

# 建立个人博客

## 目录

1. Zotero
2. BookxNote
3. markdown语法
4. Typora:Markdown 编辑器
5. Obsidian:知识管理和笔记软件
6. VSCode:代码编辑器
7. docsify:简洁、高效的文档网站生成器
8. Gitee:代码托管平台
9. GitHub:代码托管平台

## 一、Zotero

Zotero 是一个免费且开源的引用和参考文献管理软件，广泛用于学术研究和写作中。它可以帮助用户收集、管理、引用和分享研究资料。以下是一些基本的 Zotero 使用指南：

### 安装 Zotero

1. 访问 Zotero 官方网站 <https://www.zotero.org/>
2. 下载适合你操作系统的 Zotero 版本并安装。

### 启动 Zotero

- 安装完成后，启动 Zotero。

### 创建个人文库

- 打开 Zotero 后，你可以创建一个或多个文库（library），用于存储你的文献资料。

### 添加文献

- **手动添加：**通过 "New Item" 可以手动输入文献信息。

- **从网页添加**：Zotero 可以安装浏览器插件，当你在网页上遇到想要引用的文献时，可以直接通过插件保存到 Zotero。
- **导入文件**：支持从各种文献数据库导入文献信息。

## 添加文献实操

### 1. 从知网导入单篇中文文献

可以选择自动抓取的方式  
也可以选择导入PDF的方式

以“机器学习在肾脏疾病中的应用进展”为例

### 2. 从pubmed导入单篇文献

可以选择自动抓取的方式  
也可以选择导入PDF的方式

以"Prediction and diagnosis of chronic kidney disease development and progression using machine-learning: Protocol for a systematic review and meta-analysis of reporting standards and model performance"为例

### 3. 导入多篇文献（以pubmed为例）

PubMed中与LSTM相关的心脏杂音文献

检索公式：(heart sounds OR heart murmurs) AND "classification" AND LSTM

13篇中有5篇未找到PDF

### 4. 导入多篇文献（以知网为例）

可以选择从剪切版导入或下载文献信息

## 管理文献

- **标签**：为文献添加标签，便于分类和检索。
- **文件夹**：创建文件夹，将相关文献分组管理。
- **搜索**：使用 Zotero 的搜索功能快速找到所需文献。 **亦有高级搜索功能。**

## 创建引用

- Zotero 支持多种引用格式，包括 APA、MLA、Chicago 等。
- 在文档中，你可以使用 Zotero 的插件来插入引用和生成参考文献列表。

# 同步

- Zotero 提供了数据同步功能，可以跨设备同步你的文库。

# 插件和集成

- Zotero 支持与 Microsoft Word、LibreOffice 等文字处理软件集成。
- 通过插件，Zotero 可以与许多其他应用程序和工具集成。

# 学习资源

- [白嫖这8个插件，让你的Zotero成为最强文献管理器，导师看了都说顶呱呱！-哔哩哔哩](#)
- [Zotero文献管理软件】附安装包+插件+操作笔记+思维导图-哔哩哔哩](#)
- [AI读文献轻松又简单之：Zotero+ChatGPT/Kimi 使用教程-哔哩哔哩](#)

# 注意事项

- 熟悉 Zotero 的基本操作，如添加文献、管理文献、创建引用等。
- 学习如何使用 Zotero 的各种功能，如标签、搜索、同步等。
- 了解 Zotero 支持的引用格式和如何根据需要进行切换。

通过以上步骤，你可以开始使用 Zotero 来管理你的研究资料和参考文献。随着你的使用，你将发现 Zotero 提供了许多其他有用的功能，以帮助你更高效地进行学术研究和写作。

## 二、BookxNote:一款免费的文献阅读器

BookxNote 是一个开源的电子书阅读和笔记管理软件，它支持多种电子书格式，并提供了笔记、批注、高亮等功能，方便用户在阅读电子书时进行知识管理和学习。以下是一些基本的 BookxNote 使用指南：

