

SLIDEV Docs - Français



Sliddev

Table of contents

• Fr - README	4
• Fr - TRANSLATIONS	5
• Addons - Use	8
• Addons - Write an /addon	9
• Builtin - Components	11
• Builtin - Layouts	14
• Custom - Config katex	18
• Custom - Config mermaid	19
• Custom - Config monaco	20
• Custom - Config shortcuts	23
• Custom - Config vite	25
• Custom - Config vue	27
• Custom - Config windicss	28
• Custom - Directory structure	29
• Custom - Fonts	33
• Custom - Global layers	36
• Custom - Highlighters	38
• Custom - Index	40
• Custom - Vue context	42
• Guide - Animations	44
• Guide - Drawing	48
• Guide - Editors	50
• Guide - Exporting	53
• Guide - Faq	54
• Guide - Hosting	57
• Guide - Index	60
• Guide - Install	63
• Guide - Navigation	68
• Guide - Presenter mode	70
• Guide - Recording	71
• Guide - Syntax	72
• Guide - Why	82
• Resources - Covers	85
• Resources - Learning	86

• Fr - Showcases	87
• Themes - Gallery	88
• Themes - Use	89
• Themes - Write a /theme	91

fr.sli.dev

French documentation for Slidev

Translations

	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
日本語	docs-ja	ja.sli.dev	@IkumaTadokoro

Start Server Locally

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

And then visit <http://localhost:3000>

Or install the [Vite extension for VS Code](#) to edit side-by-side.

Help on Translating

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

addons/

- [use.md](#) - How to use Slidev addons
- [write-an-addon.md](#) - Write your own addon

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!

Utiliser un Addon

Les addons sont des ensembles de composants supplémentaires, de mises en page, de styles, de configuration, etc. que vous pouvez utiliser dans votre présentation.

Ils sont assez similaires à [themes](#), mais en général :

- ils n'affectent pas les styles globaux de vos diapositives
- vous pouvez utiliser plusieurs addons dans une présentation

Pour utiliser les addons, vous devez les installer manuellement via :

```
$ npm install [slidev-addon-package1] [slidev-addon-package2]
```

Et ensuite déclarez les addons soit dans votre frontmatter :

```
---
addons:
  - slidev-addon-package1
  - slidev-addon-package2
---
```

Ou dans votre fichier `package.json` :

```
// package.json
{
  "slidev": {
    "addons": [
      "slidev-addon-package1",
      "slidev-addon-package2",
    ]
  }
}
```

Écrire un Addon

Disponible depuis v0.32.1

Aptitude

Un addon peut contribuer aux points suivants :

- Styles globaux (à utiliser avec prudence c'est plus le rôle de [themes](#))
- Fournir des mises en page personnalisées ou remplacer celle existante
- Fournir des composants personnalisés ou remplacer celui existant
- Étendre les configurations Windi CSS
- Configurer des outils comme Monaco et Prism

Conventions

Les addons sont publiés dans le registre npm et doivent suivre les conventions ci-dessous :

- Le nom du package doit commencer par `slidev-addon-`, par exemple : `slidev-addon-awesome`
- Ajoutez `slidev-addon` et `slidev` dans le champ `keywords` de votre `package.json`

Installer

Initialisation

Pour créer votre addon, commencez par créer un répertoire avec créer un fichier `package.json` (vous pouvez utiliser `npm init`).

Ensuite, installez les dépendances slidev :

```
$ npm install -D @slidev/cli
```

Essai

Pour configurer le terrain de jeu de test pour votre addon, vous pouvez créer un fichier `example.md` avec du contenu.

Et éventuellement, vous pouvez également ajouter des scripts à votre `package.json`

```
// package.json
{
  "scripts": {
    "dev": "slidev example.md",
    "build": "slidev build example.md",
    "export": "slidev export example.md",
```

```
        "screenshot": "slidev export example.md --format png"
    }
}
```

Pour publier votre addon, lancez simplement `npm publish` et vous êtes prêt à partir. Aucun processus de construction n'est requis (ce qui signifie que vous pouvez publier directement les fichiers `.vue` et `.ts`, Slidev est assez intelligent pour les comprendre).

Les points de contribution supplémentaires suivent les mêmes conventions que la personnalisation locale, veuillez vous référer à [la documentation pour les conventions de nommage](#).

Métadonnées complémentaires

Version Diapositive

Si l'addon s'appuie sur une fonctionnalité spécifique de Slidev qui vient d'être introduite, vous pouvez définir la version minimale de Slidev requise pour que votre addon fonctionne correctement :

```
// package.json
{
  "engines": {
    "slidev": ">=0.32.1"
  }
}
```

Si les utilisateurs utilisent des versions plus anciennes de Slidev, une erreur sera générée.

Composants

Composants Intégrés

Les documentations de cette section sont toujours en cours d'élaboration. Avant cela, vous pouvez directement jeter un coup d'oeil au [code source](#)

Toc

Insérez une table des matières.

Si vous souhaitez qu'une diapositive n'apparaisse pas dans le composant `<Toc>`, vous pouvez utiliser dans le bloc de présentation de la diapositive :

```
---  
  hideInToc: true  
---
```

Les titres sont affichés à l'aide du composant `<Titres>`

Usage

```
<Toc />
```

Parameters:

- `columns` (`string | number`, défaut : `1`): Le nombre de colonnes de l'affichage
- `listClass` (`string | string[]`, défaut : `''`): Classes à appliquer à la table des matières
- `maxDepth` (`string | number`, défaut : `Infinity`): Le niveau de profondeur maximum du titre à afficher
- `minDepth` (`string | number`, défaut : `1`): Le niveau de profondeur minimum du titre à afficher
- `mode` (`'all' | 'onlyCurrentTree' | 'onlySiblings'`, défaut : `'all'`):
 - `'all'` : Afficher tous les articles
 - `'onlyCurrentTree'` : Afficher uniquement les éléments qui sont dans l'arborescence actuelle (élément actif, parents et enfants de l'élément actif)
 - `'onlySiblings'` : Afficher uniquement les éléments qui sont dans l'arborescence actuelle et leurs frères et sœurs directs

Link

Insérez un lien que vous pouvez utiliser pour accéder à une diapositive donnée.

Usage

```
<Link to="42">Aller à la diapositive 42</Link>
<Link to="42" title="Aller à la diapositive 42"/>
```

Paramètres:

- `to (string | number)`: Le chemin de la diapositive vers laquelle naviguer (les diapositives commencent à partir de "1")
- `title (string)`: Le titre à afficher

Titles

Insérez le titre principal d'une diapositive analysée au format HTML.

Les titres et les niveaux de titre sont automatiquement récupérés à partir du premier élément de titre de chaque diapositive.

Vous pouvez remplacer ce comportement automatique pour une diapositive à l'aide de la syntaxe frontale :

```
---  
  title: Amazing slide title  
  level: 2  
---
```

Usage

Le composant `<Titles>` est un composant virtuel avec lequel vous pouvez importer :

```
import Titles from '@/slidev/titles.md'
```

Ensuite, vous pouvez l'utiliser avec :

```
<Titles no="42" />
```

Paramètres:

- `no (string | number)`: Le numéro de la diapositive à partir de laquelle afficher le titre (les diapositives commencent à partir de `1`)

Composants Customisés

Créer un répertoire `components/` sous la racine de votre projet, et placez simplement vos composants Vue personnalisés sous celui-ci, puis vous pouvez l'utiliser avec le même nom dans votre fichier markdown!

Pour en savoir plus, consultez la section [Customisation](#)

Composants fournis par thème

Les thèmes peuvent également fournir des composants. Veuillez lire leurs documentations pour découvrir ce qu'ils fournissent.

En savoir plus sur la section [structure de répertoires](#)

Mises en page

Mises en page intégrées

Comme les thèmes peuvent remplacer le comportement des mises en page, le meilleur moyen de connaître exactement l'usage, les paramètres et les exemples est de se référer à leur documentation

center

Affiche le contenu au milieu de l'écran.

cover

Utilisé pour afficher la page de couverture pour la présentation, peut contenir le titre de la présentation, la contextualisation, etc.

default

La mise en page la plus basique, pour afficher n'importe quel type de contenu.

end

La page finale de la présentation.

fact

Pour montrer un fait ou des données avec beaucoup d'importance à l'écran.

full

Utilise tout l'espace de l'écran pour afficher le contenu.

image-left

Affiche une image sur le côté gauche de l'écran, le contenu sera placé sur le côté droit.

Usage

```
---  
layout: image-left  
  
# la source de l'image  
image: ./path/to/the/image  
  
# un nom de classe personnalisé du contenu  
class: my-cool-content-on-the-right  
---
```

image-right

Affiche une image sur le côté droit de l'écran, le contenu sera placé sur le côté gauche.

Usage

```
---
layout: image-right

# la source de l'image
image: ./path/to/the/image

# un nom de classe personnalisé au contenu
class: my-cool-content-on-the-left
---
```

image

Affiche une image comme contenu principal de la page.

Usage

```
---
layout: image

# la source de l'image
image: ./path/to/the/image
---
```

iframe-left

Affiche une page Web sur le côté gauche de l'écran, le contenu sera placé sur le côté droit.

Usage

```
---
layout: iframe-left

# la source de la page Web
url: https://github.com/slidesjs/slides

# un nom de classe personnalisé au contenu
class: my-cool-content-on-the-right
---
```

iframe-right

Affiche une page Web sur le côté droit de l'écran, le contenu sera placé sur le côté gauche.

Usage

```
---
layout: iframe-right

# la source de la page Web
url: https://github.com/slidesjs/slides
```

```
---  
# un nom de classe personnalisé au contenu  
class: my-cool-content-on-the-left
```

iframe

Affiche une page Web comme contenu principal de la page.

Usage

```
---  
layout: iframe  
  
# la source de la page Web  
url: https://github.com/slidesjs/slides  
---
```

intro

Pour introduire la présentation, habituellement avec le titre de présentation, une courte description, l'auteur, etc.

none

Une mise en page sans style existant.

quote

Pour afficher une citation avec proéminence.

section

Utilisé pour marquer le début d'une nouvelle section de présentation.

statement

Faire une affirmation/déclaration comme contenu de la page principale.

two-cols

Séparer le contenu de la page en deux colonnes.

Usage

```
---  
layout: two-cols  
---  
  
# Left  
Cela apparaît sur la gauche  
::right::
```

Right

Cela apparaît sur la droite

Mises en page personnalisées

Créer un répertoire `layouts/` sous la racine de votre projet, et placez simplement vos composants Vue personnalisés sous celui-ci.

Pour en savoir plus, consultez la section [Customisation](#)

Mises en page fournies par thème

Les thèmes peuvent fournir des mises en page ou remplacer celles existantes. Veuillez lire leurs documentations pour découvrir ce qu'ils fournissent.

Configurer KaTeX

Créer `./setup/katex.ts` avec le contenu suivant :

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

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

Avec la configuration, vous pouvez fournir le paramètre personnalisé pour [Options KaTeX](#). Reportez-vous aux définitions de type et à leur documentation pour plus de détails.

Configurer Mermaid

Créer `./setup/mermaid.ts` avec le contenu suivant :

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

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

Avec la configuration, vous pouvez fournir un paramètre par défaut personnalisé pour [Mermaid](#). Reportez-vous aux définitions de type et à sa documentation pour plus de détails.

Configurer Monaco

Par défaut, uniquement Monaco est activé uniquement en mode développement. Pour le faire fonctionner sur la version SPA, ajoutez `monaco: true` à vos configurations frontmatter.

Créer `./setup/monaco.ts` avec le contenu suivant :

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

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

En savoir plus sur [la configuration de Monaco](#).

Utilisation

Pour utiliser Monaco dans vos diapositives, ajoutez simplement `{monaco}` à vos extraits de code :

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

console.log(plusOne.value) // 2

plusOne.value++ // error
//````
```

À

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

console.log(plusOne.value) // 2

plusOne.value++ // error
//````
```

Exporter

Par défaut, Monaco fonctionnera UNIQUEMENT en mode `dev`. Si vous souhaitez également l'avoir disponible dans le SPA exporté, vous pouvez le configurer dans votre frontmatter :

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

Types d'installation automatique

Lorsque vous utilisez TypeScript avec Monaco, les types de dépendances sont installés automatiquement côté client.

