



用 Slidev 写 PPT



我是页眉

PPT 的功能

我是页脚

什么是 Slidev, 有哪些特点?

简而言之, Slidev 就是一个工具库, 基于 Node.js、Vue.js , 使用 Markdown 语法 辅以 tailwindcss 等模块来制作 PPT。

-  **Markdown 基本语法支持**
-  **TailwindCSS 灵活样式**
-  **可选主题** - 当前只有官方主题可用 可访问几乎所有的开源图标集
-  **图标** —— 能够直接从任意图标库中获取图标
-  **可交互** - 嵌入 Vue 组件
-    **开发者友好** - 内置代码高亮(无自动补全功能)
-  **演讲者模式**
-  **跨平台** - 导出 PDF、PNG 和单页面应用

1. 安装和启动

Slidev 需要 Node.js 的版本 $\geq 14.0.0$

①. npm init slidev@latest

```
MINGW64:/d/FHW/sliderv_test
Lenovo@DESKTOP-VQ3V17E MINGW64 /d/FHW/sliderv_test
$ npm init slidev@latest
npm WARN exec The following package was not found and will be installed: create-slidev@latest

  •▲
slidev Creator v0.28.10

? Project name: » sliderv_sliderv_test
✓ Project name: ... sliderv_test
Scaffolding project in sliderv_test ...
Done.

? Install and start it now? » (Y/n)y
✓ Install and start it now? ... yes
? Choose the agent » - Use arrow-keys. Return to submit.
>   npm
    yarn
    pnpm
✓ Choose the agent » npm
@稀土掘金技术社区
```

②. npm run dev

```
PS D:\SiJiBao\sliderv\sliderv_test> npm run dev
> dev
> sliderv --open

  •▲
Slidev v0.28.10

theme  @sliderv/theme-Serif
entry   D:\SiJiBao\sliderv\sliderv_test\slides.md

slide show      > http://localhost:3030/
presenter mode > http://localhost:3030/presenter
remote control > pass --remote to enable

shortcuts       > _restart | _open | _edit  @稀土掘金技术社区
```

2. 项目结构

```
your-slides/
├── components/      # 自定义组件
├── global-bottom.vue # 页脚
└── global-top.vue   # 页眉
└── slides.md        # 幻灯片主入口
```

I. components

此目录中的组件可以在`markdown`中直接使用，其组件名与文件名相同。

```
your-slides/
├── ...
└── components/
    └── counter.ts
```

- 10 +

II. 页眉页脚

约定：根目录新增文件名为 *global-top.vue* / *global-bottom.vue* 即为页眉页脚

```
your-slides/
├── ...
├── components/
└── global-bottom.vue
└── global-top.vue
```

III. slides.md，入口文件

Markdown 语法 + TailwindCss

1. 常用 MarkDown 语法

1.1 分页

第一页

第二页

第三页

1.2 代码块、内联样式

```
interface TableSearch {  
    settlementBillNo: string | number; //采购结算单号  
    warehouseBillNo: string | number; //所属入库单  
    merchandiseName: string; //商品名称  
    supplierName: string; //供应商名称  
    paymentMode: string; //是否可线上支付  
}
```

内联样式

1.3. 自定义布局

layout:two-cols(自定义布局名称)



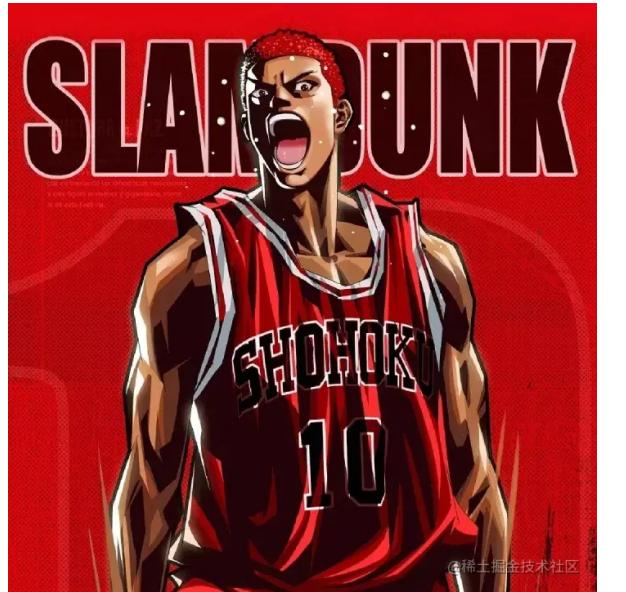
333

999ee



我是页眉

1.4. 静态资源



我是页脚

1.5. 图表

right / space

1 下一个动画或幻灯片

left / shift space

2 上一个动画或幻灯片

up

3 上一张幻灯片

down

4 下一张幻灯片

2. 导航

将鼠标悬停在左下角以查看导航的控制面板

快捷键	按钮	说明
f		切换全屏
right / space		下一动画或幻灯片
left		上一动画或幻灯片
up	-	上一张幻灯片
down	-	下一张幻灯片
o		切换 幻灯片总览
d		切换 暗黑模式
-		切换 摄像头视图
-		演讲录制
-		进入 演讲者模式
-		切换 集成编辑器
-		下载幻灯片 (仅在 单页 (SPA) 构建 中支持)
-		显示该演示文稿的信息
-		显示设置菜单

3. 动画.