### 安装 BookxNote

1. 访问 [BookxNote 的官方网站](#)或者 GitHub 仓库页面，根据你的操作系统下载相应的安装包。
2. 安装完成后，启动 BookxNote。

### 界面介绍

BookxNote 的界面通常包括以下几个主要部分：

- **书架**：展示所有已添加的电子书，可以快速找到并打开书籍。

- **阅读器**：用于阅读电子书，提供翻页、放大缩小等操作。
- **笔记管理**：管理你的笔记和批注，可以查看所有书籍的笔记或特定书籍的笔记。

## 添加电子书

- 点击界面上方的“添加书籍”按钮，选择你的电子书文件，将其添加到书架。

## 阅读和批注

- 双击书架中的电子书，打开阅读器开始阅读。
- 使用阅读器工具栏中的批注工具，可以进行高亮、划线、添加批注等操作。

## 笔记管理

- 在阅读过程中，可以实时添加笔记和批注。
- 笔记可以导出为 Markdown 格式，方便进行进一步的编辑和分享。

## 同步和备份

- BookxNote 支持云端同步，可以将你的笔记和批注同步到云端，方便在不同设备间切换。
- 定期备份你的电子书和笔记，防止数据丢失。

## 高级功能

- **目录导航**：通过电子书的目录快速跳转到指定章节。
- **搜索**：在整本电子书中搜索关键词，快速定位内容。

## 学习资源

## 注意事项

- **熟悉电子书格式**：BookxNote 支持 EPUB、PDF、MOBI 等多种电子书格式。
- **学习如何使用批注工具**：高亮、划线、批注等，这些工具可以帮助你更好地进行阅读和学习。
- **定期同步和备份**：确保你的笔记和批注不会丢失。

通过以上步骤，你可以开始使用 BookxNote 来阅读电子书和管理笔记。随着你的使用，你将发现 BookxNote 提供了许多其他有用的功能，以帮助你更高效地学习和研究。

## 三、markdown

# 语法规则

Markdown 是一种轻量级的标记语言，它允许人们使用易读易写的纯文本格式编写文档，然后转换成结构化的 HTML 页面。

以下是一些基本的 Markdown 语法：

## 标题

使用 # 来创建标题。一个井号表示一级标题，两个井号表示二级标题，以此类推，直到六级标题。

```
# 一级标题
## 二级标题
### 三级标题
#### 四级标题
##### 五级标题
##### 六级标题
```