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

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

Dans l'exemple ci-dessus, assurez-vous simplement que `vue` et `@vueuse/core` sont installés localement en tant que dependencies / devDependencies, Slidev s'occupera du reste et votre éditeur fonctionnera !

Configure Themes

Le thème est contrôlé par Slidev basé sur le thème clair/sombre. Si vous souhaitez le customiser, vous pouvez passer l'identifiant du thème dans la fonction `setup` :

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

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

Si vous souhaitez charger des thèmes personnalisés :

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

Si vous créez un thème pour Slidev, utilisez dynamiquement `import()` au sein de la fonction de setup afin d'avoir le meilleur résultat d'arborescence et de fractionnement de code possible.

[Go to TOC](#)

Configurer les raccourcis

Disponible depuis v0.20

Since v0.35.6 (excluded), you decide which base shortcuts to keep (see `...base`, below).

Créez `./setup/shortcuts.ts` avec le contenu suivant :

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

export default defineShortcutsSetup((nav: NavOperations, base: ShortcutOptions[]) => {
  return [
    ...base, // keep the existing shortcuts
    {
      key: 'enter',
      fn: () => nav.next(),
      autoRepeat: true,
    },
    {
      key: 'backspace',
      fn: () => nav.prev(),
      autoRepeat: true,
    },
  ],
})
```

Avec la configuration, vous pouvez fournir le paramètre personnalisé pour les raccourcis mentionnés dans [Navigation](#). La configuration ci-dessus lie l'animation ou la diapositive suivante à `enter` et l'animation ou la diapositive précédente à `backspace`.

La fonction de configuration reçoit un objet avec certaines méthodes de navigation et renvoie un tableau contenant une configuration de raccourci. Reportez-vous aux définitions de type pour plus de détails.

Le type `key` n'autorise que les chaînes, mais vous pouvez toujours lier plusieurs clés en utilisant la convention suivante :

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

export default defineShortcutsSetup((nav: NavOperations, base: ShortcutOptions[]) => {
  return [
    ...base,
    {
      key: 'ShiftLeft+ArrowRight',
      fn: () => nav.next(),
      autoRepeat: true,
    },
  ],
})
```

```
    }  
  ]  
})
```

Reportez-vous à [useMagicKeys](#) | [VueUse](#) pour plus de détails sur l'événement de pression de touche.

[Go to TOC](#)

Configure Vite

Slidev est alimenté par [Vite](#) sous le capot. Cela signifie que vous pouvez tirer parti de l'excellent système de plugins de Vite pour personnaliser encore plus vos diapositives.

La configuration `vite.config.ts` sera respectée si vous en avez une.

Slidev a les plugins suivants préconfigurés :

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

Découvrez le mode sur les [pré-configurations ici](#).

Configurer les plugins internes

Disponible depuis v0.21

Pour configurer la liste des plugins intégrés ci-dessus, créez `vite.config.ts` avec le contenu suivant. Veuillez noter que Slidev a quelques options de préconfiguration pour ces plugins, cette utilisation remplacera certaines d'entre elles, ce qui pourrait potentiellement provoquer la panne de l'application. Veuillez considérer cela comme **une fonctionnalité avancée**, assurez-vous de savoir ce que vous faites avant de continuer.

```
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 */
    },
  },
})
```

Voir les [déclarations de types](#) pour plus d'options.

[Go to TOC](#)

Configurer Vue

Slidev utilise [Vue 3](#) pour rendre l'application côté client. Vous pouvez étendre l'application pour ajouter des plugins ou des configurations personnalisés.

Créer `./setup/main.ts` avec le contenu suivant :

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

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

Cela peut également être utilisé comme entrée principale de votre application Slidev pour effectuer des initialisations avant le démarrage de l'application.

En savoir plus: [Vue Application API](#).

Configurer Windi CSS

Markdown prend naturellement en charge les balises HTML intégrées. Vous pouvez donc styliser votre contenu comme vous le souhaitez. Pour plus de commodité, nous avons [Windi CSS](#) intégré, vous pouvez donc styliser le balisage directement à l'aide des utilitaires de classe.

Par exemple :

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

Le [Mode Attributify](#) dans [Windi CSS v3.0](#) est activé par défaut.

Configurations

Pour configurer Windi CSS, créez `setup/windicss.ts` avec le contenu suivant pour étendre les configurations intégrées

```
// setup/windicss.ts

import { defineWindiSetup } from '@slidev/types'

// étendre les configurations de windics intégrées
export default defineWindiSetup(() => ({
  shortcuts: {
    // personnaliser l'arrière-plan par défaut
    'bg-main': 'bg-white text-[#181818] dark:(bg-[#121212] text-[#ddd])',
  },
  theme: {
    extend: {
      // les polices peuvent être remplacées ici, n'oubliez pas de mettre à jour
      // les liens de polices Web dans `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',
      },
    },
  },
}))
```

En savoir plus sur les [configurations WindiCSS](#)

[Go to TOC](#)

Structure du répertoire

Slidev utilise certaines conventions de structure de répertoires pour minimiser la surface de configuration et rendre les extensions de fonctionnalités flexibles et intuitives.

La structure de base est la suivante :

```
your-slidev/
  └── components/      # composants personnalisés
  └── layouts/         # mises en page personnalisées
  └── public/          # assets statiques
  └── setup/           # configuration personnalisée / hooks
  └── styles/          # style personnalisé
  └── index.html       # injections dans index.html
  └── slides.md        # l'entrée principale des diapositives
  └── vite.config.ts   # extension de vite config
```

Tous sont facultatifs.

Composants

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

Les composants à l'intérieur de ce répertoire peuvent être directement utilisés dans les diapositives Markdown avec le même nom de composant que le nom de fichier.

Par exemple :

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

```
<!-- slides.md -->
# My Slide
<MyComponent :count="4"/>
<!-- Les deux appellations fonctionnent -->
<hello-world foo="bar">
  Slot
</hello-world>
```

Cette fonctionnalité est alimentée par `unplugin-vue-components`, apprenez-en plus ici.

Slidev fournit également des [composants intégrés](#) que vous pouvez utiliser.

Mises en page

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

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

Vous pouvez utiliser n'importe quel nom de fichier pour votre mise en page. Vous référez ensuite votre mise en page dans votre en-tête YAML en utilisant le nom de fichier.

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

Si la mise en page que vous fournissez porte le même nom qu'une mise en page intégrée ou une mise en page de thème, votre mise en page personnalisée prévaudra sur la mise en page intégrée / thème. L'ordre de priorité est `local > thème > intégré`.

Dans le composant de mise en page, utilisez `<slot />` pour le contenu de la diapositive. Par exemple :

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

Public

Conventions : `./public/*`

Les actifs de ce répertoire seront servis à la racine du chemin `/` pendant le développement, et copiés à la racine du répertoire dist tels quels. En savoir plus sur [le répertoire public de Vite](#).

Style

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

Les fichiers suivant cette convention seront injectés à la racine de l'application. Si vous devez importer plusieurs entrées CSS, vous pouvez créer la structure suivante et gérer vous-même l'ordre d'importation.

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

```
// styles/index.ts

import './base.css'
import './code.css'
import './layouts.css'
```

Les styles seront traités par [Windi CSS](#) et [PostCSS](#), vous pouvez donc utiliser l'imbrication css et [at-directives](#) prêt à l'emploi. Par exemple :

```
.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');
  }
}
```

[En savoir plus sur la syntaxe.](#)

index.html

Conventions : `index.html`

Le `index.html` offre la possibilité d'injecter des balises meta et/ou des scripts dans le `index.html` principal

Par exemple, pour le `index.html` personnalisé suivant :

```
<!-- ./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>
```

Le fichier `index.html` final hébergé sera :

```
<!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>
```

Couches globales

Conventions : `global-top.vue` | `global-bottom.vue`

En savoir plus: [Couches globales](#)

[Go to TOC](#)

Polices

Disponible depuis v0.20

Bien que vous puissiez utiliser HTML et CSS pour personnaliser les polices et le style de vos diapositives à votre guise, Slidev offre également un moyen pratique de les utiliser sans effort.

Dans votre fontmatter, configurez comme suit :

```
---  
  fonts:  
    # le texte  
    sans: 'Robot'  
    # utiliser avec la classe css `font-serif` de windicss  
    serif: 'Robot Slab'  
    # pour les blocs de code, le code en ligne, etc.  
    mono: 'Fira Code'  
---
```

Et c'est tout.

Les polices seront **importées automatiquement depuis Google Fonts**. Cela signifie que vous pouvez utiliser directement toutes les polices disponibles sur Google Fonts.

Polices locales

Par défaut, Slidev suppose que toutes les polices spécifiées via les configurations `fonts` proviennent de Google Fonts. Si vous souhaitez utiliser des polices locales, spécifiez `fonts.local` pour désactiver l'importation automatique.

```
---  
  fonts:  
    # comme font-family en css, vous pouvez utiliser `,` pour séparer plusieurs  
    # polices de secours  
    sans: 'Helvetica Neue, Robot'  
    # marquer 'Helvetica Neue' comme police locale  
    local: 'Helvetica Neue'  
---
```

Poids et italique

Par défaut, Slidev importe trois poids "200", "400", "600" pour chaque police. Vous pouvez les configurer en :

```
---  
  fonts:  
    sans: 'Robot'  
    # défaut  
    weights: '200,400,600'
```

```
# importer des polices italiques, par défaut `false`
italic: false
---
```

Cette configuration est appliquée à toutes les polices web. Pour des contrôles plus précis des poids de chaque police, vous devrez les importer manuellement avec [HTML](#) et CSS.

Police de remplacement

Pour la plupart des scénarios, il vous suffit de spécifier la "police spéciale" et Slidev ajoutera les polices de secours pour vous, par exemple :

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

aura pour résultat

```
.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;
}
```

Si vous souhaitez désactiver les polices de secours, configurez comme suit

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

Fournisseurs

- Options : `google` | `none`
- Par défaut : `google`

Actuellement, seules les polices Google sont prises en charge, nous prévoyons d'ajouter d'autres fournisseurs à l'avenir. Spécifier `none` désactivera complètement la fonction d'importation automatique et traitera toutes les polices en local.

```
---  
  fonts:  
    provider: 'none'  
---
```

Couches globales

Disponible depuis v0.17

Les couches globales vous permettent d'avoir des composants personnalisés qui **persistent** sur les diapositives. Cela pourrait être utile pour avoir des pieds de page, des animations de diapositives croisées, des effets globaux, etc.

Slidev fournit deux couches pour cet usage, créez `global-top.vue`, `global-bottom.vue` ou `custom-nav-controls.vue` sous la racine de votre projet et il reprendra automatiquement.

Relation des couches :

- Global Top (`global-top.vue`)
- Diapositives
- Global Bottom (`global-bottom.vue`)
- NavControls
 - Customized Navigation Controls (`custom-nav-controls.vue`)

Exemple

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

Le texte `Votre nom` apparaîtra sur toutes vos diapositives.

```
<!-- custom-nav-controls -->
<template>
  <button class="icon-btn" title="Next" @click="$slidev.nav.next">
    <carbon:arrow-right />
  </button>
</template>
```

Le bouton `Next` apparaîtra dans NavControls.

Pour l'activer conditionnellement, vous pouvez l'appliquer avec le [Contexte Global Vue](#).

```
<!-- masquer le pied de page de la page 4 -->
<template>
  <footer
    v-if="$slidev.nav.currentPage !== 4"
    class="absolute bottom-0 left-0 right-0 p-2">
    >
      Votre name
    </footer>
  </template>
```

```
<!-- masquer le pied de page de la mise en page "cover" -->
<template>
  <footer>
    <v-if="$slidev.nav.currentLayout !== 'cover'">
      <div>{{Votre nom}}</div>
    </v-if>
  </footer>
</template>
```

```
<!-- un exemple de pied de page pour les pages -->
<template>
  <footer>
    <v-if="$slidev.nav.currentLayout !== 'cover'">
      <div>{{ $slidev.nav.currentPage }} / {{ $slidev.nav.total }}</div>
    </v-if>
  </footer>
</template>
```

```
<!-- custom-nav-controls -->
<!-- hide the button in Presenter model -->
<template>
  <button v-if="!$slidev.nav.isPresenter" class="icon-btn" title="Next"
    @click="$slidev.nav.next">
    <carbon:arrow-right />
  </button>
</template>
```

Surligneurs

Slidev est livré avec deux surligneurs de syntaxe parmi lesquels vous pouvez choisir :

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

Prism est l'un des surligneurs de syntaxe les plus populaires. La mise en évidence est effectuée en ajoutant des classes de jetons au code et il est coloré à l'aide de CSS. Vous pouvez parcourir leurs [thèmes officiels](#), ou en créer/personnaliser un vous-même très facilement en utilisant `prism-theme-vars`.

