

SLIDEV Docs - Việt Nam



Sliddev

Table of contents

• Vn - README	4
• Vn - TRANSLATIONS	5
• Builtin - Components	8
• Builtin - Layouts	9
• Custom - Config katex	13
• Custom - Config mermaid	14
• Custom - Config monaco	15
• Custom - Config shortcuts	19
• Custom - Config vite	20
• Custom - Config vue	22
• Custom - Config windicss	23
• Custom - Directory structure	24
• Custom - Fonts	28
• Custom - Global layers	31
• Custom - Highlighters	33
• Custom - Index	35
• Custom - Vue context	37
• Guide - Animations	39
• Guide - Drawing	43
• Guide - Editors	45
• Guide - Exporting	48
• Guide - Faq	51
• Guide - Hosting	54
• Guide - Index	57
• Guide - Install	62
• Guide - Navigation	64
• Guide - Presenter mode	67
• Guide - Recording	68
• Guide - Syntax	69
• Guide - Why	80
• Resources - Covers	83
• Resources - Learning	84
• Vn - Showcases	85
• Themes - Gallery	86

- Themes - Use _____ 87
- Themes - Write a /theme _____ 89

sli.dev

Tài liệu cho [Slidev](#)

Ngôn ngữ

	Repo	Site	Maintainers
English	docs	sli.dev	@antfu
简体中文	docs-cn	cn.sli.dev	@QC-L @Ivocin
Français	docs-fr	fr.sli.dev	@ArthurDanjou
Español	docs-es	es.sli.dev	@owlnai
Русский	docs-ru	ru.sli.dev	@xesjkeee
Việt Nam	docs-vn	vn.sli.dev	@bongudth
Deutsch	docs-de	de.sli.dev	@fabiankachlock
Português (BR)	docs-br	br.sli.dev	@luisfelipesdn12
Ελληνικά	docs-el	el.sli.dev	@GeopJr

Start Server Locally

```
npm i -g pnpm
pnpm i
pnpm run dev
```

Và sau đó truy cập <http://localhost:3000>

Hoặc cài đặt [Vite extension cho VS Code](#) để chỉnh sửa song song.

Trợ giúp về phiên dịch

See [TRANSLATIONS.md](#)

[Go to TOC](#)

Help on Translating

First of all, thank you for being interested in contributing to translations!

You can find the repositories for each existing translation in [README.md](#). To help improve them, simply sending a Pull Request to their repo.

If the language you want to contribute isn't on the list, join [our Discord server](#), and find the `#translations` channel to see if someone is already working on the language you want, consider joining them and translate together. If not, you can start a new translation project with the following steps.

In case it's already been translated but you're wondering how to maintain it, skip to the end. ## Some tips before you get started

- It is recommended that you use your IDE of choice (e.g VSCode) paired with a development server running, so you can see your translation changes in real-time.
- You can mark these checkmarks as the translation progresses or use your own workflow. The translations don't need to be made in any particular order.
- Translations don't need to be literal, but they should convey the same message. In case you're not sure how to translate something, you can either leave it as it is or use online tools like WordReference or Linguee to aid you.
- Most translations will simply consist in editing Markdown files. Certain areas are buried under Vue components, which will be listed below. You can also use your IDE to find the string to translate.

Getting started

- Fork the main docs repo: [slidevjs/docs](#)
- Translate README.md, you can take one of the already translated repositories as an example.
- Share your repo's link to the `#translations` channel telling people you are working on it and find collaborators.

Translating Markdown files

- `showcases.md` - A gallery showcase of Slidev presentations.
- `index.md` - Mainpage content, note that some of it is buried under Vue components listed further below.

.vitepress/

- `config.js` - Sitemap
- `/theme/components/WorkingInProgress.vue` - WIP notice shown in mainpage
- `/theme/components/demo/Demo.vue` - Animated demo shown in mainpage
- `/theme/components/Environment.vue` - Describes the environment of a setting.

builtin/

- `components.md` - Use [Vue components](#) inside Slidev
- `layouts.md` - Use Vue layouts inside Slidev

custom/

- `config-katex.md` - Configuring Katex
- `config-mermaid.md` - Configuring Mermaid
- `config-monaco.md` - Configuring Monaco
- `config-shortcuts.md` - Configuring Shortcuts
- `config-vite.md` - Configuring Vite
- `config-vue.md` - Configuring Vue
- `config-windicss.md` - Configuring Windicss
- `directory-structure.md` - Configuring the directory structure
- `fonts.md` - Configuring fonts
- `global-layers.md` - Configuring the global layers
- `highlighters.md` - Configuring code highlighters
- `index.md` - Customizations index page
- `vue-context.md` - The Vue global context

guide/

- `animations.md` - Animations and transitions
- `editors.md` - Editor integrations
- `exporting.md` - Exporting your slides
- `faq.md` - Frequent Answered Questions
- `index.md` - Getting started with Slidev
- `navigation.md` - Navigation across slides
- `presenter-mode.md` - Toggling presenter mode
- `recording.md` - Recording your presentation
- `syntax.md` - Markdown syntax
- `why.md` - *Why Slidev?*

resources/

- `covers.md` - Curated covers for Slidev

themes/

- `gallery.md` - Theme gallery
- `use.md` - How to use Slidev themes
- `write-a-theme.md` - Write your own theme

Publishing your translations

- When you finish the translation (at least 90%), `@antfu` in the Discord and we will invite you to the org and make the translation official.
- Once the transferring is done, we will set up the subdomain, auto-deployment, and a daily sync-up bot to keep the translation up-to-date with the latest English docs.
- The site is live, and we will send a shout-out tweet on [our Twitter account](#).

Maintaining the translations up-to-date

- `docschina-bot` will periodically submit merge requests from the `slidev/docs` repository. Switch to the branch created in the pull request, make any changes necessary and merge it. [example](#).
- Sometimes it will occur that a merge request is made and you haven't merged the previous one. The latest PR always checks your main branch against the English one; so you can just close the previous PR(s), move your work to the latest one and merge it.

[Working-in-progress translation list](#)

Thanks again!

[Go to TOC](#)

Component

Tích hợp Component

Các tài liệu của phần này vẫn đang được hoàn thiện. Trước đó, bạn có thể xem trực tiếp tại [source code](#).

Tùy chỉnh Component

Tạo một thư mục `components/` trong dự án gốc của bạn và chỉ cần đặt các component Vue tùy chỉnh của bạn bên trong nó, sau đó bạn có thể sử dụng nó với cùng tên trong file markdown của mình!

Đọc thêm trong phần [Tùy chỉnh](#)

Component do Chủ đề cung cấp

Chủ đề cũng có thể cung cấp các component. Vui lòng đọc tài liệu về nó để biết những gì nó cung cấp.

Kiểm tra thêm trong phần [Cấu trúc thư mục](#).

[Go to TOC](#)

Layout

Tích hợp Layout

<<<<< HEAD

**Các tài liệu của phần này vẫn đang
được hoàn thiện. Trước đó, bạn có thể
xem trực tiếp tại [source code](#).**

As themes may override layouts behaviour, the best way to know exactly the usage, parameters and examples is referring their documentation.

center

Displays the content in the middle of the screen.

cover

Used to display the cover page for the presentation, may contain the presentation title, contextualization, etc.

default

The most basic layout, to display any kind of content.

end

The final page for the presentation.

fact

To show some fact or data with a lot of prominence on the screen.

full

Use all the space of the screen to display the content.

image-left

Shows an image on the left side of the screen, the content will be placed on the right side.

Usage

```
---  
layout: image-left  
  
# the image source  
image: ./path/to/the/image  
  
# a custom class name to the content  
class: my-cool-content-on-the-right  
---
```

image-right

Shows an image on the right side of the screen, the content will be placed on the left side.

Usage

```
---  
layout: image-right  
  
# the image source  
image: ./path/to/the/image  
  
# a custom class name to the content  
class: my-cool-content-on-the-left  
---
```

image

Shows an image as the main content of the page.

Usage

```
---  
layout: image  
  
# the image source  
image: ./path/to/the/image  
---
```

iframe-left

Shows a web page on the left side of the screen, the content will be placed on the right side.

Usage

```
---  
layout: iframe-left  
  
# the web page source  
url: https://github.com/slidesjs/slides  
  
# a custom class name to the content  
class: my-cool-content-on-the-right  
---
```

iframe-right

Shows a web page on the right side of the screen, the content will be placed on the left side.

Usage

```
---
layout: iframe-right
# the web page source
url: https://github.com/slidesjs/slides
# a custom class name to the content
class: my-cool-content-on-the-left
---
```

iframe

Shows a web page as the main content of the page.

Usage

```
---
layout: iframe
# the web page source
url: https://github.com/slidesjs/slides
---
```

intro

To introduce the presentation, usually with the presentation title, a short description, the author, etc.

none

A layout without any existent styling.

quote

To display a quotation with prominence.

section

Used to mark the beginning of a new presentation section.

statement

Make an affirmation/statement as the main page content.

two-cols

Separates the page content in two columns.

Usage

```
---  
layout: two-cols  
---  
  
# Left  
  
This shows on the left  
  
::right::  
  
# Right  
  
This shows on the right
```

ee7ae42035591cb6565a72f5217129c670a59b0c

Tùy chỉnh Layout

Tạo một thư mục `layouts/` trong dự án gốc của bạn và chỉ cần đặt các component layout Vue tùy chỉnh của bạn bên trong nó.

Đọc thêm trong phần [Các tùy chỉnh](#)

Layout do Chủ đề cung cấp

Chủ đề có thể cung cấp các layout hoặc ghi đè các layout hiện có. Vui lòng đọc tài liệu về nó để biết những gì nó cung cấp.

[Go to TOC](#)

Cấu hình KaTeX

Tạo `./setup/katex.ts` với nội dung sau:

```
import { defineKatexSetup } from '@slidev/types'

export default defineKatexSetup(() => {
  return {
    /* ... */
  }
})
```

Trong quá trình thiết lập, bạn có thể cung cấp cài đặt tùy chỉnh cho [Các lựa chọn KaTeX](#). Tham khảo định nghĩa loại và tài liệu của chúng để biết thêm chi tiết.

Cấu hình Mermaid

Tạo `./setup/mermaid.ts` với nội dung sau:

```
import { defineMermaidSetup } from '@slidev/types'

export default defineMermaidSetup(() => {
  return {
    theme: 'forest',
  }
})
```

Trong quá trình thiết lập, bạn có thể cung cấp cài đặt mặc định tùy chỉnh cho [Mermaid](#). Tham khảo định nghĩa loại và tài liệu của chúng để biết thêm chi tiết.

Cấu hình Monaco

<<<<< HEAD

Theo mặc định, Monaco chỉ được bật trên chế độ dev. Để nó hoạt động trên bản dựng SPA, hãy thêm `monaco: true` vào cấu hình frontmatter của bạn.

Tạo `./setup/monaco.ts` với nội dung sau:

Create `./setup/monaco.ts` with the following content:

```
ee7ae42035591cb6565a72f5217129c670a59b0c
```

```
import { defineMonacoSetup } from '@slidev/types'

export default defineMonacoSetup(async (monaco) => {
  // use `monaco` to configure
})
```

Tìm hiểu nhiều hơn tại [Cấu hình Monaco](#).

Sử dụng

Để sử dụng Monaco trong các trang trình bày của bạn, chỉ cần thêm `{monaco}` vào các đoạn code của bạn:

```
//```js
const count = ref(1)
const plusOne = computed(() => count.value + 1)

console.log(plusOne.value) // 2
plusOne.value++ // error
//```

```

Và

```
//```js {monaco}
const count = ref(1)
const plusOne = computed(() => count.value + 1)

console.log(plusOne.value) // 2
plusOne.value++ // error
//```

```

Xuất trang trình bày

<<<<< HEAD Theo mặc định, Monaco sẽ CHỈ hoạt động ở chế độ `dev`. Nếu bạn cũng muốn có nó trong SPA đã xuất, bạn có thể định cấu hình nó trong frontmatter của mình: ===== By default, Monaco will ONLY work on `dev` mode. If you would like to have it available in the exported SPA, configure it in your frontmatter:

ee7ae42035591cb6565a72f5217129c670a59b0c

```
---
monaco: true # default "dev"
---
```

Các loại cài đặt tự động

<<<<< HEAD Khi bạn sử dụng TypeScript với Monaco, các kiểu dependencie sẽ được cài đặt tự động cho phía client. ===== When use TypeScript with Monaco, types for dependencies will be installed to the client-side automatically.

ee7ae42035591cb6565a72f5217129c670a59b0c

```
//``ts {monaco}
import { ref } from 'vue'
import { useMouse } from '@vueuse/core'

const counter = ref(0)
//``
```

<<<<< HEAD Trong ví dụ trên, chỉ cần đảm bảo rằng `vue` và `@vueuse/core` được cài đặt cục bộ dưới dạng dependencies / devDependencies, Slidev sẽ xử lý phần còn lại và trình soạn thảo của bạn sẽ hoạt động! ===== In the example above, make sure `vue` and `@vueuse/core` are installed locally as dependencies / devDependencies, Slidev will handle the rest to get the types working for the editor automatically!

