

个人网站搭建

使用MkDocs搭建个人网站流程

前期准备

- 拥有github账号
- 已下载Git(我一开始并未下载，后面补的，具体配置方法可以b站搜索)
- 配置了python环境
- 下载了vscode
- 新建一个文件夹A，然后复制A的地址

搭建过程

本地初步搭建

- 1.在vscode中打开终端(Terminal),在页面上方，未直接显示可以点开旁边的三个点，有显示
- 2.在终端中依次输入以下命令，每输一次按一次回车

```
pip install mkdocs
```

安装mkdocs

```
pip install mkdocs-material
```

安装插件，后续使用更美观的主题

```
cd 你复制的A的地址
```

定位到A文件夹

```
mkdocs new 取一个名称（比如mygarden，后续称为B吧）
```

创建项目文件夹，你会发现A文件夹中包含了你所建的B文件夹，复制B的地址

```
cd B的地址
```

定位到B文件夹

mkdocs server

按住ctrl点击那个网址就可以预览了

网址应该是这个：http://127.0.0.1:8000/

现在我们已经可以在本地看到我们搭建的一个默认网站了

主题更改

- 1.复制B的路径(主要是为了方便第二步快速找到目标文件夹)
- 2.在vscode中打开文件夹,用刚复制的文件路径定位到目标文件夹：文件（File）->打开文件夹（Open Folder）
- 选择打开文件夹B
- 在vscode中点击mkdocs.yml进行编辑,编辑后代码应为

```
site_name: My Docs
theme:
  name: material
```

- 保存后可以回到前面的网站查看效果，刷新一下，若无法打开，再次在终端中输入mkdocs serve即可，这个时候页面相对而言就更好看了,然后可以在site_name那里更改自己的网站名称呀。（如果没有成功，可以刷新几次，或者重新打开终端输入mkdocs serve，我就试了好几次，哈哈）
- 以后如果碰到更改theme后回到网站刷新也没有变化，可以新建终端或ctrl C之后再次输入mkdocs serve然后按住ctrl单击网址进入到网站就好了

将本地站点推到Github

将文件传至github

- 进入github，新建一个仓库（在home点击new），然后给仓库（repository）起一个名字（只要available就好，注意username.github.io只能创建一个网站，其他的可以创建多个（网址后面会加上/项目名），起好名之后直接点Create repository
- 在vscode中开启终端，依次输入以下代码

git init

初始化仓库

```
git add .
```

增加更改内容(注意上面的空格和点)

```
git commit -m"first"
```

添加推送时的注释

- 接下来回到github上的仓库，向下滑可以看见两个标题，选择``

```
...or push an existing repository from the command line
```

下的三行代码，依次复制到vscode的终端中（每复制一次都要回车一次，共三次回车，不要一次性复制三行代码）

- 再次回到github中刷新就可以看见我们的代码已经传过去了

部署

- 在vscode终端中输入以下代码

```
mkdocs gh-deploy
```

出现warning可以不用管

回到Github后点击Actions就可以看见我们的workflow（工作流）了，等到有绿色的√号时，我们点击Settings，左侧找到pages并点击就可以看到我们部署的网址了

网站优化

界面布置

- 我每次都会把当前进度的所有代码传上去，如果你想了解每一个代码在做什么，可以一步步按我的走，如果不想，可以直接到页面布置的最后一步复制所有代码到mkdocs.yml上，然后再根据我的注释对特定地方进行更改，在此只列举了一些常用的，如果需要其他的，可以到Material for Mkdocs的官网上去看

网站名称更改

- 这个在上文中已经提及，即在mkdocs.yml中对site_name进行编辑，想要预览可以在终端中输入mkdocs serve然后回车，点击本地的那个网址查看

网站图标更改

- 在mkdocs.yml中进行更改，以下是我以github图标作为logo的代码（mkdocs.yml中的所有代码），其他图标可以前往material for mkdocs这个官网查看，在此不过多赘述

```
site_name: Lumner的haven

theme:

  name: material

  icon:

    logo: material/github
```

