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?

Interactive Data Visualization

Slidev supports powerful data visualization through addons like **PlotlyFigure**, allowing you to embed interactive charts and graphs directly into your presentations.

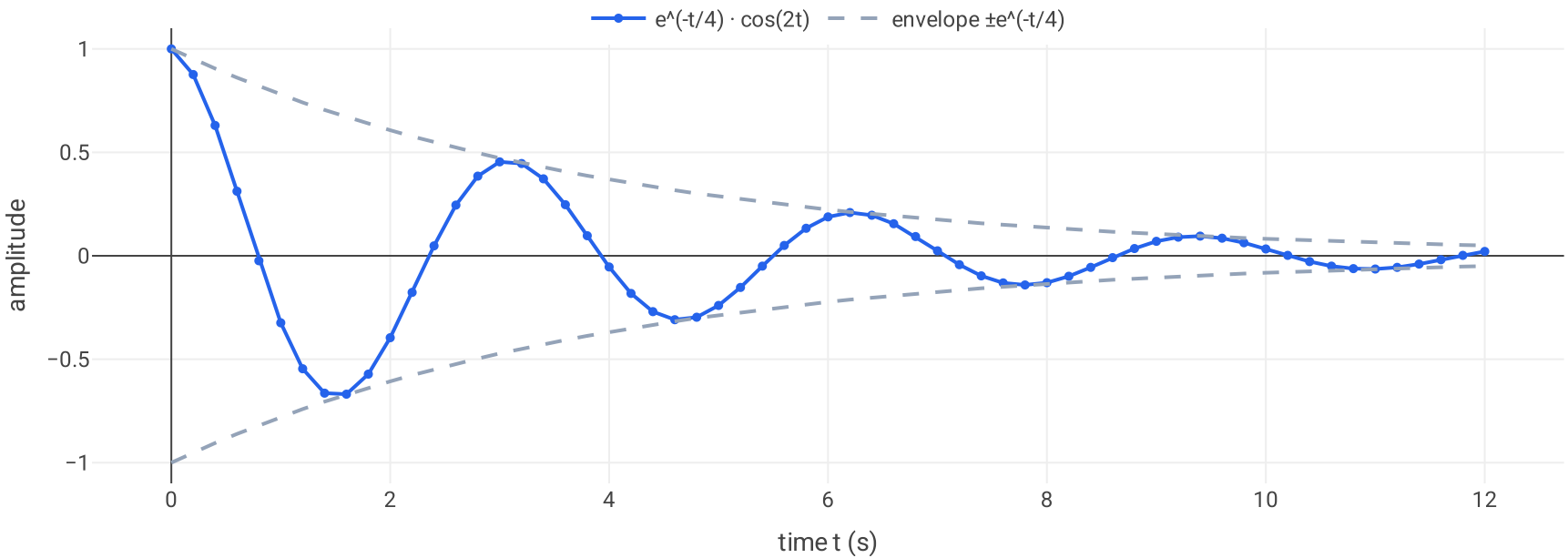


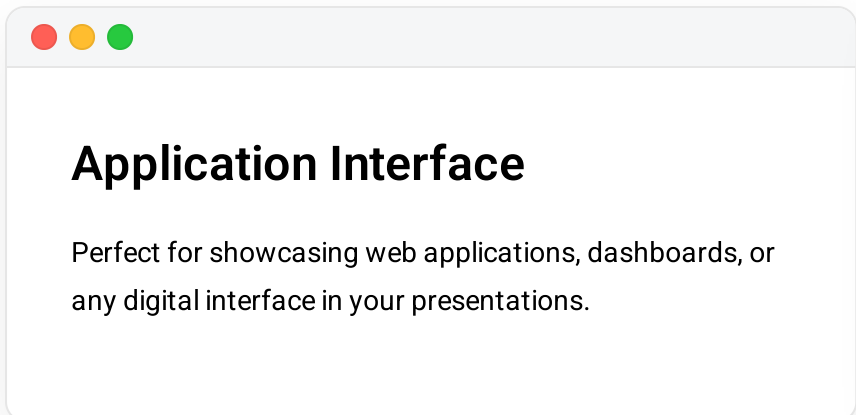
Figure 1: Interactive Plotly Chart - Hover over data points for details

Perfect for showcasing research data, analytics, or any quantitative information with full interactivity.