Configure Themes

The theme is controlled by Slidev based on the light/dark theme. If you want to customize it, you can pass the theme id to the setup function:

```
// ./setup/monaco.ts
import { defineMonacoSetup } from '@slidev/types'

export default defineMonacoSetup(() => {
  return {
    theme: {
```

```
        dark: 'vs-dark',
        light: 'vs',
    },
})
```

If you want to load custom themes:

```
import { defineMonacoSetup } from '@slidev/types'

// change to your themes
import dark from 'theme-vitesse/themes/vitesse-dark.json'
import light from 'theme-vitesse/themes/vitesse-light.json'

export default defineMonacoSetup((monaco) => {
    monaco.editor.defineTheme('vitesse-light', light as any)
    monaco.editor.defineTheme('vitesse-dark', dark as any)

    return {
        theme: {
            light: 'vitesse-light',
            dark: 'vitesse-dark',
        },
    }
})
```

If you are creating a theme for Slidev, use dynamic `import()` inside the setup function to get better tree-shaking and code-splitting results.

ee7ae42035591cb6565a72f5217129c670a59b0c

Cấu hình Phím tắt

Có hiệu lực từ v0.20

Tạo `./setup/shortcuts.ts` với nội dung sau:

```
import { defineShortcutsSetup, NavOperations } from '@slidev/types'

export default defineShortcutsSetup((nav: NavOperations) => {
  return [
    {
      key: 'enter',
      fn: () => nav.next(),
      autoRepeat: true,
    },
    {
      key: 'backspace',
      fn: () => nav.prev(),
      autoRepeat: true,
    },
  ]
})
```

Với việc cài đặt, bạn có thể cung cấp cài đặt tùy chỉnh cho các phím tắt được đề cập trong [Navigation](#) Cấu hình trên liên kết animation hoặc trang chiếu tiếp theo với `enter` và animation hoặc trang chiếu trước đó với `backspace`.

Hàm cấu hình nhận một đối tượng với một số hàm navigation và trả về một mảng chứa một số cấu hình phím tắt. Tham khảo định nghĩa loại để biết thêm chi tiết.

Tham khảo [useMagicKeys | VueUse](#) để biết thêm chi tiết về sự kiện được nhấn phím.

Cấu hình Vite

Slidev được cung cấp bởi [Vite](#). Điều này có nghĩa là bạn có thể tận dụng hệ thống plugin tuyệt vời của Vite để tùy chỉnh các trang trình bày của mình hơn nữa.

Nếu có file `vite.config.js` trong dự án, nó sẽ được đọc.

Slidev có các plugin sau được cấu hình sẵn:

- [@vitejs/plugin-vue](#)
- [unplugin-vue-components](#)
- [unplugin-icons](#)
- [vite-plugin-md](#)
- [vite-plugin-windicss](#)
- [vite-plugin-remote-assets](#)

<<<<< HEAD Tìm hiểu thêm về [Cấu hình trước đó](#). ====== Learn more about the [pre-configurations here](#).

Configure Internal Plugins

Available since v0.21

To configure the built-in plugins list above, create `vite.config.ts` with the following content. Please note Slidev has some preconfigure options for those plugins, this usage will override some of them, which could potentially cause the app to break. Please treat this as **an advanced feature**, make sure you know what you are doing before moving on.

```
import { defineConfig } from 'vite'

export default defineConfig({
  slidev: {
    vue: {
      /* vue options */
    },
    markdown: {
      /* markdown-it options */
      markdownItSetup(md) {
        /* custom markdown-it plugins */
        md.use(/* ... */)
      },
      /* options for other plugins */
    },
  },
})
```

See the [type declarations](#) for more options.

ee7ae42035591cb6565a72f5217129c670a59b0c

Cấu hình Vue

Slidev sử dụng [Vue 3](#) để hiển thị ứng dụng ở phía client. Bạn có thể mở rộng ứng dụng để thêm các cấu hình hoặc plugin tùy chỉnh.

Tạo `./setup/main.ts` với nội dung sau:

```
import { defineAppSetup } from '@slidev/types'

export default defineAppSetup(({ app, router }) => {
  // Vue App
  app.use(YourPlugin)
})
```

Đây cũng có thể được sử dụng làm lối vào chính của ứng dụng Slidev của bạn, để thực hiện một số thao tác khởi tạo trước khi ứng dụng khởi động.

Tìm hiểu thêm về: [Vue Application API](#).

Cấu hình Windi CSS

Markdown hỗ trợ các thẻ HTML được nhúng một cách tự nhiên. Do đó, bạn có thể tạo nội dung của mình theo cách bạn muốn. Để mang lại sự tiện lợi, chúng tôi đã tích hợp sẵn [Windi CSS built-in](#), vì vậy bạn có thể tạo thẻ HTML trực tiếp bằng các class.

Ví dụ:

```
<div class="grid pt-4 gap-4 grids-cols-[100px,1fr]>
  ### Name
  - Item 1
  - Item 2
</div>
```

Attributify Mode trong [Windi CSS v3.0](#) mặc định được bật.

Cấu hình

Để định cấu hình CSS Windi, hãy tạo `setup/windicss.ts` với nội dung sau để mở rộng cấu hình tích hợp

```
// setup/windicss.ts

import { defineWindiSetup } from '@slidev/types'

// extending the builtin windicss configurations
export default defineWindiSetup(() => ({
  shortcuts: {
    // custom the default background
    'bg-main': 'bg-white text-[#181818] dark:(bg-[#121212] text-[#ddd])',
  },
  theme: {
    extend: {
      // fonts can be replaced here, remember to update the web font links in
      `index.html`
      fontFamily: {
        sans: 'ui-sans-serif,system-ui,-apple-system,BlinkMacSystemFont,"Segoe
        UI",Roboto,"Helvetica Neue",Arial,"Noto Sans",sans-serif,"Apple Color
        Emoji","Segoe UI Emoji","Segoe UI Symbol","Noto Color Emoji"',
        mono: '"Fira Code", monospace',
      },
    },
  },
}))
```

Tìm hiểu thêm về: [Cấu hình Windi CSS](#)

[Go to TOC](#)

Cấu trúc thư mục

Slidev sử dụng một số quy ước cấu trúc thư mục để thu nhỏ bề mặt cấu hình và làm cho các phần mở rộng chức năng trở nên linh hoạt và trực quan.

Cấu trúc cơ bản như sau:

```
your-slidev/
  └── components/      # custom components
  └── layouts/         # custom layouts
  └── public/          # static assets
  └── setup/           # custom setup / hooks
  └── styles/          # custom style
  └── index.html        # injections to index.html
  └── slides.md         # the main slides entry
  └── vite.config.ts    # extending vite config
```

Tất cả chúng đều là tùy chọn.

Component

Quy ước: `./components/*.{vue,js,ts,jsx,tsx,md}`

Các component bên trong thư mục này có thể được sử dụng trực tiếp trong các slide Markdown với component cùng tên với tên file.

Ví dụ:

```
your-slidev/
  └── ...
  └── components/
      └── MyComponent.vue
      └── HelloWorld.ts
```

```
<!-- slides.md -->
# Slide của tôi
<MyComponent :count="4"/>
<!-- both namings work -->
<hello-world foo="bar">
  Slot
</hello-world>
```

Tính năng này được cung cấp bởi `vite-plugin-components`, hãy tìm hiểu thêm tại đây.

Slidev cũng cung cấp một số [component tích hợp](#) để bạn sử dụng.

Layout

Quy ước: `./layouts/*.{vue,js,ts,jsx,tsx}`

```
your-slidev/
  ...
  └── layouts/
    ├── cover.vue
    └── my-cool-theme.vue
```

Bạn có thể sử dụng bất kỳ tên file nào cho layout của mình. Sau đó, bạn tham chiếu layout của mình trong tiêu đề YAML bằng tên file.

```
---  
layout: my-cool-theme  
---
```

Nếu layout bạn cung cấp có cùng tên với layout cài sẵn hoặc layout chủ đề, layout tùy chỉnh của bạn sẽ được ưu tiên hơn layout tích hợp / layout chủ đề. Thứ tự ưu tiên là `cục bộ > chủ đề > tích hợp sẵn`.

Trong layout component, sử dụng `<slot/>` cho nội dung trang chiếu. Ví dụ:

```
<!-- default.vue -->
<template>
  <div class="slidev-layout default">
    <slot />
  </div>
</template>
```

Public

Quy ước: `./public/*`

Các asset trong thư mục này sẽ được dùng tại đường dẫn gốc `/` trong quá trình dev và được sao chép vào thư mục gốc của thư mục dist nguyên trạng. Đọc thêm về [Thư mục public của Vite](#).

Style

Quy ước: `./style.css | ./styles/index.{css,js,ts}`

Các file theo quy ước này sẽ được đưa vào App. Nếu bạn cần import nhiều mục css, bạn có thể tạo cấu trúc sau và tự quản lý thứ tự import.

```
your-slidev/
  ...
  └── styles/
    ├── index.ts
    ├── base.css
    ├── code.css
    └── layouts.css
```

```
// styles/index.ts
import './base.css'
import './code.css'
import './layouts.css'
```

Style sẽ được thực thi bởi Windi CSS và PostCSS, vì vậy bạn có thể sử dụng lồng ghép css và [at-directives](#).
Ví dụ:

```
.slidev-layout {
  @apply px-14 py-10 text-[1.1rem];

  h1, h2, h3, h4, p, div {
    @apply select-none;
  }

  pre, code {
    @apply select-text;
  }

  a {
    color: theme('colors.primary');
  }
}
```

Tìm hiểu thêm về cú pháp.

index.html

Quy ước: `index.html`

`index.html` cung cấp khả năng đưa thẻ meta và/hoặc tập lệnh vào `index.html` chính

Ví dụ: đổi với `index.html` tùy chỉnh sau:

```
<!-- ./index.html -->
<head>
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link href="https://fonts.googleapis.com/css2?family=Fira+Code:wght@400;600&family=Nunito+Sans:wght@200;400;600&display=swap" rel="stylesheet">
</head>

<body>
  <script src=".//your-scripts"></script>
</body>
```

`index.html` cuối cùng sẽ là:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="icon" type="image/png" href="https://cdn.jsdelivr.net/gh/slidevjs/slidev/assets/favicon.png">
  <!-- injected head -->
```

```
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?
family=Fira+Code:wght@400;600&family=Nunito+Sans:wght@200;400;600&display=swap"
rel="stylesheet">
</head>
<body>
  <div id="app"></div>
  <script type="module" src="__ENTRY__"></script>
  <!-- injected body -->
  <script src=".//your-scripts"></script>
</body>
</html>
```

Global Layers

Quy ước: [global-top.vue](#) | [global-bottom.vue](#)

Tìm hiểu thêm về: [Global Layers](#)

Fonts

Có hiệu lực từ v0.20

Mặc dù bạn có thể sử dụng HTML và CSS để tùy chỉnh phông chữ và kiểu cho các trang trình bày của mình theo ý muốn, nhưng Slidev cũng cung cấp một cách để sử dụng chúng dễ dàng.

<<<<< HEAD Trong phông chữ của bạn, hãy định cấu hình như sau ===== In your frontmatter, configure as following

ee7ae42035591cb6565a72f5217129c670a59b0c

```
---  
  fonts:  
    # basically the text  
    sans: 'Robot'  
    # use with `font-serif` css class from windicss  
    serif: 'Robot Slab'  
    # for code blocks, inline code, etc.  
    mono: 'Fira Code'  
---
```

Và đó là tất cả.

Phông chữ sẽ được **nhập tự động** từ [Google Fonts](#). Điều đó có nghĩa là bạn có thể sử dụng trực tiếp bất kỳ phông chữ nào có sẵn trên Google Fonts.

Local Fonts

Theo mặc định, Slidev giả định tất cả các phông chữ được chỉ định thông qua cấu hình `fonts` đến từ Google Fonts. Nếu bạn muốn sử dụng font cục bộ, hãy chỉ định `fonts.local` để chọn không tự động nhập.

```
---  
  fonts:  
    # like font-family in css, you can use ` ,` to separate multiple fonts for  
    fallback  
    sans: 'Helvetica Neue, Robot'  
    # mark 'Helvetica Neue' as local font  
    local: 'Helvetica Neue'  
---
```

In đậm & In nghiêng

Theo mặc định, Slidev nhập ba trọng số `200`, `400`, `600` cho mỗi font. Bạn có thể cấu hình chúng bằng cách:

```
---  
  fonts:  
    sans: 'Robot'  
    # default  
    weights: '200,400,600'  
    # import italic fonts, default `false`  
    italic: false  
---
```

Cấu hình này áp dụng cho tất cả các font web. Để có thể chi tiết hơn về trọng số của từng font, bạn cần nhập chúng theo cách thủ công bằng [HTML](#) và CSS.

Font dự phòng

Đối với hầu hết các tình huống, bạn chỉ cần chỉ định "special font" và Slidev sẽ thêm các font dự phòng cho bạn, ví dụ:

```
---  
  fonts:  
    sans: 'Robot'  
    serif: 'Robot Slab'  
    mono: 'Fira Code'  
---
```

sẽ cho kết quả

```
.font-sans {  
  font-family: "Robot", ui-sans-serif, system-ui, -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", Arial, "Noto Sans", sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol", "Noto Color Emoji";  
}  
.font-serif {  
  font-family: "Robot Slab", ui-serif, Georgia, Cambria, "Times New Roman", Times, serif;
```

```

}
.font-mono {
  font-family: "Fira Code", ui-monospace, SFMono-
Regular, Menlo, Monaco, Consolas, "Liberation Mono", "Courier New", monospace;
}

```

Nếu bạn muốn tắt font dự phòng, hãy định cấu hình như sau

```

---
fonts:
  mono: 'Fira Code, monospace'
  fallback: false
---

```

Providers

- Tùy chọn: `google` | `none`
- Mặc định: `google`

Hiện tại, chỉ Google Fonts được hỗ trợ, chúng tôi dự kiến sẽ bổ sung thêm nhiều nhà cung cấp khác trong tương lai. Chỉ định thành `none` sẽ vô hiệu hóa hoàn toàn tính năng tự động nhập và xử lý tất cả các font cục bộ.