Shiki, d'autre part, est un surlieur de syntaxe basé sur la grammaire TextMate. Il génère des jetons colorés, il n'y a donc pas besoin de CSS supplémentaire. Comme il prend en charge la grammaire, les couleurs générées sont très précises, tout comme ce que vous verrez dans VS Code. Shiki est également livré avec [un tas de thèmes intégrés](#). L'inconvénient de Shiki est qu'il nécessite également des thèmes TextMate (compatibles avec le thème VS Code) pour faire la mise en évidence, ce qui peut être un peu plus difficile à personnaliser.

Les thèmes Slidev prennent généralement en charge Prism et Shiki, mais selon le thème que vous utilisez, il se peut qu'il ne prenne en charge que l'un d'entre eux.

Lorsque vous avez le choix, le compromis est essentiellement :

- **Prism** pour une personnalisation plus facile
- **Shiki** pour une mise en évidence plus précise

Par défaut, Slidev utilise Prism. Vous pouvez le changer en modifiant votre frontmatter :

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

Configurer Prism

Pour configurer votre Prism, vous pouvez simplement importer le thème css ou utiliser `prism-theme-vars` pour configurer des thèmes pour les modes clair et sombre. Reportez-vous à sa documentation pour plus de détails.

Configurer Shiki

Créer un fichier `./setup/shiki.ts` avec le contenu suivant :

```
/* ./setup/shiki.ts */  
import { defineShikiSetup } from '@slidev/types'  
  
export default defineShikiSetup(() => {  
  return {  
    theme: {  
      dark: 'min-dark',  
      light: 'min-light',  
    },  
  };  
});
```

```
  },
})
```

Reportez-vous à la [documentation de Shiki](#) pour les noms de thèmes disponibles.

Ou si vous souhaitez utiliser votre propre thème :

```
/* ./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'),
    },
  }
})
```

Customisations

Slidev est entièrement personnalisable, du style aux configurations d'outillage. Il vous permet de configurer les outils ci-dessous ([Vite](#), [Windi CSS](#), [Monaco](#), etc.)

Configurations du frontmatter

Vous pouvez configurer Slidev dans le frontmatter de votre première diapositive, ce qui suit montre la valeur par défaut pour chaque option.

```
---  
# identifiant du thème ou nom du package  
# En savoir plus : https://sli.dev/themes/use.html  
theme: 'default'  
# titre de votre diapositive, sera automatiquement déduit du premier en-tête s'il  
n'est pas spécifié  
title: 'Slidev'  
# titleTemplate pour la page Web, `'%s` sera remplacé par le titre de la page  
titleTemplate: '%s - Slidev'  
# informations pour vos diapositives, peut être une chaîne de démarquage  
info: false  
  
# téléchargement de pdf activé dans la version SPA, peut également être une URL  
personnalisée  
download: false  
# nom de fichier du fichier d'exportation  
exportFilename: 'slidev-exported'  
# surligneur de syntaxe, peut être 'prism' ou 'shiki'  
highlighter: 'prism'  
# afficher les numéros de ligne dans les blocs de code  
lineNumbers: false  
# activer l'éditeur monaco, peut être booléen, 'dev' ou 'build'  
monaco: 'dev'  
# télécharger les actifs distants en local à l'aide de vite-plugin-remote-assets,  
peut être booléen, 'dev' ou 'build'  
remoteAssets: false  
# contrôle si les textes des diapositives sont sélectionnables  
selectable: true  
# activer l'enregistrement de diapositives, peut être booléen, 'dev' ou 'build'  
record: 'dev'  
  
# forcer le schéma de couleur pour les diapositives, peut être 'auto', 'light' ou  
'dark'  
colorSchema: 'auto'  
# mode routeur pour vue-router, peut être 'history' ou 'hash'  
routerMode: 'history'  
# ratio d'aspect pour les diapositives  
aspectRatio: '16/9'  
# largeur réelle du canvas, unité en px  
canvasWidth: 980  
# utilisé pour la personnalisation du thème, injectera les styles racine comme `--  
slidev-theme-x` pour l'attribut `x`  
themeConfig:  
  primary: '#5d8392'  
  
# favicon, peut être un chemin de fichier local ou une URL
```

```

favicon: 'https://cdn.jsdelivr.net/gh/slideserve/slideserve/assets/favicon.png'
# URL du serveur PlantUML utilisé pour afficher les diagrammes
plantUmlServer: 'https://www.plantuml.com/plantuml'
# polices seront automatiquement importées à partir des polices Google
# En savoir plus : https://sli.dev/custom/fonts
fonts:
  sans: 'Roboto'
  serif: 'Roboto Slab'
  mono: 'Fira Code'

# le frontmatter par défaut s'applique à toutes les diapositives
defaults:
  layout: 'default'
  # ...

# options de dessin
# En savoir plus : https://sli.dev/guide/drawing.html
drawings:
  enabled: true
  persist: false
  presenterOnly: false
  syncAll: true

```

Consultez les [définitions de type](#) pour plus d'options.

Structure du répertoire

Slidev utilise des conventions de structure de répertoires pour minimiser la surface de configuration et rendre les extensions de fonctionnalités flexibles et intuitives.

Reportez-vous à la section [Structure des répertoires](#).

vite.config.ts

Reportez-vous à la section [Configurer Vite](#).

windicss.config.ts

Reportez-vous à la section [Configurer Windi CSS](#).

Configuration des outils

- [Surligneurs](#)
- [Configurer Vue](#)
- [Configurer Vite](#)
- [Configurer Windi CSS](#)
- [Configurer Monaco](#)
- [Configurer KaTeX](#)
- [Configurer Mermaid](#)

[Go to TOC](#)

Contexte Global Vue

Slidev a injecté un [contexte global Vue](#) `$slidev` pour les conditions avancées ou les commandes de navigation.

Usage

Vous pouvez y accéder n'importe où dans votre markdown et modèle Vue, avec la [syntaxe "Mustache"](#).

```
<!-- slides.md -->
# Page 1
La page actuelle est : {{ $slidev.nav.currentPage }}

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

Propriétés

`$slidev.nav`

Un objet réactif contenant les propriétés et les contrôles de la navigation des diapositives. Par exemple :

```
$slidev.nav.next() // passer à l'étape suivante
$slidev.nav.nextSlide() // passer à la diapositive suivante (saute les v-clicks)
$slidev.nav.go(10) // aller à la diapositive #10

$slidev.nav.currentPage // numéro de la diapositive actuelle
$slidev.nav.currentLayout // ID de mise en page actuelle
$slidev.nav.clicks // nombre de clics en cours
```

Pour plus de propriétés disponibles, reportez-vous aux exportations [nav.ts](#).

`$slidev.configs`

Un objet réactif contenant les [configurations analysées dans le premier frontmatter](#) de votre `slides.md`. Par example :

```
---
title: Ma première Slidev
---

{{ $slidev.configs.title }} // Ma première Slidev
```

\$slidev.themeConfigs

Un objet réactif contenant les configurations de thème analysées.

```
---  
title: Ma première Slidev  
themeConfig:  
  primary: #213435  
---
```

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

Animations

Animations de Clics

v-click

Pour appliquer des "animations de clic" aux éléments, vous pouvez utiliser la directive `v-click` ou les composants `<v-click>`

```
# Hello

<!-- Utilisation des composants : ce sera invisible jusqu'à ce que vous appuyiez
sur "suivant" -->
<v-click>

Hello World

</v-click>

<!-- Utilisation de la directive : elle sera invisible jusqu'à ce que vous
appuyiez sur "suivant" la deuxième fois -->
<div v-click class="text-xl p-2">

Bonjour!

</div>
```

v-after

L'utilisation de `v-after` est similaire à `v-click` mais cela rendra l'élément visible lorsque le `v-click` précédent sera déclenché.

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

Lorsque vous cliquez sur le bouton "suivant", `Hello` et `World` s'affichent ensemble.

v-click-hide

Same as `v-click` but instead of making the element appear, it makes the element invisible after clicking.

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

v-clicks

`v-clicks` n'est fourni qu'en tant que composant. C'est un raccourci pour appliquer la directive `v-click` à tous ses éléments enfants. C'est particulièrement utile lorsque vous travaillez avec des listes.

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

```
- Item 3
- Item 4

</v-clicks>
```

Un élément deviendra visible à chaque fois que vous cliquerez sur "suivant".

Nombre de clics personnalisés

Par défaut, Slidev compte le nombre d'étapes nécessaires avant de passer à la diapositive suivante. Vous pouvez remplacer ce paramètre en passant l'option frontmatter `clicks` :

```
---  
# 10 clicks in this slide, before going to the next  
clicks: 10  
---
```

Ordre

En passant l'index de clic à vos directives, vous pouvez personnaliser l'ordre de la révélation

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

```
<!-- the order is reversed -->
<div v-click="3">1</div>
<div v-click="2">2</div>
<div v-click="1">3</div>
```

```
---  
clicks: 3  
---  
  
<!-- visible après 3 clics -->  
<v-clicks at="3">  
  <div>Bonjour</div>  
</v-clicks>
```

Transitions d'éléments

Lorsque vous appliquez la directive `v-click` à vos éléments, elle y attache le nom de classe `slidev-vclick-target`. Lorsque les éléments sont masqués, le nom de classe `slidev-vclick-hidden` sera également attaché. Par exemple :

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

Après un clic, il deviendra

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

Par défaut, une transition d'opacité subtile est appliquée à ces classes :

```
// the default

.slidev-vclick-target {
  transition: opacity 100ms ease;
}

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

Vous pouvez les remplacer pour personnaliser les effets de transition dans vos feuilles de style personnalisées.

