

# SLIDEV Docs - Русский



Sliddev

# Table of contents

• Ru - README	4
• Ru - TRANSLATIONS	5
• Builtin - Components	8
• Builtin - Layouts	9
• Custom - Config katex	13
• Custom - Config mermaid	14
• Custom - Config monaco	15
• Custom - Config shortcuts	17
• Custom - Config vite	18
• Custom - Config vue	20
• Custom - Config windicss	21
• Custom - Directory structure	22
• Custom - Fonts	26
• Custom - Global layers	29
• Custom - Highlighters	31
• Custom - Index	33
• Custom - Vue context	35
• Guide - Animations	37
• Guide - Drawing	41
• Guide - Editors	43
• Guide - Exporting	46
• Guide - Faq	49
• Guide - Hosting	52
• Guide - Index	55
• Guide - Install	60
• Guide - Navigation	62
• Guide - Presenter mode	64
• Guide - Recording	65
• Guide - Syntax	67
• Guide - Why	76
• Resources - Covers	79
• Resources - Learning	80
• Ru - Showcases	81
• Themes - Gallery	82

- Themes - Use \_\_\_\_\_ 83
- Themes - Write a /theme \_\_\_\_\_ 85

# sli.dev

Documentation for [Slidev](#)

## Translations

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

## Start Server Locally

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

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

## Publishing your translations

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

## Maintaining the translations up-to-date

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

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

Thanks again!

---

[Go to TOC](#)

# Компоненты

## Встроенные компоненты

Документация в процессе разработки. А пока, вы можете взглянуть напрямую [в исходный код](#).

## Кастомные компоненты

Создайте папку `components/` в корне вашего проекта, и просто вставьте в неё ваши кастомные Vue-компоненты, после этого вы сможете использовать их по тому же имени в ваших markdown-файлах!

Подробнее в разделе [Кастомизация](#)

## Компоненты темы

Темы также могут содержать компоненты. Прочтайте документацию по ним, чтобы узнать, что они предоставляют.

Подробнее в [структуре каталогов](#).

---

[Go to TOC](#)

# Шаблоны

## Встроенные шаблоны

Поскольку темы могут переопределять поведение макетов, лучший способ узнать точное использование, параметры и примеры - обратиться к их документации.

### center

Отображает содержимое в центре экрана.

### cover

Используется для отображения титульной страницы презентации, может содержать название презентации, контекстуализацию и т.д.

### default

Самый простой макет для отображения любого вида контента.

### end

Заключительная страница для презентации.

### fact

Показать на экране какой-либо факт или данные с большим акцентом.

### full

Использует все пространство экрана для отображения содержимого.

### image-left

Показывает изображение в левой части экрана, содержимое будет размещено в правой части.

## Использование

```
---  
layout: image-left  
  
# источник изображения  
image: ./path/to/the/image  
  
# пользовательское имя класса для содержимого  
class: my-cool-content-on-the-right  
---
```

## image-right

Показывает изображение в правой части экрана, содержимое будет размещено в левой части.

### Использование

```
---
layout: image-right
# источник изображения
image: ./path/to/the/image
# пользовательское имя класса для содержимого
class: my-cool-content-on-the-left
---
```

## image

Показывает изображение в качестве основного содержимого страницы.

### Использование

```
---
layout: image
# источник изображения
image: ./path/to/the/image
---
```

## iframe-left

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

### Usage

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

## iframe-right

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

### Usage

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

```
--  
# a custom class name to the content  
class: my-cool-content-on-the-left  
---
```

## iframe

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

### Usage

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

## intro

Представление презентации, обычно с указанием названия презентации, краткого описания, автора и т.д.

## none

Макет без какой-либо существующей стилизации.

## quote

Отображает цитату на видном месте.

## section

Используется для обозначения начала нового раздела презентации.

## statement

Делает утверждение/заявление основным содержанием страницы.

## two-cols

Разделяет содержимое страницы на две колонки.

### Использование

```
--  
layout: two-cols  
--  
  
# Слева  
  
Это отобразится слева  
::right::
```

# Справа

Это отобразится справа

## Кастомные шаблоны

Создайте папку `layouts/` в корне вашего проекта, и просто вставьте в неё ваши кастомные Vue-компоненты.

Подробнее в разделе [Кастомизация](#)

## Шаблоны темы

Темы также могут содержать макеты, либо заменять существующие. Пожалуйста, прочтите документацию, чтобы узнать, что они предоставляют.

---

[Go to TOC](#)

# Конфигурация KaTeX

Создайте `./setup/katex.ts` со следующим содержанием:

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

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

С помощью этой записи вы можете указать пользовательские настройки для [параметров KaTeX](#). Помимо определения типов и их документацию для получения более подробной информации.

# Конфигурация Mermaid

Создайте `./setup/mermaid.ts` со следующим содержанием:

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

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

С помощью этой записи вы можете указать дефолтные настройки для [Mermaid](#). Посмотрите определения типов и их документацию для получения более подробной информации.

# Конфигурация Monaco

Создайте `./setup/monaco.ts` со следующим содержанием:

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

export default defineMonacoSetup(async (monaco) => {
  // используйте `monaco` для конфигурации
})
```

Подробнее о [конфигурации Monaco](#).

## Использование

Чтобы использовать Monaco в своих слайдах, просто добавьте `{monaco}` в фрагменты кода:

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

console.log(plusOne.value) // 2

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

На

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

console.log(plusOne.value) // 2

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

## Экспорт

По умолчанию Monaco будет работать ТОЛЬКО в `dev` режиме. Если вы хотите, чтобы он был доступен в экспортированном SPA, вы можете настроить его в своём frontmatter:

```
---
monaco: true # по умолчанию "dev"
---
```

## Автоматическая установка типов

Когда вы используете TypeScript с Monaco, типы для зависимостей будут автоматически установлены на клиентской стороне.

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

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

В приведённом выше примере убедитесь, что `vue` и `@vueuse/core` установлены локально как `dependencies` / `devDependencies`, Slidev сделает все остальное, чтобы типы для редактора работали автоматически!

## Конфигурация тем

Тема контролируется Slidev на основе светлой/темной темы. Если вы хотите настроить ее, вы можете передать идентификатор темы в функцию настройки:

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

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

Если вы хотите загрузить собственные темы:

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

// измените на свои темы
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',
    },
  }
})
```

Если вы создаете тему для Slidev, используйте динамический `import()` внутри функции настройки, чтобы получить лучшие результаты при tree-shaking и code-splitting.

---

[Go to TOC](#)

# Конфигурация горячих клавиш

Доступно с версии v0.20

Создайте `./setup/shortcuts.ts` со следующим содержанием:

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

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

С помощью этой настройки вы можете задать кастомные горячие клавиши, упомянутые в [Навигации](#). Приведённая выше конфигурация привязывает переход на следующую анимацию или слайд к `enter` и переход на предыдущую анимацию или слайд к `backspace`.

Функция получает объект с некоторыми методами навигации и возвращает массив, содержащий некоторую конфигурацию горячих клавиш. Посмотрите определения типов для получения более подробной информации.

Подробнее о событиях нажатия клавиш в [useMagicKeys | VueUse](#).

# Конфигурация Vite

Под капотом Slidev работает на [Vite](#). Это означает, что вы можете использовать отличную систему плагинов Vite, чтобы ещё больше настраивать свои слайды.

Создайте `vite.config.ts`, если хотите настроить Vite.

В Slidev предварительно настроены следующие плагины:

- `@vitejs/plugin-vue`
- `unplugin-vue-components`
- `unplugin-icons`
- `vite-plugin-md`
- `vite-plugin-windicss`
- `vite-plugin-remote-assets`

Узнайте больше о [предварительных настройках](#) здесь.

## Конфигурация внутренних плагины

Доступно с версии v0.21

Чтобы настроить список встроенных плагинов выше, создайте `vite.config.ts` со следующим содержимым. Обратите внимание, что у Slidev есть некоторые параметры предварительной настройки для этих плагинов, это использование переопределит некоторые из них, что потенциально может привести к поломке приложения. Относитесь к этому как к **расширенной функции**. Прежде чем двигаться дальше, убедитесь, что вы знаете, что делаете.

```
import { defineConfig } from 'vite'

export default defineConfig({
  slidev: {
    vue: {
      /* vue параметры */
    },
    markdown: {
      /* markdown-it параметры */
      markdownItSetup(md) {
        /* кастомные markdown-it плагины */
        md.use(/* ... */)
      },
      /* параметры для других плагинов */
    },
  },
})
```

См. дополнительные параметры в [объявлении типов](#).

---

[Go to TOC](#)

# Конфигурация Vue

Slidev использует [Vue 3](#) для рендеринга приложения на стороне клиента. Вы можете расширить приложение, добавив собственные плагины или конфигурации.

Создайте `./setup/main.ts` со следующим содержанием:

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

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

Его также можно использовать в качестве главного входа в ваше приложение Slidev для выполнения некоторых инициализаций перед запуском приложения.

Подробнее: [API Vue приложений](#).

# Конфигурация Windi CSS

Markdown, естественно, поддерживает встроенные разметки HTML. Таким образом, вы можете стилизовать свой контент так, как захотите. Для некоторого удобства у нас есть встроенный [Windi CSS](#), так что вы можете стилизовать разметку напрямую с помощью utility-классов.

Например:

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

Режим атрибутов в [Windi CSS v3.0](#) включён по умолчанию.