网站语言更改

- 在mkdocs.yml中进行更改，一下是我改为中文的代码(要注意对齐呀)，然后就可以看到网站的search和content改成中文了

```
site_name: Lumner的haven

theme:

  name: material

  icon:

    logo: material/github

  language: zh
```

添加版权声明

- 代码如下（我用的Lumner，你们改自己的就好），copyright:后面的可以随意写哈，不一定要我这个格式

```
site_name: Lumner的haven

theme:

  name: material

  icon:

    logo: material/github

  language: zh

copyright: Copyright © 2026~2026 Lumner All Rights Reserved
```

添加顶部分类标签和返回顶部按钮

- 代码如下，注意对齐，- 与 navigation 之间的空格

```
site_name: Lumner的haven

theme:

  name: material

  icon:

    logo: material/github

  language: zh

  features:

    - navigation.tabs #顶部分类标签

    - navigation.top  #返回顶部按钮

copyright: Copyright © 2026~2026 Lumner All Rights Reserved
```

添加搜索高亮

- 代码如下

```
site_name: Lumner的haven

theme:

  name: material

  icon:

    logo: material/github

  language: zh

  features:

    - navigation.tabs  #顶部分类标签

    - navigation.top   #返回顶部按钮

    - search.highlight #搜索结果高亮

copyright: Copyright © 2026~2026 Lumner All Rights Reserved
```

导航栏颜色更改

- 代码如下，我选择了蓝色，你可以根据个人喜好进行更改

```
site_name: Lumner的haven

theme:

  name: material

  icon:

    logo: material/github    #网站logo更改

  language: zh              #语言更改

  palette:

    primary: blue           #导航栏颜色更改
```

features:

- navigation.tabs #顶部分类标签
- navigation.top #返回顶部按钮
- search.highlight #搜索结果高亮

copyright: Copyright © 2026~2026 Lumner All Rights Reserved #版权声明

明暗背景更改

- 代码如下，如果使用明暗背景，记得把上个环节导航栏颜色更改代码删掉或者剪切，并分别加在明暗模式的代码中，明暗模式对应的导航栏颜色可以不一样哦(我light mode选了蓝色，dark mode选了灰色)，更改的按钮在搜索框边上，我选择的是日月样式的

site_name: Lumner的haven

theme:

name: material

icon:

logo: material/github #网站logo更改

language: zh #语言更改

palette: #明暗模式和颜色更改

- media: "(prefers-color-scheme: light)" #明
scheme: default

primary: blue #导航栏颜色

toggle:

icon: material/brightness-7

name: Switch to dark mode #改提示词处（光标放在按钮上显示的提示

词）

- media: "(prefers-color-scheme: dark)" #暗

scheme: slate

primary: grey #导航栏颜色

toggle:

icon: material/brightness-4

name: Switch to light mode #改提示词处（光标放在按钮上显示的提示

词)

features:

- navigation.tabs #顶部分类标签
- navigation.top #返回顶部按钮
- search.highlight #搜索结果高亮

copyright: Copyright © 2026~2026 Lumner All Rights Reserved #版权声明

文章

文章发布

首页

- 在vscode里面打开docs下的index.md，然后删除里面的内容，用markdown语言进行编辑就好了

首页索引

- 在docs文件夹中建立几个新的文件夹（根据自己需要吧）暂时记为A，B，然后建几个.md文件（这个里面的内容就是我们的文章），记为A1，A2，B1,B2
- 然后在mkdocs.yml中输入以下代码，这里我就不复制前面的那些代码了，后面需要添加时类似

nav:

- 自行起名1:
 - 自行起名2: A1的地址（删去docs/,留下A/A1.md（下面的同理））
- 自行起名3:
 - 自行起名4: B1的地址
 - 自行起名5: B2的地址