Par exemple, vous pouvez réaliser les transitions de mise à l'échelle en :

```
// styles.css

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

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

Pour spécifier des animations uniquement pour certaines diapositives ou mises en page

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

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

En savoir plus sur la [personnalisation des styles](#)

Mouvement

Slidev a [@vueuse/motion](#) intégré. Vous pouvez utiliser la directive `v-motion` sur n'importe quel élément pour y appliquer un mouvement. Par exemple

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

Le texte `Slidev` passera de `-80px` à sa position d'origine lors de l'initialisation.

Remarque : Sliddev précharge la diapositive suivante pour les performances, ce qui signifie que les animations peuvent démarrer avant que vous ne naviguez vers la page. Pour que cela fonctionne correctement, vous pouvez désactiver le préchargement pour la diapositive particulière

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

Ou contrôlez le cycle de vie de l'élément avec `v-if` pour avoir des contrôles fins

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

Mode d'apprentissage: [Démo](#) | [@vueuse/motion](#) | [v-motion](#) | [Préréglages](<https://motion.vueuse.org/presets.html>)

Transitions de pages

La prise en charge intégrée des diapositives n'est PAS ENCORE fournie dans la version actuelle. Nous prévoyons d'ajouter un support pour eux dans la prochaine version majeure. Avant cela, vous pouvez toujours utiliser vos styles et bibliothèques personnalisés pour ce faire.

Dessins & Annotations

Disponible depuis v0.23

Nous avons [drauu](#) intégré pour le dessin et l'annotation qui pourraient améliorer davantage votre présentation.

Pour commencer, cliquez sur l'icône dans la barre d'outils et commencez à dessiner. Il est également disponible dans le [Mode Présentateur](#). Les dessins et les annotations que vous avez créés seront **synchronisés** automatiquement sur toutes les instances en temps réel.

Utiliser avec stylet

Lors de l'utilisation d'un stylet sur une tablette (par exemple, iPad avec Apple Pencil), Slidev pourrait déterminer intelligemment le type d'entrée. Vous pouvez dessiner directement sur vos diapositives avec le stylet sans activer le mode dessin, tout en laissant vos doigts ou votre souris contrôler la navigation.

Dessins persistants

La configuration frontmatter suivante vous permet de conserver vos dessins en tant que SVG sous le répertoire `.slidev/drawings` et de les avoir dans votre pdf exporté ou votre site hébergé.

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

Désactiver les dessins

Entièrement :

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

Uniquement en développement :

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

Uniquement en mode Présentateur :

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

Synchronisation des dessins

Par défaut, Slidev synchronise vos dessins sur toutes les instances. Si vous partagez vos diapositives avec d'autres personnes, vous pouvez désactiver la synchronisation en :

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

Avec cette configuration, seul le dessin de l'instance du présentateur pourra se synchroniser avec les autres.

Support de l'éditeur

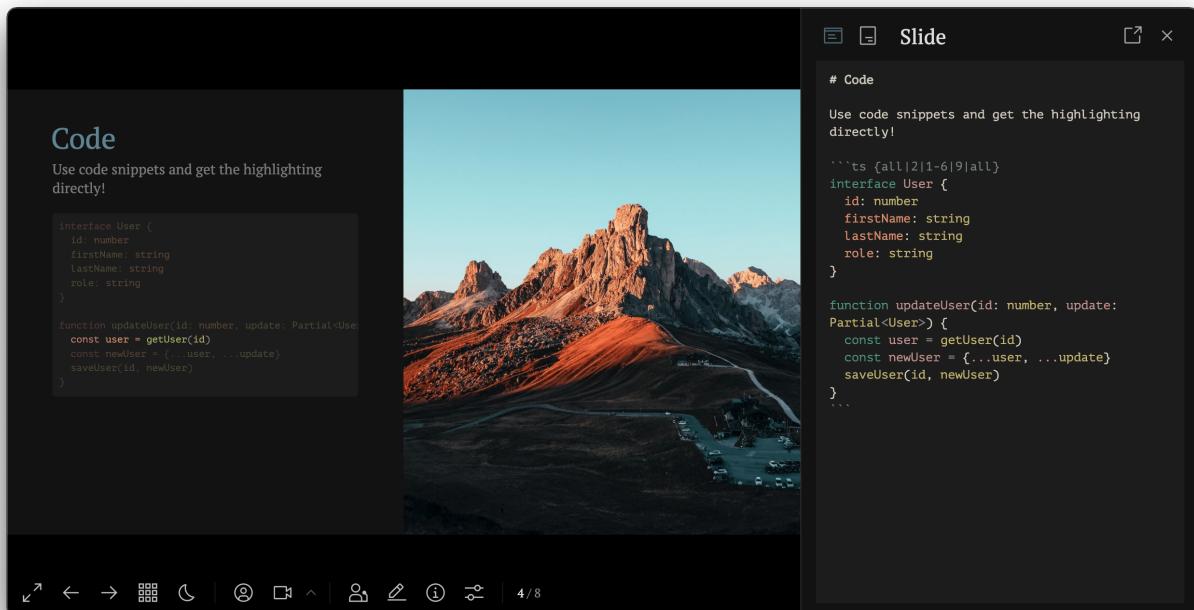
Puisque Slidev utilise Markdown comme entrée source, vous pouvez utiliser TOUS les éditeurs que vous aimez pour l'écrire.

Si vous souhaitez une gestion de haut niveau de vos diapositives, nous avons fourni les intégrations d'éditeur suivantes pour vous !

Éditeur intégré

Slidev est livré avec un éditeur intégré [CodeMirror](#) qui rechargera et enregistrera instantanément les modifications apportées à votre fichier.

Cliquez sur le bouton pour l'ouvrir.



VS Code Extension



Slidev for VS Code

 VS Code Marketplace

v0.4.1

downloads 26k

L'extension VS Code fournit certaines fonctionnalités pour vous aider à mieux organiser vos diapositives et en avoir un aperçu rapide.

Caractéristiques

- Afficher les diapositives dans le panneau latéral
- Aller aux boutons suivant / précédent
- Réorganiser les diapositives
- Pliage pour blocs coulissants
- Convertir Markdown en HTML

The screenshot shows a slide editor interface. On the left is a sidebar with a file tree titled "SLIDEV: SLIDES". The tree includes items like "Composable Vue", "Anthony Fu", "Sponsors", and several sections under "Composition API" such as "Ref", "Ref Auto Unwrapping", "Patterns & Tips", and "What's Composable Func...". On the right is a main editor window titled "slides.md — demo". The code in the editor is:

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

watch(counter, count => {
 console.log(count) // same as `counter.value`
})
```
  - Ref is auto unwrapped in the template
```
190 <template>
 <button @click="counter += 1">
```

---

[Go to TOC](#)

# Exportation

## PDF

L'exportation au format PDF ou PNG repose sur [Playwright](#) pour le rendu. Vous devrez donc installer `playwright-chromium` pour utiliser cette fonctionnalité. Si vous effectuez une exportation dans un environnement CI, [le guide du dramaturge CI](#) peut être utile.

Installez `playwright-chrome`

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

Exportez maintenant vos diapositives au format PDF à l'aide de la commande suivante

```
$ slidev export
```

Après quelques secondes, vos diapositives seront prêtes dans `./slides-exports.pdf`.

### Étapes d'exportation des clics

Disponible depuis v0.21

Par défaut, Slide export une page par diapositive avec les animations de clics désactivées. Si vous souhaitez exporter des diapositives avec plusieurs étapes dans plusieurs pages, passez l'option `--with-clicks`.

```
$ slidev export --with-clicks
```

## PNGs

En passant l'option `--format png`, Slidev exportera des images PNG pour chaque diapositive au lieu d'un PDF.

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

## Single-Page Application (SPA)

Voir [Hébergement statique](#).

---

[Go to TOC](#)

# FAQ

## Grilles

Étant donné que Slidev est basé sur le Web, vous pouvez appliquer n'importe quelle disposition de grille à votre guise. [Grilles CSS](#), [flexboxes](#), ou même [Masonry](#), vous obtenez tous les contrôles.

Puisque nous avons [Windi CSS](#) intégré, voici un moyen simple de référence :

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

 La première colonne

</div>
<div>

 La seconde colonne

</div>
</div>
```

Pour aller plus loin, vous pouvez personnaliser la taille de chaque colonne comme :

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

 La première colonne (200px)

</div>
<div>

 La seconde colonne (ajustement automatique)

</div>
<div>

 La troisième colonne (10% de largeur jusqu'au conteneur parent)

</div>
</div>
```

En savoir plus sur [les grilles Windi CSS](#).

## Positionnement

Les diapositives sont définies dans des tailles fixes (par défaut `980x552px`) et à l'échelle pour s'adapter à l'écran de l'utilisateur. Vous pouvez utiliser en toute sécurité la position absolue dans vos diapositives car elles évolueront avec l'écran.

Par exemple :

```
<div class="absolute left-30px bottom-30px">
 Ceci est un pied de page aligné en bas à gauche
</div>
```

Pour changer la taille réelle du canvas, vous pouvez passer les options `canvasWidth` dans votre premier frontmatter :

```

 canvasWidth: 800

```

## Taille de police

Si vous pensez que la taille de la police de vos diapositives est trop petite, vous pouvez l'ajuster de plusieurs manières :

### Remplacer le style local

Vous pouvez remplacer les styles de chaque diapositive avec la balise `<style>` en ligne.

```
Page 1
<style>
 h1 {
 font-size: 10em;
 }
</style>

Page 2
Cela ne sera pas affecté.
```

En savoir plus: [Styles intégrés](#)

### Remplacer le style global

Vous pouvez fournir des styles globaux personnalisés en créant `./style.css`, par exemple

```
/* style.css */
h1 {
 font-size: 10em !important;
}
```

En savoir plus: [Style Global](#)

## Mettre à l'échelle le canvas

Changer la taille réelle du canvas mettra à l'échelle tout votre contenu (texte, images, composants, etc.) et les diapositives

```

defaut : 980
puisque la toile devient plus petite, la taille visuelle deviendra plus grande
canvasWidth: 800

```

## Utiliser la transformation

Nous fournissons un composant intégré `<Transform />`, qui est un wrapper de la propriété de transformation CSS.

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

# Hébergement statique

## Créer des applications monopage (SPA)

Vous pouvez également intégrer les diapositives dans une SPA autohébergée :

```
$ slidev build
```

L'application générée sera disponible sous `dist/` et vous pourrez ensuite l'héberger sur [Pages GitHub](#), [Netlify](#), [Vercel](#), ou ce que vous voulez. Vous pouvez maintenant partager vos diapositives avec le reste du monde avec un seul lien.

### Chemin de base

Pour déployer vos slides sous des sous-routes, vous devrez passer l'option `--base`. Par exemple :

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

Reportez-vous à la [documentation de Vite](#) pour plus de détails.

### Fournir un PDF téléchargeable

Vous pouvez fournir un PDF téléchargeable aux spectateurs de votre SPA avec la configuration suivante :

```

download: true

```

Slidev générera un fichier pdf avec la construction et un bouton de téléchargement sera affiché dans le SPA.

Vous pouvez également fournir une URL personnalisée au PDF. Dans ce cas, le processus de rendu sera ignoré.

```

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

```

## Exemples

Voici quelques exemples de SPA exportées :

- [Modèle de démarrage](#)
- [Vue composable par Anthony Fu](#)

Pour en savoir plus, consultez [Showcases](#).

# Hébergement

Nous vous recommandons d'utiliser `npm init slidev@lastest` pour échafauder votre projet, qui contient les fichiers de configuration nécessaires pour héberger les services prêts à l'emploi.

## Netlify

- [Netlify](#)

Créez `netlify.toml` dans la racine de votre projet avec le contenu suivant.

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

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

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

Ensuite, accédez à votre tableau de bord Netlify, créez un nouveau site avec le repository.

## Vercel

- [Vercel](#)

Créez `vercel.json` dans la racine de votre projet avec le contenu suivant.

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

Accédez ensuite à votre tableau de bord Vercel, créez un nouveau site avec le repository.

## GitHub Pages

- [GitHub Pages](#)

Pour déployer vos diapositives sur GitHub Pages :

- téléchargez tous les fichiers du projet dans votre repository (c'est-à-dire nommé `name_of_repo`)
- créez `.github/workflows/deploy.yml` avec le contenu suivant pour déployer vos diapositives sur GitHub Pages via GitHub Actions. Dans ce fichier, remplacez `<name_of_repo>` par `name_of_repo`.
- 

```
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: Install slidev
 run: npm i -g @slidev/cli
 - name: Build
 run: slidev build --base <name_of_repo>
 - name: Deploy pages
 uses: crazy-max/ghaction-github-pages@v2
 with:
 build_dir: dist
 env:
 GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
```

- Dans votre repository, allez dans Paramètres > Pages. Sous « Build and Deployment », sélectionnez « Deploy from a branch », sélectionnez « gh-pages » et « root ». Cliquez sur enregistrer.
- Enfin, une fois tous les flux de travail exécutés, un lien vers les diapositives doit apparaître sous Paramètres > Pages.

# Commencer

## Aperçu

Slidev (slide + dev, `/slайдив/`) est un créateur et présentateur de diapositives basé sur le Web. Il est conçu pour que les développeurs se concentrent sur l'écriture de contenu dans Markdown tout en ayant la puissance des composants HTML et Vue pour fournir des mises en page et des conceptions parfaites au pixel près avec des démos interactives intégrées dans vos présentations.

Il utilise un fichier de démarque riche en fonctionnalités pour générer de belles diapositives avec une expérience de recharge instantané, ainsi que de nombreuses intégrations intégrées telles que le codage en direct, l'exportation de PDF, l'enregistrement de présentations, etc. Comme il est alimenté par le Web, vous pouvez tout faire avec Slidev - les possibilités sont infinies.

Vous pouvez en savoir plus sur la raison d'être du projet dans la section [Pourquoi Slidev](#).