## Конфигурации

Чтобы настроить Windi CSS и расширить встроенные конфигурации, создайте `setup/windicss.ts` со следующим содержимым

```
// setup/windicss.ts

import { defineWindiSetup } from '@slidev/types'

// расширение встроенных конфигураций windicss
export default defineWindiSetup(() => ({
  shortcuts: {
    // кастомный дефолтный фон
    'bg-main': 'bg-white text-[#181818] dark:(bg-[#121212] text-[#ddd])',
  },
  theme: {
    extend: {
      // здесь можно заменить шрифты, не забудьте обновить ссылки на веб-шрифты в
      `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',
      },
    },
  },
}))
```

Learn more about [Windi CSS configurations](#)

---

[Go to TOC](#)

# Структура каталогов

Slidev использует некоторые соглашения о структуре каталогов, чтобы минимизировать конфигурацию и сделать расширения функциональности гибкими и интуитивно понятными.

Базовая структура выглядит следующим образом:

```
your-slidev/
  └── components/      # кастомные компоненты
  └── layouts/         # кастомные шаблоны
  └── public/          # static assets
  └── setup/           # кастомные настройки / хуки
  └── styles/          # кастомные стили
  └── index.html        # инъекции в index.html
  └── slides.md         # описание слайдов
  └── vite.config.ts    # расширение конфигурации vite
```

Все они необязательны.

## Компоненты

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

Компоненты внутри этого каталога можно напрямую использовать в Markdown слайдов с тем же именем компонента, что и имя файла.

Например:

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

```
<!-- slides.md -->

# Мой слайд

<MyComponent :count="4"/>

<!-- оба названия работают -->

<hello-world foo="bar">
  Слот
</hello-world>
```

Эта функция работает благодаря `vite-plugin-components`.

Slidev также предоставляет вам некоторые [встроенные компоненты](#).

## Шаблоны

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

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

Вы можете использовать любое имя файла для своего макета. Затем вы можете указать его в шаблоне в заголовке YAML, используя имя файла.

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

Если шаблон имеет то же имя, что и встроенный шаблон или шаблон темы, то ваш шаблон будет иметь приоритет над ними. Порядок приоритетов - `local > theme > built-in`.

В компоненте шаблона используйте `<slot/>` для содержимого слайда. Например:

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

## Public

Соглашения: `./public/*`

Ресурсы в этой папке будут находиться по корневому пути `/` во время разработки и копироваться в корень каталога `dist` как есть. Узнайте больше о `public` каталоге Vite.

## Стили

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

Файлы, соответствующие этому соглашению, будут внедрены в корень приложения. Если вам нужно импортировать несколько css записей, вы можете создать следующую структуру и самостоятельно управлять порядком импорта.

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

```
// styles/index.ts

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

Стили будут обрабатываться через [Windi CSS](#) и [PostCSS](#), поэтому вы можете использовать вложенные css стили и [at-директивы](#) прямо из коробки. Например:

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

[Подробнее о синтаксисе.](#)

## index.html

Соглашения: `index.html`

`index.html` позволяет вставлять метатеги и/или скрипты в основной `index.html`

Например, для следующего кастомного файла `index.html`:

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

Финальный файл `index.html` будет выглядеть:

```
<!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>
```

## Глобальные слои

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

Подробнее: [Глобальные слои](#)

---

[Go to TOC](#)

# Шрифты

Доступно с версии v0.20

Хотя вы можете использовать HTML и CSS для настройки шрифтов и стилей для своих слайдов, Slidev также предоставляет удобный способ их использования без особых усилий.

В вашем fontmatter настройте следующим образом

```
---  
fonts:  
  # основной текст  
  sans: 'Robot'  
  # использование с css классом `font-serif` от windicss  
  serif: 'Robot Slab'  
  # для блоков кода, inline-кода и т.д.  
  mono: 'Fira Code'  
---
```

И это всё.

Шрифты будут **автоматически импортированы из Google Fonts**. Это означает, что вы можете напрямую использовать любые шрифты, доступные в Google Fonts.

## Локальные шрифты

По умолчанию Slidev считает, что все шрифты, указанные в конфигурациях `fonts`, взяты из Google Fonts. Если вы хотите использовать локальные шрифты, укажите `fonts.local`, чтобы отключить автоматический импорт.

```
---  
fonts:  
  # вы можете использовать `,` чтобы использовать несколько шрифтов для fallback'а  
  # (как font-family в css)  
  sans: 'Helvetica Neue, Robot'  
  # пометить 'Helvetica Neue' как локальный шрифт  
  local: 'Helvetica Neue'  
---
```

## Начертание шрифта и курсив

По умолчанию Slidev импортирует три начертания: `200, 400, 600` для каждого шрифта. Вы можете настроить их следующим образом:

```
---  
fonts:  
  sans: 'Robot'  
  # по умолчанию
```

```

weights: '200,400,600'
# импорт курсивных шрифтов, по умолчанию `false`
italic: false
---
```

Эта конфигурация применяется ко всем веб-шрифтам. Для более детального управления начертанием каждого шрифта вам необходимо вручную импортировать их с помощью [HTML](#) и CSS.

## Fallback шрифтов

Для большинства сценариев вам нужно только указать «специальный шрифт», и Slidev добавит вам fallback шрифты, например:

```

---  

fonts:  

  sans: 'Robot'  

  serif: 'Robot Slab'  

  mono: 'Fira Code'  

---
```

сгенерируется CSS

```

.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;
}
```

Если вы хотите отключить fallback шрифты, то настройте следующим образом

```

---  

fonts:  

  mono: 'Fira Code, monospace'  

  fallback: false  

---
```

## Провайдеры

- Параметры: `google` | `none`
- По умолчанию: `google`

В настоящее время поддерживаются только Google Fonts, в будущем мы планируем добавить больше провайдеров. Если указать `none`, то функция автоимпорта будет полностью отключена, и все шрифты будут обрабатываться локально.

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

---

[Go to TOC](#)

# Глобальные слои

Доступно с версии v0.17

Глобальные слои позволяют иметь кастомные компоненты, **постоянно** доступные в слайдах. Это может быть полезно для футера, анимации смены слайдов, глобальных эффектов и т.д.

Slidev предоставляет два слоя для этого: создайте `global-top.vue` или `global-bottom.vue` в корне вашего проекта, и он будет загружен автоматически.

Связь слоёв:

- Глобальный Top (`global-top.vue`)
- Слайды
- Глобальный Bottom (`global-bottom.vue`)

## Примеры

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

Текст `Ваше имя` будет отображаться на всех ваших слайдах.

Чтобы использовать при определённых условиях, вы можете применить его с помощью [глобального контекста Vue](#).

```
<!-- футер будет скрыт на четвёртой странице -->
<template>
  <footer
    v-if="$slidev.nav.currentPage !== 4"
    class="absolute bottom-0 left-0 right-0 p-2">
    >
    Ваше имя
  </footer>
</template>
```

```
<!-- футер будет скрыт на "обложке" шаблона -->
<template>
  <footer
    v-if="$slidev.nav.currentLayout !== 'cover'"
    class="absolute bottom-0 left-0 right-0 p-2">
    >
    Ваше имя
  </footer>
</template>
```

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

# Подсветка

Slidev поставляется с двумя подсветками синтаксиса на выбор:

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

**Prism** - один из самых популярных инструментов подсветки синтаксиса. Подсветка выполняется путём добавления токенов в виде классов и стилизации их через CSS. Вы можете просмотреть [официальные темы](#) или очень легко создать/настроить их самостоятельно, используя `prism-theme-vars`.

**Shiki**, с другой стороны, представляет собой средство выделения синтаксиса на основе грамматики TextMate. Он генерирует цветовые токены, поэтому дополнительный CSS не требуется. Поскольку он имеет отличную поддержку грамматики, сгенерированные цвета очень точны, как, например, в VS Code. Shiki также поставляется с [набором встроенных тем](#). Обратной стороной Shiki является то, что для выделения также требуются темы TextMate (совместимые с темой VS Code), что может быть немногого сложнее настроить.

Темы Slidev обычно поддерживают как Prism, так и Shiki, но в зависимости от темы, которую вы используете, он может поддерживать только одну из них.

Когда у вас есть выбор, компромисс заключается в следующем:

- **Prism** – для лёгкой кастомизации
- **Shiki** – для более аккуратной подсветки

По умолчанию Slidev использует Prism. Вы можете изменить его в frontmatter:

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

## Конфигурация Prism

Чтобы настроить Prism, вы можете просто импортировать css тему или использовать `prism-theme-vars` для настройки тем как для светлого, так и для тёмного режима. Обратитесь к документации Prism для получения более подробной информации.

## Конфигурация Shiki

Создайте файл `./setup/shiki.ts` со следующим содержанием:

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

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

Доступные имена тем можно посмотреть в [документации Shiki](#).

Или, если вы хотите использовать свою собственную тему:

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

# Кастомизация

Slidev полностью кастомизируем, начиная от стилей, заканчивая инструментами конфигурации. Это позволяет вам сконфигурировать инструменты под ([Vite](#), [Windi CSS](#), [Monaco](#), и т.д.)

## Frontmatter Configures

Вы можете настроить Slidev в frontmatter блоке вашего первого слайда, ниже показаны значения по умолчанию для каждого параметра.