## 强调

使用星号 \* 或下划线 \_ 来强调文本。

```
*斜体*
_斜体_

**粗体**
__粗体__

**_同时斜体和粗体_**
```

## 列表

使用星号 \*、加号 + 或减号 - 来创建无序列表。

```
* 列表项一
* 列表项二
* 列表项三
```

有序列表则使用数字后跟一个点来创建：

- 1. 第一项
- 2. 第二项
- 3. 第三项

链接

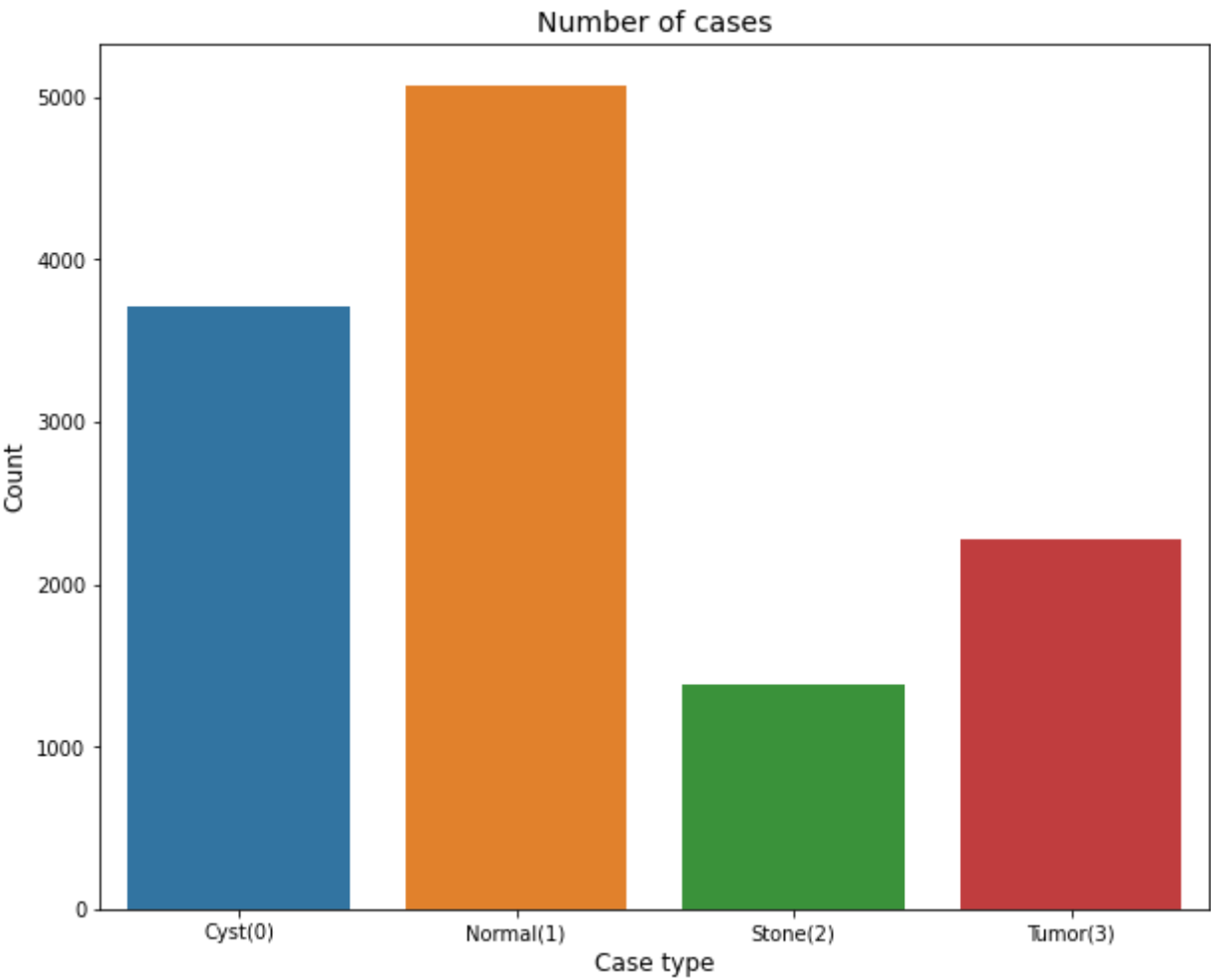
使用方括号包围链接文本，圆括号内输入链接地址。

```
[链接文本](http://example.com)
```

图片

图片的语法和链接类似，但是前面要加一个感叹号。

```
![替代文本](http://example.com/image.jpg)
```



代码

行内代码使用反引号包围：

``代码内容``

代码块使用三个反引号或缩进四个空格：

```
代码内容
```

或者使用缩进：

```
    代码内容
```

# 引用

使用大于号 > 来创建引用文本。

> 引用文本

# 分隔线

使用三个星号 \*\*\* 或三个短划线 --- 来创建分隔线。

\*\*\*  
---

# 表格

使用管道符 | 和连字符 - 来创建表格。

标题1	标题2	标题3
单元格1	单元格2	单元格3
单元格4	单元格5	单元格6

这些是 Markdown 的一些基本语法。Markdown 的目的是要让文档易于阅读和编写，同时又能轻松地转换成 HTML。随着你的实践，你将能够更熟练地使用 Markdown 来编写各种文档。