## Caractéristiques

- 📝 **basé sur Markdown** - utilisez vos éditeurs et flux de travail préférés
- 💻 **Convivial pour les développeurs** - coloration syntaxique intégrée, codage en direct, etc.
- 🎨 **Themable** - le thème peut être partagé et utilisé avec les packages npm
- 🌈 **Élégant** - utilitaires à la demande via [Windi CSS](#) ou [UnoCSS](<https://github.com/unocss/unocss>).
- 🎭 **Interactive** - intégration transparente des composants Vue
- 📱 **Mode Présentateur** - utilisez une autre fenêtre, ou même votre téléphone pour contrôler vos diapositives
- 🖌️ **Dessin** - dessinez et annotez sur vos diapositives
- TeX **LaTeX** - prise en charge intégrée des équations mathématiques LaTeX
- Diagram **Diagrams** - crée des diagrammes avec des descriptions textuelles
- Icons **Icons** - Accès direct aux icônes de n'importe quel jeu d'icônes
- Editors **Editors** - éditeur intégré, ou [extension pour VS Code](#)
- Recording **Enregistrement** - enregistrement intégré et vue caméra
- Portable **Portable** - exportez en PDF, PNG ou même un SPA hébergeable
- ⚡ **Fast** - recharge instantané alimenté par [Vite](#)
- Hackable **Hackable** - en utilisant les plugins Vite, les composants Vue ou tout package npm

## Pile technologique

Slidev est rendu possible en combinant ces outils et technologies.

- [Vite](#) - Un outil frontal extrêmement rapide
- [Vue 3](#) propulsé [Markdown](#) - Concentrez-vous sur le contenu tout en ayant la puissance des composants HTML et Vue chaque fois que nécessaire
- [Windi CSS](#) ou [UnoCSS](#) - Utilitaire à la demande-premier cadre CSS, stylisez vos diapositives à l'aise
- [Prism](#), [Shiki](#), [Monaco Editor](<https://github.com/Microsoft/monaco-editor>) - Prise en charge des extraits de code de première classe avec capacité de codage en direct

- [RecordRTC](#) - Enregistrement intégré et vue caméra
- Famille [VueUse](#) - `@vueuse/core`, [ `@vueuse/head` ](<https://github.com/vueuse/head>), `@vueuse/motion`, etc.
- [Iconify](#) (<https://iconify.design/>) - Collection d'ensembles d'icônes.
- [Drauu](#) - Prise en charge du dessin et des annotations
- [KaTeX](#) - Rendu mathématique LaTeX.
- [\[Mermaid\]](#) (<https://mermaid-js.github.io/mermaid>) - Diagrammes textuels.

## Création de votre première présentation

### Essayez en ligne

[sli.dev/new](#)

 Open in StackBlitz

### Créer localement

Avec NPM:

```
$ npm init slidev
```

Avec Yarn :

```
$ yarn create slidev
```

Suivez les instructions et commencez à créer vos diapositives maintenant ! Pour plus de détails sur la syntaxe de démarquage, lisez le [guide de syntaxe](#).

## Interface de ligne de commande

Dans un projet où Slidev est installé, vous pouvez utiliser le binaire `slidev` dans vos scripts npm.

```
{
 "scripts": {
 "dev": "slidev", // lancer le serveur de développement
 "build": "slidev build", // construire pour la production SPA
 "export": "slidev export" // exporter les diapositives en pdf
 }
}
```

Sinon, vous pouvez l'utiliser avec `npx`

```
$ npx slidev
```

Exécutez `slidev --help` pour plus d'options disponibles.

## Syntaxe de Markdown

Slidev lit votre fichier `slides.md` sous la racine de votre projet et les convertit en diapositives. Chaque fois que vous y apportez des modifications, le contenu des diapositives est mis à jour immédiatement. Par exemple :

```
Slidev
Hello World

Page 2
Utilisez directement des blocs de code pour la mise en évidence
//```ts
console.log('Hello, World!')
//```

Page 3
```

En savoir plus sur la syntaxe Slidev Markdown dans le [guide de syntaxe](#).

# Installation

## Modèle de démarrage

Slidev nécessite **Node.js > = 14.0**

La meilleure façon de commencer est d'utiliser notre modèle de démarrage officiel.

Avec NPM :

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

Avec Yarn :

```
$ yarn create slidev
```

Suivez les invitations et il ouvrira le diaporama à <http://localhost:3030/> automatiquement pour vous.

Il contient également la configuration de base et une courte démo avec des instructions sur la façon de démarer avec Slidev.

## Installer manuellement

Si vous préférez toujours installer Slidev manuellement ou souhaitez l'intégrer dans vos projets existants, vous pouvez faire :

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

```
$ touch slides.md
```

```
$ npx slidev
```

Veuillez noter que si vous utilisez **pnpm**, vous devrez activer l'option **shamefully-hoist** pour que Slidev fonctionne correctement :

```
echo 'shamefully-hoist=true' >> .npmrc
```

## Installer globalement

Vous pouvez installer SLidev globallement avec la commande suivante :

Disponible depuis v0.14

Vous pouvez installer Slidev globalement avec la commande suivante

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

Et puis utilisez `slidev` partout sans créer de projet à chaque fois.

```
$ slidev
```

Cette commande essaiera également d'utiliser local `@slidev/cli` s'il a été trouvé dans les `node_modules`.

## Installer sur Docker

Si vous avez besoin d'un moyen rapide d'exécuter une présentation avec des conteneurs, vous pouvez utiliser l'image prédefinie `docker` maintenue par [tangramor](#), ou créez le vôtre.

Exécutez simplement la commande suivante dans votre dossier de travail :

```
docker run --name slidev --rm -it \
--user node \
-v ${PWD}:/slidev \
-p 3030:3030 \
tangramor/slidev:latest
```

Si votre dossier de travail est vide, il générera un modèle `slides.md` et d'autres fichiers associés sous votre dossier de travail, et lancera le serveur sur le port `3030`.

Vous pouvez accéder à vos diapositives à partir de `http://localhost:3030/`

## Créer une image déployable

Ou vous pouvez créer votre propre projet slidev sur une image docker avec Dockerfile :

```
FROM tangramor/slidev:latest
ADD . /slidev
```

Créez l'image Docker : `docker build -t myppt .`

Et lancez le conteneur : `docker run --name myslides --rm --user node -p 3030:3030 myppt`

Vous pouvez visiter vos diapositives à partir de `http://localhost:3030/`

## Construire une SPA hébergeable (application monopage)

Exécutez la commande `docker exec -i slidev npx slidev build` sur le conteneur en cours d'exécution `slidev`. Il générera des fichiers HTML statiques sous le dossier `dist`.

## Héberger sur Github Page

Vous pouvez héberger `dist` dans un site Web statique tel que [Github Pages](#) ou des pages Gitlab.

Parce que dans les pages Github, l'URL peut contenir un sous-dossier, vous devez donc modifier le `index.html` généré pour changer `href="/assets/xxx` en `href=".assets/xxx"`. Ou vous pouvez utiliser l'option `--base=/<sous-dossier>/` pendant le processus de construction, comme : `docker exec -i slidev npx slidev build --base=slidev_docker/`.

Et pour éviter le processus de construction de Jekyll, vous devez ajouter un fichier vide `.nojekyll`.

## Hébergé par docker

Vous pouvez également l'héberger par vous-même avec docker :

```
docker run --name myslides --rm -p 80:80 -v ${PWD}/dist:/usr/share/nginx/html
nginx:alpine
```

Ou créez une image statique avec le Dockerfile suivant :

```
FROM nginx:alpine
COPY dist /usr/share/nginx/html
```

Créez l'image docker : `docker build -t mystaticppt .`

Et exécutez le conteneur : `docker run --name myslides --rm -p 80:80 mystaticppt`

Vous pouvez visiter vos diapositives à partir de `http://localhost/`

Reportez-vous au [tangramor/slides\\_docker](#) pour plus de détails.

## Command Line Interface (CLI)

`@slidev/cli` Exposez quelques commandes que vous pouvez utiliser avec `npx slidev ...` ou en ajoutant des scripts dans votre `package.json` :

```
{
 "script": {
 "dev": "slidev"
 }
}
```

Dans ce cas, vous pourrez exécuter `npm run dev`.

Vous pouvez passer des options à n'importe quelle commande :

- les options booléennes sont `true` si elles sont présentes, `false` sinon (exemple : `slidev --open`)
- certaines options peuvent avoir des valeurs que vous pouvez ajouter juste après l'option ou en utilisant le caractère `=` (exemple : `slidev --port 8080` ou `slidev --port=8080`)

Si vous utilisez des scripts npm, n'oubliez pas d'ajouter `--` après la commande npm :

```
npm run slidev -- --open
```

## slidev [entry]

Démarrez un serveur local pour Slidev.

- `[entry] ( string , défaut : slides.md )`: chemin d'accès à l'entrée des diapositives markdown.

Options:

- `--port , -p ( number , défaut : 3030 )`: numéro de port.
- `--open , -o ( boolean , défaut : false )`: ouvrir dans le navigateur.
- `--remote [password] ( string )`: écouter l'hôte public et activer le contrôle à distance, si une valeur est transmise, le mode présentateur est privé et accessible uniquement en transmettant le mot de passe donné dans le paramètre `password` de la requête d'URL.
- `--log ( 'error' , 'warn' , 'info' , 'silent' , défaut : 'warn' )`: niveau de log.
- `--force , -f ( boolean , défaut false )`: forcer l'optimiseur à ignorer le cache et à regrouper.
- `--theme , -t ( string )`: thème de remplacement.

## slidev build [entry]

Construire une SPA hébergeable.

- `[entry] ( string , défaut : slides.md )`: chemin d'accès à l'entrée des diapositives markdown.

Options:

- `--watch , -w ( boolean , défaut : false )`: build watch.
- `--out , -o ( string , défaut : dist )`: rép. de sortie
- `--base ( string , défaut : / )`: URL de base (voir <https://cli.vuejs.org/config/#publicpath>)
- `--download ( boolean , défaut : false )`: permettent de télécharger les diapositives au format PDF à l'intérieur du SPA.
- `--theme , -t ( string )`: thème de remplacement.

## slidev export [entry]

Exportez les diapositives au format PDF (ou autre format).

- `[entry] ( string , défaut : slides.md )`: chemin d'accès à l'entrée des diapositives markdown.

Options:

- `--output ( string , défaut : use exportFilename (voir https://sli.dev/custom/#frontmatter-configures) ou utiliser [entry]-export )`: chemin vers la sortie.
- `--base ( 'pdf' , 'png' , 'md' , défaut : 'pdf' )`: format de sortie.
- `--timeout ( number , défaut : 30000 )`: délai d'attente pour le rendu de la page d'impression (voir <https://playwright.dev/docs/api/class-page#page-goto>).
- `--range ( string )`: plages de pages à exporter (exemple : `'1,4-5,6'` ).
- `--dark ( boolean , défaut : false )`: exporter en tant que thème sombre.

- `--with-clicks`, `-c` (`boolean`, défaut : `false`): exporter des pages pour chaque clic (voir <https://sli.dev/guide/animations.html#click-animations>).
- `--theme`, `-t` (`string`): thème de remplacement.

## **slidev format [entry]**

Formatez le fichier Markdown.

- `[entry]` (`string`, défaut : `slides.md`): chemin d'accès à l'entrée des diapositives markdown.

## **slidev theme [subcommand]**

Opérations liées au thème.