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



Slidev

Slidev 会预加载下一张幻灯片以提高性能，导致动画无法被看见，可以禁用幻灯片预加载或使用v-if来控制

```
--- preload: false ---  
<div  
  v-if="$slidev.nav.currentPage === 12"  
  v-motion  
  :initial="{ x: -80 }"  
  :enter="{ x: 0 }"  
>  
  Slidev  
</div>
```

4.部署(Netlify、Vercel、GitHub Pages)

4.1 Netlify 部署

The screenshot shows the Netlify site overview for the deployment 'bucolic-brioche-876425'. At the top, there's a blue diamond icon followed by the text 'Far > bucolic-brioche-876425'. Below this is a navigation bar with links: Site overview (underlined), Deployments, Plugins, Functions, Identity, Forms, Large Media, and Split Testing. The main content area displays the deployment name 'bucolic-brioche-876425' and its URL '<https://bucolic-brioche-876425.netlify.app>'. It also mentions that it deploys from GitHub and was last published at 3:25 PM. To the right of this text is a small thumbnail image showing a silhouette of a person holding balloons next to a large '2022' with a crown on top. At the bottom of the screenshot, there are two buttons: 'Site settings' and 'Domain settings'. A watermark '@稀土掘金技术社区' is visible at the bottom right.

4.2 Vercel 部署

4.3 通过 Github Action 部署到 GitHub Pages

a. 创建 .github/workflows/deploy.yml 文件

Github 只要发现该目录中由 yml 文件就会自动运行该文件。

```
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 }} //下一页中配置
```

b. 获取密钥并存储到 Github 仓库中

Settings / Developer settings

GitHub Apps
OAuth Apps
Personal access tokens

Personal access tokens

Tokens you have generated that can be used to access the GitHub API.

通过 GitHub Action 将你的幻灯片部署到 GitHub Pages — repo
Last used within the last week [Delete](#)
Expires on Thu, May 5 2022.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Select scopes

Scopes define the access for personal tokens. Read more about C

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflow

Generate new token [Revoke all](#)

admin:gpg_key Full control of public user
 write:gpg_key Write public user GPG key:
 read:gpg_key Read public user GPG keys

Generate token [Cancel](#)

获取到token。

@稀土掘金技术社区

c. 给项目设置 token

The screenshot shows the 'General' settings page for a GitHub repository. The 'Secrets' section is highlighted with a red box. Other sections visible include 'Access', 'Collaborators', 'Moderation options', 'Code and automation' (with 'Branches', 'Tags', 'Actions', 'Webhooks', 'Environments', and 'Pages'), 'Security' (with 'Code security and analysis' and 'Deploy keys'), and a sidebar with 'Actions' and 'Dependabot'.

Actions secrets / New secret

Name
ACCESS_TOKEN

Value
和deploy.yml配置的token一致

Add secret

d.代码提交到 GitHub 后，进入.github/workflows/deploy.yml 点击 View runs 查看部署情况

The screenshot shows the GitHub repository `far_okr_sliderv` with the file `.github/workflows/deploy.yml` open. The file contains a workflow for deploying pages:

```

1 name: Deploy pages
2 on: push
3 jobs:
4   deploy:
5     runs-on: ubuntu-latest
6     steps:
7       - uses: actions/checkout@v2
8       - uses: actions/setup-node@v2
9         with:
10           node-version: "14"
11       - name: Install dependencies
12         run: npm install
13       - name: Build
14         run: npm run build
15       - name: Deploy pages
16         uses: crazy-max/ghaction-github-pages@v2
17         with:
18           build_dir: dist
19         env:
20           GITHUB_TOKEN: ${{ secrets.ACCESS_TOKEN }}
  
```

The `View runs` button is highlighted with a red box.

The GitHub Actions page for the repository shows the `Actions` tab selected. It lists the workflow `pages-build-deployment` with 14 runs. One specific run is highlighted with a red box:

- pages build and deployment** (pages-build-deployment #14: by github-pages (bot))

The run was triggered 8 hours ago and completed 1m 0s ago.

The detailed view of the workflow run shows the following steps:

```

graph TD
    build[build] --> reportBuildStatus[report-build-status]
    reportBuildStatus --> deploy[deploy]
    deploy --> artifact[github-pages]
  
```

The `deploy` step is highlighted with a red box. The artifact produced is named `github-pages` and is 2.35 MB in size.

我是页眉

我是页脚