## 四、Typora

Typora 是一个支持即时渲染的 Markdown 编辑器，它允许你一边写 Markdown，一边看到最终排版后的样式。这使得写作和格式化文本变得更加直观和方便。以下是一些关于如何使用 Typora 的基本指南：

### 安装和启动

1. 访问 Typora 的官方网站 <https://typora.io/> 或者通过应用商店下载并安装 Typora。
2. 安装完成后，启动 Typora。

### 界面介绍

Typora 的界面非常简洁，主要包括以下几个部分：

- **顶部菜单栏**：包含文件、编辑、段落、格式等选项。
- **左侧边栏**：显示文档结构，如标题和列表。
- **编辑区域**：你可以在这里编写和格式化文本。

### 基本操作

- **打开文件**：通过顶部菜单栏的 文件 > 打开，或者直接拖拽文件到 Typora 窗口中打开 Markdown 文件。
- **保存文件**：通过 文件 > 保存 或者 文件 > 另存为...。

### 编辑和格式化文本

Typora 支持直接使用 Markdown 语法来编辑文本，同时也提供了一些快捷键和工具栏按钮来帮助格式化文本。

- **标题**：选中文本后，点击工具栏上的 H1 到 H6 来设置不同级别的标题。
- **加粗和斜体**：选中文本，点击工具栏上的 B（加粗）或 I（斜体）按钮。
- **链接和图片**：选中文本，点击工具栏上的链接或图片按钮，输入链接地址或图片路径。
- **列表**：输入 - 或 \* 来创建无序列表，输入 1. 来创建有序列表。
- **表格**：输入表格的格式，例如：



标题1	标题2	
-----	-----	
单元格	单元格	

- **代码块**：输入三个反引号后跟语言名称，然后输入代码，例如：

```
print("Hello, World!")
```

- **引用**：输入 > 来创建引用文本。

## 高级功能

- **文件管理**：Typora 支持文件夹和文件的拖拽操作，方便管理文档。
- **主题和样式**：通过顶部菜单栏的 主题 选项，可以切换不同的主题和样式。
- **预览模式**：点击工具栏上的 专注模式 或 源代码模式 按钮，可以在不同的视图之间切换。
- **导出功能**：通过 文件 > 导出，可以将 Markdown 文档导出为 HTML、PDF、Word 等多种格式。

## 快捷键

Typora 支持大量的快捷键，以提高编辑效率。一些常用的快捷键包括：

- Ctrl + N：新建文档
- Ctrl + S：保存文档
- Ctrl + Z：撤销
- Ctrl + Shift + Z：重做
- Ctrl + B：加粗
- Ctrl + I：斜体
- Ctrl + P：预览

通过不断练习和探索，你将能够更熟练地使用 Typora 来编写和格式化 Markdown 文档。祝你学习愉快！

## 五、Obsidian

Obsidian 是一款知识管理和笔记软件，它利用 Markdown 语法来编写和组织笔记，并通过链接将不同笔记相互关联，形成个人知识库。以下是一些关于如何使用 Obsidian 的基本指南：

## 安装和启动

1. 访问 Obsidian 的官方网站 <https://obsidian.md/> 下载并安装 Obsidian。
2. 安装完成后，启动 Obsidian。

## 界面介绍

Obsidian 的界面通常分为三个主要部分：

- **左侧边栏**：显示所有笔记的列表，可以创建文件夹来组织笔记。
- **中间主面板**：显示选中笔记的内容，你可以在这里编辑和查看笔记。
- **右侧面板**：用于显示笔记的反向链接、大纲、标签等信息。

## 创建和编辑笔记

- **创建笔记**：点击左侧边栏的 "+" 按钮，选择 "Create a new note" 创建新笔记，或者直接拖拽 Markdown 文件到 Obsidian 窗口中。
- **编辑笔记**：Obsidian 支持 Markdown 语法，你可以在中间主面板中直接编辑笔记内容。