```
---
# id темы или название пакета
theme: 'default'
# заголовок слайда, если не указан, то будет автоматически подставлен из первого найденного заголовка
title: ''
# шаблон заголовка для веб-страницы, `%s` будет заменен заголовком страницы
titleTemplate: '%s - Slidev'

# загрузка pdf в SPA сборке, также может содержать кастомный URL
download: true
# подсветка синтаксиса, может быть 'prism' или 'shiki'
highlighter: 'prism'
# отображение номеров строк в блоках кода
lineNumbers: false
# включение Monaco редактора, по умолчанию только в дев режиме
monaco: 'dev'

# цветовая схема для слайдов, может быть 'auto', 'light', или 'dark'
colorSchema: 'auto'
# режим роутера для vue-router, может быть 'history' или 'hash'
routerMode: 'history'
# соотношение сторон слайдов
aspectRatio: '16/9'
# реальная ширина canvas, единица измерения в пикселях
canvasWidth: 980

# шрифты будут автоматически импортированы из Google fonts
# Подробнее: https://sli.dev/custom/fonts
fonts:
  sans: 'Roboto'
  serif: 'Roboto Slab'
  mono: 'Fira Code'

# дефолтные настройки frontmatter для всех слайдов
defaults:
  layout: 'default'
  # ...

# информация для слайдов, может быть markdown-строкой
info: |
  ## Slidev
  My first [Slidev](http://sli.dev/) presentations!
---
```

Подробнее в [определении типов](#).

## Структура каталогов

Slidev использует структуру каталогов для минимальной и поверхностной конфигурации, и делает расширения гибкими и интуитивными в функциональности.

Подробнее в [Структура каталогов](#).

## Конфигурация инструментов

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

---

[Go to TOC](#)

# Глобальный контекст Vue

Slidev внедряет `$slidev` в [глобальный контекст Vue](#) для расширенных условий или элементов управления навигацией.

## Использование

Вы можете получить доступ к нему в любом месте markdown-файла или шаблона Vue, используя "Mustache" синтаксис.

```
<!-- slides.md -->
# Страница 1
Текущая страница: {{ $slidev.nav.currentPage }}
```

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

## Параметры

### `$slidev.nav`

Реактивный объект, содержащий свойства и элементы управления для навигации по слайдам. Например:

```
$slidev.nav.next() // переход к следующему шагу
$slidev.nav.nextSlide() // переход к следующему слайду (пропуская v-clicks)
$slidev.nav.go(10) // переход к слайду #10
```

```
$slidev.nav.currentPage // номер текущего слайда
$slidev.nav.currentLayout // текущий id шаблона
$slidev.nav.clicks // текущее количество кликов
```

Дополнительные доступные свойства смотрите в экспортах файла `nav.ts`.

### `$slidev.configs`

Реактивный объект, содержащий обработанные [настройки из первого frontmatter](#) вашего `slides.md`. Например

```
---  
title: Мой первый Slidev!
```

```
---
```

```
{{ $slidev.configs.title }} // 'Мой первый Slidev!'
```

## \$slidev.themeConfigs

Реактивный объект, содержащий обработанные настройки темы.

```
---  
title: Мой первыйSlidev!
```

```
themeConfig:
```

```
  primary: #213435
```

```
---
```

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

# Анимация

## Анимация по клику

### v-click

Чтобы применить «анимацию по клику» для элементов, вы можете использовать директиву `v-click` или `<v-click>` компоненты

```
# Привет
<! -- Использование компонента: этот блок не будет видно, пока вы не нажмёте
"Далее" -->
<v-click>

Привет, мир!

</v-click>

<! -- Использование директивы: этот блок не будет видно, пока вы не нажмёте "Далее"
второй раз -->
<div v-click class="text-xl p-2">

Хэй!

</div>
```

### v-after

Использование `v-after` очень похоже на `v-click`, но элемент появится только при срабатывании предыдущего `v-click`.

```
<div v-click>Привет</div>
<div v-after>Мир</div>
```

Когда вы нажмёте кнопку «Далее», Привет и Мир появятся вместе.

### v-click-hide

То же, что и `v-click`, но вместо того, чтобы отображать элемент, он скроет его после клика.

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

### v-clicks

`v-clicks` используется только как компонент. Это сокращение для директивы `v-click` ко всем её дочерним элементам. Особенно полезно при работе со списками.

```
<v-clicks>
  - Пункт 1
```

```
- Пункт 2
- Пункт 3
- Пункт 4

</v-clicks>
```

Каждый пункт будет отображаться по клику на «Далее».

## Кастомный счётчик кликов

По умолчанию Slidev подсчитывает, сколько шагов нужно сделать, прежде чем перейти к следующему слайду. Вы можете переопределить этот параметр, передав frontmatter опцию `clicks`:

```
---  
# 10 кликов на этом слайде, прежде чем перейти к следующему  
clicks: 10  
---
```

## Порядок

Передав индекс кликов в свои директивы, вы можете настроить порядок раскрытия

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

```
<!-- обратный порядок -->
<div v-click="3">1</div>
<div v-click="2">2</div>
<div v-click="1">3</div>
```

```
---  
clicks: 3  
---  
  
<!-- отобразится после 3 кликов -->  
<v-clicks at="3">  
  <div>Привет</div>  
</v-clicks>
```

## Переходы элементов

Когда вы применяете директиву `v-click` к элементам, она добавляет к ним класс `slidev-vclick-target`. Когда элементы скрыты, будет добавлен класс `slidev-vclick-hidden`. Например:

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

После клика он станет

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

По умолчанию к этим классам применяется мягкий переход прозрачности:

```
// по умолчанию

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

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

Вы можете переопределить их, чтобы настроить эффекты перехода в своих стилях.

Например, вы можете добавить переход масштабирования следующим образом:

```
// styles.css

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

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

Чтобы указать анимацию только для определённого слайда или макета

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

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

Подробнее в [кастомизации стилей](#)

## Перемещение

Slidev имеет встроенный `@vueuse/motion`. Вы можете использовать директиву `v-motion` к любым элементам, чтобы применить к ним перемещение. Например

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

Текст `Slidev` переместится с `-80px` в исходное положение при инициализации.

Примечание: Slidev предзагружает следующий слайд для повышения производительности, что означает, что анимация может начаться до того, как вы перейдёте на страницу. Чтобы она работала правильно, вы можете отключить предварительную загрузку для конкретного слайда.

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

Или управляйте жизненным циклом элемента с помощью `v-if`

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

Подробнее: [Демо](#) | [@vueuse/motion](#) | [v-motion](#) | [Presets](#)

## Переходы страниц

Встроенная поддержка слайдов ЕЩЁ НЕ предусмотрена в текущей версии. Мы планируем добавить их в следующей мажорной версии. Но сейчас вы всё ещё можете использовать свои собственные стили и библиотеки для этого.

---

[Go to TOC](#)

# Drawing & Annotations

Available since v0.23

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

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

## Use with Stylus Pen

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

## Persist Drawings

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

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

## Disable Drawings

Entirely:

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

Only in Development:

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

Only in Presenter Mode:

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

## Drawing Syncing

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

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

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

# Поддержка редакторов

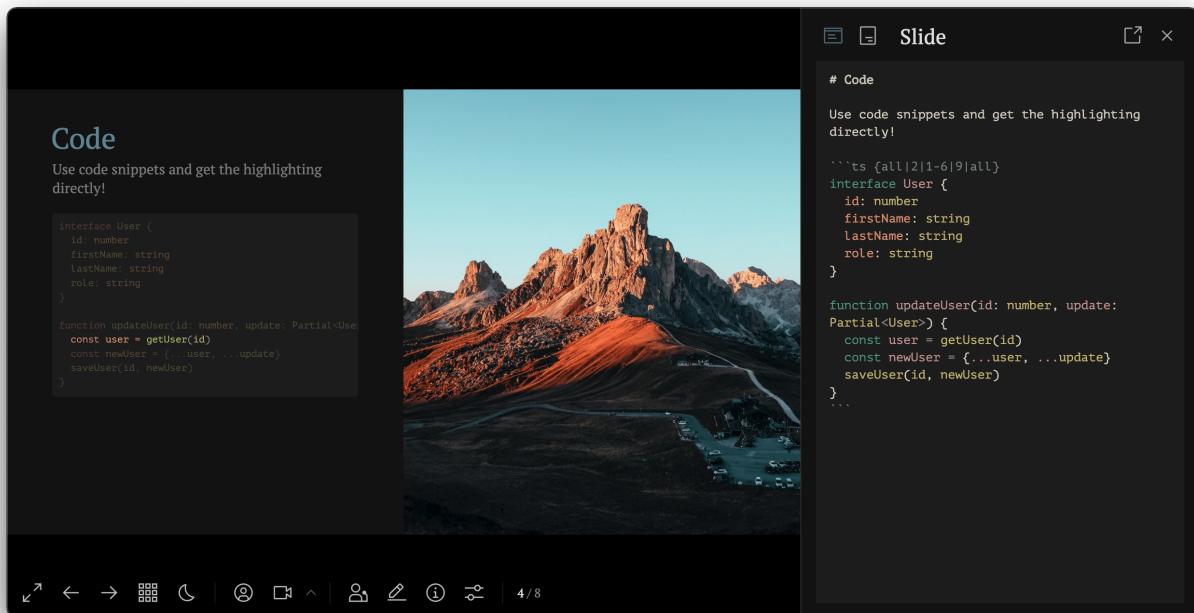
Поскольку Slidev использует Markdown в качестве источника, вы можете использовать ЛЮБЫЕ редакторы, которые вам нравятся.

Если вам нужно высокоуровневое управление слайдами, мы подготовили для вас следующие интеграции редакторов!

## Встроенный редактор

Slidev поставляется со встроенным редактором [CodeMirror](#), который мгновенно перезагружает и сохраняет изменения в вашем файле.

Нажмите на кнопку , чтобы открыть его.



## VS Code расширения



### Slidev for VS Code

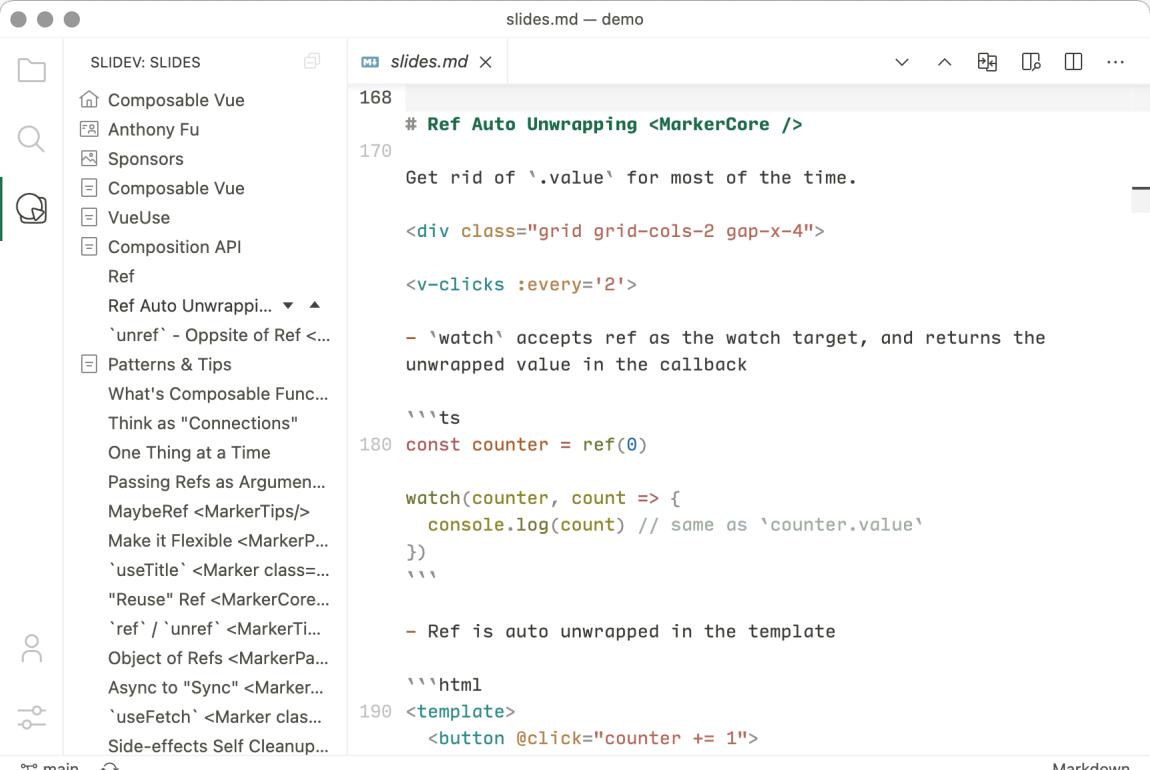
VS Code Marketplace v0.4.1

downloads 26k

Расширение VS Code предоставляет некоторые функции, которые помогут вам лучше организовать слайды и быстро просмотреть их.

#### Особенности

- Просмотр слайдов на боковой панели
- Переход по кнопкам "Далее" / "Назад"
- Изменение порядка слайдов
- Сворачивание блоков слайда
- Преобразование Markdown в HTML



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

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

