }

  },

  //templates
  //template: '<h2> You are welcome to the class. </h2>'

})
```

# Mouse Events

---

Activity 7

# use directive 'mouseover'

```
<br/>
<!--mouse events-->
<div class="box" v-on:mouseover="handleEvent">mouseover (enter)</div>
<div class="box" @mouseleave="handleEvent">mouseleave</div>
<div class="box" @dblclick="handleEvent">double click</div>
<div class="box"></div>
```

```
//functions
methods: {
  changeTime() {
    this.time = '16:00pm EST'
  },

  removeStudent() {
    this.students = this.students - 1
  },

  toggleShowTimes(){
    this.showTime = !this.showTime
  },

  handleEvent(e){
    console.log(e.type)
  }
}
```



# add data to event handle

```
<!--mouse events-->
<div class="box" v-on:mouseover="handleEvent($event, 5)">mouseover (enter)</div>
<div class="box" @mouseleave="handleEvent">mouseleave</div>
<div class="box" @dblclick="handleEvent">double click</div>
<div class="box"></div>
```

```
//functions
methods: {
  changeTime() {
    this.time = '16:00pm EST'
  },

  removeStudent() {
    this.students = this.students - 1
  },

  toggleShowTimes(){
    this.showTime = !this.showTime
  },

  handleEvent(e, data){
    console.log(e.type)
    if (data){
      console.log(data)
    }
  }
}
```

# Mouse Move Events -> Offset

```
<!--mouse events-->
<div class="box" v-on:mouseover="handleEvent($event, 5)">mouseover (enter)</div>
<div class="box" @mouseleave="handleEvent">mouseleave</div>
<div class="box" @dblclick="handleEvent">double click</div>
<div class="box" @mousemove="handleMouseMove">Position: {{x}}, {{y}} </div>
```

```
data() {
  return {
    name: 'Trends in Technology',
    time: '14:00pm EST',
    students: 1,
    showTime: true,
    x: 0,
    y: 0
  },
}
```

```
handleEvent(e, data){
  console.log(e.type)
  if (data){
    console.log(data)
  }
},

handleMouseMove(e){
  this.x = e.offsetX
  this.y = e.offsetY
}
```

# Outputting List

---

Activity 8

# Dynamic list rendering

```
<div id="app">
  <h2> You are welcome to the class: {{ name }} </h2>

  <div v-if="showClasses">
    <ul>
      <ul v-for="item in classes">
        Course: {{item.name}}, Time: {{item.time}}
      </ul>
    </ul>
  </div>

  <button v-on:click="toggleShowClasses()">
    <span v-if="showClasses">Hide Classes</span>
    <span v-if="!showClasses">Show Classes</span>
  </button>
```

```
//data
data() {
  return {
    students: 1,
    showClasses: true,
    x: 0,
    y: 0,
    classes: [
      {name: 'Trends in Technology', time: '14:00pm EST'},
      {name: 'Artificial Intelligence', time: '16:00pm EST'},
      {name: 'Mathematics for Machine Learning', time: '20:00pm EST' }
    ]
  }
},
```

```
//functions
methods: {

  removeStudent() {
    this.students = this.students - 1
  },

  toggleShowClasses(){
    this.showClasses = !this.showClasses
  },

  handleEvent(e, data){
    console.log(e.type)
    if (data){
      console.log(data)
    }
  },

  handleMouseMove(e){
    this.x = e.offsetX
    this.y = e.offsetY
  }
}
```

# Attribute Binding

---

Activity 9

# Dynamic attribute

```
<div id="app">
  <h2> You are welcome to
  |   <a v-bind:href="website">LaSalle College</a>
  </h2>

  <p>Available Courses:</p>

  <div v-if="showClasses">
    <ul>
      <ul v-for="item in classes">
        |   Course: {{item.name}}, Time: {{item.time}}
        </ul>
      </ul>
    </div>
  </div>
```

```
<div id="app">
  <h2> You are welcome to
  |   <a :href="website">LaSalle College</a>
  </h2>

  <p>Available Courses:</p>
```

```
//data
data() {
  return {
    website: 'https://lasallecollege.lcieducation.com/',
    students: 1,
    showClasses: true,
    x: 0,
    y: 0,
    classes: [
      {name: 'Trends in Technology', time: '14:00pm EST'},
      {name: 'Artificial Intelligence', time: '16:00pm EST'},
      {name: 'Mathematics for Machine Learning', time: '20:00pm EST' }
    ]
  }
},
```

# Dynamic attribute

```
<div v-if="showClasses">
  <ul>
    <ul v-for="item in classes">
      Course: {{item.name}}, Time: {{item.time}}
      
    </ul>
  </ul>
</div>
```

```
//data
data() {
  return {
    website: 'https://lasallecollege.lcieducation.com/',
    students: 1,
    showClasses: true,
    x: 0,
    y: 0,
    classes: [
      {name: 'Trends in Technology', time: '14:00pm EST', img: 'assets/1.png'},
      {name: 'Artificial Intelligence', time: '16:00pm EST', img: 'assets/2.jpg'},
      {name: 'Mathematics for Machine Learning', time: '20:00pm EST', img: 'assets/3.jpg' }
    ]
  }
},
```

# Dynamic Class

---

Activity 10



# Dynamic Classes

```
<p>Available Courses:</p>

<div v-if="showClasses">
  <ul>
    <li v-for="item in classes" :class="{ closed: item.isClosed }">
      
      <h3>Course: {{item.name}}</h3>
      <p>Time: {{item.time}}</p>
    </li>
  </ul>
</div>
```

```
p, h3, ul{
  margin: 0;
  padding: 0;
}

li{
  list-style-type:none;
  background: #fff;
  margin: 20px auto;
  padding: 10px 20px;
  border-radius: 10px;
  display: flex;
  align-items: center;
  justify-content: space-between;
}

li.closed{
  background-color: beige;
}
```

```
//data
data() {
  return {
    website: 'https://lasallecollege.lcieducation.com/',
    students: 1,
    showClasses: true,
    x: 0,
    y: 0,
    classes: [
      {name: 'Trends in Technology', time: '14:00pm EST', img:'assets/1.png', isClosed:true},
      {name: 'Artificial Intelligence', time: '16:00pm EST', img:'assets/2.jpg', isClosed:true},
      {name: 'Mathematics for Machine Learning', time: '20:00pm EST', img:'assets/3.jpg', isClosed:false }
    ]
  }
},
```

# Dynamic Toggling

```
<p>Available Courses:</p>

<div v-if="showClasses">
  <ul>
    <li v-for="item in classes" :class="{ closed: item.isClosed }" @click="toggleClosed(item)">
      
      <h3>Course: {{item.name}}</h3>
      <p>Time: {{item.time}}</p>
    </li>
  </ul>
</div>
```

```
handleMouseMove(e){
  this.x = e.offsetX
  this.y = e.offsetY
},

toggleClosed(item){
  item.isClosed = !item.isClosed
}
```

# Computed Properties

---

Activity 11

# Dynamic Filtering – Data Changes

```
//computed
computed: {
  filteredClasses(){
    return this.classes.filter((item) => item.isClosed )
  }
}
```

```
<div v-if="showClasses">
  <ul>
    <li v-for="item in filteredClasses" :class="{ closed: item.isClosed }">
      
      <h3>Course: {{item.name}}</h3>
      <p>Time: {{item.time}}</p>
    </li>
  </ul>
</div>
```