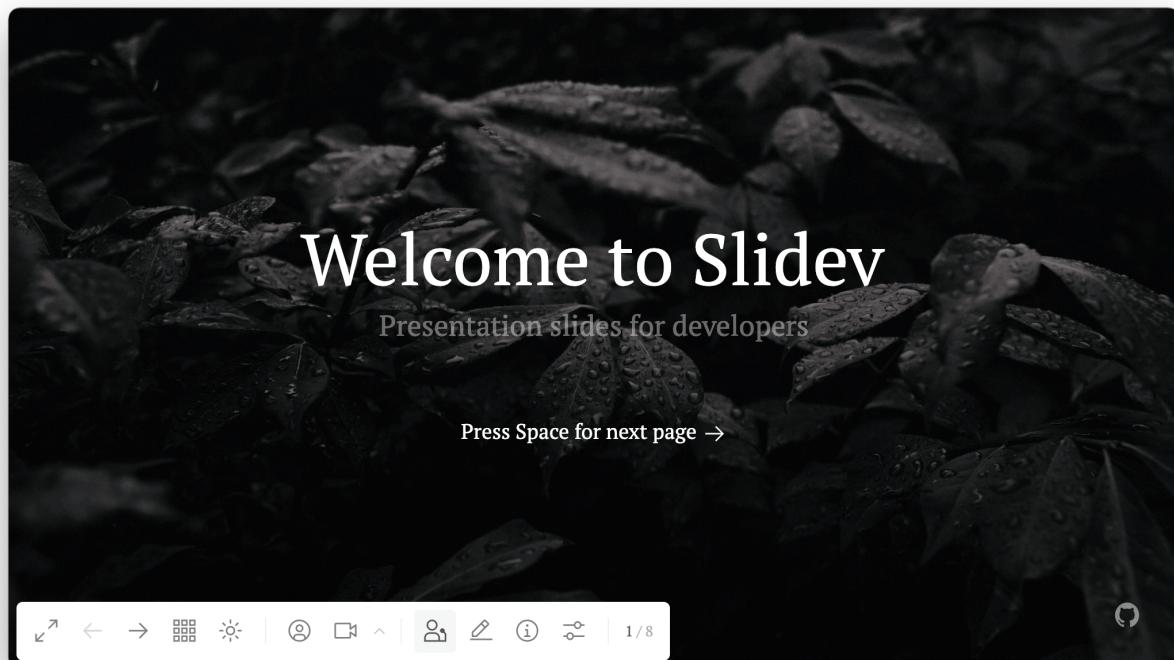
Sous-commandes :

- `eject [entry]` : Éjecter le thème actuel dans le système de fichiers local
  - `[entry]` (`string`, défaut : `slides.md`): chemin d'accès à l'entrée des diapositives markdown.
  - Options: `--dir` (`string`, défaut : `theme`): rép. de sortie `--theme`, `-t` (`string`): thème de remplacement.

# Navigation

## Barre de navigation

Déplacez votre souris vers le coin inférieur gauche de la page Slidev, la barre de navigation apparaîtra.

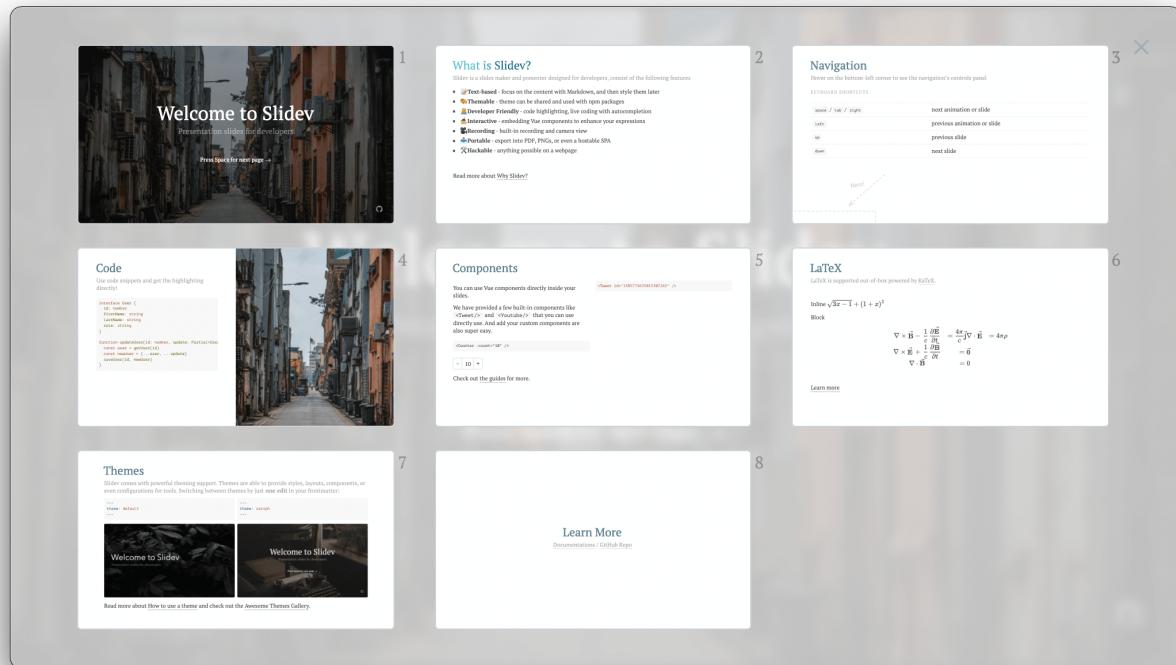


Raccourcis	Bouton	Description
f		basculer en plein écran
droite / espace		prochaine animation ou diapositive
gauche		animation ou diapositive précédente
haut	-	diapositive précédente
bas	-	diapositive suivante
o		bascule <a href="#">présentation des diapositives</a>
d		basculer le mode sombre
-		basculer <a href="#">vue caméra</a>
-		<a href="#">enregistrement</a>
-		entrer <a href="#">mode présentateur</a>
-		toggle <a href="#">éditeurs intégrés</a>

Raccourcis	Bouton	Description
-		télécharger les diapositives (n'apparaissent que dans <a href="#">SPA build</a> )
-		afficher des informations sur les diapositives
-		afficher le menu des paramètres
g	-	montrer <code>goto</code> ...

## Présentation des diapositives

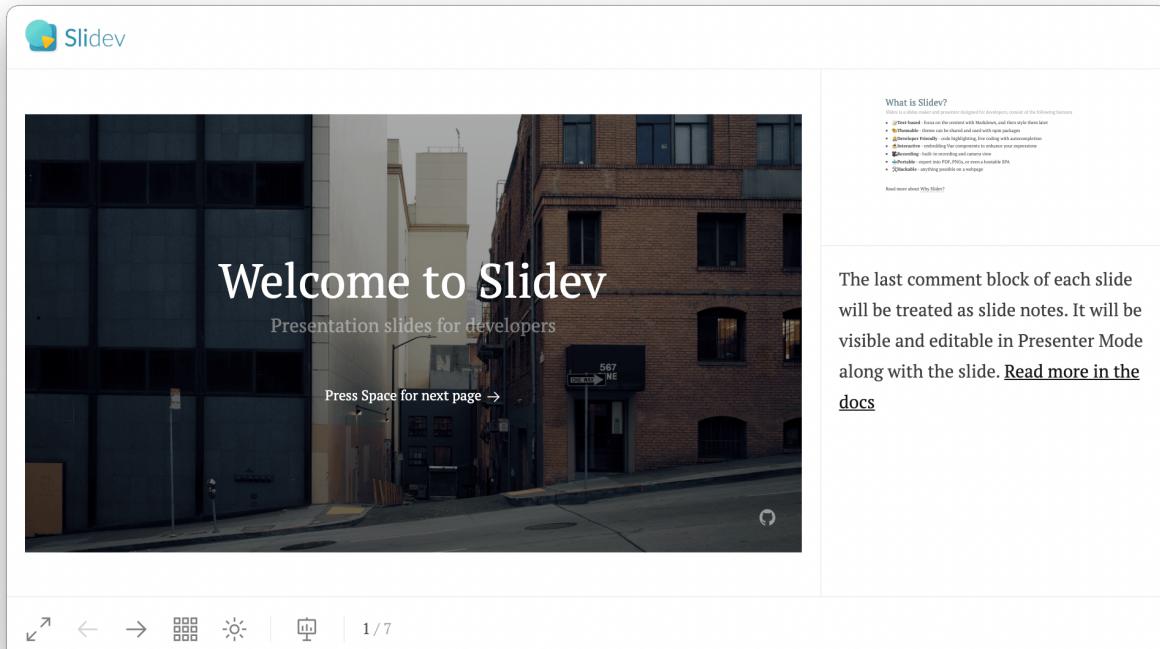
En appuyant sur `o` ou en cliquant sur le bouton dans la barre de navigation, vous pouvez avoir un aperçu de vos diapositives afin de pouvoir passer de l'une à l'autre facilement.



[Go to TOC](#)

# Mode présentateur

Cliquez sur le bouton dans le panneau de navigation ou visitez manuellement <http://localhost:3030/presenter> pour accéder au mode présentateur. Chaque fois que vous entrez en mode présentateur, les autres instances de page resteront automatiquement synchronisées avec le présentateur.



---

[Go to TOC](#)

# Enregistrement

Slidev a un enregistrement intégré et une vue de la caméra. Vous pouvez les utiliser pour enregistrer facilement votre présentation en un seul endroit.

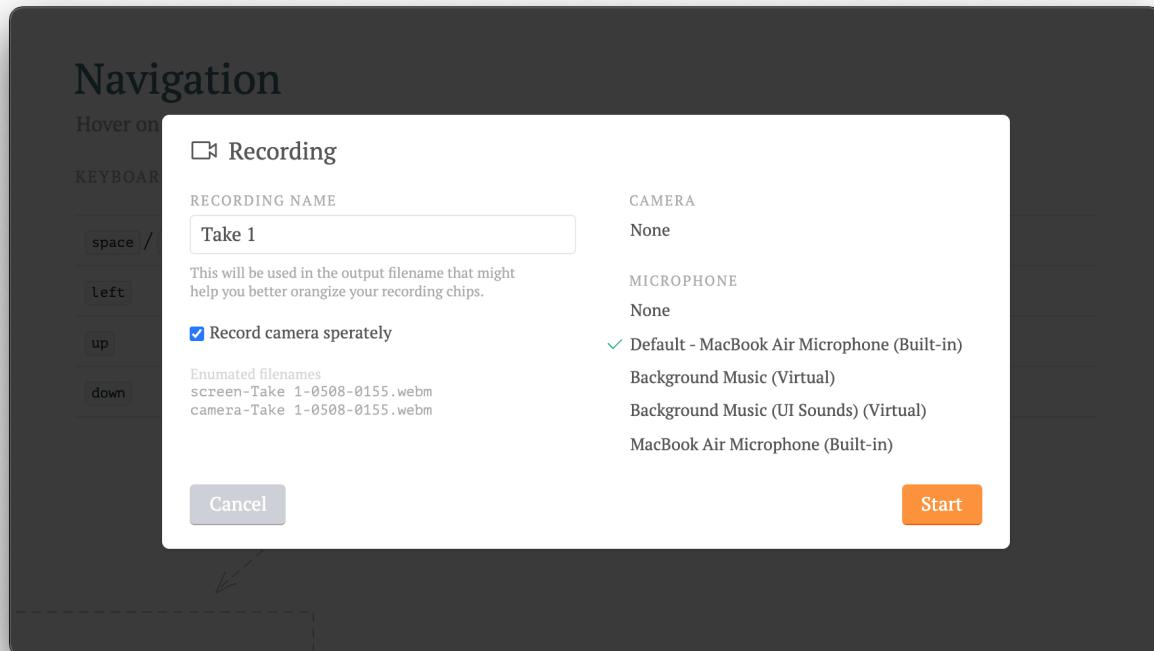
## Vue de la caméra

Cliquez sur le bouton dans le panneau de navigation pour afficher la vue de votre caméra dans la présentation. Vous pouvez faire glisser pour le déplacer et utiliser le gestionnaire dans le coin inférieur droit pour le redimensionner. La taille et la position persisteront dans `localStorage` et seront donc cohérentes sur plusieurs actualisations, donc pas besoin de s'inquiéter à ce sujet.

## Enregistrement

Cliquez sur le bouton dans le panneau de navigation pour ouvrir une boîte de dialogue. Ici, vous pouvez choisir d'enregistrer votre caméra intégrée dans vos diapositives ou de les séparer en deux fichiers vidéo.

Cette fonctionnalité est alimentée par [RecordRTC](#) et utilise l'[API WebRTC](#).




---

[Go to TOC](#)

# Syntaxe de Markdown

Les diapositives sont écrites dans **un seul fichier markdown** (par défaut `./slides.md`).

Vous pouvez utiliser [les fonctionnalités Markdown](#) comme vous le feriez normalement, avec le support supplémentaire du HTML intégré et des composants Vue. Le style à l'aide de [Windi CSS](#) est également pris en charge. Utilisez `---` avec une nouvelle ligne pour séparer vos diapositives.

```
Sliddev

Hello, World!

Page 2

Utilisez directement des blocs de code pour la mise en évidence

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

Page 3

Vous pouvez directement utiliser les composants Windi CSS et Vue pour styliser et enrichir vos diapositives.

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

## Front Matter & Layouts

Vous pouvez spécifier des mises en page et d'autres métadonnées pour chaque diapositive en convertissant les séparateurs en [blocs de présentation](#). Chaque avant-propos commence par un triple tiret et se termine par un autre. Les textes entre eux sont des objets de données au format [YAML](#). Par exemple :

```

layout: cover

Sliddev

Ceci est la page de garde.