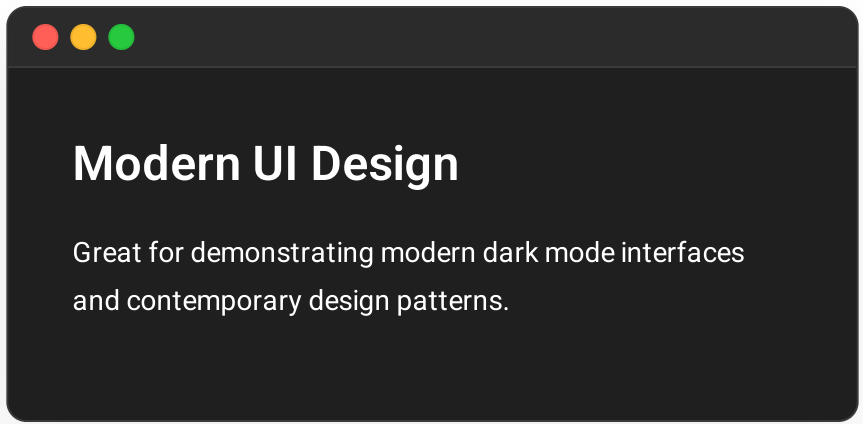
Window Mockup Components

Create professional-looking presentations with **WindowMockup** components that simulate real application interfaces.

Light Theme Mockup:



Dark Theme Mockup:



Ideal for product demos, UI/UX presentations, and showcasing digital prototypes.

Dynamic Arrow Connections

The **FancyArrow** addon allows you to create animated, customizable arrows that connect elements on your slides, perfect for illustrating relationships, workflows, and processes.



Key Features:

- **Smart positioning** - Automatically connects to element edges
- **Multiple arrow types** - Different styles and animations
- **Responsive design** - Adapts to different screen sizes

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

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

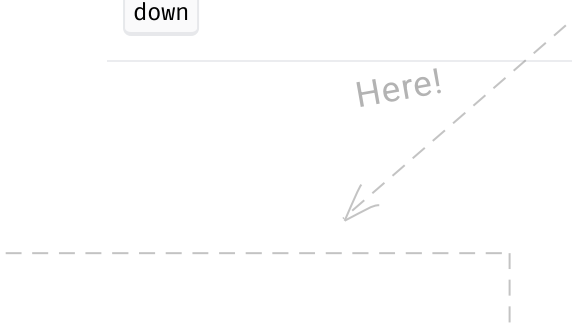


Table of contents

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. A Demo Presentation with Slidev
2. What is Slidev?
3. Interactive Data Visualization
4. Window Mockup Components
5. Dynamic Arrow Connections
    1. Navigation
6. Table of contents
7. Code
    1. Shiki Magic Move
8. Components
9. New Built-in Components
10. Themes
11. Clicks Animations
12. Click Animation Presets
13. Motions
14. LaTeX
15. Diagrams
16. Draggable Elements
17. Imported Slides
18. Monaco Editor

Code

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

TS filename-example.ts

```ts
// 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
```

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

Learn more



Code

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

TS filename-example.ts

```ts
// 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
```

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

Learn more



Code

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

TS filename-example.ts

```ts
// 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
```

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

Learn more



Code

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

TS filename-example.ts

```ts
// 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
```

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

Learn more



Code

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

TS filename-example.ts

```ts
// 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
```

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

Learn more



Code

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

TS filename-example.ts

```ts
// 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
```

```ts
// 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. A `[file name]` after it adds a title bar, and `shiki.magicMove` in the headmatter tunes the animation. For example:

Author.vue

```vue
// step 1
const author = reactive({
  name: 'John Doe',
  books: [
    'Vue 2 - Advanced Guide',
    'Vue 3 - Basic Guide',
    'Vue 4 - The Mystery'
  ]
})
```

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. A `[file name]` after it adds a title bar, and `shiki.magicMove` in the headmatter tunes the animation. For example:

Author.vue

```vue
// step 1
const author = reactive({
  name: 'John Doe',
  books: [
    'Vue 2 - Advanced Guide',
    'Vue 3 - Basic Guide',
    'Vue 4 - The Mystery'
  ]
})
```

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. A `[file name]` after it adds a title bar, and `shiki.magicMove` in the headmatter tunes the animation. For example:

Author.vue

```vue
// step 1
const author = reactive({
  name: 'John Doe',
  books: [
    'Vue 2 - Advanced Guide',
    'Vue 3 - Basic Guide',
    'Vue 4 - The Mystery'
  ]
})
```

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. A `[file name]` after it adds a title bar, and `shiki.magicMove` in the headmatter tunes the animation. For example:

Author.vue