```

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">

```
- Status Bar (Bottom):** Markdown

---

[Go to TOC](#)

# Экспорт

## PDF

Экспорт в PDF или PNG происходит за счёт [Playwright](#) рендеринга. Поэтому вам необходимо установить `playwright-chromium`, чтобы использовать этот функционал. Если вы выполняете экспорт в CI, то взгляните на [руководство по playwright CI](#).

Установите `playwright-chromium`

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

Теперь можете экспортировать слайды в PDF с помощью следующей команды

```
$ slidev export
```

Через несколько секунд ваши слайды будут находиться в `./slides-exports.pdf`.

## Экспорт с кликами

Доступно с версии v0.21

По умолчанию Slidev экспортирует одну страницу на слайд с отключенной анимацией кликов. Если вы хотите экспортировать слайды с несколькими шагами на несколько страниц, передайте параметр `--with-clicks`.

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

## PNG

При передаче параметра `--format png` Slidev будет экспортировать каждый слайд в изображения PNG вместо PDF.

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

## Single-Page Application (SPA)

<<<<< HEAD Вы также можете скомпилировать слайды в self-hostable SPA:

```
$ slidev build
```

Созданное приложение будет доступно в папке `dist/`, после чего его можно будет разместить на [GitHub Pages](#), [Netlify](#), [Vercel](#) или на любых других платформах. Теперь вы можете поделиться своими слайдами с остальным миром с помощью одной ссылки.

## Базовый путь

Чтобы развернуть слайды на определённом роутере, вам нужно передать параметр `--base`. Например:

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

Подробнее в [документации Vite](#).

## Provide Downloadable PDF

Вы можете предоставить загружаемый PDF-файл зрителям вашего SPA со следующей конфигурацией:

```
---  
  download: true  
---
```

Slidev сгенерирует файл pdf вместе со сборкой, и в SPA отобразится кнопка загрузки.

Вы также можете указать собственный URL-адрес для PDF-файла. В этом случае процесс рендеринга будет пропущен.

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

## Примеры

Вот несколько примеров экспортированного SPA:

- [Starter Template](#)

## • Composable Vue OT Anthony Fu

See [Static Hosting](#).

ee7ae42035591cb6565a72f5217129c670a59b0c

---

[Go to TOC](#)

# FAQ

## Сетки

Поскольку Slidev использует в своей работе интернет, вы можете использовать любые макеты сетки по своему усмотрению. [CSS Grids](#), [flexboxes](#), или даже [Masonry](#).

Поскольку у нас есть встроенный [Windi CSS](#), вот один из простых способов использовать его:

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

Первая колонка

</div>
<div>

Вторая колонка

</div>
</div>
```

Пойдём дальше, вы можете настроить размер каждого столбца, например:

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

Первая колонка (200px)

</div>
<div>

Вторая колонка (auto fit)

</div>
<div>

Третья колонка (10% ширины от родительского контейнера)

</div>
</div>
```

Подробнее о [Windi CSS сетках](#).

## Позиционирование

Слайды имеют фиксированный размер (по умолчанию `980x552px`) и масштабируются в соответствии с экраном пользователя. Вы можете безопасно использовать абсолютное позиционирование на своих слайдах, поскольку они будут масштабироваться вместе с экраном.

Например:

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

Чтобы изменить фактический размер холста, вы можете передать параметры `canvasWidth` в вашем первом frontmatter:

```
---  
canvasWidth: 800  
---
```

## Размер шрифта

Если вам кажется, что размер шрифта на слайдах слишком мал, вы можете изменить его несколькими способами:

### Замена локальных стилей

Вы можете переопределить стили для каждого слайда с помощью встроенного тега `<style>`.

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

---
# Страница 2
Этот слайд не будет затронут.
```

Подробнее: [Встроенные стили](#)

### Замена глобальных стилей

Вы можете указать собственные глобальные стили, например, создав `./style.css`

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

Подробнее: [Глобальные стили](#)

## Масштабирование Canvas

Изменение фактического размера canvas приведёт к масштабированию слайдов и всего вашего содержимого (текста, изображений, компонентов и т.д.).

```
---  
# default: 980  
# поскольку canvas становится меньше, визуальный размер станет больше  
canvasWidth: 800  
---
```

## Использование трансформации

Мы предоставляем встроенный компонент `<Transform />`, который представляет из себя обёртку над CSS свойством трансформации.

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

# Static Hosting

## Build Single Page Applications (SPA)

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

```
$ slidev build
```

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

### Base Path

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

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

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

### Provide Downloadable PDF

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

```
---  
  download: true  
---
```

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

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

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

## Examples

Here are a few examples of the exported SPA:

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

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

## Hosting

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

## Netlify

- [Netlify](#)

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

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

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

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

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

## Vercel

- [Vercel](#)

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

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

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

## GitHub Pages

- [GitHub Pages](#)

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

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

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

# Руководство

## Введение

Slidev (slide + dev, произносится `/slʌɪdɪv/`) – это веб-приложение для создания и презентации слайдов. Предназначен для разработчиков, которые сосредоточены на написании контента в Markdown, в то же время используя HTML и Vue-компоненты для реализации pixel-perfect шаблонов и дизайна со встроенными интерактивными демо в ваши презентации.

Slidev использует многофункциональный markdown-файл для создания красивых слайдов с мгновенной перезагрузкой, а также множество встроенных интеграций, таких как live-кодинг, экспорт PDF, запись презентаций и т.д. Поскольку он работает через интернет, вы можете делать с Slidev всё что угодно — возможности безграничны.

Вы можете узнать больше о создании проекта на странице [Почему Slidev](#).

