

# Welcome to Slidev

Presentation slides for developers

Press Space for next page →



# What is Slidev?

Slidev is a slides maker and presenter designed for developers, consist of the following features

-  **Text-based** - focus on the content with Markdown, and then style them later
-  **Themable** - themes can be shared and re-used as npm packages
-  **Developer Friendly** - code highlighting, live coding with autocompletion
-  **Interactive** - embed Vue components to enhance your expressions
-  **Recording** - built-in recording and camera view
-  **Portable** - export to PDF, PPTX, PNGs, or even a hostable SPA
-  **Hackable** - virtually anything that's possible on a webpage is possible in Slidev

Read more about [Why Slidev?](#)

# Navigation

Hover on the bottom-left corner to see the navigation's controls panel, [learn more](#)

## Keyboard Shortcuts

right / space

next animation or slide

left / shift space

previous animation or slide

up

previous slide

down

next slide

Here!



# Table of contents

You can use the `Toc` component to generate a table of contents for your slides:

```
<Toc minDepth="1" maxDepth="1" />
```

The title will be inferred from your slide content, or you can override it with `title` and `level` in your frontmatter.

1. [Welcome to Slidev](#)
2. [What is Slidev?](#)
  1. [Navigation](#)
3. [Table of contents](#)
4. [Code](#)
  1. [Shiki Magic Move](#)
5. [Components](#)
6. [Themes](#)
7. [Clicks Animations](#)
8. [Motions](#)
9. [LaTeX](#)
10. [Diagrams](#)
11. [Draggable Elements](#)
12. [Imported Slides](#)
13. [Monaco Editor](#)
14. [Learn More](#)

# Code

Use code snippets and get the highlighting directly, and even types hover!

```
// TwoSlash enables TypeScript hover information
// and errors in markdown code blocks
// More at https://shiki.style/packages/twoslash

import { computed, ref } from 'vue'

const count = ref(0)
const doubled = computed(() => count.value * 2)

doubled.value = 2
    Cannot assign to 'value' because it is a read-only

// Inside ./snippets/external.ts
export function emptyArray<T>(length: number) {
  return Array.from<T>({ length })
}
```

[Learn more](#)



# Shiki Magic Move

Powered by shiki-magic-move, Slidev supports animations across multiple code snippets.

Add multiple code blocks and wrap them with ````md magic-move` (four backticks) to enable the magic move. For example:

```
1 <!-- step 4 -->
2 <script setup>
3 const author = {
4   name: 'John Doe',
5   books: [
6     'Vue 2 - Advanced Guide',
7     'Vue 3 - Basic Guide',
8     'Vue 4 - The Mystery'
9   ]
10 }
11 </script>
```

# Components

You can use Vue components directly inside your slides.

```
<Tweet id="1390115482657726468" />
```

We have provided a few built-in components like `<Tweet/>` and `<Youtube/>` that you can use directly. And adding your custom components is also super easy.

```
<Counter :count="10" />
```

-	10	+
---	----	---

Check out the guides for more.

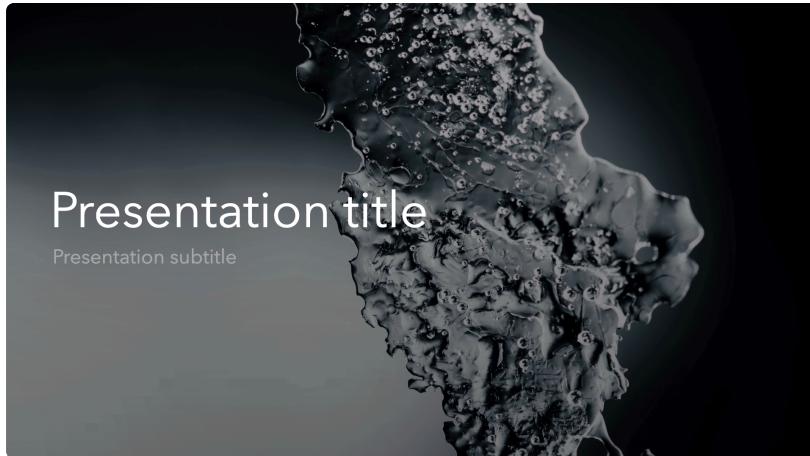
# Themes

Slidev comes with powerful theming support. Themes can provide styles, layouts, components, or even configurations for tools. Switching between themes by just **one edit** in your frontmatter:

```
---
```

```
theme: default
```

```
--
```



```
---
```

```
theme: serifh
```

```
--
```



Read more about [How to use a theme](#) and check out the [Awesome Themes Gallery](#).

# Clicks Animations

You can add `v-click` to elements to add a click animation.

This shows up when you click the slide:

```
<div v-click>This shows up when you click the slide.</div>
```

The `v-mark` directive also allows you to add inline marks, powered by Rough Notation:

```
<span v-mark.underline.orange>inline markers</span>
```

[Learn more](#)

# Motions

Motion animations are powered by [@vueuse/motion](#), triggered by `v-motion` directive.

```
<div
  v-motion
  :initial="{ x: -80 }"
  :enter="{ x: 0 }"
  :click-3="{ x: 80 }"
  :leave="{ x: 1000 }"
>
  Slidev
</div>
```



# Slidev

[Learn more](#)

# LaTeX

LaTeX is supported out-of-box. Powered by [KaTeX](#).

