

03. 使用Obsidian发布个人笔记

配置 MkDocs 以兼容 Obsidian 语法

安装插件

1. 打开Obsidian设置，点击第三方插件（Community plugins）,关闭安全模式
2. 点击浏览（Browse）,搜索 Enveloppe
3. 点击后安装（install）并启用（enable）

获取Github Token

1. 登录Github
2. 点击右上角头像 -> Settings -> Developer settings (最下面) -> Personal access tokens-> Tokens (classic)
3. 点击Generate new tokens(classic)
4. 登录
5. Note随意填一个就好了
6. Expiration (过期时间) 选 No expiration , 或者视你个人情况
7. Scopes(权限)勾选：
repo (全选, 包含 repo:status, repo_deployment 等)
workflow (如果要触发自动构建)
8. 划到最下方，点击生成，复制 ghp_ 开头的字符串，保存好，只会出现一次

配置插件

回到Obsidian，打开 Enveloppe 的设置页,配置方面我改了又改，感觉AI前后说法不一，可能是我对功能要求不一样吧

GitHub config

1. 依次输入GitHub username, Repository name, GitHub token (你刚复制的那串字符)
2. Main branch选择main
3. 点击下方test connection检查，右上角报绿就连接成功了

File paths

1. File tree in repository选择Obsidian Path

2. Root folder输入docs

Content

1. 打开 [[Wikilinks]] to [MDlinks](links) ,选择strict
2. 打开Markdown hard line break

Attachment & embeds

- Default attachment folder写docs/assets(如果没有就建一个assets文件夹)

发布文章

前置准备

这个我一开始没弄，搞了很久(Vscode里面改了之后可以用Vscode里面的源代码管理上传，也不强求在两个地方都写)

1. 在项目文件夹下（可以同时看到docs和mkdocs.yml）建一个requirements.txt，在Github仓库中也要
2. requirements.txt中输入

```
mkdocs-material  
mkdocs-awesome-pages-plugin  
mkdocs-ezlinks-plugin
```

3. 终端运行

```
pip install -r requirements.txt
```

4. 在.yml中添加如下代码

```
plugins:  
  - search  
  - ezlinks  
  - awesome-pages
```

5. 在Github仓库中的code页面再建一个文件夹，名称为： .github/workflows/publish.yml ，注意前面还有个点
文件中输入：

```

name: Publish to GitHub Pages

on:
  push:
    branches:
      - main # 盯着你的 main 分支，一旦有变动就开始工作

permissions:
  contents: write

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4

      - name: Configure Git Credentials
        run: |
          git config user.name github-actions[bot]
          git config user.email 41898282+github-
actions[bot]@users.noreply.github.com

      - uses: actions/setup-python@v5
        with:
          python-version: 3.x

      - name: Install dependencies
        # 这里会读取你之前建的 requirements.txt 安装插件
        # 如果你没建 requirements.txt，会自动安装基础包
        run: |
          if [ -f requirements.txt ]; then pip install -r requirements.txt; else
            pip install mkdocs-material mkdocs-awesome-pages-plugin; fi

      - name: Deploy site
        # 这一步把 Markdown 变成 HTML 并推送到 gh-pages 分支
        run: mkdocs gh-deploy --force

```

正常步骤

- 在文章开头输入以下代码

```

---
share: true
---
```

- 停留在笔记页面，按Ctrl P(mac:Cmd P)打开命令面板

3. 输入关键词Enveloppe，选择 Upload single current active note (发布当前文件)
然后等待即Github Actions变绿即可

设置快捷键

- 还是要输入上面那个代码
1. 打开Obsidian设置
 2. 在左侧找到快捷键 (Hotkeys)
 3. 在搜索框中输入Enveloppe
 4. 选择 Upload single current active note
 5. 输入喜欢的快捷键就好，我的是 Alt S

一些问题

- 用这个方法可能会把你仓库所有.md文档都传上去，所以建议仓库中无用的.md文件删除
- 图片如果是拉到笔记中注意图片格式，生成的图片是.JPG而拉进去的代码是.jpg，还是 Gemini帮我发现的
- 如果遇到一些传输上的问题(Obsidian的)可以Ctrl shift i调出来之后把console中的报红扔给AI，让它帮忙解释和给出修改方案
- 遇到一些其他问题也可以找AI，我结合Obsidian实操+写教程+与AI对线花了将近一天，AI真的能解决很多问题