## Особенности

<<<<< HEAD

-  **Markdown-based** - используйте свои любимые редакторы и рабочие процессы
-  **Удобство разработки** - встроенная подсветка кода, live-кодинг, и т.д.
-  **Темизация** - темы могут распространяться и использоваться через npm-пакеты
-  **Стилизация** - благодаря [Windi CSS](#) очень просто работать со стилями
-  **Интерактивность** - плавное встраивание Vue-компонентов
-  **Режим докладчика** - используйте другое окно или даже телефон для управления слайдами
-  **LaTeX** - встроенная поддержка LaTeX математических уравнений
-  **Диаграммы** - создание диаграмм с текстовым описанием
-  **Иконки** - доступ к иконкам напрямую из любого набора иконок
-  **Редакторы** - встроенные редакторы и [расширения для VS Code](#)
-  **Запись** - встроенная запись и просмотр камеры
-  **Портативность** - экспорт в PDF, PNGs, или даже в hostable SPA
-  **Скорость** - мгновенная перезагрузка благодаря [Vite](#)

## **Расширяемость – использование Vite плагинов, Vue-компонентов, либо других npm-пакетов**

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

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



ee7ae42035591cb6565a72f5217129c670a59b0c

## Технический стек

Slidev разработан благодаря набору этих инструментов и технологий.

<<<<< HEAD

- [Vite](#) - Чрезвычайно быстрый инструмент frontend разработки.
- [Vue 3](#) на основе [Markdown](#) - Фокусировка на контенте, при этом используя всю мощность HTML и Vue компонентов в любом месте.
- [Windi CSS](#) - CSS-фреймворк на основе utility-классов, позволяет легко стилизовать слайды.
- [Prism](#), [Shiki](#), [Monaco Editor](#) - Первоклассная поддержка фрагментов кода с возможностью live-кодинг.
- [RecordRTC](#) - Встроенная запись и просмотр камеры.
- [VueUse](#) - Набор `@vueuse/core`, `@vueuse/head`, `@vueuse/motion`, и т.д.
- [Iconify](#) - Коллекция иконок.

- [KaTeX](#) - Математический рендеринг LaTeX.

## • **Mermaid - Текстовые диаграммы.**

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



ee7ae42035591cb6565a72f5217129c670a59b0c

## **Создание вашей первой презентации**

<<<<< HEAD Чрез NPM: =====

**Try it Online**

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



[Open in StackBlitz](#)

## Create Locally

With NPM:

```
ee7ae42035591cb6565a72f5217129c670a59b0c
```

```
$ npm init slidev
```

Через Yarn:

```
$ yarn create slidev
```

Следуйте инструкциям и начните создавать слайды прямо сейчас! Подробнее о markdown-синтаксисе читайте в [руководстве о синтаксисе](#).

## Command Line Interface

В проекте, где установлен Slidev, вы можете использовать бинарную версию `slidev` в своих пром-скриптах.

```
{
  "scripts": {
    "dev": "slidev", // запуск dev сервера
    "build": "slidev build", // билд SPA для production
    "export": "slidev export" // экспорт слайдов в pdf
  }
}
```

В противном случае вы можете использовать `npx`

```
$ npx slidev
```

Запустите `slidev --help` для просмотра дополнительный опций.

## Markdown синтаксис

Slidev считывает ваш `slides.md` файл в корне проекта и конвертирует его в файлы. Всякий раз, когда вы вносите в него изменения, содержание слайдов будет моментально обновлено. Например:

```
# Slidev

Привет, мир!

---

# Страница 2

Использование блоков кода для подсветки

//```ts
console.log('Привет, мир!')
//```

---

# Страница 3
```

Подробнее о Markdown синтаксисе Slidev читайте в [руководстве по синтаксису](#).

# Установка

## Стартовый шаблон

Для Slidev требуется [Node.js >=14.0](#)

Наилучшим способом начать, будет использование нашего официального стартового шаблона.

Через NPM:

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

Через Yarn:

```
$ yarn create slidev
```

Следуйте подсказкам и презентация автоматически откроется на <http://localhost:3030/>

Шаблон также содержит базовую настройку и короткую демонстрацию с инструкциями о том, как начать работу со Slidev.

## Ручная установка

Если вы по-прежнему хотите установить Slidev вручную или хотите интегрировать его в свои существующие проекты, вы можете сделать:

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

```
$ touch slides.md
```

```
$ npx slidev
```

Обратите внимание, если вы используете [pnpm](#), вам нужно включить [shamefully-hoist](#) опцию для корректной работы Slidev:

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

## Глобальная установка

Доступно с версии v0.14

Вы можете установить Slidev глобально с помощью следующей команды

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

И далее использовать `slidev` в любом месте, без создания проекта каждый раз.

```
$ slidev
```

Эта команда также попытается использовать локальный `@slidev/cli`, если найдёт его в `node_modules`.

## Установка на Docker

Если вам нужен быстрый способ запуска презентации в контейнерах, вы можете использовать предварительно созданный `docker` образ, поддерживаемый [stig124](#), либо создать свой собственный.

Более подробно в [slidevjs/container repo](#).

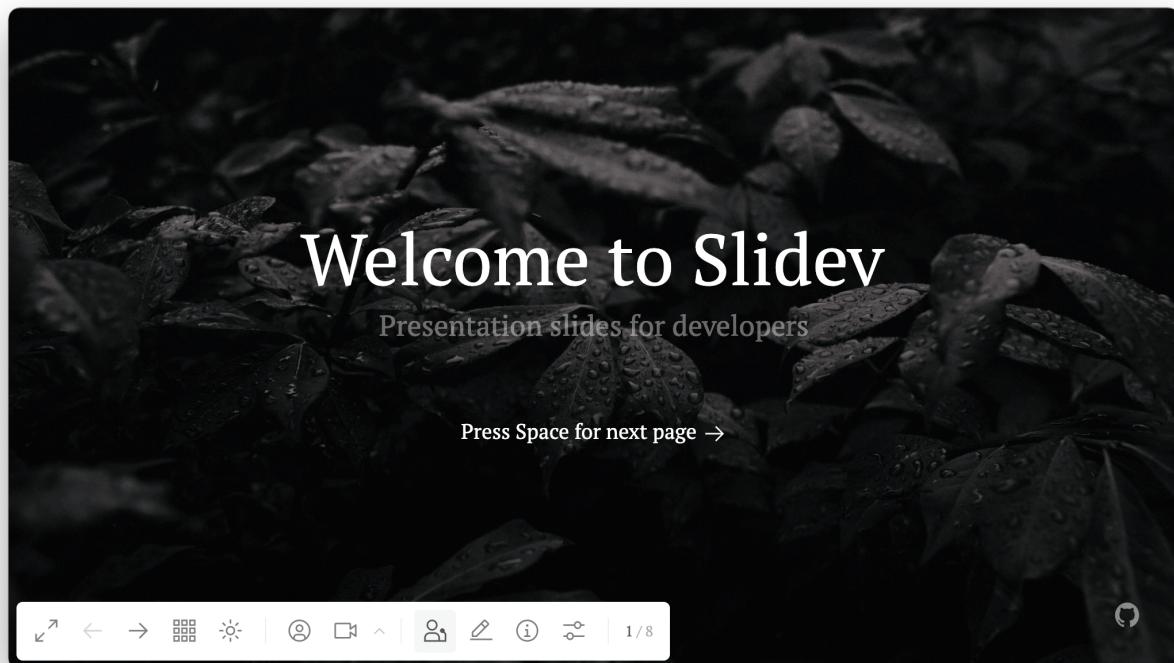
---

[Go to TOC](#)

# Навигация

## Панель навигации

Переместите указатель мыши в левый нижний угол страницы Slidev, и у вас появится панель навигации.



Комбинации	Кнопки	Описание
f		переключение на полный экран
right / space		следующая анимация слайда
left		предыдущая анимация слайда
up	-	предыдущий слайд
down	-	следующий слайд
o		переключение <a href="#">обзора слайдов</a>
d		переключение тёмной темы
-		переключение <a href="#">просмотра камеры</a>
-		<a href="#">запись</a>
-		открытие <a href="#">режима докладчика</a>

Комбинации Кнопки	Описание	
-	переключение <a href="#">встроенного редактора</a>	
-	загрузка слайдов (появляется только в <a href="#">SPA сборке</a> )	
-	отображение информации о слайдах	
-	отображение меню настроек	
g	-	переход к ...

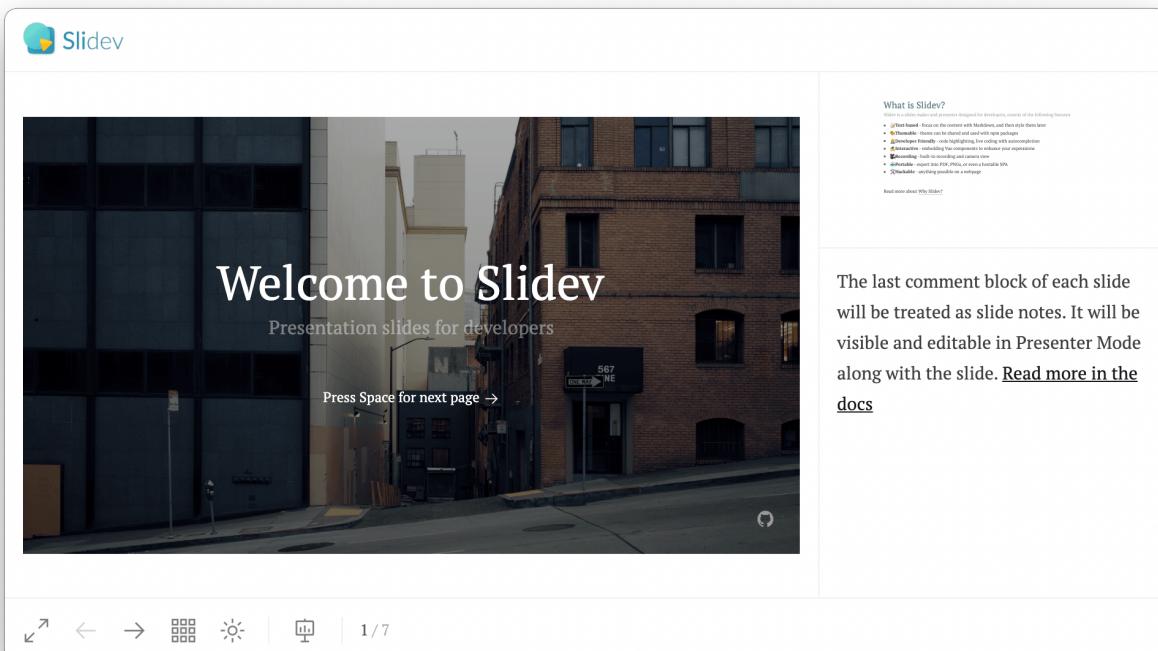
## Обзор слайдов

Нажав на о или кликнув на кнопку в панели навигации, вы можете просмотреть свои слайды и без труда переключиться на них