```

---
fonts:
  provide: 'none'
---

```

Global Layers

Có hiệu lực từ v0.17

Các Global layers cho phép bạn có các component tùy chỉnh **liên tục** trên các trang trình bày. Điều này có thể hữu ích để có chân trang, animation trang trình bày chéo, hiệu ứng toàn bộ, v.v...

Slidev cung cấp hai layer cho cách sử dụng này, tạo `global-top.vue` hoặc `global-bottom.vue` trong dự án gốc của bạn và nó sẽ tự động nhận.

Mỗi quan hệ layer:

- Global Top (`global-top.vue`)
- Slides
- Global Bottom (`global-bottom.vue`)

Ví dụ

```
<!-- global-top.vue -->
<template>
  <footer class="absolute bottom-0 left-0 right-0 p-2">Your Name</footer>
</template>
```

Dòng chữ `Your Name` sẽ xuất hiện trên tất cả các slide của bạn.

Để bật nó có điều kiện, bạn có thể áp dụng nó với [Vue Global Context](#).

```
<!-- hide the footer from Page 4 -->
<template>
  <footer
    v-if="$slidev.nav.currentPage !== 4"
    class="absolute bottom-0 left-0 right-0 p-2">
    Your Name
  </footer>
</template>
```

```
<!-- hide the footer from "cover" layout -->
<template>
  <footer
    v-if="$slidev.nav.currentLayout !== 'cover'"
    class="absolute bottom-0 left-0 right-0 p-2">
    Your Name
  </footer>
</template>
```

```
<!-- an example footer for pages -->
<template>
  <footer>
```

```
v-if="$slidev.nav.currentLayout !== 'cover'"  
  class="absolute bottom-0 left-0 right-0 p-2"  
>    {{ $slidev.nav.currentPage }} / {{ $slidev.nav.total }}  
  </footer>  
</template>
```

Đánh dấu

Slidev đi kèm với hai công cụ đánh dấu cú pháp để bạn lựa chọn:

- [Prism](#)
- [Shiki](#)

Prism là một trong những công cụ đánh dấu cú pháp phổ biến nhất. Việc đánh dấu được thực hiện bằng cách thêm các lớp token vào code và nó được đánh dấu bằng cách sử dụng CSS. Bạn có thể duyệt qua [chủ đề chính thức](#), hoặc tự tạo/tùy chỉnh một chủ đề rất dễ dàng bằng cách sử dụng `prism-theme-vars`.

Mặt khác, **Shiki**, à một công cụ đánh dấu cú pháp hỗ trợ ngữ pháp TextMate. Nó tạo ra các token có màu, vì vậy không cần thêm CSS. Vì nó có hỗ trợ ngữ pháp tuyệt vời, màu sắc được tạo ra rất chính xác, giống như những gì bạn sẽ thấy trong VS Code. Shiki cũng đi kèm với [một loạt các chủ đề cài sẵn](#). Nhược điểm của Shiki là nó cũng yêu cầu các chủ đề TextMate (tương thích với chủ đề VS Code) để làm nổi bật, có thể khó tùy chỉnh hơn một chút.

Chủ đề slidev thường hỗ trợ cả Prism và Shiki, nhưng tùy thuộc vào chủ đề bạn đang sử dụng, nó có thể chỉ hỗ trợ một trong số chúng.

Khi bạn có sự lựa chọn, sự cân bằng về cơ bản là:

- **Prism** để tùy chỉnh dễ dàng hơn
- **Shiki** để đánh dấu chính xác hơn

Theo mặc định, Slidev sử dụng Prism. Bạn có thể thay đổi nó bằng cách sửa đổi frontmatter của mình:

```
---  
highlighter: shiki  
---
```

Cấu hình Prism

Để định cấu hình Prism, bạn chỉ có thể nhập css chủ đề hoặc sử dụng `prism-theme-vars` để định cấu hình chủ đề cho cả chế độ sáng và tối. Tham khảo tài liệu của nó để biết thêm chi tiết.

Cấu hình Shiki

Tạo file `./setup/shiki.ts` với nội dung sau

```
/* ./setup/shiki.ts */
import { defineShikiSetup } from '@slidev/types'

export default defineShikiSetup(() => {
  return {
    theme: {
      dark: 'min-dark',
      light: 'min-light',
    }
  }
})
```

```
    },
  },
})
```

Tham khảo [tài liệu của Shiki](#) để biết các tên chủ đề có sẵn.

Hoặc nếu bạn muốn sử dụng chủ đề của riêng mình:

```
/* ./setup/shiki.ts */

import { defineShikiSetup } from '@slidev/types'

export default defineShikiSetup(async({ loadTheme }) => {
  return {
    theme: {
      dark: await loadTheme('path/to/theme.json'),
      light: await loadTheme('path/to/theme.json'),
    },
  }
})
```

Các tùy chỉnh

Slidev hoàn toàn có thể tùy chỉnh, từ kiểu dáng đến cấu hình công cụ. Nó cho phép bạn định cấu hình các công cụ bên dưới ([Vite](#), [Windi CSS](#), [Monaco](#), v.v...)

Cấu hình Frontmatter

Bạn có thể định cấu hình Slidev trong frontmatter của trang trình bày đầu tiên của mình, phần sau hiển thị giá trị mặc định cho mỗi tùy chọn.

```
---
# theme id or package name
theme: 'default'
# title of your slide, will auto infer from the first header if not specified
title: ''
# titleTemplate for the webpage, `'%s'` will be replaced by the page's title
titleTemplate: '%s - Slidev'

# enabled pdf downloading in SPA build, can also be a custom url
download: true
# syntax highlighter, can be 'prism' or 'shiki'
highlighter: 'prism'
# show line numbers in code blocks
lineNumbers: false
# enable monaco editor, default to dev only
monaco: 'dev'

# force color schema for the slides, could be 'auto', 'light', or 'dark'
colorSchema: 'auto'
# router mode for vue-router, could be "history" or "hash"
routerMode: 'history'
# aspect ratio for the slides
aspectRatio: '16/9'
# real width of the canvas, unit in px
canvasWidth: 980

# fonts will be auto imported from Google fonts
# Learn more: https://sli.dev/custom/fonts
fonts:
  sans: 'Roboto'
  serif: 'Roboto Slab'
  mono: 'Fira Code'

# default frontmatter applies to all slides
defaults:
  layout: 'default'
  # ...

# information for your slides, can be a markdown string
info: |
  ## Slidev
  Trang trình bày [Slidev](http://sli.dev/) đầu tiên của tôi!
---
```

<<<<< HEAD Xem [định nghĩa loại](#) để có thêm tùy chọn. ===== Check out the [type definitions](#) for more options.

ee7ae42035591cb6565a72f5217129c670a59b0c

Cấu trúc thư mục

Slidev sử dụng các quy ước cấu trúc thư mục để thu nhỏ bề mặt cấu hình và làm cho các phần mở rộng trong chức năng trở nên linh hoạt và trực quan.

Tham khảo phần [Cấu trúc thư mục](#).

Config Tools

- [Highlighters](#)
- [Configure Vue](#)
- [Configure Vite](#)
- [Configure Windi CSS](#)
- [Configure Monaco](#)
- [Configure KaTeX](#)
- [Configure Mermaid](#)

[Go to TOC](#)

Vue Global Context

Slidev đã đưa vào `global Vue context $slidev` cho các điều kiện nâng cao hoặc điều khiển navigation.

Sử dụng

Bạn có thể truy cập nó ở bất kỳ đâu trong markdown và Vue template của mình, với cú pháp "Mustache".

```
<!-- slides.md -->
# Trang 1
Trang hiện tại là: {{ $slidev.nav.currentPage }}
```

```
<!-- Foo.vue -->
<template>
  <div>Title: {{ $slidev.configs.title }}</div>
  <button @click="$slidev.nav.next">Next Page</button>
</template>
```

Thuộc tính

\$slidev.nav

Một đối tượng phản ứng giữ các thuộc tính và điều khiển của điều hướng trang trình bày. Ví dụ như:

```
$slidev.nav.next() // go next step
$slidev.nav.nextSlide() // go next slide (skip v-clicks)
$slidev.nav.go(10) // go slide #10
```

```
$slidev.nav.currentPage // current slide number
$slidev.nav.currentLayout // current layout id
$slidev.nav.clicks // current clicks count
```

Để biết thêm các thuộc tính khả dụng, hãy tham khảo [nav.ts](#).

\$slidev.configs

Một đối tượng phản ứng giữ [các cấu hình đã được phân tích](#) cú pháp trong frontmatter đầu tiên của `slides.md`. Ví dụ

```
---
title: My First Slidev!
---
```

```
{{ $slidev.configs.title }} // 'My First Slidev!'
```

\$slidev.themeConfigs

Một đối tượng phản ứng giữ các cấu hình chủ đề đã được phân tích cú pháp.

```
---  
title: My First Slidev!  
themeConfig:  
  primary: #213435  
---
```

```
{} $slidev.themeConfigs.primary } } // '#213435'
```

[Go to TOC](#)

Animation

Click Animation

v-click

Để áp dụng "click animation" cho các element, bạn có thể sử dụng `v-click` hoặc các component `<v-click>`

```
# Hello

<!-- Sử dụng Component: cái này sẽ ẩn cho đến khi bạn nhấn "next" -->
<v-click>

Hello World

</v-click>

<!-- Sử dụng Directive: cái này sẽ ẩn cho đến khi bạn nhấn "next" lần thứ hai -->
<div v-click class="text-xl p-2">

Hey!

</div>
```

v-after

Việc sử dụng `v-after` tương tự như `v-click` nhưng nó sẽ hiển thị element khi kích hoạt `v-click` trước đó.

```
<div v-click>Hello</div>
<div v-after>World</div>
```

Khi bạn nhấp vào nút "next", cả hai `Hello` và `World` sẽ cùng hiển thị.

v-click-hide

Tương tự như `v-click` nhưng thay vì làm cho phần tử xuất hiện, nó làm cho element ẩn sau khi nhấn.

```
<div v-click-hide>Hello</div>
```

v-clicks

`v-clicks` chỉ được cung cấp như một component. Đó là một cách viết tắt để áp dụng `v-click` cho tất cả các element con của nó. Nó đặc biệt hữu ích khi làm việc với list.

```
<v-clicks>
  - Item 1
  - Item 2
  - Item 3
```

```
- Item 4
```

```
</v-clicks>
```

Một mục sẽ hiển thị mỗi khi bạn nhấp vào "next".

Custom Số lần Click

Theo mặc định, Slidev đếm số bước cần thiết trước khi chuyển sang trang chiếu tiếp theo. Bạn có thể ghi đè cài đặt này bằng cách chuyển tùy chọn frontmatter `clicks`:

```
---  
# 10 lần nhấp vào trang trình bày này, trước khi chuyển sang trang tiếp theo  
clicks: 10  
---
```

Ordering

Chuyển chỉ số nhấp chuột vào directive của bạn, bạn có thể tùy chỉnh thứ tự của việc revealing

```
<div v-click>1</div>  
<div v-click>2</div>  
<div v-click>3</div>
```

```
<!-- thứ tự bị đảo ngược -->  
<div v-click="3">1</div>  
<div v-click="2">2</div>  
<div v-click="1">3</div>
```

```
---  
clicks: 3  
---  
  
<!-- hiển thị sau 3 lần click -->  
<v-clicks at="3">  
  <div>Hi</div>  
</v-clicks>
```

Chuyển đổi Element

Khi bạn áp dụng `v-click` cho các element của mình, nó sẽ đính kèm tên class `slidev-vclick-target` vào đó. Khi các element bị ẩn, tên class `slidev-vclick-hidden` cũng sẽ được đính kèm. Ví dụ:

```
<div class="slidev-vclick-target slidev-vclick-hidden">Text</div>
```

Sau khi click, nó sẽ trở thành

```
<div class="slidev-vclick-target">Text</div>
```

Theo mặc định, một quá trình chuyển đổi opacity được áp dụng cho các class đó:

```
// mặc định  
.slidev-vclick-target {  
  transition: opacity 100ms ease;
```

```

}
.slidev-vclick-hidden {
  opacity: 0;
  pointer-events: none;
}

```

Bạn có thể ghi đè chúng để tùy chỉnh các hiệu ứng chuyển tiếp trong các stylesheet tùy chỉnh của mình.

Ví dụ: bạn có thể đạt được các chuyển đổi scale bằng cách:

```

// styles.css
.slidev-vclick-target {
  transition: all 500ms ease;
}

.slidev-vclick-hidden {
  transform: scale(0);
}

```

Để chỉ định animation cho một slide hoặc layout nhất định

```

.slidev-page-7,
.slidev-layout.my-custom-layout {
  .slidev-vclick-target {
    transition: all 500ms ease;
  }

  .slidev-vclick-hidden {
    transform: scale(0);
  }
}

```

Tìm hiểu thêm về [customizing styles](#)

Motion

Slidev được tích hợp sẵn [@vueuse/motion](#). Bạn có thể sử dụng `v-motion` cho bất kỳ element nào để thực hiện motion áp dụng cho chúng. Ví dụ

```

<div
  v-motion
  :initial="{ x: -80 }"
  :enter="{ x: 0 }">
  Slidev
</div>

```

Văn bản `Slidev` sẽ di chuyển từ `-80px` về vị trí ban đầu khi khởi tạo.

Lưu ý: Slidev tải trước slide tiếp theo để trình bày, có nghĩa là các animation có thể bắt đầu trước khi bạn điều hướng đến trang. Để làm cho nó hoạt động bình thường, bạn có thể tắt tính năng tải trước cho slide cụ thể

```
---  
preload: false  
---
```

Hoặc kiểm soát vòng đời của element bằng `v-if` để có các điều khiển chi tiết

```
<div  
  v-if="$slidev.nav.currentPage === 7"  
  v-motion  
  :initial="{ x: -80 }"  
  :enter="{ x: 0 }">  
  Slidev  
</div>
```

Tìm hiểu chế độ [Demo](#) | [@vueuse/motion](#) | [v-motion](#) | [Presets](#)

Chuyển trang

Hỗ trợ tích hợp cho các slide KHÔNG ĐƯỢC cung cấp trong phiên bản hiện tại. Chúng tôi đang có kế hoạch bổ sung hỗ trợ cho họ trong phiên bản chính tiếp theo. Trước đó, bạn vẫn có thể sử dụng các style và librarie tùy chỉnh của mình để làm điều đó.

[Go to TOC](#)

Drawing & Annotations

Available since v0.23

We have [drauu](#) built-in for drawing and annotation that could enhance your presentation further.

To start, click the icon in the toolbar and start drawing. It's also available in the [Presenter Mode](#). Drawings and annotations you created will be **synced up** automatically across all instances in real-time.

Use with Stylus Pen

When using a stylus pen on a tablet (for example, iPad with Apple Pencil), Slidev could smartly detect the input type. You can directly draw on your slides with the pen without turning on the drawing mode, while having your fingers or mouse control the navigation.

Persist Drawings

The following frontmatter configuration allows you to persist your drawings as SVGs under `.slidev/drawings` directory and have them inside your exported pdf or hosted site.