```vue
// step 2
export default {
  data() {
    return {
      author: {
        name: 'John Doe',
        books: [
          'Vue 2 - Advanced Guide',
          'Vue 3 - Basic Guide',
          'Vue 4 - The Mystery'
        ]
      }
    }
  }
}
```

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. A `[file name]` after it adds a title bar, and `shiki.magicMove` in the headmatter tunes the animation. For example:

Author.vue

```vue
// step 2
export default {
  data() {
    return {
      author: {
        name: 'John Doe',
        books: [
          'Vue 2 - Advanced Guide',
          'Vue 3 - Basic Guide',
          'Vue 4 - The Mystery'
        ]
      }
    }
  }
}
```

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. A `[file name]` after it adds a title bar, and `shiki.magicMove` in the headmatter tunes the animation. For example:

Author.vue

```vue
// step 2
export default {
  data() {
    return {
      author: {
        name: 'John Doe',
        books: [
          'Vue 2 - Advanced Guide',
          'Vue 3 - Basic Guide',
          'Vue 4 - The Mystery'
        ]
      }
    }
  }
}
```

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. A `[file name]` after it adds a title bar, and `shiki.magicMove` in the headmatter tunes the animation. For example:

Author.vue

```vue
// step 2
export default {
  data() {
    return {
      author: {
        name: 'John Doe',
        books: [
          'Vue 2 - Advanced Guide',
          'Vue 3 - Basic Guide',
          'Vue 4 - The Mystery'
        ]
      }
    }
  }
}
```

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. A `[file name]` after it adds a title bar, and `shiki.magicMove` in the headmatter tunes the animation. For example:

Author.vue

```vue
// step 3
export default {
  data: () => ({
    author: {
      name: 'John Doe',
      books: [
        'Vue 2 - Advanced Guide',
        'Vue 3 - Basic Guide',
        'Vue 4 - The Mystery'
      ]
    }
  })
}
```

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. A `[file name]` after it adds a title bar, and `shiki.magicMove` in the headmatter tunes the animation. For example:

Author.vue

```vue
<!-- step 4 -->
<script setup>
const author = {
  name: 'John Doe',
  books: [
    'Vue 2 - Advanced Guide',
    'Vue 3 - Basic Guide',
    'Vue 4 - The Mystery'
  ]
}
</script>
```

Components

You can use Vue components directly inside your slides.

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" />
```

<Counter :count="10" />

Check out the guides for more.

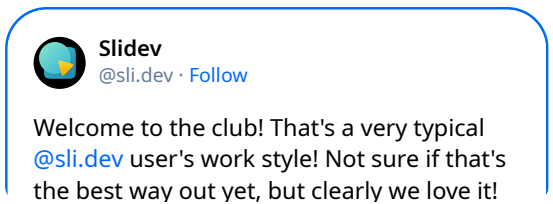
```
<Tweet url="https://x.com/i/web/status/1390115482657726468"
```

<Tweet url="https://x.com/i/web/status/1390115482657726468" />

New Built-in Components

`<BlueSky>` embeds a Bluesky post from its URL or `at://` URI:

```
<BlueSky uri="https://bsky.app/profile/sli.dev/post/3lgwg2v
```



`<AutoFitText>` scales its text to fill the box. Try typing:

Fits any box

Fits any box

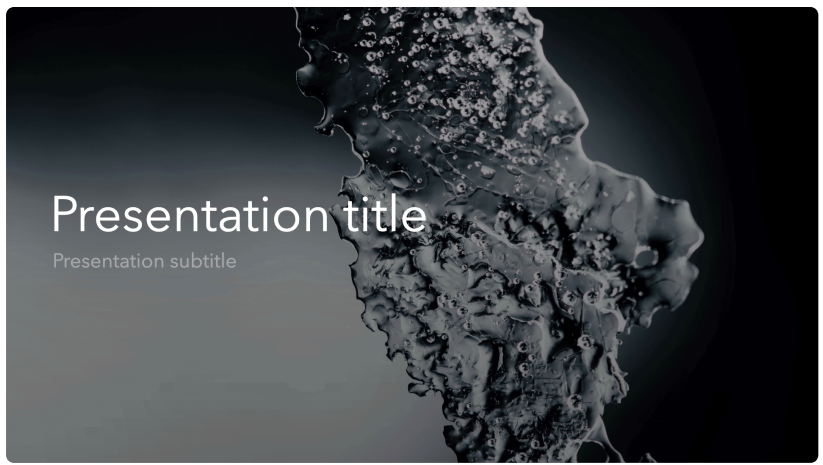
```
<AutoFitText v-model="text" :min="12" :max="96" />
```

