



# 快速开始指南

这份指南将帮助你在 5 分钟内 运行起 PDF 预览插件项目！



## 前置要求

在开始之前，请确保你的电脑上已安装：

- **Node.js** (v16 或更高版本)
- **npm** (通常随 Node.js 一起安装)
- **代码编辑器** (推荐 VS Code)

## 检查环境

打开终端（命令提示符），运行以下命令：

```
node -v      # 应该显示 v16.x.x 或更高  
npm -v      # 应该显示 8.x.x 或更高
```

如果没有安装，请访问 <https://nodejs.org/> 下载安装。

## 🎯 方式一：Vue3 版本（推荐）

### 步骤 1: 进入项目目录

```
cd vue3-version
```

### 步骤 2: 安装依赖

```
npm install
```

这可能需要 1-2 分钟，请耐心等待... 🚧

### 步骤 3: 启动项目

```
npm run dev
```

### 步骤 4: 打开浏览器

看到类似这样的提示：

```
VITE v5.0.0  ready in 500 ms  
→ Local:  http://localhost:3000/  
→ Network: use --host to expose
```

浏览器会自动打开，或手动访问 <http://localhost:3000>

## 步骤 5: 测试功能

1. 点击 "加载示例 PDF" 按钮
2. 等待 PDF 加载完成
3. 尝试以下操作：
  - 点击翻页按钮
  - 点击放大/缩小按钮
  - 点击缩略图按钮
  - 使用键盘方向键翻页

🎉 恭喜！Vue3 版本运行成功！

## 🎯 方式二：原生 JS 版本

### 步骤 1: 进入项目目录

```
cd vanilla-js-version
```

### 步骤 2: 安装依赖

```
npm install
```

### 步骤 3: 启动项目

```
npm run dev
```

### 步骤 4: 打开浏览器

访问显示的 URL (通常是 <http://localhost:3000>)

## 步骤 5: 测试功能

点击 "加载示例 PDF" 开始使用！

🎉 恭喜！原生 JS 版本运行成功！

## 💡 使用本地 PDF 文件

### 方法 1: 通过界面上传

1. 点击 "选择 PDF 文件" 按钮
2. 从你的电脑中选择一个 PDF 文件
3. 文件将立即显示

## 方法 2: 修改代码

编辑 `App.vue` 或 `index.html`, 将 URL 改为你的 PDF 文件路径:

```
// Vue3 版本 (App.vue)
const pdfUrl = ref("/path/to/your.pdf");

// 或使用自动加载示例
onMounted(async () => {
  try {
    // 自动加载默认PDF文件
    const defaultFilePath = '/sample.pdf'; // 确保文件位于public目录
    const response = await fetch(defaultFilePath);
    const arrayBuffer = await response.arrayBuffer();
    const blob = new Blob([arrayBuffer], { type: 'application/pdf' });
    await loadPdf(blob);
  } catch (error) {
    console.error('加载默认PDF失败:', error);
  }
});

// 原生 JS 版本 (index.html)
const viewer = new PdfViewer({
  container: "#pdf-viewer",
  url: "/path/to/your.pdf",
  // 添加更多配置选项
  enableWebGL: true,      // 启用WebGL加速
  enableThumbnails: true, // 启用缩略图
  enableSearch: true       // 启用搜索功能
});
```

## 键盘快捷键

按键	功能
←	上一页
→	下一页
+	放大
-	缩小
0	重置缩放
Enter	跳转页面 (输入页码后)

# ！常见问题

## 问题 1: npm install 失败

解决方案:

```
# 清理缓存  
npm cache clean --force  
  
# 使用淘宝镜像（中国用户）  
npm install --registry=https://registry.npmirror.com  
  
# 或使用 cnpm  
npm install -g cnpm --registry=https://registry.npmirror.com  
cnpm install
```

## 问题 2: 端口 3000 已被占用

解决方案:

```
# 修改端口号  
# 编辑 vite.config.js，将 port: 3000 改为其他端口  
port: 3001
```

## 问题 3: PDF 加载失败

可能原因:

- PDF 文件损坏
- PDF 文件受密码保护
- 网络连接问题（在线 PDF）
- 跨域问题

解决方案:

- 尝试使用其他 PDF 文件
- 确保网络连接正常
- 使用本地 PDF 文件

## 问题 4: 页面显示空白

解决方案:

1. 打开浏览器开发者工具 (F12)
2. 查看 Console 标签页的错误信息
3. 检查是否有 Worker 加载失败的错误
4. 如果遇到Canvas相关错误，检查浏览器是否支持Canvas API
5. 确保PDF.js Worker配置正确

```
// vue3版本正确配置
import * as pdfjsLib from 'pdfjs-dist';
import pdfjsworker from 'pdfjs-dist/build/pdf.worker.entry';

pdfjsLib.GlobalWorkerOptions.workersSrc = pdfjsworker;
```

## 6. 对于中文乱码问题，添加cMap配置

```
const pdfDoc = await pdfjsLib.getDocument({
  data: arrayBuffer,
  cMapUrl: 'https://unpkg.com/pdfjs-dist@3.4.120/cmaps/',
  cMapPacked: true
}).promise;
```



现在你已经成功运行了项目！接下来可以：

### 1. 阅读核心代码:

- Vue3 版本: `src/components/PdfViewer.vue` (重点关注Composition API的使用)
- 原生 JS 版本: `src/pdf-viewer.js` (学习类封装和事件处理)
- 配置文件: `vite.config.js` 和 `package.json` (了解构建配置)

### 2. 自定义样式: 修改 CSS 文件，打造你自己的风格

- Vue3版本: `src/styles/pdf-viewer.css`
- 原生JS版本: `src/styles/main.css`

### 3. 添加高级功能:

- 文本搜索和高亮
- 页面旋转和注释
- 导出和打印功能
- 水印添加和移除

### 4. 性能优化实践:

- 实现页面缓存机制
- 添加预加载功能
- 优化内存使用

### 5. 兼容性测试:

- 在不同浏览器中测试
- 在移动设备上验证
- 测试不同大小和复杂度的PDF文档

### 6. 准备面试:

- 总结项目亮点和技术难点
- 准备技术问题的回答
- 演示项目功能和你的贡献

# 学习路径建议

## 第 1 天: 熟悉项目

- 运行两个版本
- 测试所有功能
- 阅读 README 文档

## 第 2-3 天: 理解代码

- 阅读 PdfViewer 组件/类
- 理解 PDF.js 使用方法
- 了解 Canvas 渲染原理

## 第 4-5 天: 修改定制

- 修改样式和 UI
- 调整功能逻辑
- 添加自己的想法

## 第 6-7 天: 扩展功能

- 添加水印功能
- 实现文本搜索
- 支持移动端手势

## 第 8+ 天: 总结提升

- 编写项目文档
- 准备面试问答
- 上传到 GitHub



## 需要帮助?

如果遇到问题:

1. 检查本文档的「常见问题」部分
2. 阅读 `实施指南.md` 获取详细信息
3. 查看浏览器控制台的错误信息
4. 搜索相关技术文档

## ✿ 开始你的前端之旅!

记住: 每个优秀的开发者都是从第一行代码开始的。

现在就开始探索这个项目, 理解每一行代码的含义, 然后把它变成真正属于你自己的作品!

祝你学习愉快, 求职顺利! 