[Go to TOC](#)

# Режим докладчика

Нажмите на кнопку на панели навигации или перейдите вручную на <http://localhost:3030/presenter>, чтобы войти в режим докладчика. Каждый раз, когда вы входите в режим докладчика, другие экземпляры страницы будут автоматически синхронизироваться с докладчиком.



---

[Go to TOC](#)

# Запись

Slidev имеет встроенную поддержку записи и просмотра камеры. Вы можете использовать их для простой записи презентации.

## Просмотр камеры

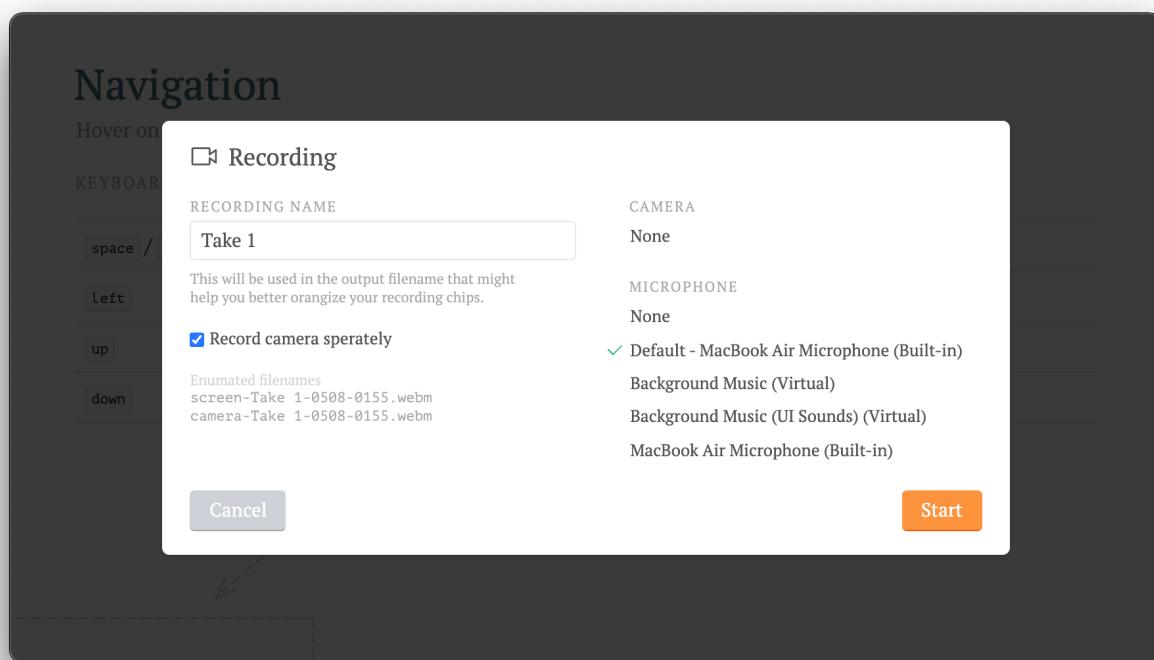
Нажмите кнопку на панели навигации, чтобы отобразить изображение с камеры в презентации. Вы можете переместить его и настроить его размер, используя меню в правом нижнем углу. Размер и позиция будут сохраняться в `localStorage` и, следовательно, будут иметь актуальные настройки при обновлениях страницы, так что не нужно об этом беспокоиться.

## Запись

Clicking the button in the navigation panel will bring up a dialog for you. Here you can choose to either record your camera embedded in your slides or to separate them into two video files.

При нажатии на кнопку на панели навигации откроется диалоговое окно. Здесь вы можете выбрать запись камеры, встроенную в слайды, либо разделить их на два видеофайла.

Эта функция работает на [RecordRTC](#) и использует [WebRTC API](#).



---

[Go to TOC](#)

# Markdown синтаксис

Слайды описываются в **одном markdown-файле** (по умолчанию `./slides.md`).

Вы можете использовать обычный [Markdown](#), с дополнительной поддержкой встроенного HTML и Vue-компонентов. Также поддерживается стилизация с использованием [Windi CSS](#). Используйте `---` с добавлением новой строки для разделения слайдов.

```
# Slidev

Привет, мир!

---

# Page 2

Использование блоков кода для подсветки

//```ts
console.log('Привет, мир!')
//```

---

# Page 3

Вы можете напрямую использовать Windi CSS и Vue-компоненты для стилизации и
улучшения своих слайдов.

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

## Front Matter и Шаблоны

Укажите макеты и другие метаданные для каждого слайда, преобразовав разделители в [front matter блоки](#). Каждый frontmatter блок начинается с тройного тире и заканчивается ими же. Тексты между ними представляют собой объекты данных в формате [YAML](#). Например:

```
---
layout: cover
---

# Slidev

Это титульная страница.

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

# Page 2
```

Это страница с макетом `center` и фоновым изображением.

---

# Page 3

Это дефолтная страница без каких-либо дополнительных метаданных.

Подробная информация описана на странице [кастомизации](#).

## Блоки кода

Одна из главных причин, по которой я создала Slidev – это то, что мой код должен хорошо выглядеть на слайдах. Так, как вы и ожидали, вы можете использовать блоки кода Markdown, чтобы подсветить свой код.

```
//``ts
console.log('Привет, мир!')
//``
```

## Подсветка строк

Мы поддерживаем [Prism](#) и [Shiki](#) в качестве подсветки синтаксиса. Обратитесь к [разделу подсветки](#) для получения более подробной информации.

Чтобы выделить определённые строки, просто добавьте номера строк в скобки `{}`. Нумерация строк начинается с 1.

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

Чтобы изменить выделение в несколько шагов, вы можете использовать `|` для их разделения. Например

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

Сначала будет выделено `a: Ref<number> | number` и `b: Ref<number> | number`, а затем `return computed(() => unref(a) + unref(b))` после одного клика, и наконец, весь блок. Подробнее в [руководстве анимации по кликам](#).

## Monaco редактор

Всякий раз, когда вы хотите внести какие-либо изменения в презентацию, просто добавьте `{monaco}` после указания языка подсветки - это превратит блок в полнофункциональный Monaco редактор!

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

Подробнее в [конфигурации Monaco](#).

## Встроенные стили

Вы можете использовать тег `<style>` напрямую в Markdown, чтобы переопределить стили для **текущего слайда**.

```
# Это красный заголовок

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

---

# На следующем слайде стили заголовка останутся прежними
```

Тег `<style>` в Markdown всегда является [scoped](#). Чтобы изменить глобальные стили, прочтите [раздел кастомизации](#)

Работая с [Windi CSS](#), вы можете напрямую использовать вложенные CSS и [directives](#) (например, `@apply`)

```
# Slides

> Привет, `мир`

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

## Статические ресурсы

Как и в случае с markdown, вы можете вставлять изображения, использующие удаленный или локальный URL.

Для удаленных ресурсов встроенный `vite-plugin-remote-assets` кэширует их на диск при первом запуске, поэтому позже у вас будет мгновенная загрузка даже для больших изображений.

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

Для локальных ресурсов поместите их в папку `public` и укажите их через **ведущий слэш**.

! [Local Image](pic.png)

Если вы хотите применить кастомные размеры или стили, вы можете вставить их через `<img>` тег

```

```

## Заметки

Вы также можете делать заметки для каждого слайда. Они будут отображаться в [Режиме презентации](#), чтобы вы могли ссылаться на них во время презентаций.

В Markdown последний блок комментария в каждом слайде будет считаться как примечание.

```
---
layout: cover
---

# Страница 1

Это титульная страница.

<!-- Это заметка -->

---

# Страница 2

<!-- Это НЕ заметка, потому что стоит перед содержанием слайда. -->

Это вторая страница

<!--
А это другая заметка
-->
```

## Иконки

Slidev позволяет вам получить доступ практически ко всем популярным наборам иконок с открытым исходным кодом **прямо** в вашем markdown. Работает это через `vite-plugin-icons` и [Iconify](#).

Именование следует [Iconify](#) преобразованию `{collection-name}-{icon-name}`. Например:

- `<mdi-account-circle />` - из [Material Design Icons](#)
- `<carbon-badge />` - из [Carbon](#)
- `<uim-rocket />` - из [Unicons Monochrome](#)
- `<twemoji-cat-with-tears-of-joy />` - из [Twemoji](#)
- `<logos-vue />` - из [SVG Logos](#)
- И многое другое...