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: seriph
---
```





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.

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>
```

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. A `[start, end]` range removes the mark again:

```
<span v-mark.underline.orange>inline markers</span>
<span v-mark.circle.orange="[4, 5]">marked from click 4 until click 5</span>
```

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. A `[start, end]` range removes the mark again:

```
<span v-mark.underline.orange>inline markers</span>
<span v-mark.circle.orange="[4, 5]">marked from click 4 until click 5</span>
```

Learn more

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. A `[start, end]` range removes the mark again:

```
<span v-mark.underline.orange>inline markers</span>
<span v-mark.circle.orange="[4, 5]">marked from click 4 until click 5</span>
```

Learn more

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. A `[start, end]` range removes the mark again:

```
<span v-mark.underline.orange>inline markers</span>
<span v-mark.circle.orange="[4, 5]">marked from click 4 until click 5</span>
```

Learn more

Click Animation Presets

Pick how hidden elements enter with `clickAnimation` in the headmatter or a slide's frontmatter. This slide uses `fade-in up`; modifiers override it per element.

`v-click.fade` (dimmed until shown)

```
---
clickAnimation: fade-in up   # fade, fade-in, up, down, left, right, scale, none
---
```

Click Animation Presets

Pick how hidden elements enter with `clickAnimation` in the headmatter or a slide's frontmatter. This slide uses `fade-in up`; modifiers override it per element.

`v-click` (slide default)

`v-click.fade` (dimmed until shown)

```
---
clickAnimation: fade-in up   # fade, fade-in, up, down, left, right, scale, none
---
```

Click Animation Presets

Pick how hidden elements enter with `clickAnimation` in the headmatter or a slide's frontmatter. This slide uses `fade-in up`; modifiers override it per element.

`v-click` (slide default)

`v-click.fade-in.left`

`v-click.fade` (dimmed until shown)

```
---
clickAnimation: fade-in up   # fade, fade-in, up, down, left, right, scale, none
---
```

Click Animation Presets

Pick how hidden elements enter with `clickAnimation` in the headmatter or a slide's frontmatter. This slide uses `fade-in up`; modifiers override it per element.

`v-click` (slide default)

`v-click.fade-in.left`

`v-click.fade-in.right`

`v-click.fade` (dimmed until shown)

```
---
clickAnimation: fade-in up   # fade, fade-in, up, down, left, right, scale, none
---
```

Click Animation Presets

Pick how hidden elements enter with `clickAnimation` in the headmatter or a slide's frontmatter. This slide uses `fade-in up`; modifiers override it per element.

`v-click` (slide default)

`v-click.fade-in.left`

`v-click.fade-in.right`

`v-click.fade-in.down`

`v-click.fade` (dimmed until shown)

```
---
clickAnimation: fade-in up   # fade, fade-in, up, down, left, right, scale, none
---
```

Click Animation Presets

Pick how hidden elements enter with `clickAnimation` in the headmatter or a slide's frontmatter. This slide uses `fade-in up`; modifiers override it per element.

`v-click` (slide default)

`v-click.fade-in.left`

`v-click.fade-in.right`

`v-click.fade-in.down`

`v-click.fade-in.scale`

`v-click.fade` (dimmed until shown)

```
---
clickAnimation: fade-in up   # fade, fade-in, up, down, left, right, scale, none
---
```

Click Animation Presets

Pick how hidden elements enter with `clickAnimation` in the headmatter or a slide's frontmatter. This slide uses `fade-in up`; modifiers override it per element.

`v-click` (slide default)

`v-click.fade-in.left`

`v-click.fade-in.right`

`v-click.fade-in.down`

`v-click.fade-in.scale`

`v-click.fade` (dimmed until shown)

```
---
clickAnimation: fade-in up   # fade, fade-in, up, down, left, right, scale, none
---
```

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

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

LaTeX

LaTeX is supported out-of-box. Powered by KaTeX.