```
---  
drawings:  
  persist: true  
---
```

Disable Drawings

Entirely:

```
---  
drawings:  
  enabled: false  
---
```

Only in Development:

```
---  
drawings:  
  enabled: dev  
---
```

Only in Presenter Mode:

```
---  
drawings:  
  presenterOnly: true  
---
```

Drawing Syncing

By default, Slidev syncs up your drawings across all instances. If you are sharing your slides with others, you might want to disable the syncing by:

```
---  
  drawings:  
    syncAll: false  
---
```

With this config, only the drawing from the presenter instance will be able to sync with others.

Hỗ trợ trình soạn thảo

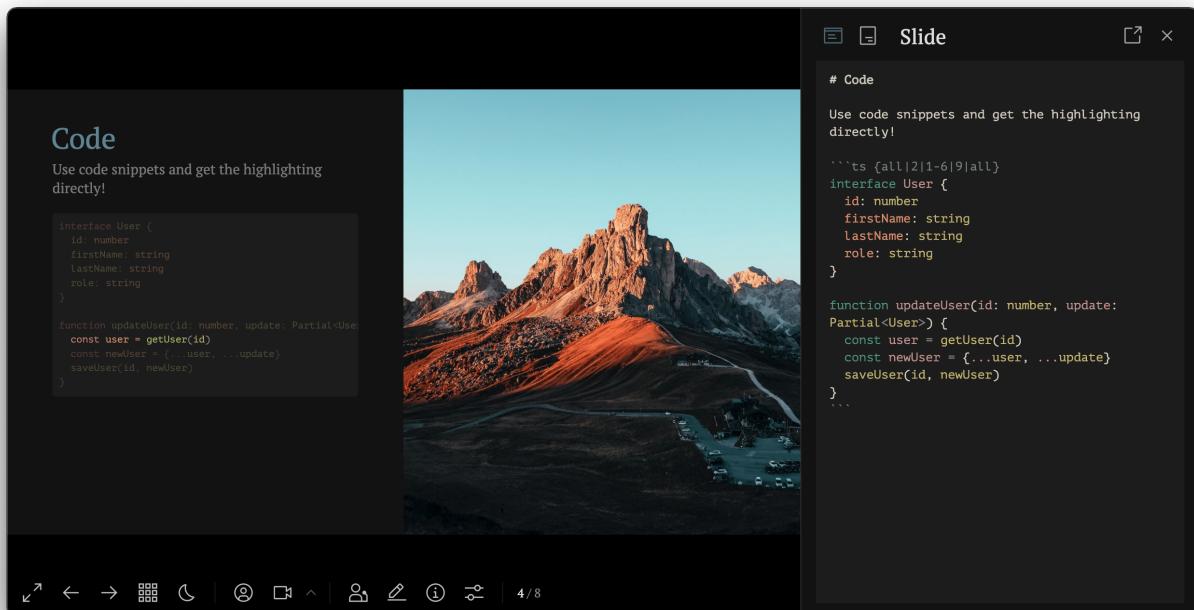
Vì Slidev đang sử dụng Markdown làm mục nhập nguồn, bạn có thể sử dụng **BẤT KỲ** trình soạn thảo nào bạn thích để viết nó.

Nếu bạn muốn một số quản lý hiệu quả cao cho các slide của mình, chúng tôi cung cấp các trình soạn thảo tích hợp cho bạn!

Trình soạn thảo tích hợp

Slidev đi kèm với trình soạn thảo tích hợp [CodeMirror](#) sẽ tải lại ngay lập tức và lưu các thay đổi vào file của bạn.

Click vào nút để mở.



VS Code Extension



Slidev for VS Code

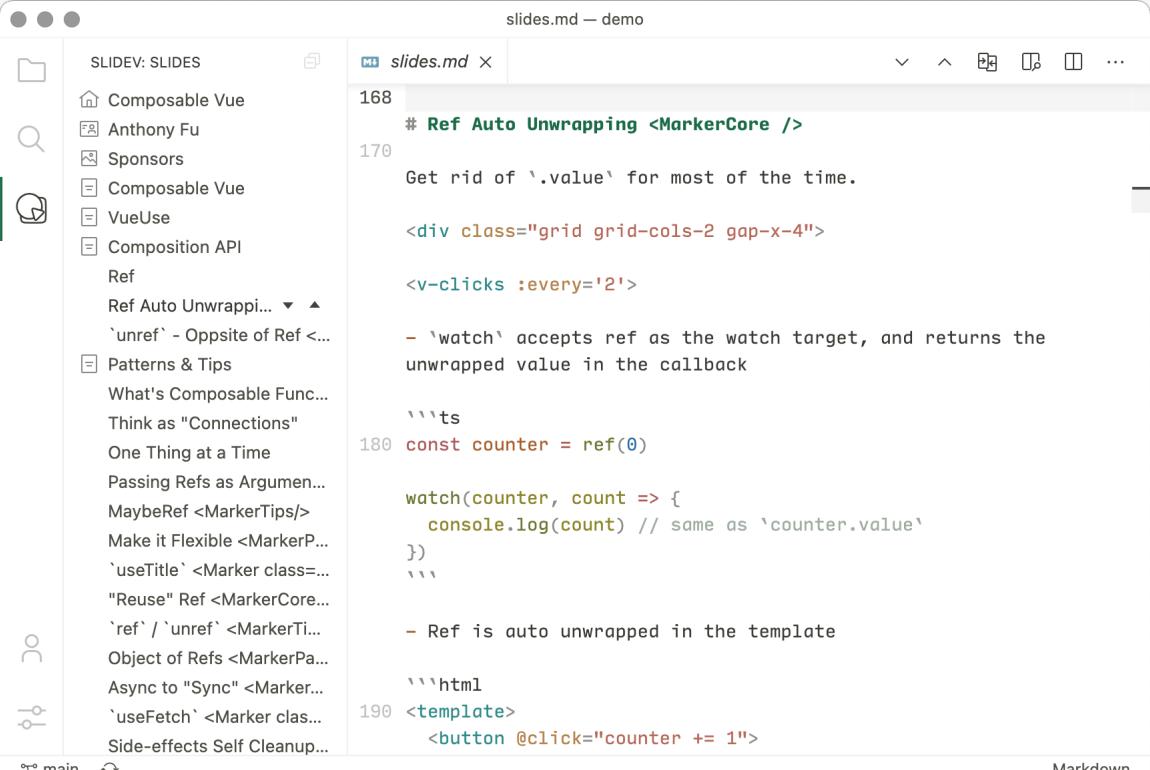
VS Code Marketplace v0.4.1

downloads 26k

The VS Code extension cung cấp một số tính năng để giúp bạn tổ chức các slide của mình tốt hơn và có cái nhìn tổng quan nhanh về chúng.

Tính năng

- Xem các slide trong panel bên cạnh
- Chuyển đến các nút next / prev
- Sắp xếp lại các slide
- Đóng gói các khối slide
- Chuyển đổi Markdown sang HTML



The screenshot shows a code editor window with the following details:

- Title Bar:** slides.md — demo
- File Tree:** SLIDEV: SLIDES / Composable Vue, Anthony Fu, Sponsors, Composable Vue, VueUse, Composition API / Ref, Ref Auto Unwrapping... (highlighted), 'unref' - Oppsite of Ref <..., Patterns & Tips / What's Composable Func..., Think as "Connections", One Thing at a Time, Passing Refs as Argumen..., MaybeRef <MarkerTips/>, Make it Flexible <MarkerP..., 'useTitle' <Marker class=..., "Reuse" Ref <MarkerCore..., 'ref' / 'unref' <MarkerTi..., Object of Refs <MarkerPa..., Async to "Sync" <Marker..., 'useFetch' <Marker clas..., Side-effects Self Cleanup...
- Code Editor:**

```
168 # Ref Auto Unwrapping <MarkerCore />
170 Get rid of `value` for most of the time.

<div class="grid grid-cols-2 gap-x-4">
  <v-clicks :every='2'>
    - `watch` accepts ref as the watch target, and returns the
      unwrapped value in the callback

    ````ts
180 const counter = ref(0)

 watch(counter, count => {
 console.log(count) // same as `counter.value`
 })
  ````

  - Ref is auto unwrapped in the template

    ````html
190 <template>
 <button @click="counter += 1">
```
- Status Bar:** Markdown

---

[Go to TOC](#)

# Xuất slide

## PDF

Xuất sang PDF hoặc PNG dựa vào [Playwright](#) để hiển thị. Do đó, bạn sẽ cần cài đặt `playwright-chromium` để sử dụng tính năng này. Nếu bạn đang xuất trong môi trường CI, [hướng dẫn về CI](#) có thể hữu ích.

Cài đặt `playwright-chromium`

```
$ npm i -D playwright-chromium
```

Bây giờ xuất các slide của bạn sang PDF bằng lệnh sau

```
$ slidev export
```

Sau một vài giây, các slide của bạn sẽ có sẵn tại `./slides-exports.pdf`.

<<<<< HEAD ## PNGs

Khi chuyển vào tùy chọn `--format png`, Slidev sẽ xuất hình ảnh PNG cho từng trang chiếu thay vì PDF.

```
$ slidev export --format png
```

===== ## Export Clicks Steps

ee7ae42035591cb6565a72f5217129c670a59b0c

Available since v0.21

<<<<< HEAD Bạn cũng có thể xây dựng các slide thành một SPA tự lưu trữ: ===== By default, Slidev exports one page per slide with clicks animations disabled. If you want export slides with multiple steps into multiple pages, pass the `--with-clicks` options.

```
ee7ae42035591cb6565a72f5217129c670a59b0c
```

`$ slidev export --with-clicks`

<<<<< HEAD Ứng dụng đã tạo sẽ có sẵn trong `dist/` và sau đó bạn có thể lưu trữ nó trên [GitHub](#), [Netlify](#), [Vercel](#), hoặc bất cứ thứ gì bạn muốn. Giờ đây, bạn có thể chia sẻ các slide của mình với mọi người bằng một liên kết duy nhất.

## Base Path

Để deploy các slide của bạn theo các sub-route, bạn cần phải chuyển tùy chọn `--base`. Ví dụ:

`$ slidev build --base /talks/my-cool-talk/`

Tham khảo [Vite's documentation](#) để biết thêm chi tiết.

## Cung cấp file PDF cho phép tải xuống

Bạn có thể cung cấp một file PDF cho phép tải xuống cho người xem SPA của bạn. Bạn có thể kích hoạt nó bằng cách cấu hình sau:

```

download: true

```

Bây giờ, Slidev sẽ tạo một file pdf cùng với bản dựng và một nút tải xuống sẽ xuất hiện trong SPA.

Bạn cũng có thể cung cấp một url tùy chỉnh cho PDF. Trong trường hợp đó, quá trình render sẽ bị bỏ qua.

```

download: 'https://myside.com/my-talk.pdf'

```

## Ví dụ

Dưới đây là một số ví dụ về SPA đã xuất:

- [Starter Template](#)

## . Composable Vue bởi Anthony Fu PNGs

When passing in the `--format png` option, Slidev will export PNG images for each slide instead of a PDF.

```
$ slidev export --format png
```

## Single-Page Application (SPA)

See [Static Hosting](#).

ee7ae42035591cb6565a72f5217129c670a59b0c

---

[Go to TOC](#)

# FAQ

## Grids

Vì Slidev dựa trên Web nên bạn có thể áp dụng bất kỳ bố cục lưới nào tùy thích. [CSS Grids](#), [flexboxes](#), hoặc thậm chí [Masonry](#), bạn có toàn quyền kiểm soát.

Vì chúng tôi đã tích hợp sẵn [Windi CSS](#), đây là một cách đơn giản để bạn tham khảo:

```
<div class="grid grid-cols-2 gap-4">
<div>
The first column

</div>
<div>
The second column

</div>
</div>
```

<<<<< HEAD Hơn nữa, bạn có thể tùy chỉnh kích thước của từng cột như: ===== Go further, you can customize the size of each column like:

ee7ae42035591cb6565a72f5217129c670a59b0c

```
<div class="grid grid-cols-[200px,1fr,10%] gap-4">
<div>
```

The first column (200px)

```
</div>
<div>

The second column (auto fit)

</div>
<div>

The third column (10% width to parent container)

</div>
</div>
```

Tìm hiểu thêm về [Windi CSS Grids](#).

## Positioning

Slide được xác định theo kích thước cố định (mặc định `980x552px`) và chia tỷ lệ để vừa với màn hình người dùng. Bạn có thể an toàn sử dụng position absolute trong slide của mình vì chúng sẽ scale cùng với màn hình.