Просмотрите и найдите все иконки, доступные с [Icônes](#).

## Стилизация иконок

Вы можете стилизовать иконки так же, как и другие HTML элементы. Например:

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

## Слоты

Доступны с версии v0.18

Некоторые шаблоны могут предоставлять несколько дополнительных точек, используя [именованные слоты Vue](#).

Например, в [шаблоне two-cols](#) у вас может быть два столбца рядом, левый (`default` слот) и правый (`right` слот).

```
---
layout: two-cols
---

<template v-slot:default>
# Левый
Это отобразится слева
</template>
<template v-slot:right>
# Правый
Это отобразится справа
</template>
```

### Левый

Это отобразится слева

### Правый

Это отобразится справа

Мы также предоставляем сокращённый синтаксический сахар `::name::` для имени слота. Следующий пример работает точно так же, как и предыдущий.

```
---  
layout: two-cols  
---  
  
# Левый  
  
Это отобразится слева  
  
::right::  
  
# Правый  
  
Это отобразится справа
```

Вы также можете явно указать слот по умолчанию и в произвольном порядке

```
---  
layout: two-cols  
---  
  
::right::  
  
# Правый  
  
Это отобразится справа  
  
::default::  
  
# Левый  
  
Это отобразится слева
```

## Конфигурации

Все необходимые конфигурации можно определить в Markdown файле. Например:

```
---  
theme: serif  
layout: cover  
background: 'https://source.unsplash.com/1600x900/?nature,water'  
---  
  
# Slidev  
  
Это титульная страница.
```

Подробнее о [конфигурации frontmatter](#)

## LaTeX

Slidev поставляется с встроенной поддержкой LaTeX на базе [KaTeX](#).

## Строка

Оберните своё LaTeX выражение одним `$` с каждой стороны для строчного рендеринга.

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

## Блок

Используйте 2 ( `$$` ) для блочного рендеринга. В этом режиме используются более крупные символы и результат центрируется.

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

Подробнее: [Демо](#) | [KaTeX](#) | [markdown-it-katex](#)

## Диаграммы

Вы также можете создавать диаграммы / графики из текстовых описаний в вашем Markdown, используя [Mermaid](#).

Блоки кода, помеченные как `mermaid`, будут преобразованы в диаграммы, например:

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

Кроме того, вы можете передать ему объект параметров, чтобы указать масштабирование и тему. Синтаксис объекта - это литерал объекта JavaScript, вам нужно будет добавить кавычки ( `'` ) для строк и использовать запятую ( `,` ) между ключами.

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

Подробнее: [Демо](#) | [Mermaid](#)

# Множественные вхождения

Доступны с версии v0.15

Вы можете разделить свой `slides.md` на несколько файлов и организовать их по своему усмотрению.

`slides.md` :

```
# Страница 1
Это обычная страница
---
src: ./subpage2.md
---

<!-- эта страница будет загружена из './subpage2.md' -->
Контент в этом месте будет проигнорирован
```

`subpage2.md` :

```
# Страница 2
Эта страница из другого файла
```

## Frontmatter слияние

Вы можете использовать frontmatters как со своей основной страницы, так и с внешней markdown страницы. Если в них есть одинаковые ключи, то **приоритет будут иметь ключи из основной записи**. Например

`slides.md` :

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

`cover.md` :

```
---
layout: cover
background: https://sli.dev/foo.png
---
# Обложка
Страница обложки
```

Они будут эквивалентны следующей странице:

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

# Обложка

Страница обложки

## Переиспользование страниц

Благодаря поддержке множественных вхождений переиспользование страниц может быть довольно простым. Например:

```
---  
src: ./cover.md  
---  
  
---  
src: ./intro.md  
---  
  
---  
src: ./content.md  
---  
  
---  
# переиспользование  
src: ./content.md  
---
```

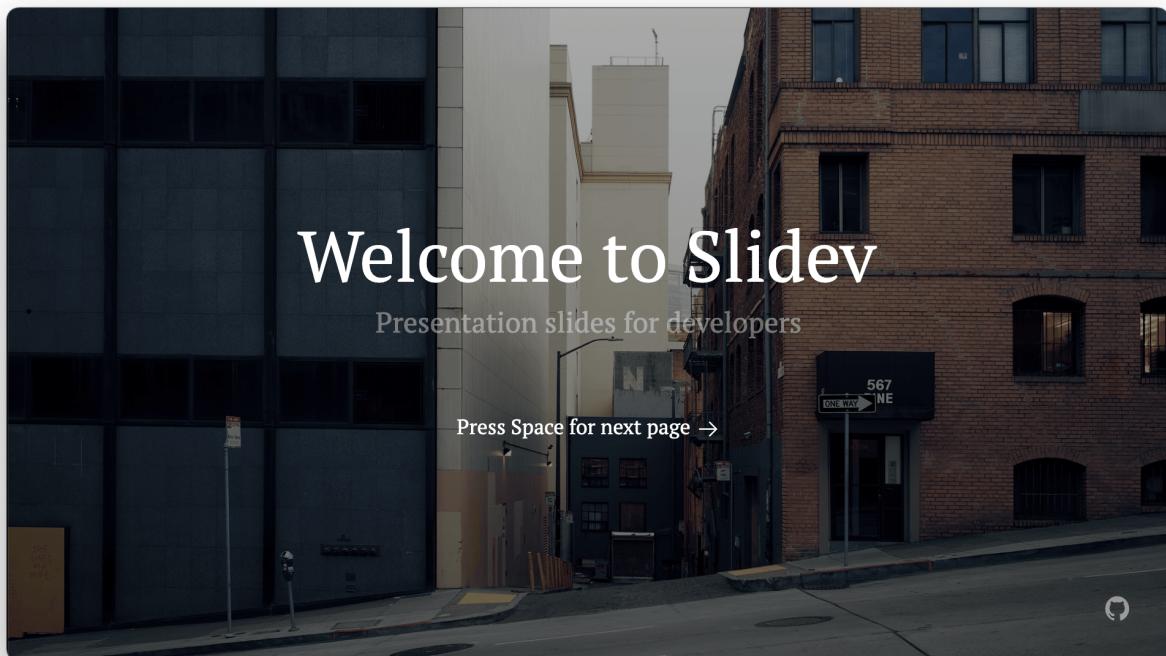
# Почему Slidev

Есть множество богатых функционалом, универсальных WYSIWYG редакторов презентаций, например [Microsoft PowerPoint](#) или [Apple Keynote](#). Они не плохо работают для создания красивых презентаций с анимациями, графиками, и множества других штук, оставаясь при этом очень интуитивными и лёгкими в изучении. Так зачем вообще нужен Slidev?

Slidev направлен на поддержку гибкости и интерактивности для разработчиков, чтобы делать их презентации более интересными, выразительными и привлекательными. Используя инструменты и технологии, с которыми они уже знакомы.

При работе с WYSIWYG редакторами очень легко отвлечься на варианты стилизации. Slidev исправляет это, разделяя контент и визуальную часть. Что позволяет вам сфокусироваться на чем-то одном, а также даёт возможность переиспользовать темы из сообщества. Slidev не стремится полностью заменить другие конструкторы презентаций. Скорее, он фокусируется на разнообразии сообщества разработчиков новыми возможностями.

## Slidev



Это лишь часть ключевых особенностей Slidev:

## Markdown-based

Slidev использует расширенный Markdown формат для хранения и организации ваших презентаций в простом текстовом файле. Это даёт вам сфокусироваться на создании контента. Поскольку контент и стили разделены, это позволяет вам переключаться между темами без особых усилий.

Подробнее о [Markdown синтаксисе Slidev](#).

## Темизация

Темы для Slidev могут распространяться и устанавливаться, используя npm-пакеты. Вам понадобится одна строка в конфигурации, чтобы применить их.

Посмотрите [галерею тем](#) или [узнайте, как написать свою тему](#).

## Удобство разработки

Slidev предоставляет разработчикам первоклассную поддержку фрагментов кода. Он поддерживает как [Prism](#), так и [Shiki](#), чтобы добиться идеальной pixel-perfect подсветки синтаксиса, при этом сохранив возможность изменять код в любое время. Благодаря встроенному редактору [Monaco](#), он также даёт вам возможность выполнять кодирование / демонстрацию в реальном времени в вашей презентации с автозаполнением, наведением курсора и даже с поддержкой проверки типов TypeScript.

Узнайте больше о [подсветки и конфигурации Monaco](#).

## Скорость

Slidev разработан на [Vite](#), [Vue 3](#) и [Windi CSS](#), которые в совокупности дают прекрасный опыт разработки. Каждое внесённое вами изменение **мгновенно** отразится на слайдах.

Почитайте подробнее о [нашем техническом стеке](#)

## Интерактивность

Вы можете написать кастомные Vue-компоненты и использовать их непосредственно внутри вашего markdown-файла. Вы также можете взаимодействовать с ними внутри вашей презентации, чтобы исполнить вашу идею в более интересной и интуитивной форме.

## Поддержка записи

Slidev обеспечивает встроенную запись и просмотр с камеры. Вы можете поделиться своей презентацией с камерой внутри или записать и сохранить их отдельно для экрана и камеры. Всё сразу, никаких дополнительных инструментов не требуется.

Узнайте больше о [записи здесь](#).

## Портативность

Экспортируйте ваши слайды в PDF, PNGs, или даже a hostable Single-page Application (SPA) with a single command, and share them anywhere.