索引设置

第一阶段：准备工作 (必做)

在开始改文件名之前，必须先清理掉 GitHub 上“手动挡”的残留，否则数字前缀不会生效。

1. 打开 GitHub 网页，找到你的 `mkdocs.yml` 文件。
 2. 点击编辑 (📝)。
 3. 删除 `nav:` 模块：找到 `nav:` 这行字，把它以及下面缩进的所有内容统统删掉。
 - 原理：如果不删，MkDocs 会优先听 `nav` 的指挥，忽略你的数字排序。
 4. 保存提交 (Commit changes)。
-

第二阶段：在 Obsidian 中排兵布阵

现在的核心逻辑是：文件名决定顺序，文件内容决定显示名称。

1. 开启自动更新链接 (防止断链)

因为我们要修改文件名，为了防止你文章里的图片或引用失效，请先检查这个设置：

- 设置 (Settings) -> 文件与链接 (Files & Links)。
- 确保“自动更新内部链接” (Automatically update internal links) 是开启 (On) 的。
- 确保“自动更新外部链接” (Automatically update external links) 是开启 (On) 的。

2. 给文件和文件夹编号

回到 Obsidian 的左侧文件列表，按照你想要的顺序，给文件名前面加上 `01.`、`02.` 等前缀。

实操演示：

假设你现在的结构很乱：

- 📁 docs
 - 📄 index.md (首页)
 - 📁 mkdocs网站搭建教程
 - 📄 关于我.md
 - 📁 Python学习

修改后的结构 (建议)：

- 📁 docs
 - 📄 index.md (首页通常不需要加数字，或者加 `00.index.md` 也可以，但一般保留原

- **docs**
 - `index.md` (首页通常不需要加数字，或者加 `00.index.md` 也可以，但一般保留原名)
 - `01. 关于我.md` (想让它排在首页后面)
 - `02. mkdocs网站搭建教程` (文件夹也可以加数字！)
 - `01. 前期准备.md` (文件夹内部的排序)
 - `02. 搭建过程.md`
 - `03. 常见问题.md`
 - `03. Python学习`
-

第三阶段：让菜单变漂亮 (隐藏数字)

你肯定不希望网站导航栏上显示 "01. 关于我"，太丑了。我们利用 MkDocs 的特性来“骗”过它。

. 这里的原理

- **排序时：**插件看文件名 (比如 `01. 关于我.md`)，所以顺序对了。
- **排序时：**插件看文件名 (比如 `01. 关于我.md`)，所以顺序对了。
- **显示时：**插件会看文件里的 H1 标题 (第一行 `# 标题`)。如果文件里有标题，它就用标题作为菜单名，忽略文件名。

2. 检查你的笔记内容

请点开你改名后的每一篇笔记，确保它们的第一行是这样的：

文件： `01. 关于我.md`

Markdown



```
# 关于我本人    <-- 网站菜单上会显示这几个字
```

这里是正文内容...

文件： `02. 搭建过程.md`

Markdown



```
# 详细搭建步骤    <-- 网站菜单上会显示这几个字
```

这里是正文...

注：Awesome Pages 插件通常也会自动移除文件夹名上的数字，所以文件夹叫 `02. mkdocs` 网站
搭建教程，网站上通常会自动显示为 `Mkdocs` 网站搭建教程）

第四阶段：发布与验收

1. **一键发布：** 在 Obsidian 里按下 `Alt + S` (Obsidian Mkdocs Publisher 插件)。
2. **等待构建：** 等待 GitHub Actions 跑完 (约 2 分钟)。
3. **查看效果：** 刷新你的网站。

你会看到：

1. **导航栏顺序** 变成了：首页 -> 关于我本人 -> Mkdocs网站搭建教程 -> Python学习。
2. **丑陋的数字** (`01.`, `02.`) 全部消失了。
2. **丑陋的数字** (`01.`, `02.`) 全部消失了。
3. **层级结构** 清晰明了。

💡 进阶小技巧

- **以后想插入文章怎么办？** 比如你想在 `01` 和 `02` 之间加一篇。
 - 你可以把新文章命名为 `01-1. 新文章.md`。
 - 或者乖乖把后面的 `02` 改成 `03`，`03` 改成 `04`... (这是数字法的唯一缺点，类似在图书馆插书)。
- **index.md 怎么处理？** `index.md` 永远默认是该文件夹的“封面”，它会自动排在所在文件夹的第一位，通常不需要加数字。