

Vue.js

The Progressive Framework

KATHIRAVAN K

Kathiravan K

Senior Software Engineer With 11+ Years of experience in
Web and UI Development And 5+ Years of experience in
Vue.js.



 kathirr007

 kathirr007

 kathirr_007



Day 1 - Agenda

- Introduction
- Get Started



Introduction

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 } from 'vue'  
createApp({  
  data() {  
    return {  
      count: 0  
    }  
  }  
}).mount('#app')  
  
<div id="app">  
  <button @click="count++">  
    Count is: {{ count }}  
  </button>  
</div>
```

Result:



Introduction

The Progressive Framework

Vue is a framework and ecosystem that covers most of the common features needed in frontend development. But the web is extremely diverse - the things we build on the web may vary drastically in form and scale. With that in mind, Vue is designed to be flexible and incrementally adoptable. Depending on your use case, Vue can be used in different ways:

- Enhancing static HTML without a build step
- Embedding as Web Components on any page
- Single-Page Application (SPA)
- Fullstack / Server-Side Rendering (SSR)
- Jamstack / Static Site Generation (SSG)
- Targeting desktop, mobile, WebGL, and even the terminal

If you are an experienced developer interested in how to best integrate Vue into your stack, or you are curious about what these terms mean, we discuss them in more detail in [Ways of Using Vue](#).

Despite the flexibility, the core knowledge about how Vue works is shared across all these use cases. Even if you are just a beginner now, the knowledge gained along the way will stay useful as you grow to tackle more ambitious goals in the future. If you are a veteran, you can pick the optimal way to leverage Vue based on the problems you are trying to solve, while retaining the same productivity. This is why we call Vue "The Progressive Framework": it's a framework that can grow with you and adapt to your needs.



Introduction

Single File Component

In most build-tool-enabled Vue projects, we author Vue components using an HTML-like file format called Single-File Component (also known as *.vue files, abbreviated as SFC). A Vue SFC, as the name suggests, encapsulates the component's logic (JavaScript), template (HTML), and styles (CSS) in a single file. Here's the previous example, written in SFC format:

SFC Example for Options API

```
<script>
export default {
  data() {
    return {
      count: 0
    }
  }
}
</script>

<template>
  <button @click="count++">Count is: {{ count }}</button>
</template>

<style scoped>
```



Introduction

API Styles

Vue components can be authored in two different API styles: [Options API](#) and [Composition API](#).

Options API

With Options API, we define a component's logic using an object of options such as `data`, `methods`, and `mounted`. Properties defined by options are exposed on this inside functions, which points to the component instance:

```
<script>
export default {
  // Properties returned from data() become reactive state
  // and will be exposed on `this`.
  data() {
    return {
      count: 0
    }
  },
  // Methods are functions that mutate state and trigger updates.
  // They can be bound as event handlers in templates.
}
```

TypeScript Serverless Server Components

Middleware Hybrid Rendering Composables Auto Imports

Modules Ecosystem Hot Module Replacement Nitro SEO

Developer Experience

Server-side Rendering File-based Routing ESM First Vite Powered

Zero-config Edge Rendering Components Auto Imports

Static Site Generation Layouts Server API

Conventions

Abstractions

Sensible Defaults

Normalizations

Transparency

Transparency

- Easy to learn & understand
- Easy to modify
- Easy to debug

Introducing



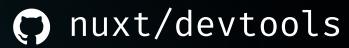
The goal of



- Enhance your DX with Nuxt
- Improve Transparency
- Performance & analysis
- Interactive & playful
- Personalized documentations

Demo time!

Open Sourced at



Q&A

Thank You!

Slides on antfu.me