文章内容

- 文章的撰写地方如前文所说，在A1,B2这些.md文件中，不过mkdocs对Markdown的代码要求会比Vscode高一些，Vscode正常渲染的.md文件到了mkdocs上可能就不行了，可以把自己写的扔给 比较聪明 的AI,至于为什么要强调聪明点的AI，哈哈
- 为了后续渲染，我对一些文件进行了修改，现在分享出来

1. 在你的 docs 文件夹下，新建一个文件夹叫 javascripts ，并在里面新建 mathjax.js 并加入以下代码：用于渲染数学公式（用不到可以不添加）

```

window.MathJax = {
  tex: {
    inlineMath: [["$","$"], ["\\(", "\\)"]],
    displayMath: [["$$","$$"], ["\\[", "\\]"]],
    processEscapes: true,
    processEnvironments: true
  },
  options: {
    ignoreHtmlClass: ".*|",
    processHtmlClass: "arithmatex"
  }
};

document$.subscribe(() => {
  MathJax.typesetPromise()
})

```

2. 在你的 docs 文件夹下，新建一个文件夹叫 stylesheets ，并在里面新建 extra.css 并加入以下代码：用于让表格显示边框（用不到可以不添加）

```

/* 让表格强制显示边框和线条 */
.md-typeset table:not([class]) {
  border: 1px solid var(--md-default-fg-color--lightest);
  border-collapse: collapse;
  display: inline-block; /* 修复某些情况下的宽度溢出 */
  max-width: 100%;
}
.md-typeset table:not([class]) th,
.md-typeset table:not([class]) td {
  border: 1px solid var(--md-default-fg-color--lightest); /* 添加内部边框 */
  padding: 9px;
}
/* 修复数学公式在深色模式下的显示问题（防止变黑看不见） */
.md-typeset .arithmatex, .MathJax {
  color: var(--md-default-fg-color) !important;
}

```

3. mkdocs.yml:（如果你没有建我上面的两个文件夹，记得删除下方代码中最后六行，视你建的情况决定）

```

# Markdown 扩展 (核心部分, 对应你的 .md 语法需求)
markdown_extensions:
    # 目录支持
    - toc:
        permalink: true # 生成锚点链接
    # 数学公式
    - pymdownx.arithmatex:
        generic: true
    # “代码”部分
    - pymdownx.highlight:
        anchor_linenums: true # 代码行号
        line_spans: __span
        pygments_lang_class: true
    - pymdownx.inlinehilite # 行内代码高亮
    - pymdownx.superfences # 支持复杂的代码块嵌套
    - pymdownx.snippets # 支持代码片段导入
    # 列表
    - pymdownx.tasklist:
        custom_checkbox: true # 漂亮的复选框样式
    # 字体设置
    - pymdownx.betterem: # 优化粗体和斜体
        smart_enable: all
    - pymdownx.caret: # 支持上标 ^^text^^ 和下划线
    - pymdownx.mark: # 支持高亮 ==text==
    - pymdownx.tilde: # 支持删除线 ~~text~~
    # 脚注
    - footnotes
    # 图片允许调整大小
    - attr_list # 允许使用 {: width="300" } 语法调整图片
    # 额外
    - admonition # 漂亮的提示框 (!!! note)
    - pymdownx.details # 折叠详情
    - pymdownx.emoji:
        emoji_index: !!python/name:material.extensions.emoji.twemoji
        emoji_generator: !!python/name:material.extensions.emoji.to_svg
# 额外的 JavaScript (用于数学公式渲染)
extra_javascript:
    - javascripts/mathjax.js # 必须是你自己创建的那个本地文件
    - https://polyfill.io/v3/polyfill.min.js?features=es6
    - https://cdn.jsdelivr.net/npm/mathjax@3/es5/tex-mml-compiler.js
extra_css:
    - stylesheets/extra.css # 新增: 用于给表格加横线

```

最后，祝你能够建成属于自己的网站呀