Ví dụ:

```
<div class="absolute left-30px bottom-30px">
This is a left-bottom aligned footer
</div>
```

Để thay đổi kích thước thực của canvas, bạn có thể chuyển các tùy chọn `canvasWidth` trong frontmatter đầu tiên của mình:

```

canvasWidth: 800

```

## Kích thước Font

Nếu bạn cảm thấy kích thước font trong các slide của mình quá nhỏ, bạn có thể điều chỉnh nó theo một số cách:

### Ghi đè Style cục bộ

Bạn có thể ghi đè style cho từng trang slide bằng thẻ `<style>`.

```
Page 1

<style>
h1 {
 font-size: 10em;
}
</style>

```

# Page 2

This will not be affected.

Tìm hiểu thêm về [Embedded Styles](#)

## Ghi đè Style toàn cục

Bạn có thể cung cấp các style toàn cục tùy chỉnh bằng cách tạo `./style.css`, ví dụ

```
/* style.css */

h1 {
 font-size: 10em !important;
}
```

Tìm hiểu thêm về: [Global Style](#)

## Scale Canvas

Thay đổi kích thước thực tế của canvas sẽ chia tỷ lệ tất cả nội dung của bạn (văn bản, hình ảnh, components, v.v.) và các slide

```

mặc định: 980
vì canvas nhỏ hơn, kích thước hình ảnh sẽ trở nên lớn hơn
canvasWidth: 800

```

## Sử dụng Transform

Chúng tôi cung cấp một component tích hợp sẵn `<Transform />`, là một lớp wrapper của thuộc tính transform CSS.

```
<Transform :scale="1.4">
 - Item 1
 - Item 2
</Transform>
```

---

[Go to TOC](#)

# Static Hosting

## Build Single Page Applications (SPA)

You can also build the slides into a self-hostable SPA:

```
$ slidev build
```

The generated application will be available under `dist/` and then you can host it on [GitHub Pages](#), [Netlify](#), [Vercel](#), or whatever you want. Now you can share your slides with the rest of the world with a single link.

### Base Path

To deploy your slides under sub-routes, you will need to pass the `--base` option. For example:

```
$ slidev build --base /talks/my-cool-talk/
```

Refer to [Vite's documentation](#) for more details.

### Provide Downloadable PDF

You can provide a downloadable PDF to the viewers of your SPA with the following config:

```

 download: true

```

Slidev will generate a pdf file along with the build, and a download button will be displayed in the SPA.

You can also provide a custom url to the PDF. In that case, the rendering process will be skipped.

```

 download: 'https://myside.com/my-talk.pdf'

```

## Examples

Here are a few examples of the exported SPA:

- [Starter Template](#)
- [Composable Vue by Anthony Fu](#)

For more, check out [Showcases](#).

## Hosting

We recommend to use `npm init slidev@lastest` to scaffolding your project, which contains the necessary configuration files for hosting services out-of-box.

## Netlify

- [Netlify](#)

Create `netlify.toml` in your project root with the following content.

```
[build.environment]
 NODE_VERSION = "14"

[build]
 publish = "dist"
 command = "npm run build"

[[redirects]]
 from = "/"
 to = "/index.html"
 status = 200
```

Then go to your Netlify dashboard, create new site with the repository.

## Vercel

- [Vercel](#)

Create `vercel.json` in your project root with the following content.

```
{
 "rewrites": [
 { "source": "/(.*)", "destination": "/index.html" }
]
}
```

Then go to your Vercel dashboard, create new site with the repository.

## GitHub Pages

- [GitHub Pages](#)

Create `.github/workflows/deploy.yml` with following content to deploy your slides to GitHub Pages via GitHub Actions.

```
name: Deploy pages
on: push
jobs:
 deploy:
 runs-on: ubuntu-latest
 steps:
 - uses: actions/checkout@v2
 - uses: actions/setup-node@v2
 with:
 node-version: '14'
 - name: Install dependencies
 run: npm install
 - name: Build
 run: npm run build
 - name: Deploy pages
```

```
uses: crazy-max/ghaction-github-pages@v2
with:
 build_dir: dist
env:
 GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
```

# Bắt đầu

## Tổng quan

Slidev (slide + dev, /slʌɪdɪv/) là một trình tạo và trình chiếu slide dựa trên web. Nó được thiết kế để các nhà lập trình tập trung vào việc viết nội dung trong Markdown đồng thời có sự hỗ trợ của các component HTML và Vue để cung cấp layout và thiết kế hoàn hảo trên từng pixel trong bản trình bày của bạn.

Nó sử dụng file markdown giàu tính năng để tạo các slide đẹp với trải nghiệm tải lại tức thì, cùng với nhiều tích hợp tích hợp như code trực tiếp, xuất PDF, ghi bản trình bày, v.v... Vì nó được cung cấp bởi web, bạn có thể làm bất cứ điều gì với Slidev - khả năng là vô tận.

Bạn có thể tìm hiểu thêm về cơ sở lý luận đằng sau dự án trong phần [Tại sao chọn Slidev](#).

## Tính năng

<<<<< HEAD

- **Dựa trên Markdown** - sử dụng trình soạn thảo và quy trình làm việc yêu thích của bạn
- **Thân thiện với nhà lập trình** - đánh dấu cú pháp tích hợp, code trực tiếp, v.v...
- **Themable** - chủ đề có thể được chia sẻ và sử dụng với package npm
- **Stylish** - Windi CSS tiện ích theo yêu cầu, stylesheet được nhúng dễ sử dụng
- **Tương tác** - nhúng các component Vue một cách liền mạch
- **Chế độ trình bày** - sử dụng cửa sổ khác hoặc thậm chí điện thoại của bạn để điều khiển các slide của bạn
- **LaTeX** - tích hợp LaTeX hỗ trợ phương trình toán học
- **Diagrams** - tạo sơ đồ với mô tả bằng văn bản
- **Icons** - truy cập trực tiếp vào các icon từ bất kỳ bộ icon nào
- **Trình soạn thảo** - trình soạn thảo tích hợp, hoặc [extension cho VS Code](#)
- **Recording** - tích hợp recording and xem camera
- **Portable** - xuất sang PDF, PNGs, hoặc thậm chí là một SPA có thể lưu trữ
- **Nhanh** - tải lại tức thì được cung cấp bởi [Vite](#)
  
- **Hackable - sử dụng plugin Vite, component Vue hoặc bất kỳ package npm nào**

- **Markdown-based** - use your favorite editors and workflow
- **Developer Friendly** - built-in syntax highlighting, live coding, etc.
- **Themable** - theme can be shared and used with npm packages
- **Stylish** - Windi CSS on-demand utilities, easy-to-use embedded stylesheets
- **Interactive** - embedding Vue components seamlessly

- 🎤 **Presenter Mode** - use another window, or even your phone to control your slides
- 🖌 **Drawing** - draw and annotate on your slides
- 📄 **LaTeX** - built-in LaTeX math equations support
- 📀 **Diagrams** - creates diagrams with textual descriptions
- 🌟 **Icons** - Access to icons from any iconset directly
- 💻 **Editors** - integrated editor, or [extension for VS Code](#)
- 📹 **Recording** - built-in recording and camera view
- 🛠 **Portable** - export into PDF, PNGs, or even a hostable SPA
- ⚡ **Fast** - instant reloading powered by [Vite](#)
- 🧩 **Hackable** - using Vite plugins, Vue components, or any npm packages

ee7ae42035591cb6565a72f5217129c670a59b0c

## Tech Stack

Slidev có thể thực hiện được bằng cách kết hợp các công cụ và công nghệ này.

<<<<< HEAD

- [Vite](#) - một công cụ frontend cực kỳ nhanh chóng
- [Vue 3](#) cung cấp [Markdown](#) - tập trung vào nội dung khi có sự hỗ trợ của các component HTML và Vue bất cứ khi nào cần thiết
- [Windi CSS](#) - khung CSS ưu tiên tiện ích theo yêu cầu, tạo style cho các slide của bạn một cách dễ dàng
- [Prism](#), [Shiki](#), [Monaco Editor](#) - hỗ trợ các đoạn code first-class với khả năng code trực tiếp
- [RecordRTC](#) - tích hợp recording và chế độ xem camera
- [VueUse](#) - `@vueuse/core`, `@vueuse/head`, `@vueuse/motion`, v.v...
- [Iconify](#) - bộ sưu tập icon.
- [KaTeX](#) - render toán học LaTeX.

## • **Mermaid - Sơ đồ văn bản.**

- [Vite](#) - An extremely fast frontend tooling
- [Vue 3](#) powered [Markdown](#) - Focus on the content while having the power of HTML and Vue components whenever needed
- [Windi CSS](#) - On-demand utility-first CSS framework, style your slides at ease
- [Prism](#), [Shiki](#), [Monaco Editor](#) - First-class code snippets support with live coding capability
- [RecordRTC](#) - Built-in recording and camera view
- [VueUse](#) family - `@vueuse/core`, `@vueuse/head`, `@vueuse/motion`, etc.
- [Iconify](#) - Icons collection.
- [Drauu](#) - Drawing and annotations support
- [KaTeX](#) - LaTeX math rendering.
- [Mermaid](#) - Textual Diagrams.

ee7ae42035591cb6565a72f5217129c670a59b0c

## Dàn dựng bài thuyết trình đầu tiên của bạn

<<<<< HEAD Với NPM: =====

**Try it Online**

[sli.dev/new](https://sli.dev/new)



Open in StackBlitz

## Create Locally

With NPM:

```
ee7ae42035591cb6565a72f5217129c670a59b0c
```

```
$ npm init slidev
```

Với Yarn:

```
$ yarn create slidev
```

Làm theo lời nhắc và bắt đầu tạo slide của bạn ngay bây giờ! Để biết thêm chi tiết về cú pháp markdown, hãy đọc qua [hướng dẫn cú pháp](#).

## Giao diện Command Line

Trong một dự án có cài đặt Slidev, bạn có thể sử dụng hệ nhị phân `slidev` trong các tập lệnh npm của mình.

```
{
 "scripts": {
 "dev": "slidev", // start dev server
 "build": "slidev build", // build for production SPA
 "export": "slidev export" // export slides to pdf
 }
}
```

Nếu không, bạn có thể sử dụng nó với `npx`

```
$ npx slidev
```

Chạy `slidev --help` để có thêm tùy chọn.

## Cú pháp Markdown

Slidev đọc file `slides.md` trong dự án gốc của bạn và chuyển đổi chúng thành các slide. Bất cứ khi nào bạn thực hiện thay đổi với nó, nội dung của các slide sẽ được cập nhật ngay lập tức. Ví dụ:

```
Slidev

Hello World

Page 2

Directly use code blocks for highlighting

//```ts
console.log('Hello, World!')
//```

Page 3
```

Đọc thêm về cú pháp Slidev Markdown trong [hướng dẫn cú pháp](#).

# Cài đặt

## Mẫu dành cho người mới bắt đầu

Slidev yêu cầu **Node.js >=14.0**

Cách tốt nhất để bắt đầu là sử dụng mẫu chính thức của chúng tôi.

Với NPM:

```
$ npm init slidev@latest
```

Với Yarn:

```
$ yarn create slidev
```

Làm theo lời nhắc và nó sẽ tự động mở trình chiếu tại <http://localhost:3030/> cho bạn.

Nó cũng bao gồm thiết lập cơ bản và một bản demo ngắn với hướng dẫn về cách bắt đầu với Slidev.

## Cài đặt thủ công

Nếu bạn vẫn muốn cài đặt Slidev theo cách thủ công hoặc muốn tích hợp nó vào các dự án hiện có của mình, bạn có thể thực hiện:

```
$ npm install @slidev/cli @slidev/theme-default
```

```
$ touch slides.md
```

```
$ npx slidev
```

Xin lưu ý rằng nếu bạn đang sử dụng `pnpm`, bạn sẽ cần bật tùy chọn `shamefully-hoist` để Slidev hoạt động bình thường:

```
$ echo 'shamefully-flatten=true' >> .npmrc
```

## Cài đặt trên toàn cục

Có hiệu lực từ v0.14

Bạn có thể cài đặt Slidev trên toàn cục bằng lệnh sau

```
$ npm i -g @slidev/cli
```

Và sau đó sử dụng `slidev` ở bất cứ nơi nào mà không cần tạo dự án.

```
$ slidev
```

Lệnh này cũng sẽ sử dụng cục bộ `@slidev/cli` nếu nó được tìm thấy trong `node_modules`.