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

Block

$$\nabla \cdot \vec{E} = \frac{\rho}{\varepsilon_0}$$

$$\nabla \cdot \vec{B} = 0$$

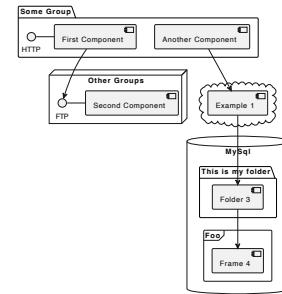
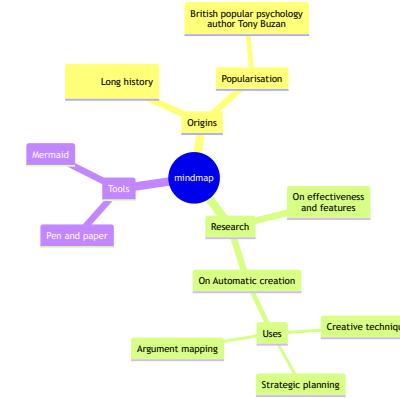
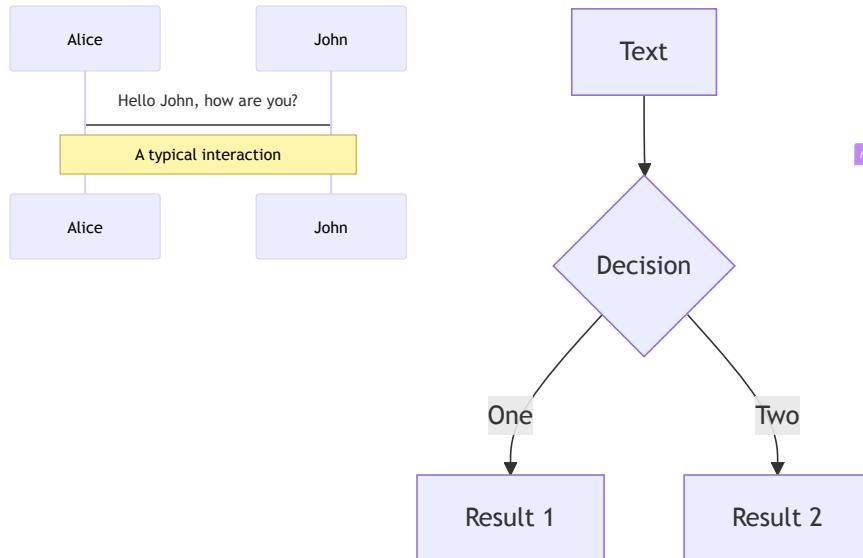
$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$

$$\nabla \times \vec{B} = \mu_0 \vec{J} + \mu_0 \varepsilon_0 \frac{\partial \vec{E}}{\partial t}$$

[Learn more](#)

# Diagrams

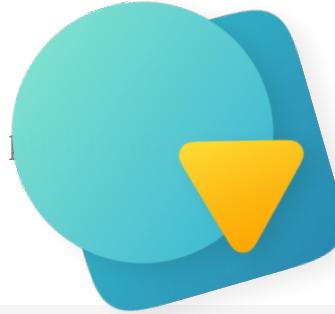
You can create diagrams / graphs from textual descriptions, directly in your Markdown.



Learn more: [Mermaid Diagrams](#) and [PlantUML Diagrams](#)

# Draggable Elements

Double-click on the draggable elements to edit their position.



## DIRECTIVE USAGE

```

```

## COMPONENT USAGE

```
<v-drag text-3xl>
  <div class="i-carbon:arrow-up" />
  Use the `v-drag` component to have a draggable container!
</v-drag>
```

## DRAGGABLE ARROW

```
<v-drag-arrow two-way />
```

Double-click me!



# Imported Slides

You can split your slides.md into multiple files and organize them as you want using the `src` attribute.

## slides.md

```
# Page 1  
  
Page 2 from main entry.  
  
---  
  
## src: ./subpage.md
```

## subpage.md

```
# Page 2  
  
Page 2 from another file.
```

[Learn more](#)

# Monaco Editor

Slidev provides built-in Monaco Editor support.

Add `{monaco}` to the code block to turn it into an editor:

```
import { ref } from 'vue'  
import { emptyArray } from './external'  
  
const arr = ref(emptyArray(10))
```

Use `{monaco-run}` to create an editor that can execute the code directly in the slide:

```
import { version } from 'vue'  
import { emptyArray, sayHello } from './external'  
  
sayHello()  
console.log(`vue ${version}`)  
console.log(emptyArray<number>(10).reduce(fib => [...fib, fib.at(-1)! + fib.at(-2)!], [1, 1]))
```

vue 3.5.13  
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144]

# Learn More

[Documentation](#) · [GitHub](#) · [Showcases](#)

Powered by  Sliddev