## 链接和关联笔记

Obsidian 的核心功能之一是将不同笔记通过链接关联起来，形成知识网络。

- **创建内部链接**：在笔记中引用另一篇笔记，使用 `[[Note Title]]` 的格式，Obsidian 会自动创建链接。
- **反向链接**：在右侧面板的 "Backlinks" 部分，可以查看所有链接到当前笔记的其他笔记。

## 标签和搜索

- **添加标签**：在笔记中添加 `#标签名`，可以为笔记添加标签。
- **搜索笔记**：在左侧边栏的搜索框中，可以搜索笔记标题或标签。

## 文件和文件夹管理

- **创建文件夹**：在左侧边栏，右键点击空白处，选择 "Create Folder" 创建文件夹。
- **移动笔记**：将笔记从左侧边栏拖拽到目标文件夹中，可以移动笔记。

## 主题和插件

- **更改主题**：通过 "Settings" > "Appearance" 可以更改 Obsidian 的主题。
- **安装插件**：Obsidian 支持插件扩展功能，通过 "Settings" > "Third-party plugins" 可以安装和管理插件。

## 导出和分享

- **导出笔记**：选中笔记，点击 "File" > "Export"，可以将笔记导出为 Markdown、PDF 等格式。
- **分享笔记**：Obsidian 支持将笔记分享为链接，方便与他人共享。

## 高级功能

- **每日笔记**：Obsidian 提供 "Daily Notes" 功能，可以快速创建和查看每日笔记。
- **任务管理**：Obsidian 支持 - [ ] 和 - [x] 格式的任务列表，方便管理待办事项。

## 学习资源

通过不断练习和探索，你将能够更熟练地使用 Obsidian 来管理你的个人知识库。祝你学习愉快！

## 六、VScode

Visual Studio Code（简称 VSCode）是一个由微软开发的免费、开源的代码编辑器。它支持多种编程语言，具有强大的功能和可扩展性，是许多开发者喜爱的工具之一。以下是一些基本的 VSCode 使用指南：

### 安装和启动

1. 访问 Visual Studio Code 的官方网站 <https://code.visualstudio.com/> 下载并安装 VSCode。
2. 安装完成后，启动 VSCode。

### 界面介绍

VSCode 的界面主要由以下几个部分组成：

- **侧边栏**：包含文件资源管理器、搜索、Git 等功能。
- **编辑区**：这是你编写代码的地方。
- **面板**：位于编辑区的下方，包含终端、调试视图、输出等。
- **状态栏**：位于窗口底部，显示文件编码、语言和其他状态信息。

### 基本操作

- **打开文件/文件夹**：点击侧边栏的文件夹图标，选择要打开的文件或文件夹。
- **新建文件**：在侧边栏的文件资源管理器中，右键点击空白处，选择 "New File"。
- **保存文件**：使用快捷键 Ctrl + S（Windows/Linux）或 Command + S（Mac）。
- **关闭文件**：点击编辑区右上角的关闭按钮。

## 编辑和导航

- **撤销和重做**：使用快捷键 `Ctrl + Z` 和 `Ctrl + Y` (Windows/Linux) 或 `Command + Z` 和 `Command + Shift + Z` (Mac) 。
- **查找和替换**：使用快捷键 `Ctrl + F` (Windows/Linux) 或 `Command + F` (Mac) 打开查找框；使用 `Ctrl + H` 或 `Command + Option + F` 打开查找和替换框。
- **多光标编辑**：按住 `Alt` (Windows/Linux) 或 `Option` (Mac) 键并点击，可以添加多个光标。

## 代码编辑增强功能

- **自动完成**：VSCode 会根据你输入的代码提供自动完成建议。
- **代码片段**：使用预定义的代码模板来加速编码。
- **语法高亮**：VSCode 支持多种语言的语法高亮。
- **代码格式化**：使用快捷键 `Shift + Alt + F` (Windows/Linux) 或 `Shift + Option + F` (Mac) 格式化代码。