## Cài đặt trên Docker

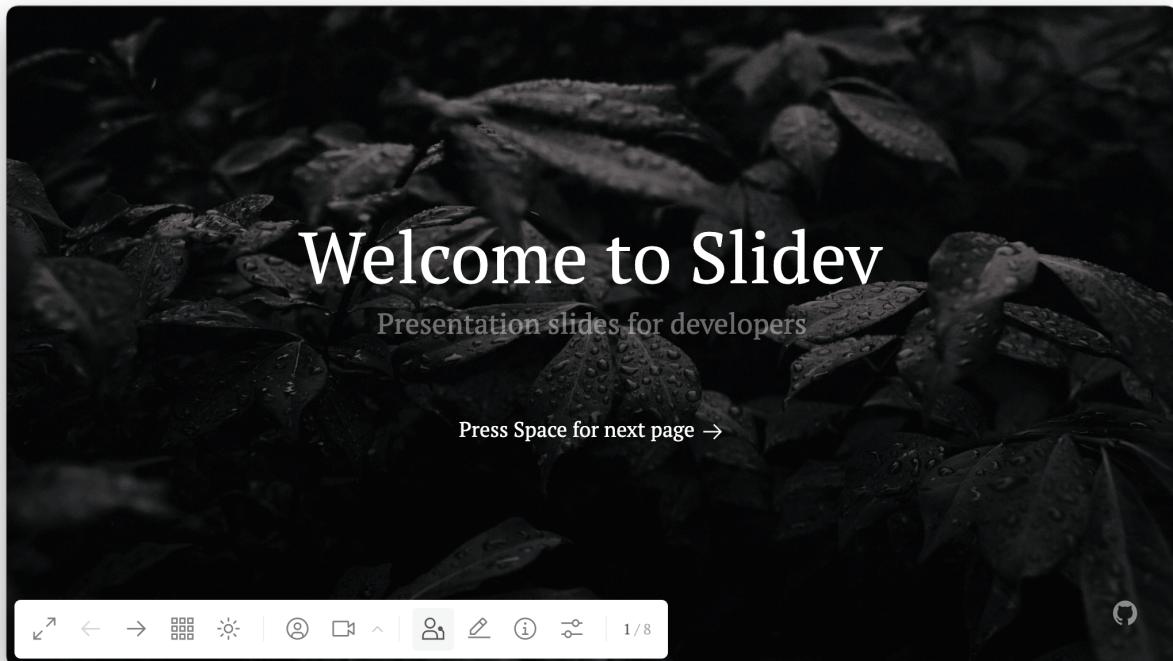
Nếu bạn cần chạy bản trình bày với container một cách nhanh chóng, bạn có thể sử dụng hình ảnh `docker` được duy trì bởi [stig124](#), hoặc xây dựng của riêng bạn.

Tham khảo [slidevjs/container repo](#) để biết thêm chi tiết.

# Điều hướng

## Thanh điều hướng

Di chuyển chuột đến góc dưới cùng bên trái của trang Slidev, thanh điều hướng sẽ hiện ra.



Phím tắt	Nút	Mô tả
f		bật chế độ toàn màn hình
right / space		animation hoặc slide tiếp theo
left		animation hoặc slide tiếp theo
up	-	slide trước
down	-	slide tiếp
o		chuyển đổi <a href="#">tổng quan slide</a>
d		chuyển đổi chế độ tối
-		chuyển đổi <a href="#">xem camera</a>
-		<a href="#">recording</a>
-		enter <a href="#">chế độ trình bày</a>
-		chuyển đổi <a href="#">trình soạn thảo tích hợp</a>

Phím tắt	Nút	Mô tả
-		tải xuống các slide (chỉ xuất hiện trong SPA build)
-		hiển thị thông tin về các slide
-		hiển thị menu cài đặt
g	-	hiển thị goto...

## Tổng quan Slide

<<<<< HEAD Bằng cách nhấn o hoặc nhấn vào nút trong thanh điều hướng, bạn có thể có tổng quan về các slide của mình, vì vậy bạn có thể chuyển đổi giữa chúng một cách dễ dàng. ===== By pressing o or clicking the button in the navigation bar, you can have the overview of your slides so you can jump between them easily.

ee7ae42035591cb6565a72f5217129c670a59b0c

1

Welcome to Slidify  
Presentation slide for developers  
Press space for next page ↗

What is Slidify?  
Slidify is a presentation designed for developers, consisting of the following features:

- Fast-based - built on the syntax with Markdown, and then render them later
- Themes - themes can be shared and used with open packages
- Developer Friendly - code highlighting, live coding with auto-completion
- Interactive - embedding Vue components to enhance your presentations
- Portable - export into PDF, PNG, or even a hostable SPA
- Hackable - anything possible on a webpage

Read more about Why Slidify?

2

**Navigation**  
Hover over the bottom-left corner to see the navigation's controls panel.

Keyboard Shortcuts	Action
Space / tab / right	next animation or slide
Shift	next
Up	previous animation or slide
Down	previous slide
Page	next slide

3

Navigation

Hover over the bottom-left corner to see the navigation's controls panel.

4

**Code**  
Use code snippets and get the highlighting directly!

```
interface User {
 id: string;
 name: string;
 email: string;
 address: string;
}

function calculateTotal(user: User, address: string): number {
 const userAddress = user.address;
 const total = userAddress.length * user.id.length;
 return total;
}
```

5

**Components**  
You can use Vue components directly inside your slides.

We have provided a few built-in components like `<code>`, `<math>` and `<img>` which you can directly use. And add your custom components are also super easy.

6

**LaTeX**  
LaTeX is supported out-of-box powered by KaTeX.

Rock

$$\text{Leftrightarrow } 3x - 1 + (1+x)^2 \\ \nabla \times \vec{B} - \frac{1}{c} \frac{\partial \vec{E}}{\partial t} = \frac{4\pi}{c} \nabla \cdot \vec{E} = 4\pi\rho \\ \nabla \times \vec{E} + \frac{1}{c} \frac{\partial \vec{B}}{\partial t} = 0 \\ \nabla \cdot \vec{B} = 0$$

Learn more

7

**Themes**  
Themes offer powerful styling support. Themes are able to provide styles, layouts, components, or even configurations for tools. Switching between themes by just one edit in your frontmatter:

8

**Learn More**  
Documentation | GitHub Repo

---

[Go to TOC](#)

# Chế độ Trình bày

Nhấn vào nút trong bảng điều hướng hoặc truy cập <http://localhost:3030/presenter> theo cách thủ công, để vào chế độ người thuyết trình. Bất cứ khi nào bạn vào chế độ người thuyết trình, các phiên bản trang khác sẽ tự động đồng bộ với người trình bày.

The screenshot shows a presentation slide titled "Welcome to Sliderv" with the subtitle "Presentation slides for developers". It includes a "Press Space for next page →" instruction and a "567" address sign on a building. The right side of the slide contains a sidebar with the title "What is Sliderv?" and a list of features:

- **Drive based:** Focus on the content and backbone, and their style sheets last
- **Developer friendly:** code highlighting, live coding with autocompletion
- **Mobile first:** responsive design, and a mobile first experience
- **Recording:** built-in recording and camera view
- **Export:** export file PDF, PPTX, or even a executable IPA
- **Shareable:** get the link or a embed

Below the sidebar, a note states: "The last comment block of each slide will be treated as slide notes. It will be visible and editable in Presenter Mode along with the slide. [Read more in the docs](#)". At the bottom, there are navigation icons and a page number "1 / 7".

---

[Go to TOC](#)

# Recording

Slidev được tích hợp chế độ record và chế độ xem camera. Bạn có thể sử dụng chúng để ghi lại bản trình bày của mình một cách dễ dàng ở một nơi.

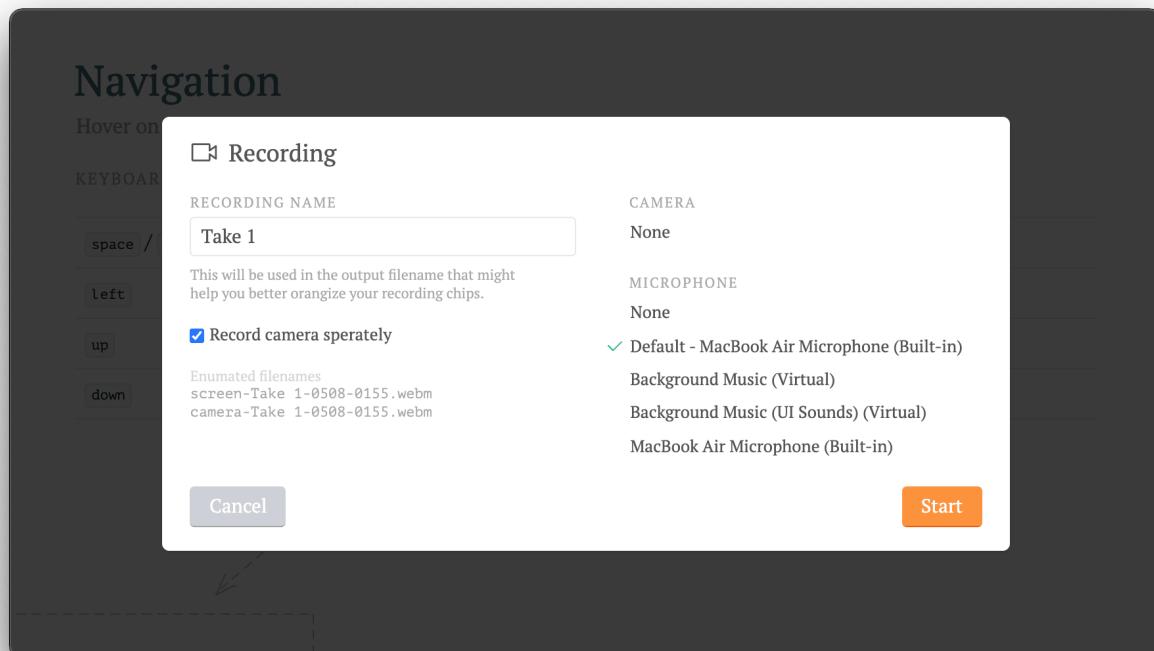
## Xem Camera

Nhấn vào nút trong bảng điều hướng để hiển thị chế độ xem camera của bạn trong bản trình bày. Bạn có thể kéo để di chuyển nó và sử dụng trình xử lý ở góc dưới bên phải để thay đổi kích thước. Kích thước và vị trí sẽ tồn tại trong `localStorage` và do đó sẽ nhất quán qua nhiều lần làm mới, vì vậy không cần phải lo lắng về điều đó.

## Recording

Nhấn vào nút trong bảng điều hướng sẽ xuất hiện hộp thoại cho bạn. Tại đây, bạn có thể chọn ghi lại máy ảnh được nhúng vào các slide của bạn hoặc tách chúng thành hai file video.

Tính năng này được cung cấp bởi [RecordRTC](#) and uses the [WebRTC API](#).




---

[Go to TOC](#)

# Cú pháp Markdown

Slide được viết trong **một file markdown duy nhất** (theo mặc định là `./slides.md`).

Bạn có thể sử dụng [các tính năng Markdown](#) như bình thường, với sự hỗ trợ bổ sung của các component HTML và Vue. Tạo style bằng [Windi CSS](#) cũng được hỗ trợ. Sử dụng dấu `---` được đệm bằng một dòng mới để tách các slide của bạn.

```
Slidev

Hello, World!

Page 2

Directly use code blocks for highlighting

//```ts
console.log('Hello, World!')
//```

Page 3

You can directly use Windi CSS and Vue components to style and enrich your slides.

<div class="p-3">
 <Tweet id="20" />
</div>
```

## Front Matter & Layout

<<<<< HEAD Bạn có thể chỉ định layout và siêu dữ liệu khác cho mỗi slide bằng cách chuyển đổi các dấu phân cách thành [khối front matter](#). Mỗi front matter bắt đầu bằng dấu ba gạch ngang và kết thúc bằng dấu gạch ngang khác. Văn bản giữa chúng là các đối tượng dữ liệu ở định dạng [YAML](#). Ví dụ: ===== Specify layouts and other metadata for each slide by converting the separators into [front matter blocks](#). Each frontmatter starts with a triple-dash and ends with another. Texts between them are data objects in [YAML](#) format. For example:

ee7ae42035591cb6565a72f5217129c670a59b0c

```

layout: cover

Slidev
This is the cover page.

layout: center
background: './images/background-1.png'
class: 'text-white'

Page 2
This is a page with the layout `center` and a background image.

Page 3
This is a default page without any additional metadata.
```

Tham khảo [customization](#) để biết thêm chi tiết.

## Code Blocks

Một lý do lớn mà tôi xây dựng Slidev là làm cho code của tôi trông vừa phải trong các slide. Vì vậy, đúng như bạn mong đợi, bạn có thể sử dụng khối code Markdown để làm nổi bật code của mình.

```
//``ts
console.log('Hello, World!')
//``
```

<<<<< HEAD ### Đánh dấu dòng ===== We support [Prism](#) and [Shiki](#) as syntax highlighters.  
Refer to [the highlighters section](#) for more details.

## Line Highlighting

```
ee7ae42035591cb6565a72f5217129c670a59b0c
```

Để đánh dấu các dòng cụ thể, chỉ cần thêm số dòng trong dấu ngoặc nhọn `{}`. Số dòng bắt đầu đếm từ 1.

```
//```ts {2,3}
function add(
 a: Ref<number> | number,
 b: Ref<number> | number
) {
 return computed(() => unref(a) + unref(b))
}
//```
```

Để thay đổi vùng đánh dấu trong nhiều bước, bạn có thể sử dụng dấu `|` để tách chúng. Ví dụ

```
//```ts {2-3|5|all}
function add(
 a: Ref<number> | number,
 b: Ref<number> | number
) {
 return computed(() => unref(a) + unref(b))
}
//```
```

Đầu tiên, điều này sẽ làm nổi bật `a: Ref<number> | number` và `b: Ref<number> | number`, và sau đó là `return computed(() => unref(a) + unref(b))` sau một cú nhấp chuột, và cuối cùng là toàn bộ khối.  
Tìm hiểu thêm trong [hướng dẫn về animation click](#).

## Trình soạn thảo Monaco

Bất cứ khi nào bạn muốn thực hiện một số sửa đổi trong bản trình bày, chỉ cần thêm `{monaco}` sau ngôn ngữ id - nó biến khối thành một trình soạn thảo Monaco đầy đủ tính năng!

```
//``ts {monaco}
console.log('HelloWorld')
//``
```

Tìm hiểu thêm về [configuring Monaco](#).

