Chapter 1 - Introduction to Vue

Asst.Prof. Dr. Umaporn Supasitthimethee
ผศ.ดร.อุมาพร สุภสิทธิเมธี

Client-Side Technology Stack

JavaScript Framework

- Vue.js (version 3.2), https://blog.vuejs.org/posts/vue-3.2.html
- Composition API Style (*script setup syntax*)
- Vite Build Tool
- Pinia store library for state management

CSS Framework

- TailwindCSS Framework
- Bootstrap Framework

Other dev tools

- VsCode Editor
- Vue Devtools
- Git version control

Official Web Resources

- JavaScript
  https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference
- Vue3
  https://v3.vuejs.org/guide/introduction.html
  https://blog.vuejs.org/posts/vue-3.2.html
- Vue Composition API
  https://v3.vuejs.org/api/composition-api.html#setup
  https://staging.vuejs.org/
- Single File Components <script setup>
  https://v3.vuejs.org/api/sfc-script-setup.html
- Vue Devtools
  https://devtools.vuejs.org/guide/installation.html
- Vite Build Tool
  https://vitejs.dev/
- Pinia store library for Vue
  https://pinia.vuejs.org/introduction.html
- CSS Framework
  https://tailwindcss.com/
  https://getbootstrap.com/
- Images
  https://unsplash.com/
- Icons
  https://material.io/resources/icons/?style=baseline
  https://iconmonstr.com/iconicfont/
- Colors
  https://coolors.co/

About Vue.js

- **Vue.js: The Documentary**, https://youtu.be/OrxmtDw4pVI
- **Why Vue.js**, https://v3.vuejs.org/

VSCode Extension Recommendation

- *Prettier* Code Formatter
- *Vue Language Features* - Vue3 Syntax Highlight and Auto Complete (Choose One *Volar* |*Vetur*)
- *Live Share* – Real-time Collaborative Development

Vue.js

- Vue (pronounced /vju:/, like view)
- A JavaScript Framework for building user interfaces
- It builds on top of standard HTML, CSS and JavaScript, and provides a declarative and component-based programming model that helps you efficiently develop user interfaces, be it simple or complex.
- The Vue official guide assumes intermediate level knowledge of HTML, CSS, and JavaScript.
- If you are totally new to frontend development, it might not be the best idea to jump right into a framework as your first step

Vue Installation

There are **four primary ways** of adding **Vue.js** to a project:

1. Import it as a CDN package on the page
2. Download the JavaScript files and host them yourself
3. Install it using npm
4. Use the official CLI or Vite to scaffold a project, which provides batteries-included build setups for a modern frontend workflow (e.g., hot-reload, lint-on-save, and much more

Vite (Veet)

- a build tool that aims to provide a faster and leaner development experience for modern web projects

https://vitejs.dev/

Vite Requirement

1. Install node.js, https://nodejs.org/en/ (**require version >=12.0.0 for Vite**)

```
>node -v
```

2. npm package manager (**recommend version 6 up**)

```
>npm -v
```

Your First Vite Project

```
>npm init vite@latest <your project name>
>cd <your project name>
>npm install
>npm run dev
```

Single File Components

- Vue Single File Components (aka *.*vue* files, abbreviated as **SFC**) is a special file format that allows us to encapsulate ***the template, logic, and styling*** of a Vue component in a single file.
- SFC is a defining feature of Vue as a framework, and is the recommended approach for using Vue

```
//Single File Components

<script setup>
//JavaScript Variables, functions, Vue libraries
</script>

<template>
//html
</template>

<style>
//styling
</style>
```

https://v3.vuejs.org/guide/single-file-component.html#introduction

VSCode Snippet SFC

File > Preferences > User Snippets > Vue.json

```
{
    "Vue SFC Script Setup": {
    "prefix": "sfc",
    "body": [
      "<script setup>",
      "$1",
      "</script>",
      " ",
      "<template>",
      "$2",
      "</template>",
      " ",
      "<style>",
      "$3",
      "</style>"
    ],
    "description": "Making simple Vue SFC Script Setup"
  }
}
```

https://code.visualstudio.com/docs/editor/userdefinedsnippets#_create-your-own-snippets

The Application Instance

- Every Vue application starts by creating a new **application instance** with the *createApp* function:

```
//main.js
import { createApp } from 'vue'
// import the root component App from a single-file component.
import App from './App.vue'

createApp(App).mount('#app')
```

```
<!—index.html-->
<body>
    <div id="app"></div>
    <script type="module“
src="/src/main.js"></script>
  </body>
```

- The object we are passing into ***createApp*** is in fact a component. Every app requires a **"root component"** that can contain other components as its children.
- An application instance won't render anything until its **.mount()** method is called.

Vue.js devtools

- Get the Chrome Extension

  https://chrome.google.com/webstore/detail/vuejs-devtools/ljjemllljcmogpfapbkkighbhhppjdbg

- Get the Firefox Addon

  https://addons.mozilla.org/en-US/firefox/addon/vue-js-devtools/

CSS Frameworks: TailwindCSS

Install Tailwind CSS with Vue 3 and Vite

- https://tailwindcss.com/docs/guides/vite

tailwindCSS

- https://tailwindcss.com/

daisyUI, Tailwind CSS Components

- https://daisyui.com/docs/install

CSS Frameworks: bootstrap

Install Bootstrap in your Node.js with the npm package:

https://getbootstrap.com/docs/5.0/getting-started/download/#npm

```
>npm install bootstrap
```

```
//main.js
import 'bootstrap/dist/css/bootstrap.min.css’
```