## Git 集成

- **提交**：在侧边栏的 Git 图标处选择要提交的文件，然后点击 "Commit"。
- **推送**：点击 "Push" 将本地更改推送到远程仓库。

## 扩展

- **扩展市场**：通过侧边栏的扩展图标或 `Ctrl + Shift + X` (Windows/Linux) 或 `Command + Shift + X` (Mac) 打开扩展市场。
- **安装扩展**：搜索需要的扩展，点击 "Install" 安装。

## 自定义设置

- **用户设置**：通过菜单栏的 `File > Preferences > Settings` 访问用户设置。
- **快捷键设置**：通过 `File > Preferences > Keyboard Shortcuts` 自定义快捷键。

## 调试

- **启动调试**：通过侧边栏的调试图标或 `Ctrl + Shift + D` (Windows/Linux) 或 `Command + Shift + D` (Mac) 打开调试视图。
- **设置断点**：在代码行号旁边点击，设置断点。

## 终端

- **打开终端**：使用 `Ctrl + ``` (Windows/Linux) 或 `Command + ``` (Mac) 打开集成终端。
- **终端命令**：在终端中，可以执行各种命令行操作。

## 学习资源

【教程】vscode优化体验篇（推荐设置 && 推荐插件）-哔哩哔哩

[https://blog.csdn.net/qq\\_51173321/article/details/126287293](https://blog.csdn.net/qq_51173321/article/details/126287293)

通过这些基本操作，你可以开始使用 VSCode 进行编程。随着你的使用，你将发现更多高级功能，以提高你的开发效率。

## 七、部署

### docsify

Docsify 是一个简洁、高效的文档网站生成器，它使用 Markdown 来编写和预览文档，并且可以轻松地将文档部署为静态网站。以下是一些基本的 Docsify 使用指南：

### B站视频指导

使用docsify搭建笔记博客

这是一篇使用docsify搭建个人笔记博客的快速入门视频

官方主页：<https://thinkaboutai.github.io>

笔记：[https://thinkaboutai.github.io/#/other/01\\_docsify/](https://thinkaboutai.github.io/#/other/01_docsify/)

<https://thinkaboutai.github.io/#>

### docsify 官网指南

官网指南

### 准备工作

- 确保你的计算机上已经安装了 Node.js 和 Git。

### 安装 Docsify CLI

1. 通过 npm 安装 Docsify 的命令行工具：

```
npm install -g docsify-cli
```

## 创建一个新项目

1. 创建一个新文件夹作为你的项目目录。
2. 在项目目录中初始化一个新项目：

```
docsify init .
```

## 文件结构

Docsify 的基本文件结构通常如下：

```
my-docs/  
├─ .nojekyll  
├─ index.html  
├─ README.md  
└─ guide.md
```

- index.html 是你的网站的入口点，通常包含对 Docsify 的引用。
- README.md 和 guide.md 是你的文档页面，可以包含更多的 .md 文件。

## 编写文档

- 使用 Markdown 语法在 .md 文件中编写你的文档内容。
- 你可以使用相对链接来连接不同的文档页面。

## 预览文档

1. 在项目目录中启动本地服务器：

```
docsify serve
```

2. 打开浏览器，访问 `http://localhost:3000` 来预览你的文档。

## 部署文档

1. 你可以将你的文档部署到 GitHub Pages、Gitee Pages 或其他静态网站托管服务。
2. 将你的项目推送到远程仓库，并按照托管服务的指南配置静态网站部署。

## 自定义主题

- Docsify 允许你通过修改 index.html 中的 CSS 或者使用插件来自定义文档的外观。

## 使用插件

- Docsify 支持插件扩展，你可以在 `index.html` 中引入插件来增加额外的功能。

## 学习资源

- **官方文档**：Docsify 官方文档提供了详细的指南和教程。
- **GitHub 仓库**：Docsify 的 GitHub 仓库中有丰富的示例和社区贡献的插件。