Read more about that in the [exporting docs](#).

## Расширяемость

Будучи сделанным на веб-технологиях, всё что вы можете сделать в веб-приложение – можно реализовать и в Slidev. Например: WebGL, API запросы, iframes, или даже live sharing. Все зависит от вашего воображения!

## Попробуйте сами

Лучше один раз увидеть, чем сто раз услышать – поиграйте со Slidev. Вас отделяет всего одна команда:

```
$ npm init slidev
```

Или посмотрите демо:

Slidev First Preview Demo

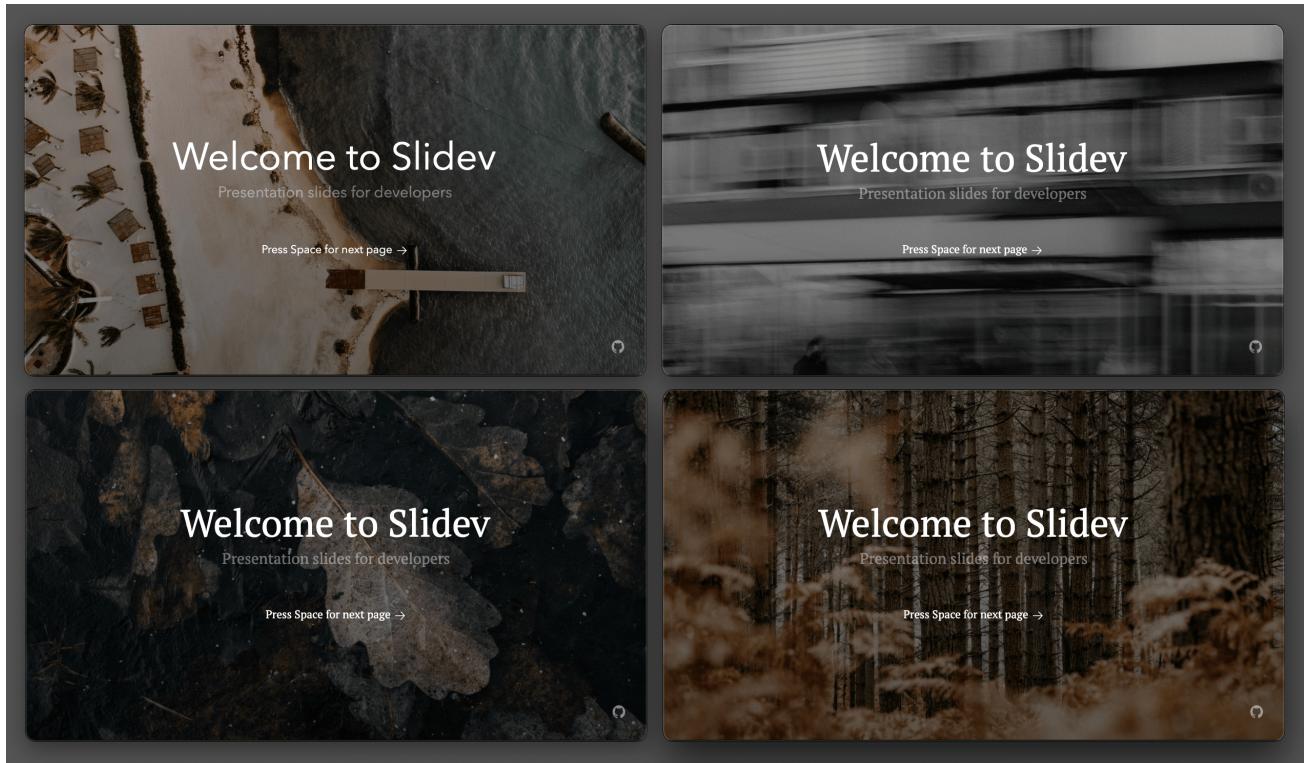


---

[Go to TOC](#)

# Модерируемые обложки

Мы отобрали несколько обложек, чтобы продемонстрировать наш стартовый шаблон.



---  
# случайное изображение из нашей коллекции  
[background: https://source.unsplash.com/collection/94734566/1920x1080](https://source.unsplash.com/collection/94734566/1920x1080)  
---

Если вам понравился какой-либо из них, ознакомьтесь с нашей [Unsplash коллекцией](#) и узнайте их авторов.

---

[Go to TOC](#)

# Образовательные ресурсы

## English

### Videos

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



### Articles

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

## 中文

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

## 日本语

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

---

[Go to TOC](#)

# Примеры

Презентации сделанные на Slidev.

---

[Go to TOC](#)

# Галерея тем

Просмотрите потрясающие темы, доступные для Slidev.

Узнайте больше о том, [как использовать тему](#) или [как написать собственную](#) и поделиться ею с сообществом!

## Официальные темы

### Темы сообщества

А это модерируемые темы, созданные сообществом.

### Больше тем

Посмотрите все [темы, доступные на NPM](#).

---

[Go to TOC](#)

# Использование тем

Менять темы в Slidev очень легко. Всё, что вам нужно сделать, это добавить поле `theme:` в frontmatter.

```
---  
theme: serif  
---
```

Вы можете запустить сервер, который предложит вам установить тему автоматически

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

или установите тему вручную через

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

И всё, наслаждайтесь новой темой! Для получения дополнительной информации об использовании вы можете прочитать README-файл темы.

Хотите поделиться своей темой? Узнайте подробнее [how to write a theme](#).

## Извлечение темы

Если вы хотите получить полный контроль над текущей темой, вы можете **извлечь** её в свою локальную файловую систему и изменить её по своему усмотрению. Выполнив следующую команду

```
$ slidev theme eject
```

Тема, которую вы сейчас используете, будет извлечена в `./theme`, а frontmatter будет изменён на

```
---  
theme: ./theme  
---
```

Это также может быть полезно, если вы хотите создать тему на основе существующей. Если собираетесь делать, не забудьте упомянуть оригинальную тему и автора :)

## Локальная тема

Как вы могли узнать из предыдущего раздела, у вас может быть локальная тема для вашего проекта. Укажите **относительный путь** в поле вашей темы.

```
---  
theme: ./path/to/theme  
---
```

Подробнее на странице [создания темы](#).

---

[Go to TOC](#)

# Создание темы

Для начала мы рекомендуем использовать наш генератор для создания вашей первой темы.

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

Затем вы можете изменить и поиграться с ней. Вы также можете посмотреть [официальные темы](#) в качестве примеров.

## Возможности

Тема может содержать:

- Глобальные стили
- Дефолтные конфигурации (шрифты, цветовые схемы, подсветки, и т.д.)
- Кастомные шаблоны или замена существующих
- Кастомные компоненты или замена существующих
- Расширение конфигурации Windi CSS
- Настройка таких инструментов, как Monaco или Prism

## Соглашения

Темы публикуются в реестре npm, и они должны соответствовать приведённым ниже соглашениям:

- Имя пакета должно начинаться с `slidev-theme-`, например: `slidev-theme-awesome`
- Содержать `slidev-theme` и `slidev` в поле `keywords` вашего `package.json`

## Настройка

Чтобы настроить тестовую песочницу для вашей темы, вы можете создать `example.md` со следующим frontmatter, чтобы сообщить Slidev, что вы используете текущий каталог в качестве темы.

```
---  
theme: ./  
---
```

При желании вы также можете добавить несколько скриптов в свой `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"
  }
}
```

Чтобы опубликовать свою тему, просто запустите `npm publish`, и всё. Процесс сборки не требуется (это означает, что вы можете напрямую публиковать файлы `.vue` и `.ts`, Slidev достаточно умён, чтобы понять их).

Пункты добавления темы следуют тем же соглашениям, что и локальная настройка, подробнее в [документации по соглашениям об именовании](#).

## Конфигурации по умолчанию

Доступно с версии v0.19

Тема может предоставлять дефолтные [конфигурации](#) через `package.json`.

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

Шрифты будут автоматически импортированы из [Google Fonts](#).

Подробнее о [шрифтах](#) и [frontmatter](#) конфигурации

## Метаданные темы

### Цветовая палитра

По умолчанию Slidev предполагает, что темы поддерживают как светлый, так и тёмный режим. Если вы хотите, чтобы ваша тема была представлена только в разработанной цветовой схеме, вам нужно будет явно указать её в `package.json`.

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

Чтобы получить доступ к тёмному режиму при стилизации вашей темы, вы можете обернуть CSS, зависящий от тёмного режима, внутри класса `dark`:

```
/* основные CSS стили */

html:not(.dark) {
    /* CSS для светлой темы */
}

html.dark {
    /* CSS для тёмной темы */
}
```

Slidev переключает класс `dark` в элементе `html` страницы для переключения цветовой схемы.

## Подсветка

В теме также предусмотрены цвета для подсветки синтаксиса. Мы поддерживаем как [Prism](#), так и [Shiki](#). Подробнее в [документации по подсветке синтаксиса](#).

Вы можете использовать как один из них, так и оба. Примеры конфигураций дефолтной темысмотрите в `./styles/prism.css` / `./setup/shiki.ts`.

Также не забудьте указать поддерживаемую подсветку в вашем `package.json`

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

## Версионирование Slidev

Если тема ссылается на определённую функцию Slidev, которая была недавно добавлена, то вы можете установить минимальную версию Slidev, необходимую для правильной работы вашей темы:

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

Если пользователи будут использовать более старую версию Slidev, то будет выведена ошибка.

# Colophon

This book is created by using the following sources:

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