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

Block

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

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

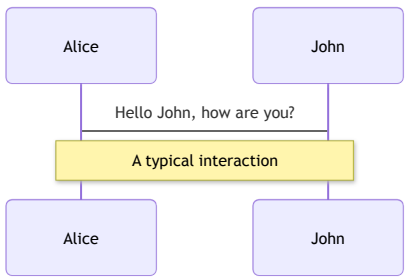
$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$

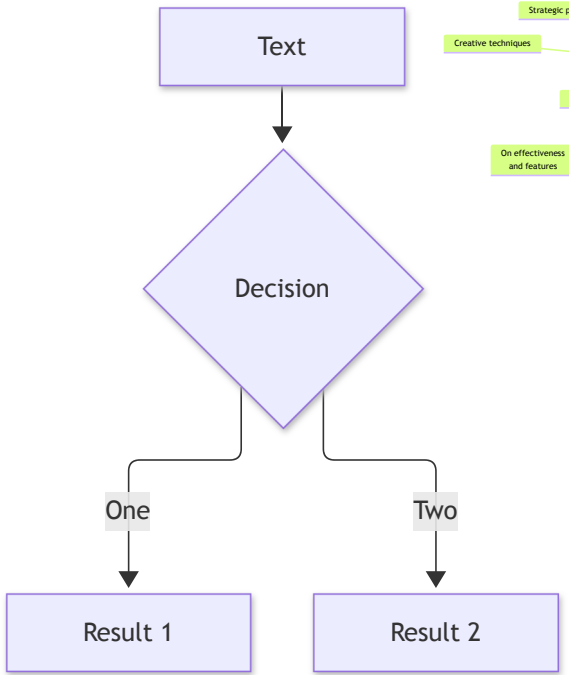
$$\nabla \times \vec{B} = \mu_0\vec{J} + \mu_0\varepsilon_0\frac{\partial \vec{E}}{\partial t}$$

Learn more

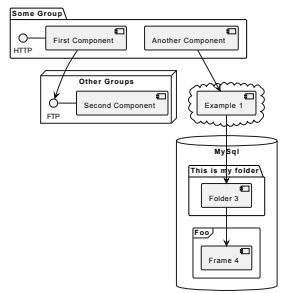
Diagrams

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 positions.

DIRECTIVE USAGE

```
<img v-drag="'square'" src="https://sli.dev/logo.png">
```

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, and `{lines:true}` to show line numbers:

```ts
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:

```ts
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]))
```

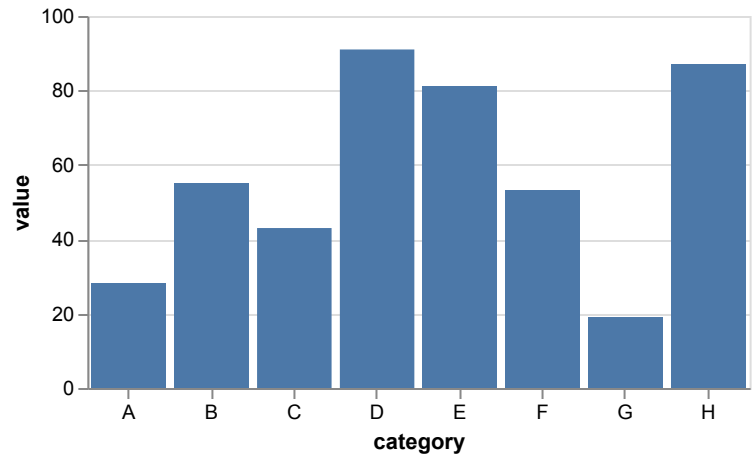
```
vue 3.5.44
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144]
```

Vega-Lite Live Editor

Edit the Vega-Lite spec on the left, then click Run to re-render on the right.

```
// Edit the Vega-Lite spec and click Run to update the cl
const spec = {
  "$schema": "https://vega.github.io/schema/vega-lite/v5
  description: "A simple bar chart with embedded data.",
  data: {
    values: [ { category: "A", value: 28 }, { category: '
    ]
  },
  mark: "bar",
  encoding: { x: { field: "category", type: "nominal", ax
}

// Render using a helper exposed by this slide
window.renderVegaLite(spec)
```



GitHub-style Alerts

A blockquote that starts with `[!NOTE]`, `[!TIP]`, `[!IMPORTANT]`, `[!WARNING]` or `[!CAUTION]` renders as an alert:

> **Note**
>
> While `slidev` runs, an MCP server at `http://localhost:<port>/__mcp` lets AI agents read and edit the deck. Turn it off with `mcp: false`.

> **Tip**
>
> Settings → Cursor Style → **Laser** turns your pointer into a laser while presenting.

> **Important**
>
> `slidev export --format pptx-editable` exports PowerPoint slides with editable text instead of screenshots.

> **Warning**
>
> `pwa: build` precaches the whole built deck for offline use, so the first visit downloads every asset.

Learn More

Documentation · GitHub · Showcases

Powered by  Slidev