Introduction of Vue.js 3

What is Vue.js?

Vue.js is a popular JavaScript framework used to build modern, interactive web applications. It helps developers create user interfaces and single-page applications efficiently and effectively.

Why Vue.js 3?

Vue.js 3 is the latest major version of Vue.js, offering several improvements over its predecessor, Vue.js 2. It introduces new features and optimizations that make developing applications even smoother.

Key Features of Vue.js 3:

1. **Performance Improvements**: Vue.js 3 is faster and more efficient, with improved performance in rendering and smaller bundle sizes, making your applications run faster.

2. **Better TypeScript Support**: Vue.js 3 has built-in support for TypeScript, a superset of JavaScript that helps catch errors early and provides better tooling.

3. **Teleport**: This feature allows you to move a component's content to a different part of the DOM, useful for modals or tooltips that need to be outside the main application layout.

How Does Vue.js Work?

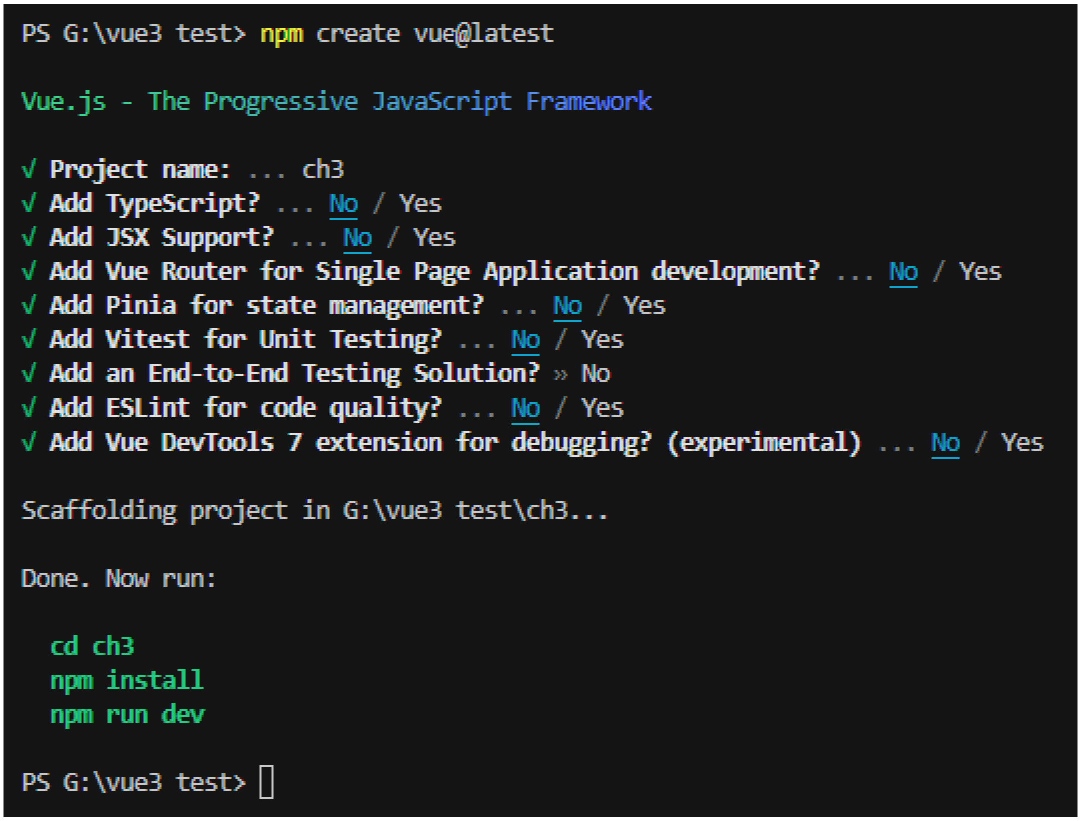
1. **Components**: Vue.js applications are built using components. A component is a reusable piece of the UI, like a button or a form. Each component has its own logic and template.

2. **Templates**: Vue.js uses HTML templates to define how your components look. You can use plain HTML and Vue's special directives to bind data and control the behavior of your components.

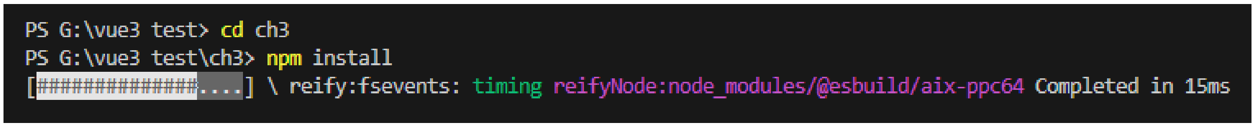
3. **Directives**: Vue.js provides special attributes called directives that you can use in your templates to perform tasks like conditional rendering, loops, and event handling.

Getting Started:

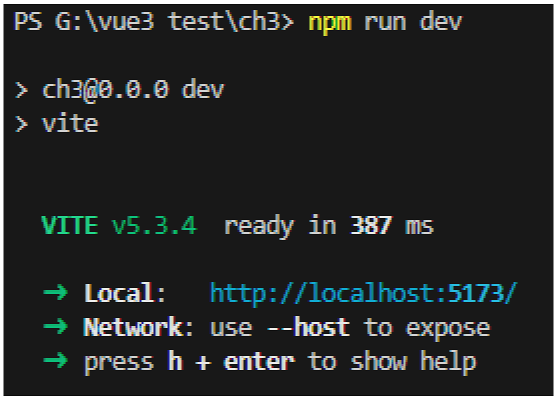
1. **Set Up Your Project**: You can quickly set up a Vue.js 3 project using the Vue CLI (Command Line Interface). Just run `npm create vue@latest` to get started.



```
PS G:\vue3 test> npm create vue@latest

Vue.js - The Progressive JavaScript Framework

√ Project name: ... ch3
√ Add TypeScript? ... No / Yes
√ Add JSX Support? ... No / Yes
√ Add Vue Router for Single Page Application development? ... No / Yes
√ Add Pinia for state management? ... No / Yes
√ Add Vitest for Unit Testing? ... No / Yes
√ Add an End-to-End Testing Solution? » No
√ Add ESLint for code quality? ... No / Yes
√ Add Vue DevTools 7 extension for debugging? (experimental) ... No / Yes

Scaffolding project in G:\vue3 test\ch3...

Done. Now run:

  cd ch3
  npm install
  npm run dev

PS G:\vue3 test> 
```



```
PS G:\vue3 test> cd ch3
PS G:\vue3 test\ch3> npm install
[##############....] \ reify:fsevents: timing reifyNode:node_modules/@esbuild/aix-ppc64 Completed in 15ms
```



```
PS G:\vue3 test\ch3> npm run dev

> ch3@0.0.0 dev
> vite


  VITE v5.3.4  ready in 387 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

2. **Create Components**: Define your components in `.vue` files. Each component consists of three parts: a template, a script, and a style section.

3. **Run Your Application**: Use the development server provided by Vue CLI to run your application locally and see your changes in real-time.

4. **Build and Deploy**: Once you're ready, you can build your application for production and deploy it to a web server.