## Embedded Styles

Bạn có thể sử dụng thẻ `<style>` trong Markdown của mình trực tiếp để ghi đè các kiểu cho **slide hiện tại**.

```
This is Red

<style>
h1 {
 color: red
}
</style>

Next slide is not affected
```

Thẻ `<style>` trong Markdown luôn **scoped**. Để ghi đè kiểu toàn cục, hãy xem [phần tùy chỉnh](#)

Được cung cấp bởi [Windi CSS](#), bạn có thể sử dụng trực tiếp css và [directives](#) lồng nhau (ví dụ: `@apply`)

```
Slides

> Hello `world`

<style>
blockquote {
 code {
 @apply text-teal-500 dark:text-teal-400;
 }
}
</style>
```

<<<<< HEAD ## Lưu ý ====== ## Static Assets

Just like you would do in markdown, you can use images pointing to a remote or local url.

For remote assets, the built-in `vite-plugin-remote-assets` will cache them into the disk at the first run so you can have instant loading even for large images later on.

! [Remote Image](<https://sli.dev/favicon.png>)

For local assets, put them into the `public` folder and reference them with **leading slash**.

! [Local Image](pic.png)

For you want to apply custom sizes or styles, you can convert them to the `<img>` tag



## Notes

ee7ae42035591cb6565a72f5217129c670a59b0c

Bạn cũng có thể ghi chú cho mỗi slide. Chúng sẽ hiển thị ở [Chế độ trình bày](#) để bạn tham khảo trong khi thuyết trình.

Trong Markdown, khối comment cuối cùng trong mỗi slide sẽ được coi là ghi chú.

```

layout: cover

Page 1

This is the cover page.

<!-- This is a note -->

Page 2

<!-- This is NOT a note because it precedes the content of the slide -->

The second page

<!--
This is another note
-->
```

## Icons

Slidev cho phép bạn có quyền truy cập vào hầu hết tất cả các bộ icon mã nguồn mở phổ biến [trực tiếp](#) trong markdown của bạn. Được cung cấp bởi [vite-plugin-icons](#) và [Iconify](#).

Việc đặt tên theo chuyển đổi của [Iconify](#) `{collection-name}-{icon-name}`. Ví dụ:

- `<mdi-account-circle />` - từ [Material Design Icons](#)
- `<carbon-badge />` - từ [Carbon](#)
- `<uim-rocket />` - từ [Unicons Monochrome](#)
- `<twemoji-cat-with-tears-of-joy />` - từ [Twemoji](#)
- `<logos-vue />` - từ [SVG Logos](#)
- Và nhiều hơn nữa...

<<<<< HEAD Bạn có thể duyệt và tìm kiếm tất cả các icon có sẵn với [Icônes](#). ===== Browse and search for all the icons available with [Icônes](#).

ee7ae42035591cb6565a72f5217129c670a59b0c

## Styling Icons

Bạn có thể tạo kiểu cho các icon giống như các element HTML khác. Ví dụ:

```
<uim-rocket />
<uim-rocket class="text-3xl text-red-400 mx-2" />
<uim-rocket class="text-3xl text-orange-400 animate-ping" />
```

## Slots

Có hiệu lực từ v0.18

Một số layout có thể cung cấp nhiều điểm đóng góp bằng cách sử dụng [slot được đặt tên của Vue](#).

Ví dụ, trong `two-cols layout`, bạn có thể có hai cột bên trái (`default slot`) và bên phải (`right slot`) cạnh nhau.

```

layout: two-cols

<template v-slot:default>
Left

This shows on the left

</template>
<template v-slot:right>
Right

This shows on the right

</template>
```

## Left

This shows on the left

## Right

This shows on the right

Chúng tôi cũng cung cấp cú pháp viết tắt `::name::` cho tên vị trí. Ví dụ sau hoạt động giống hệt như ví dụ trước.

```

layout: two-cols

Left

This shows on the left

::right::
Right

This shows on the right
```

<<<<< HEAD Bạn cũng có thể chỉ định rõ ràng vị trí mặc định và cung cấp theo thứ tự tùy chỉnh  
===== You can also explicitly specify the default slot and provide in the custom order

```
ee7ae42035591cb6565a72f5217129c670a59b0c
```

```

layout: two-cols

::right::
Right
This shows on the right

::default::
Left
This shows on the left
```

## Cấu hình

Tất cả các cấu hình cần thiết có thể được xác định trong tệp Markdown. Ví dụ:

```

theme: serif
layout: cover
background: 'https://source.unsplash.com/1600x900/?nature,water'

Slidev
This is the cover page.
```

Tìm hiểu thêm về [cấu hình frontmatter](#)

## LaTeX

Slidev đi kèm với hỗ trợ LaTeX bên ngoài, được cung cấp bởi [KaTeX](#).

### Inline

Bao quanh LaTeX của bạn với một ký tự `$` duy nhất ở mỗi bên để hiển thị trong một dòng.

```
 $ \sqrt{3x-1} + (1+x)^2 $
```

## Block

Sử dụng hai ( `$$` ) để hiển thị khối. Chế độ này sử dụng các ký hiệu lớn hơn và căn giữa kết quả.

```
 $$
 \begin{array}{c}
 \nabla \times \vec{\mathbf{B}} - \frac{1}{c}, \frac{\partial \vec{\mathbf{E}}}{\partial t} \\
 = \frac{4\pi}{c} \vec{c} \cdot \vec{\mathbf{j}} \quad \nabla \cdot \vec{\mathbf{E}} &= 4\pi \rho \\
 \\
 \nabla \times \vec{\mathbf{E}} + \frac{1}{c}, \frac{\partial \vec{\mathbf{B}}}{\partial t} &= \vec{\mathbf{0}} \\
 \nabla \cdot \vec{\mathbf{B}} &= 0 \\
 \end{array}
 $$
```

Tìm hiểu thêm: [Demo](#) | [KaTeX](#) | [markdown-it-katex](#)

## Diagrams

Bạn cũng có thể tạo sơ đồ / đồ thị từ mô tả bằng văn bản trong Markdown của mình, được cung cấp bởi [Mermaid](#).

<<<<< HEAD Code block được đánh dấu là `mermaid` sẽ được chuyển đổi thành diagram, ví dụ:  
===== Code blocks marked as `mermaid` will be converted to diagrams, for example:

```
 ee7ae42035591cb6565a72f5217129c670a59b0c
```

```
//```mermaid
sequenceDiagram
 Alice->John: Hello John, how are you?
 Note over Alice,John: A typical interaction
//```

```

Bạn có thể chuyển thêm một đối tượng tùy chọn cho nó để chỉ định tỷ lệ và chia tỷ lệ. Cú pháp của đối tượng là một đối tượng JavaScript theo nghĩa đen, bạn sẽ cần thêm dấu nháy ( ' ) cho các chuỗi và sử dụng dấu phẩy ( , ) giữa các khóa.

```
//```mermaid {theme: 'neutral', scale: 0.8}
graph TD
B[Text] --> C{Decision}
C -->|One| D[Result 1]
C -->|Two| E[Result 2]
//```

```

Tìm hiểu thêm: [Demo](#) | [Mermaid](#)

## Multiple Entries

Có hiệu lực từ v0.15

Bạn có thể chia `slides.md` của mình thành nhiều file và sắp xếp chúng theo ý muốn.

`slides.md` :

```
Page 1
This is a normal page

src: ./subpage2.md

<!-- this page will be loaded from './subpage2.md' -->
Inline content will be ignored

```

`subpage2.md` :

```
Page 2
This page is from another file

```

## Frontmatter Merging

Bạn có thể cung cấp frontmatters từ cả mục nhập chính và các trang markdown pages. bên ngoài. Nếu có các khóa giống nhau trong chúng, thì những khóa từ **mục nhập chính có mức độ ưu tiên cao hơn**. Ví dụ

`slides.md` :

```

src: ./cover.md
background: https://sli.dev/bar.png
class: text-center

```

`cover.md` :

```

layout: cover
background: https://sli.dev/foo.png

```

# Cover

Cover Page

They will end up being equivalent of the following page:

```

layout: cover
background: https://sli.dev/bar.png
class: text-center

Cover

Cover Page
```

## Page Reusing

With the multi-entries support, reusing pages could be straightforward. For example:

```

src: ./cover.md

src: ./intro.md

src: ./content.md

reuse
src: ./content.md

```

---

[Go to TOC](#)

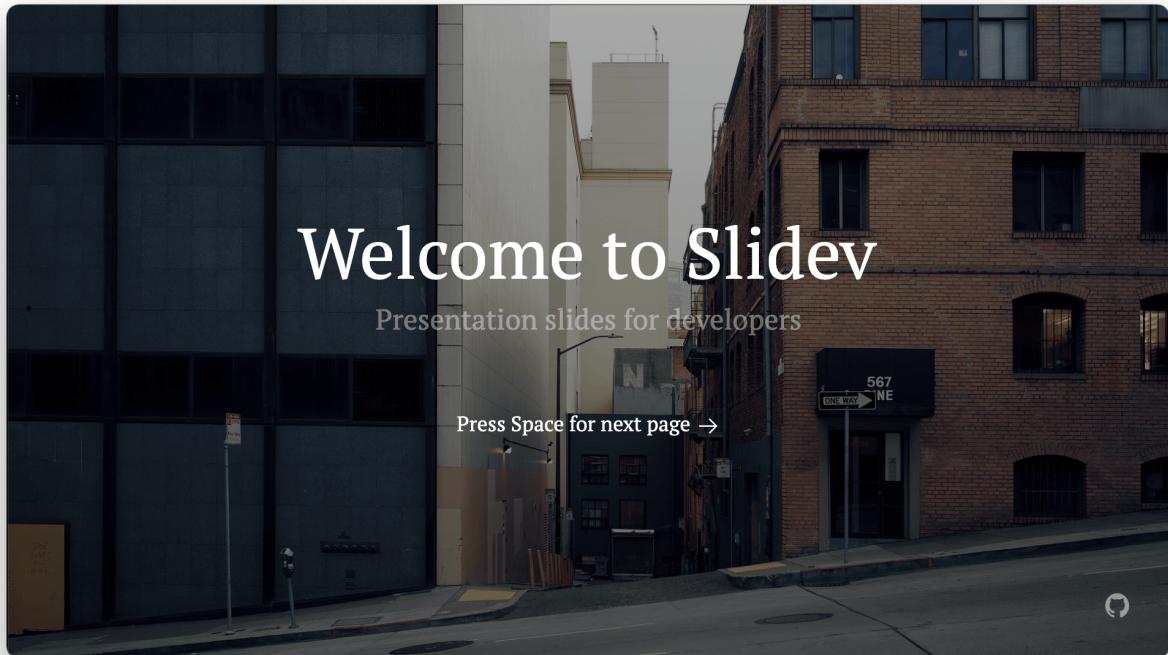
# Tại sao chọn Slidev

Có rất nhiều trình tạo slide WYSIWYG đa năng, có mục đích chung, như [Microsoft PowerPoint](#) và [Apple Keynote](#). Chúng hoạt động khá tốt để tạo các slide đẹp với animation, biểu đồ và nhiều thứ khác, đồng thời rất trực quan và dễ học. Vậy tại sao phải làm Slidev?

Slidev nhằm mục đích cung cấp tính linh hoạt và tính tương tác cho các nhà lập trình để làm cho bản trình bày của họ trở nên thú vị, biểu cảm và hấp dẫn hơn bằng cách sử dụng các công cụ và công nghệ mà họ đã quen thuộc.

Khi làm việc với các trình soạn thảo WYSIWYG, bạn rất dễ bị phân tâm bởi các tùy chọn tạo kiểu. Slidev khắc phục điều đó bằng cách tách nội dung và hình ảnh. Điều này cho phép bạn tập trung vào một thứ tại một thời điểm, đồng thời có thể sử dụng lại các chủ đề từ cộng đồng. Slidev không tìm cách thay thế hoàn toàn các slide deck builder khác. Thay vào đó, nó tập trung vào việc phục vụ cộng đồng nhà lập trình.

## Slidev



Dưới đây là một số tính năng thú vị nhất của Slidev:

## Dựa trên Markdown

Slidev sử dụng định dạng Markdown mở rộng để lưu trữ và sắp xếp các slide của bạn trong một file văn bản thuần. Điều này cho phép bạn tập trung vào việc tạo nội dung. Và vì nội dung và style được tách biệt, điều này cũng giúp bạn có thể chuyển đổi giữa các chủ đề khác nhau một cách dễ dàng.

Tìm hiểu thêm về [Cú pháp Markdown của Slidev](#).

## Themable

Các chủ đề cho Slidev có thể được chia sẻ và cài đặt bằng cách sử dụng các package npm. Sau đó, bạn áp dụng chúng chỉ với một dòng cấu hình.

Xem [thư viện chủ đề](#) hoặc [tìm hiểu cách viết chủ đề](#).

## Thân thiện với nhà lập trình

Slidev cung cấp hỗ trợ tốt nhất các đoạn code cho các nhà lập trình. Nó hỗ trợ cả [Prism](#) và [Shiki](#) để làm nổi bật cú pháp pixel hoàn hảo, trong khi vẫn có thể sửa đổi mã bất kỳ lúc nào thời gian. Với [trình soạn thảo Monaco](#) được tích hợp sẵn, nó cũng cho phép bạn viết code / trình bày trực tiếp trong bản trình bày của mình với tính năng tự động hoàn thành, gõ di chuột và thậm chí hỗ trợ kiểm tra kiểu TypeScript.

Tìm hiểu thêm về [đánh dấu và cấu hình Monaco](#).

## Nhanh

Slidev được cung cấp bởi [Vite](#), [Vue 3](#) và [Windi CSS](#), cung cấp cho bạn trải nghiệm tác giả tuyệt vời nhất. Mọi thay đổi bạn thực hiện sẽ phản ánh vào các slide của bạn **ngay lập tức**.

Tìm thêm về [our tech stack](#)

## Interactive & Expressive

Bạn có thể viết các component Vue tùy chỉnh và sử dụng chúng trực tiếp bên trong file markdown của mình. Bạn cũng có thể tương tác với chúng bên trong bản trình bày để thể hiện ý tưởng của mình theo cách trực quan và thú vị hơn.