## 注意事项

- 熟悉 Markdown 的基本语法，包括标题、链接、列表、代码块等。
- 理解如何使用相对路径来链接文档中的不同页面。
- 学习如何使用 Docsify 的配置选项来自定义你的文档网站。

通过以上步骤，你可以开始使用 Docsify 来编写和部署你的文档网站。随着你的使用，你将发现 Docsify 提供了许多其他有用的功能，以帮助你更高效地创建和管理文档。

## 和docsify相似的工具

和 Docsify 类似的工具有多种，它们都是用于生成文档网站的工具，但各有特点和优势。以下是一些与 Docsify 相似的工具：

1. **VuePress**：由 Vue.js 的创始人开发，是一个 Vue 驱动的静态站点生成器，适合生成技术文档和个人博客。
2. **GitBook**：一个基于 Node.js 的命令行工具，可以使用 GitHub/Git 和 Markdown 来制作电子书。
3. **MkDocs**：一个静态网站生成器，可以将 Markdown 文件转换为静态网页，支持 YAML 配置文件。
4. **Hexo**：一个基于 Node.js 的博客平台和静态网站生成器，支持 Markdown 写作。
5. **Docute**：类似于 Docsify，建立在 Vue.js 之上，允许在 Markdown 文件中使用 Vue 组件。
6. **Slate**：由 GitHub 开发，用于生成 REST API 文档，支持多语言选项卡。
7. **Docusaurus**：由 Facebook 创建，使用 React，适合创建开源项目文档和企业文档网站。
8. **Jekyll**：一个成熟的静态网站生成器，被广泛用于生成项目文档和个人博客。
9. **Teedoc**：一个开源静态文档网站生成工具，支持 Markdown 和多种文档格式。
10. **Docpress**：从项目文档生成网站，支持 `README.md` 和 `docs/` 中的多个 Markdown 页面。

这些工具各有千秋，选择哪个工具取决于你的具体需求、技术栈以及对特定功能的偏好。例如，如果你熟悉 Vue.js，可能会倾向于使用 VuePress；如果你的项目需要深度集成 React，Docusaurus 可能是一个好选择。

## gitee

Gitee（原名 Git@OSC，国内称为码云）是一个基于 Git 的代码托管和协作开发平台，类似于 GitHub 或 GitLab，但主要面向中文用户，提供更优的国内访问速度。以下是一些基本的 Gitee 使用指南：

## 注册账号

1. 访问 Gitee 官方网站 <https://gitee.com/>
2. 点击右上角的 "注册", 按照提示完成账号注册。

## 登录账号

1. 完成注册后, 使用账号登录 Gitee。

## 创建仓库

1. 登录后, 点击右上角的头像, 选择 "我的仓库"。
2. 点击 "新建仓库", 填写仓库名称、描述, 选择公开或私有, 然后创建。

## 初始化仓库

1. 创建仓库后, 你可以立即开始初始化你的仓库:
  - 添加一个 README.md 文件来描述你的项目。
  - 添加 .gitignore 文件来指定不需要版本控制的文件。

## 克隆仓库

1. 在你的仓库页面, 点击 "SSH" 或 "HTTPS" 复制克隆地址。
2. 在本地终端使用 `git clone` 命令加上克隆地址, 例如:

```
git clone git@gitee.com:username/repo.git
```

## 推送代码到仓库

1. 在本地创建或修改文件后, 使用以下命令将改动推送到 Gitee:

```
git add .  
git commit -m "Initial commit"  
git push -u origin main
```