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

Page 2
```

Ceci est une page avec la mise en page «centre» et une image d'arrière-plan.

---

# Page 3

Il s'agit d'une page par défaut sans métadonnées supplémentaires.

Reportez-vous à [personnalisations](#) pour plus de détails.

## Blocs de code

L'une des principales raisons pour lesquelles je construis Slidev est de faire en sorte que mon code apparaisse parfaitement dans les diapositives. Donc, comme vous vous y attendiez, vous pouvez utiliser le bloc de code aromatisé Markdown pour mettre en évidence votre code.

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

Nous supportons [Prism](#) et [Shiki](#) en tant que surligneurs de syntaxe. Se référer à [la section Surligneurs](#) pour plus de détails.

## Mise en évidence de ligne

Pour mettre en évidence des lignes spécifiques, ajoutez simplement des numéros de ligne entre crochets `{}`. Les numéros de ligne commencent à compter de 1.

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

Pour changer la surbrillance en plusieurs étapes, vous pouvez utiliser `|` pour les séparer. Par exemple

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

Cela mettra d'abord en évidence `a: Ref <number> | number` et `b: Ref <number> | number`, puis `return computed () => unref (a) + unref (b)` après un clic, et enfin, tout le bloc. Pour en savoir plus, consultez le [guide des animations de clics](#).

Pour ignorer la mise en surbrillance des lignes, vous pouvez définir le numéro de ligne sur "0". Par exemple

```
//``ts {0}
function add(
 a: Ref<number> | number,
 b: Ref<number> | number
) {
 return computed(() => unref(a) + unref(b))
}
//``
```

Si le code ne rentre pas dans une diapositive, vous pouvez passer une option maxHeight supplémentaire qui définira une hauteur fixe et activer le défilement

```
//``ts {2|3|7|12} {maxHeight:'100'}
function add(
 a: Ref<number> | number,
 b: Ref<number> | number
) {
 return computed(() => unref(a) + unref(b))
}
/// ...as many lines as you want
const c = add(1, 2)
//``
```

## Éditeur Monaco

Chaque fois que vous souhaitez apporter des modifications à la présentation, ajoutez simplement `{monaco}` après l'identifiant de la langue - cela transforme le bloc en un éditeur Monaco complet!

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

En savoir plus sur la [configuration de Monaco](#).

## Styles intégrés

Vous pouvez utiliser la balise `<style>` dans votre Markdown directement pour remplacer les styles de la **diapositive actuelle**.

```
C'est rouge

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

La prochaine diapositive n'est pas affectée
```

La balise `<style>` dans Markdown est toujours **scoped**. Pour avoir des remplacements de style globaux, consultez la [section de personnalisation](#)

Propulsé par Windi CSS, vous pouvez utiliser directement des css imbriqués et des réertoires (par exemple, `@apply`)

```
Slidev

> Hello `world`

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

## Assets statiques

Tout comme vous le feriez dans Markdown, vous pouvez utiliser des images pointant vers une URL distante ou locale.

Pour les assets statiques, le `vite-plugin-remote-assets` intégré les mettra en cache sur le disque à la première exécution afin que vous puissiez avoir un chargement instantané même pour de grandes images plus tard.

```
![Image distante](https://sli.dev/favicon.png)
```

Pour les ressources locales, placez-les dans le dossier `public` et référez-les avec **la barre oblique principale**.

```
![Image locale](pic.png)
```

Si vous souhaitez appliquer des tailles ou des styles personnalisés, vous pouvez les convertir en balise `<img>`

```

```

## Remarques

Vous pouvez également prendre des notes pour chaque diapositive. Ils apparaîtront en Mode Présentateur pour que vous puissiez vous y référer pendant les présentations.

Dans Markdown, le dernier bloc de commentaire de chaque diapositive sera traité comme une note.

```

layout: cover

Page 1

Ceci est la page de garde.

<!-- Ceci est une note --></pre>

```

```

Page 2

<!-- Ce n'est PAS une note car elle précède le contenu de la diapositive -->
La deuxième page

<!--
Ceci est une autre note
-->
```

## Icônes

Slidev vous permet d'accéder à presque tous les jeux d'icônes open-source populaires **directement** dans votre markdown. Propulsé par `unplugin-icons` et `Iconify`.

Le nom suit la conversion de `Iconify {collection-name}-{icon-name}`. Par exemple:

- `<mdi-account-circle />` - de [Material Design Icons](#)
- `<carbon-badge />` - de [Carbon](#)
- `<uim-rocket />` - de [Unicons Monochrome](#)
- `<twemoji-chat-avec-des-larmes-de-joie />` - de [Twemoji](#)
- `<logos-vue />` - de [Logos SVG](#)
- Et beaucoup plus...

Vous pouvez parcourir et rechercher toutes les icônes disponibles avec [Icônes](#).

## Icônes de style

Vous pouvez styliser les icônes comme les autres éléments HTML. Par exemple :

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

## Emplacements

Disponible depuis v0.18

Certaines mises en page peuvent fournir de multiples points de contributions utilisant [Les emplacements nommés de Vue](#).

Par exemple, dans la mise en page `two-cols`, vous pouvez avoir deux colonnes à gauche (emplacement `par défaut`) et à droite (emplacement de `droite`) côté à côté.

```

layout: two-cols

```

```
<template v-slot:default>
Gauche
Cela apparaît sur la gauche
</template>
<template v-slot:right>
Droite
Cela apparaît sur la droite
</template>
```

## Gauche

Cela apparaît sur la gauche

## Droite

Cela apparaît sur la droite

Nous fournissons également une syntaxe abrégée `::name::` pour le nom de l'emplacement. L'exemple suivant fonctionne exactement de la même manière que le précédent.

```

layout: two-cols

Gauche
Cela apparaît sur la gauche
::right::

Droite
Cela apparaît sur la droite
```

Vous pouvez également spécifier explicitement l'emplacement par défaut et fournir dans un ordre personnalisé

```

layout: two-cols

::right::
Droite
Cela apparaît sur la droite
::default::
Gauche
Cela apparaît sur la gauche
```

# Configurations

Toutes les configurations nécessaires peuvent être définies dans le fichier Markdown. Par exemple :

```

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

Slidelv

Ceci est la page de garde.
```

En savoir plus sur les [configurations frontmatter](#)

# LaTeX

Slidev est livré avec le support LaTeX prêt à l'emploi, alimenté par [KaTeX](#).

## Rendu en ligne

Entourez votre LaTeX d'un seul `$` de chaque côté pour un rendu en ligne.

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

## Block

Utilisez deux (`$$`) pour le rendu en bloc. Ce mode utilise des symboles plus grands et centre le résultat.

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

En savoir plus: [Démo](#) | [KaTeX](#) | [markdown-it-katex](#)

# Diagrammes

Vous pouvez également créer des diagrammes / graphiques à partir de descriptions textuelles dans votre Markdown, alimenté par [Mermaid](#).

Les blocs de code marqués comme `mermaid` seront convertis en digrammes, par exemple :

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

```

Vous pouvez en outre lui transmettre un objet d'options pour spécifier la mise à l'échelle et la thématisation. La syntaxe de l'objet est un littéral d'objet JavaScript, vous devrez ajouter des guillemets (` '') pour les chaînes et utiliser une virgule ( , ) entre les clés.

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

```

En savoir plus: [Démo | Mermaid](#)

## Entrées multiples

Depuis la version v0.15.0, nous avons livré le support multi-entrées. Cela signifie que vous pouvez diviser votre `slides.md` en plusieurs fichiers and les organiser comme vous le souhaitez.

Disponible depuis v0.15

Vous pouvez diviser votre `slides.md` en plusieurs fichiers et les organiser comme vous le souhaitez.

`slides.md` :

```
Page 1
Ceci est une page normale

src: ./sous-page2.md

<!-- Cette page sera chargée depuis './subpage2.md' -->
Le contenu en ligne sera ignoré

```

`sous-page2.md` :

```
Page 2
Cette page provient d'un autre fichier

```

## Fusion de frontmatter

Vous pouvez fournir des frontmatters depuis l'entrée principale mais également depuis une page markdown externe. S'il y a les mêmes clés, celles de l'**entrée principale ont la priorité la plus élevée**. Par exemple

`slides.md` :

```

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

```

`cover.md` :

```

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

Couverture

Page de couverture
```

Ils finiront par être équivalents à la page suivante :

```

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

Couverture

Page de couverture
```

## Réutilisation de la page

Avec la prise en charge des entrées multiples, la réutilisation des pages peut être simple. Par exemple :

```

src: ./cover.md

src: ./intro.md

src: ./content.md

réutilisation
src: ./content.md

```

---

[Go to TOC](#)

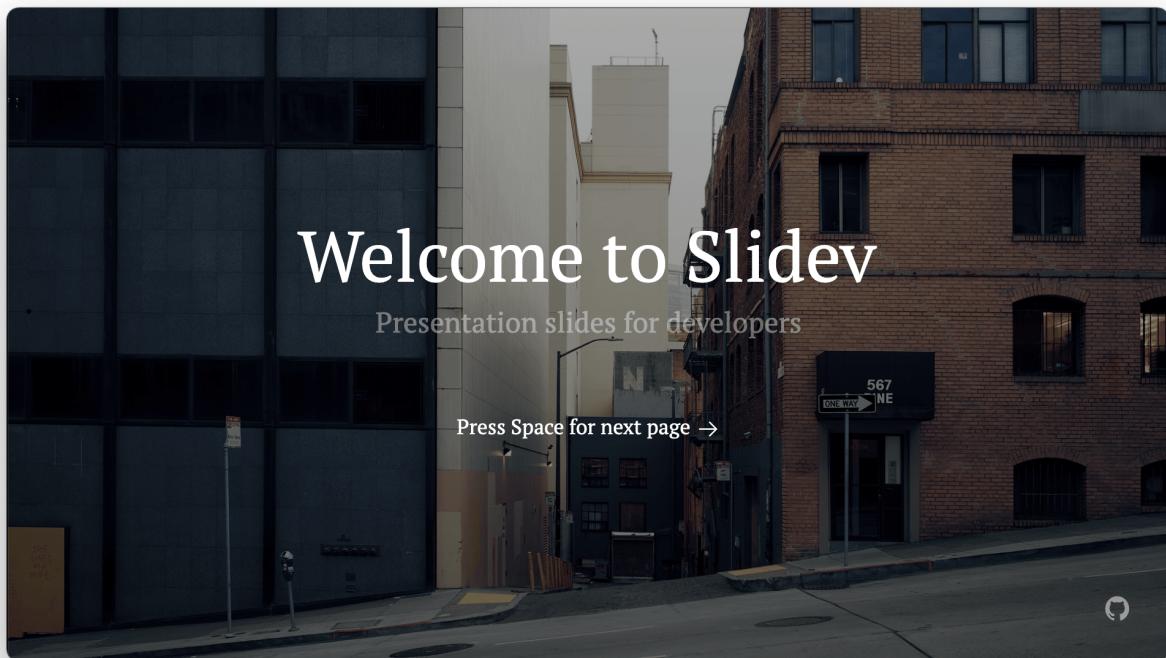
# Pourquoi Slidev

Il existe de nombreux fabricants de diapositives WYSIWYG polyvalents et riches en fonctionnalités, tels que [Microsoft PowerPoint](#) et [Apple Keynote](#). Ils fonctionnent plutôt bien pour créer de belles diapositives avec des animations, des graphiques et bien d'autres choses, tout en étant très intuitifs et faciles à apprendre. Alors pourquoi prendre la peine de faire Slidev?

Slidev vise à fournir la flexibilité et l'interactivité aux développeurs pour rendre leurs présentations encore plus intéressantes, expressives et attrayantes en utilisant les outils et les technologies avec lesquels ils sont déjà familiers.