## Hỗ trợ Recording

Slidev cung cấp tính năng record và xem camera tích hợp. Bạn có thể chia sẻ bản trình bày của mình với chế độ xem camera hoặc ghi và lưu chúng riêng cho màn hình và camera của bạn. Tất cả chỉ với một bước, không cần công cụ bổ sung.

Tìm hiểu thêm về [recording](#).

## Portable

Xuất các trang trình bày của bạn thành PDF, PNG hoặc thậm chí là Ứng dụng một trang có thể lưu trữ (SPA) chỉ với một lệnh duy nhất và chia sẻ chúng ở mọi nơi.

Đọc thêm về điều đó trong [tài liệu xuất](#).

## Hackable

Được hỗ trợ bởi công nghệ web, bất cứ điều gì có thể được thực hiện trong một ứng dụng web cũng có thể thực hiện được với Slidev. Ví dụ: WebGL, API requests, iframes hoặc thậm chí là chia sẻ trực tiếp. Nó tùy thuộc vào trí tưởng tượng của bạn!

## Hãy thử một lần

Thử với Slidev sẽ cho bạn biết hơn một nghìn từ. Bạn chỉ cần một lệnh:

```
$ npm init slidev
```

Hoặc xem trước nó:

Slidev First Preview Demo

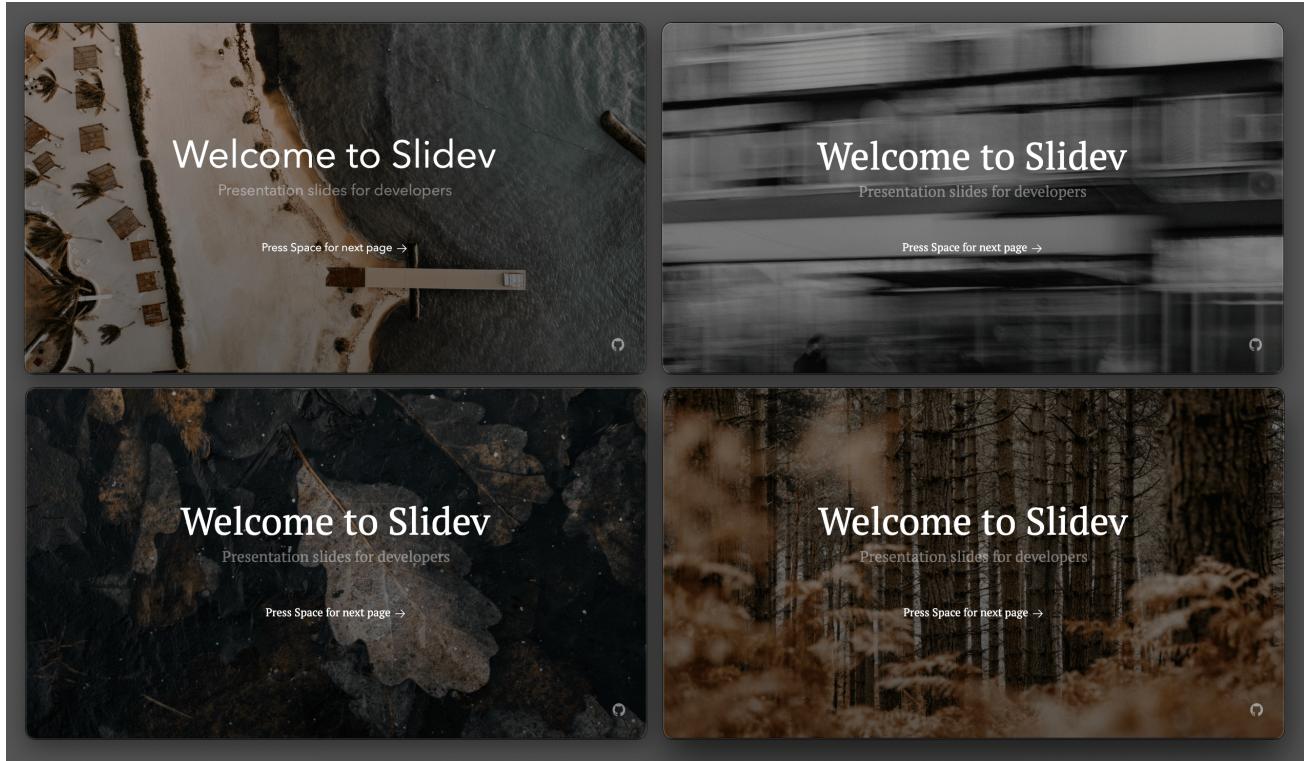


---

[Go to TOC](#)

# Bìa nổi bật

Chúng tôi đã sắp xếp một vài hình ảnh bìa để minh họa cho mẫu ban đầu của chúng tôi.



---  
# hình ảnh ngẫu nhiên từ bộ sưu tập nổi bật  
**background:** <https://source.unsplash.com/collection/94734566/1920x1080>  
---

Nếu bạn thích bất kỳ tác phẩm nào trong số chúng, hãy xem [bộ sưu tập Unsplash](#) của chúng tôi và tìm hiểu tác giả của chúng.

---

[Go to TOC](#)

# Learning Resources

## English

### Videos

Slidev - one of the best presentation software and it is free!



### Articles

- [Tips To Turn R Markdown Into Slidev Presentation](#) by Hiroaki Yutani

## 中文

- [Slidev : 一个用Markdown写slides的神器](#) by 梦里风林
- [神器！这款开源项目可以让你使用 Markdown 来做 PPT！](#) by [Github掘金计划](#)
- [【用 markdown 写 Slide!】神器 Slidev 的安装及 bug 解决](#) by HaloHoohoo

## 日本语

- [開発者のためのスライド作成ツール Slidev がすごい](#) by ryo\_kawamata
- [Markdownでオシャレなスライドを作るSli.dev](#) by Nobuko YAMADA

---

[Go to TOC](#)

# Showcases

Nói chuyện / Thuyết trình bằng Slidev.

---

[Go to TOC](#)

# Thư viện chủ đề

Duyệt qua các chủ đề tuyệt vời có sẵn cho Slidev tại đây.

Đọc thêm về [cách sử dụng chủ đề](#), hoặc [cách viết chủ đề của riêng bạn](#) và chia sẻ với cộng đồng!

## Chủ đề chính thức

## Chủ đề cộng đồng

Dưới đây là các chủ đề được quản lý bởi cộng đồng.

## Chủ đề khác

Tìm tất cả [các chủ đề có sẵn trên NPM](#).

---

[Go to TOC](#)

# Sử dụng chủ đề

Thay đổi chủ đề trong Slidev dễ dàng một cách đáng ngạc nhiên. Tất cả những gì bạn cần làm là thêm trường `theme:` vào frontmatter của bạn.

```

theme: serif

```

Bạn có thể khởi động máy chủ, máy chủ sẽ nhắc bạn cài đặt chủ đề tự động

? The theme "@slidev/theme-serif" was not found in your project, do you want to install it now? › (Y/n)

hoặc cài đặt chủ đề theo cách thủ công qua

```
$ npm install @slidev/theme-serif
```

Và đó là tất cả, hãy tận hưởng chủ đề mới! Để biết thêm chi tiết về cách sử dụng, bạn có thể tham khảo file README của chủ đề.

Muốn chia sẻ chủ đề của bạn? Tìm hiểu về [cách viết chủ đề](#).

## Eject chủ đề

Nếu bạn muốn có toàn quyền kiểm soát chủ đề hiện tại, bạn có thể **eject** nó vào hệ thống file cục bộ của bạn và sửa đổi nó theo ý muốn. Bằng cách chạy lệnh sau

```
$ slidev theme eject
```

Nó sẽ đẩy chủ đề bạn đang sử dụng hiện tại vào `./theme`, và thay đổi frontmatter của bạn thành

```

theme: ./theme

```

Điều này cũng có thể hữu ích nếu bạn muốn tạo chủ đề dựa trên chủ đề hiện có. Nếu bạn làm vậy, hãy nhớ đề cập đến chủ đề gốc và tác giả :)

## Chủ đề cục bộ

Như bạn có thể đã tìm hiểu từ phần trước, bạn có thể có một chủ đề cục bộ cho dự án của mình. Bằng cách có **đường dẫn tương đối** trong trường chủ đề của bạn.

```

theme: ./path/to/theme

```

Tham khảo [cách viết chủ đề](#) để biết thêm chi tiết.

---

[Go to TOC](#)

# Viết chủ đề mới

Để bắt đầu, chúng tôi khuyên bạn nên sử dụng trình tạo của chúng tôi để làm giàn giáo cho chủ đề đầu tiên của bạn

```
$ npm init slidev-theme
```

Sau đó, bạn có thể sửa đổi và thử với nó. Bạn cũng có thể tham khảo [chủ đề chính thức](#) làm ví dụ.

## Capability

Một chủ đề có thể đóng góp vào những điểm sau:

- Style toàn cầu
- Cung cấp cấu hình mặc định (font, lược đồ màu, đánh dấu, v.v...)
- Cung cấp layout tùy chỉnh hoặc ghi đè layout hiện có
- Cung cấp các component tùy chỉnh hoặc ghi đè component hiện có
- Mở rộng cấu hình CSS Windi
- Định cấu hình các công cụ như Monaco và Prism

## Quy ước

Chủ đề được publish lên npm registry và chúng phải tuân theo các quy ước dưới đây:

- Tên gói phải bắt đầu bằng `slidev-theme-`, ví dụ: `slidev-theme-awesome`
- Thêm `slidev-theme` và `slidev` trong trường `keywords` của `package.json`

## Cài đặt

Để thiết lập nơi thử nghiệm cho chủ đề của mình, bạn có thể tạo `example.md` với frontmatter sau để cho Slidev biết bạn đang sử dụng thư mục hiện tại làm chủ đề.

```

theme: ./
```

Theo tùy chọn, bạn cũng có thể thêm một số tập lệnh vào `package.json` của mình

```
// package.json
{
 "scripts": {
 "dev": "slidev example.md",
 "build": "slidev build example.md",
 "export": "slidev export example.md",
 "screenshot": "slidev export example.md --format png"
 }
}
```

Để xuất bản chủ đề của bạn, chỉ cần chạy `npm publish` và bạn đã sẵn sàng. Không yêu cầu quy trình xây dựng (có nghĩa là bạn có thể xuất bản trực tiếp các file `.vue` và `.ts`, Slidev đủ thông minh để hiểu chúng).

Điểm đóng góp của chủ đề tuân theo các quy ước giống như tùy chỉnh cục bộ, vui lòng tham khảo [tài liệu về quy ước đặt tên](#).

## Cấu hình mặc định

Có hiệu lực từ v0.19

Chủ đề có thể cung cấp [cấu hình](#) mặc định thông qua `package.json`.

```
// package.json
{
 "slidev": {
 "default": {
 "aspectRatio": "16/9",
 "canvasWidth": 980,
 "fonts": {
 "sans": "Robot",
 "mono": "Fira Code"
 }
 }
 }
}
```

Fonts sẽ được nhập tự động từ [Google Fonts](#).

Tìm hiểu thêm về [fonts](#) và [cấu hình frontmatter](#)

## Chủ đề Metadata

### Bảng màu

Theo mặc định, Slidev giả định các chủ đề hỗ trợ cả chế độ sáng và chế độ tối. Nếu bạn chỉ muốn chủ đề của mình được trình bày trong một lược đồ màu được thiết kế, bạn sẽ cần chỉ định nó một cách rõ ràng trong `package.json`

```
// package.json
{
 "name": "slidev-theme-my-cool-theme",
 "keywords": [
 "slidev-theme",
 "slidev"
],
 "slidev": {
 "colorSchema": "light" // or "dark" or "both"
 }
}
```

Để truy cập chế độ tối khi tạo kiểu chủ đề của bạn, bạn có thể wrap css dành riêng cho chế độ tối bên trong một class `dark`:

```
/* general css here */

html:not(.dark) {
 /* light mode css here */
}

html.dark {
 /* dark mode css here */
}
```

Slidev chuyển đổi một class `dark` trên phần tử `html` của trang để chuyển đổi bảng màu.

## Đánh dấu

Các màu đánh dấu cú pháp cũng được cung cấp trong chủ đề. Chúng tôi hỗ trợ cả `Prism` và `Shiki`. Để biết thêm thông tin, vui lòng tham khảo [tài liệu đánh dấu cú pháp](#).

Bạn có thể hỗ trợ một trong hai hoặc cả hai. Tham khảo chủ đề mặc định để biết các ví dụ về cấu hình `./styles/prism.css` / `./setup/shiki.ts`.

Ngoài ra, hãy nhớ chỉ định các điểm đánh dấu được hỗ trợ trong `package.json` của bạn

```
// package.json
{
 "slidev": {
 "highlighter": "shiki" // or "prism" or "all"
 }
}
```

## Phiên bản Slidev

Nếu chủ đề dựa trên một tính năng cụ thể của Slidev mới được giới thiệu, bạn có thể đặt phiên bản Slidev cần thiết để chủ đề của bạn hoạt động bình thường:

```
// package.json
{
 "engines": {
 "slidev": ">=0.19.3"
 }
}
```

Nếu người dùng đang sử dụng các phiên bản Slidev cũ hơn, lỗi sẽ xuất hiện.

# Colophon

This book is created by using the following sources:

- Slidev - Việt Nam
- GitHub source: [slidevjs/docs-vn](https://github.com/slidesjs/docs-vn)
- Created: 2022-11-27
- Bash v5.2.2
- Vivliostyle, <https://vivliostyle.org/>
- By: @shinokada
- GitHub repo: <https://github.com/shinokada/markdown-docs-as-pdf>