注意: `main` 可能是你仓库的主分支名称, 也可能是 `master`。

## 分支管理

1. 创建分支:

```
git branch new-branch
```

2. 切换分支:



```
git checkout new-branch
```

### 3. 合并分支：

```
git merge new-branch
```

## 问题跟踪和 Wiki

1. **问题跟踪**：可以用来跟踪和管理项目的 bug 和功能请求。
2. **Wiki**：项目的文档可以放在 Wiki 中，方便团队成员查阅。

## 协作开发

1. **团队成员管理**：可以给团队成员分配不同的权限，如管理员、写入、读取等。
2. **Pull Request**：当团队成员完成了一个新功能或修复了一个 bug，他们可以发起 Pull Request 请求将代码合并到主分支。

## 使用 Gitee Pages

1. Gitee Pages 是一个静态网页托管服务，你可以利用它来托管博客、项目页面等。

## 学习和帮助

### 注意事项

- 熟悉 Git 的基本操作，如 `git add`、`git commit`、`git push`、`git pull` 等。
- 理解 Git 的分支管理，如分支创建、合并、删除等。
- 学习 Markdown 语法，用于编写 `README.md` 和 Wiki 页面。

通过以上步骤，你可以开始使用 Gitee 进行代码托管和版本控制。随着你的使用，你将发现 Gitee 提供了许多其他有用的功能，以帮助你更好地管理项目和协作开发。

## github

GitHub 是一个非常流行的代码托管平台，同时也是一个强大的协作和代码管理工具。以下是一些基本的 GitHub 使用指南：

### 注册账号

1. 访问 GitHub 官方网站 <https://github.com/>
2. 点击右上角的 "Sign up"，按照提示完成账号注册。

## 登录账号

1. 完成注册后，使用账号登录 GitHub。

## 创建仓库

1. 登录后，点击右上角的 "+" 号，选择 "New repository" 创建新仓库。
2. 填写仓库名称、描述，选择公开或私有，然后创建。

## 初始化仓库

1. 创建仓库后，你可以立即开始初始化你的仓库：
  - 添加一个 `README.md` 文件来描述你的项目。
  - 添加 `.gitignore` 文件来指定不需要版本控制的文件。

## 克隆仓库

1. 在你的仓库页面，点击 "Code" 按钮，复制 SSH 或 HTTPS 克隆地址。
2. 在本地终端使用 `git clone` 命令加上克隆地址，例如：

```
git clone git@github.com:username/repo.git
```

## 推送代码到仓库

1. 在本地创建或修改文件后，使用以下命令将改动推送到 GitHub：

```
git add .  
git commit -m "Initial commit"  
git push -u origin main
```

注意： `main` 可能是你仓库的主分支名称，也可能是 `master`。

## 分支管理

1. 创建分支：

```
git branch new-branch
```

2. 切换分支：

```
git checkout new-branch
```

3. 合并分支：

```
git merge new-branch
```

## 问题跟踪和 Wiki

1. **问题跟踪**：可以用来跟踪和管理项目的 bug 和功能请求。
2. **Wiki**：项目的文档可以放在 Wiki 中，方便团队成员查阅。

## 协作开发

1. **团队成员管理**：可以给团队成员分配不同的权限，如管理员、写入、读取等。
2. **Pull Request**：当团队成员完成了一个新功能或修复了一个 bug，他们可以发起 Pull Request 请求将代码合并到主分支。

## 使用 GitHub Pages

1. GitHub Pages 是一个静态网页托管服务，你可以利用它来托管博客、项目页面等。

## 学习和帮助

- **官方文档**：GitHub 提供了官方文档，涵盖从基础到高级的教程。
- **社区支持**：加入 GitHub 社区，参与讨论和提问。

## 注意事项

- 熟悉 Git 的基本操作，如 `git add`、`git commit`、`git push`、`git pull` 等。
- 理解 Git 的分支管理，如分支创建、合并、删除等。
- 学习 Markdown 语法，用于编写 README.md 和 Wiki 页面。

通过以上步骤，你可以开始使用 GitHub 进行代码托管和版本控制。随着你的使用，你将发现 GitHub 提供了许多其他有用的功能，以帮助你更好地管理项目和协作开发。