Lorsque vous travaillez avec des éditeurs WYSIWYG, il est facile de se laisser distraire par les options de style. Slidev remédie à cela en séparant le contenu et les éléments visuels. Cela vous permet de vous concentrer sur une chose à la fois, tout en pouvant réutiliser les thèmes de la communauté. Slidev ne cherche pas à remplacer entièrement les autres constructeurs de diapositives. Il se concentre plutôt sur la restauration de la communauté des développeurs.

## Slidev



Voici quelques-unes des fonctionnalités les plus intéressantes de Slidev :

## Basé sur Markdown

Slidev utilise un format Markdown étendu pour stocker et organiser vos diapositives dans un seul fichier de texte brut. Cela vous permet de vous concentrer sur la création du contenu. Et comme le contenu et les styles sont séparés, cela a également permis de basculer entre différents thèmes sans effort.

En savoir plus sur la [Syntaxe Markdown de Slidev](#).

## Thème

Les thèmes pour Slidev peuvent être partagés et installés à l'aide des packages npm. Vous les appliquez ensuite avec une seule ligne de configuration.

Consultez la [galerie de thèmes](#) ou [apprenez à écrire un thème](#).

## Adapté aux développeurs

Slidev fournit un support de première classe pour les extraits de code pour les développeurs. Il prend en charge à la fois [Prism](#) et [Shiki](#) pour obtenir une coloration syntaxique parfaite des pixels, tout en étant capable de modifier le code à tout moment temps. Avec [Monaco editor](#) intégré, il vous permet également de faire du codage / démonstration en direct dans votre présentation avec la saisie semi-automatique, le survol de type et même la prise en charge de la vérification de type TypeScript.

En savoir plus sur [surligneurs](#) et [configuration Monaco](#).

## Rapide

Slidev est alimenté par [Vite](#), [Vue 3](#) et [Windi CSS](#), qui vous offrent la plus merveilleuse expérience de création. Chaque modification que vous apportez sera répercutée sur vos diapositives **instantanément**.

En savoir plus sur [notre pile technologique](#)

## Interactif et expressif

Vous pouvez écrire des composants Vue personnalisés et les utiliser directement dans votre fichier markdown. Vous pouvez également interagir avec eux dans la présentation pour exprimer votre idée de manière plus intéressante et intuitive.

## Prise en charge de l'enregistrement

Slidev fournit un enregistrement intégré et une vue de la caméra. Vous pouvez partager votre présentation avec la vue de votre caméra à l'intérieur, ou les enregistrer et les enregistrer séparément pour votre écran et votre caméra. Tout cela d'un seul coup, aucun outil supplémentaire n'est nécessaire.

En savoir plus sur [enregistrement ici](#).

## Portable

Exportez vos diapositives au format PDF, PNG ou même dans une application monopage hébergée (SPA) avec une seule commande et partagez-les n'importe où.

Apprenez-en plus à ce sujet dans la [documentation d'exportation](#).

## Hackable

Étant alimenté par les technologies Web, tout ce qui peut être fait dans une application Web est également possible avec Slidev. Par exemple, WebGL, requêtes API, iframes ou même partage en direct. C'est à votre imagination!

## Essaie

Jouer avec Slidev vous en dira plus que mille mots. Vous n'êtes qu'à une seule commande :

```
$ npm init slidev
```

Ou en avoir un aperçu rapide :

Slidev First Preview Demo

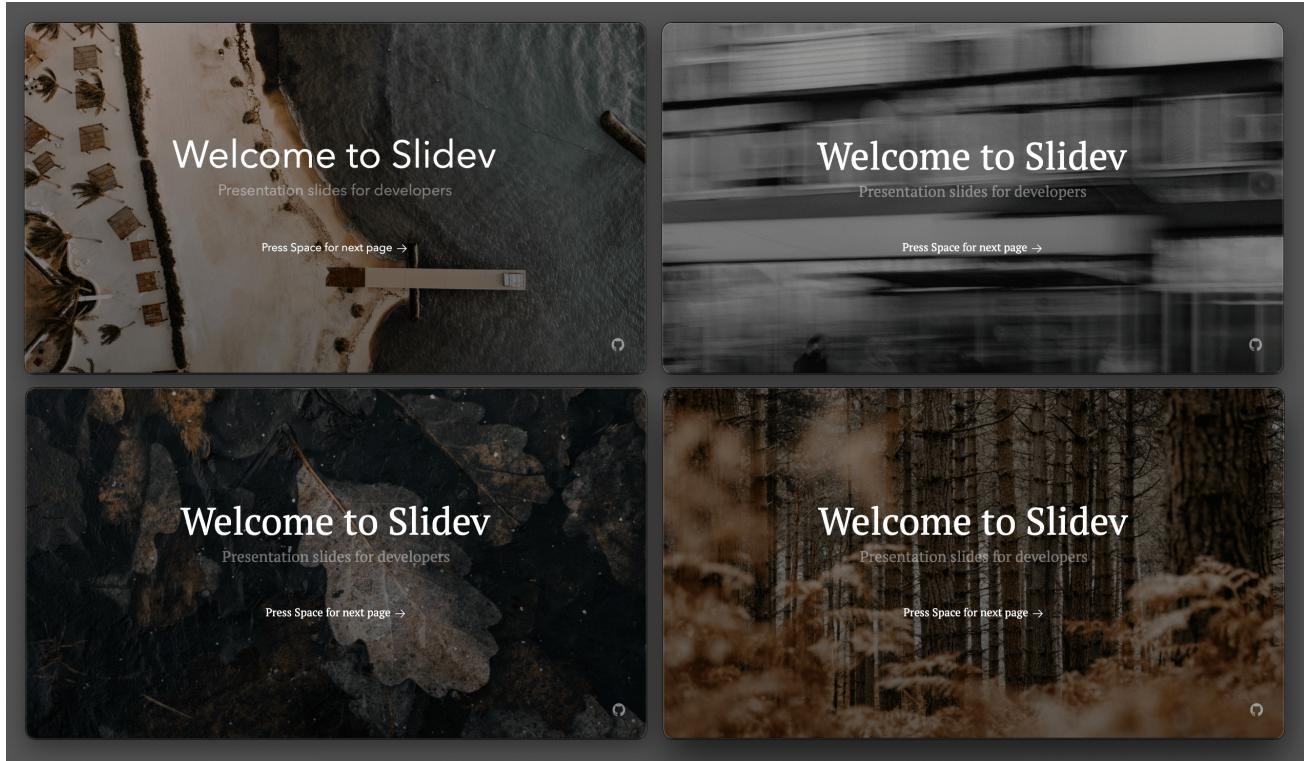


---

[Go to TOC](#)

# Couvertures organisées

Nous avons organisé quelques images de couverture pour démontrer notre modèle de démarrage.



```

image aléatoire de la collection organisée
background: https://source.unsplash.com/collection/94734566/1920x1080

```

Si vous appréciez l'un d'entre eux, consultez notre [collection Unsplash](#) et découvrez leurs auteurs.

# 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

## 日本語

- [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

Talks / Presentations using Slidev.

---

[Go to TOC](#)

# Galerie de thèmes

Parcourez les thèmes géniaux disponibles pour Slidev ici.

En savoir plus sur [comment utiliser un thème](#), ou [comment écrire le vôtre](#) et partagez avec la communauté !

## Thèmes officiels

## Thèmes de la communauté

Voici les thèmes sélectionnés par la communauté.

## Plus de thèmes

Retrouvez tous les [thèmes disponibles sur NPM](#).

---

[Go to TOC](#)

# Utiliser le thème

Changer le thème dans Slidev est étonnamment facile. Tout ce que vous avez à faire est d'ajouter le champ `theme:` dans votre frontmatter.

```

theme: serif

```

Vous pouvez démarrer le serveur, ce qui vous demandera d'installer le thème automatiquement

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

ou installez le thème manuellement via

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

Et c'est tout, profitez du nouveau thème! Pour plus de détails sur l'utilisation, vous pouvez vous référer au fichier README du thème.

Envie de partager votre thème? Découvrez [comment écrire un thème](#).

## Ejecter le thème

Si vous souhaitez obtenir le contrôle total du thème actuel, vous pouvez **l'éjecter** sur votre système de fichiers local et le modifier à votre guise. En exécutant la commande suivante

```
$ slidev theme eject
```

Il éjectera le thème que vous utilisez actuellement dans `./theme`, et changera votre frontmatter en

```

theme: ./theme

```

Cela peut également être utile si vous souhaitez créer un thème basé sur un thème existant. Si vous le faites, n'oubliez pas de mentionner le thème d'origine et l'auteur :)

## Thème local

Comme vous l'avez probablement découvert dans la section précédente, vous pouvez avoir un thème local pour votre projet. En ayant un **chemin relatif** dans votre champ de thème.

```

theme: ./path/to/theme

```

Reportez-vous à [comment écrire un thème](#) pour plus de détails.

---

[Go to TOC](#)

# Ecrire un thème

Pour commencer, nous vous recommandons d'utiliser notre générateur pour créer votre premier thème

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

Ensuite, vous pouvez le modifier et jouer avec. Vous pouvez également vous référer aux [thèmes officiels](#) comme exemples.

## Aptitude

Un thème peut contribuer aux points suivants :

- Styles globaux
- Fournir des configurations par défaut (police, schéma de couleurs, surligneurs, etc.)
- Fournir des mises en page personnalisées ou remplacer celle existante
- Fournir des composants personnalisés ou remplacer l'existant
- Étendre les configurations de WindiCSS
- Configurer des outils comme Monaco et Prism

## Conventions

Les thèmes sont publiés dans le registre npm et doivent respecter les conventions ci-dessous :

- Le nom du package doit commencer par `slidev-theme-`, par exemple : `slidev-theme-awesome`
- Ajoutez `slidev-theme` et `slidev` dans le champ `keywords` de votre `package.json`

## Installer

Pour configurer le terrain de jeu de test pour votre thème, vous pouvez créer `example.md` avec le front-matter suivant, pour dire à Slidev que vous utilisez le répertoire actuel comme thème.

```

theme: ./

```

En option, vous pouvez également ajouter des scripts à votre `package.json`

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

Pour publier votre thème, lancez simplement `npm publish` et vous êtes prêt à partir. Il n'y a pas de processus de construction requis (ce qui signifie que vous pouvez publier directement les fichiers `.vue` et `.ts`, Slidev est assez intelligent pour les comprendre).

Les points de contribution de thème suivent les mêmes conventions que la personnalisation locale, veuillez vous référer à [la documentation pour les conventions](#).

## Configurations par défaut

Disponible depuis v0.19

Un thème peut fournir des [configurations par défaut](#) via `package.json`

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

Les polices seront importées automatiquement depuis [Google Fonts](#).

En savoir plus sur les [polices](#) et sur les [configurations frontmatter](#)

## Métadonnées du thème

### Color Schema

Par défaut, Slidev suppose que les thèmes prennent en charge à la fois le mode clair et le mode sombre. Si vous voulez seulement que votre thème soit présenté dans un schéma de couleurs conçu, vous devrez le spécifier explicitement dans `package.json`

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

Pour accéder au mode sombre lors de la création de vos styles de thème, vous pouvez envelopper le css spécifique au mode sombre dans une classe `dark` :

```
/* css general ici */

html:not(.dark) {
 /* css mode clair ici */
}

html.dark {
 /* css mode sombre ici */
}
```

Slidev bascule une classe `dark` sur l'élément `html` de la page pour changer de schéma de couleur.

## Surlieur

Les couleurs de mise en évidence de la syntaxe sont également fournies dans le thème. Nous prenons en charge à la fois [Prism](#) et [Shiki](#). Pour plus d'informations, veuillez consulter [la documentation sur la coloration syntaxique](#).

Vous pouvez prendre en charge l'un d'entre eux, ou les deux. Reportez-vous au thème par défaut pour des exemples de configurations `./styles/code.css` / `./setup/shiki.ts`.

N'oubliez pas non plus de spécifier les surligneurs pris en charge dans votre `package.json`

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

## Version de Slidev

Si le thème repose sur une fonctionnalité spécifique de Slidev récemment introduite, vous pouvez définir la version minimale de Slidev requise pour que votre thème fonctionne correctement :

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

Si les utilisateurs utilisent des versions plus anciennes de Slidev, une erreur sera générée.

# Colophon

This book is created by using the following sources:

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