

附录一、Obsidian笔记系统与文本编辑器Vim

常见的笔记系统软件有：LogSeq、Obsidian、Notion、One Note

Obsidian笔记系统

Obsidian 是一款强大的**知识管理**和**笔记软件**。以下是关于 Obsidian 的详细介绍：

一、主要特点

1. 本地存储与数据安全

- Obsidian 支持**本地存储**，这意味着你的数据完全掌握在自己手中，不用担心因为云服务故障或隐私问题导致数据丢失或泄露。
- 同时，你可以随时**备份和恢复数据**，确保重要信息的安全性。

2. 双向链接与知识图谱

- 该软件的核心功能之一是**双向链接**。你可以在笔记中创建链接指向其他笔记，并且被链接的笔记也会显示有哪些笔记链接到了它。
- 这种方式构建了一个强大的知识网络，帮助你更好地组织和关联不同的知识内容，形成**知识图谱**，便于快速查找和探索相关主题。

3. 丰富的插件生态

- Obsidian 拥有活跃的**插件社区**，提供了大量的插件来扩展软件的功能。
- 例如，有用于增强搜索功能的插件、美化界面的插件、与其他工具集成的插件等，可以根据自己的需求进行定制。

4. Markdown 支持

- Obsidian 采用 **Markdown 语法**进行笔记编辑，Markdown 是一种简洁、易读易写的**标记语言**。
- 它允许你快速创建格式丰富的文本，如标题、列表、链接、代码块等，同时也方便在不同平台上进行转换和分享。

5. 跨平台使用

- 可以在 Windows、macOS、Linux 等多个操作系统上使用，让你在不同设备上都能轻松访问和编辑自己的笔记。

二、使用场景

1. 知识管理

- 无论是学生、研究者还是专业人士，都可以利用 Obsidian 来管理自己的学习笔记、研究资料和工作文档。
- 通过双向链接和知识图谱，将不同的知识点连接起来，形成系统的知识体系。

2. 写作辅助

- 对于作家、博主等创作者来说，Obsidian 可以作为一个强大的写作工具。
- 利用 Markdown 语法快速撰写文章，通过插件进行语法检查、字数统计等，同时可以方便地引用和整理相关的资料和灵感。

3. 项目管理

- 在项目管理中，Obsidian 可以用来记录项目的进展、问题和解决方案。
- 通过链接不同的任务和文档，实现项目信息的集中管理和快速检索。

三、安装与使用

1. 安装: [Obsidian - Sharpen your thinking](#)
2. 设置仓库: `E:\MathNotes\MathSoftware`
3. 浏览界面的各个功能
4. 基础设置
5. 安装everything: [下载 - voidtools](#)
6. 学习文档: [Obsidian 官方文档](#)

Markdown语法基础

[官方文档](#)

一、标题

Markdown 支持六级标题，使用 `#` 的数量表示不同级别。

一级标题
二级标题
三级标题

以此类推。

二、段落

直接输入文本即可形成段落，段落之间用空行分隔。

三、强调

- **粗体**：使用两个 ***** 或 **_** 包裹文本，如 ****这是粗体**** 或 **这是粗体**。
- **斜体**：使用一个 ***** 或 **_** 包裹文本，如 ***这是斜体*** 或 **这是斜体**。
- **高亮**：使用两个 **=** 包裹文本，如：**==高亮==**
- **删除**：使用两个 **~** 包裹文本，如：**~~删除~~**

四、列表

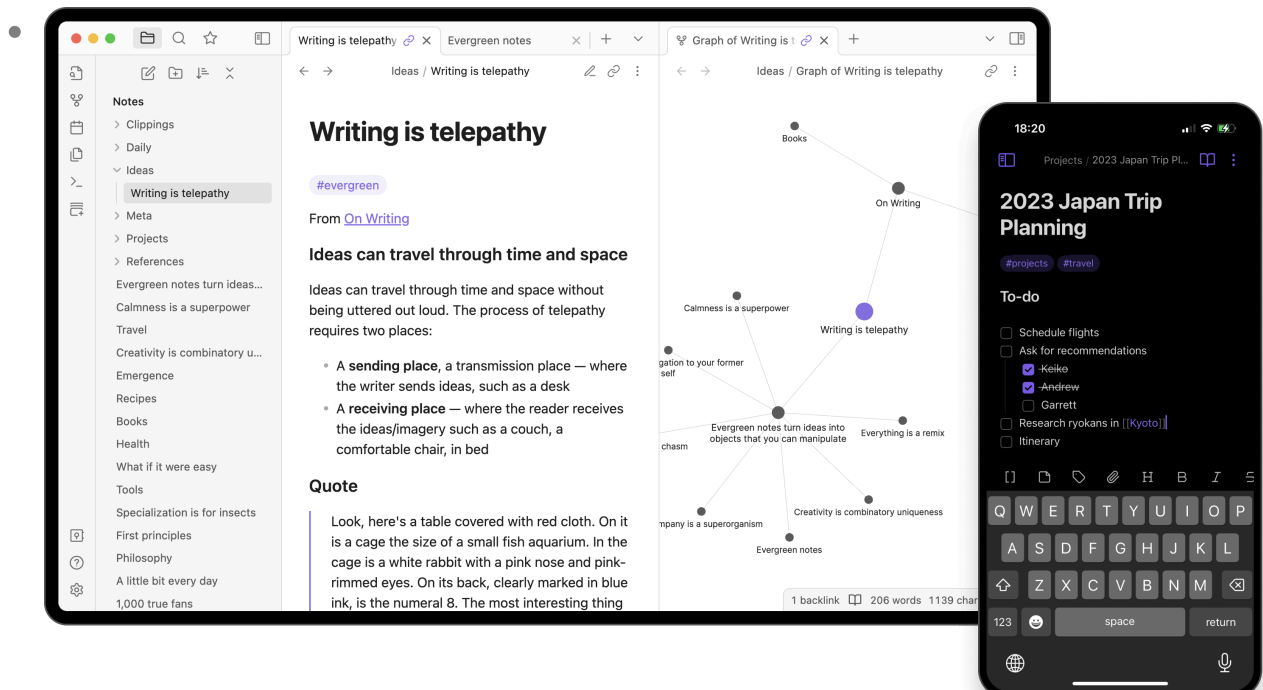
- 有序列表：使用数字加 **.**，如 **1. 第一项 2. 第二项**。
- 无序列表：使用 **-** 或 ***** 或 **+**，如 **- 第一项 * 第二项 + 第三项**。

五、链接

- 内联链接：**[链接描述](链接地址)**，例如 **[百度](https://www.baidu.com)**。
- [Obsidian](#)

六、图片

- **![图片描述](图片地址)**，例如 **![美丽的风景](图片的 URL)**



七、引用

使用 `>` 表示引用，如 `> 这是一段引用的话。`

这是一段引用的话

八、代码块

- 行内代码：使用反引号包裹，如 `这是一段code。`
- 代码块：使用三个反引号包裹，如 `这是一个代码块 多行代码可以放在这里`

```
\documentclass{amsart}
```

九、分割线

- 使用三个或以上的 `-`、`*` 或 `_`，如 `---`、`***`、`___`。

十、表格

在 Obsidian 中可以使用 Markdown 语法编辑表格，具体方法如下：

- 创建表格
 - 使用 `|` 来分隔单元格，使用 `-` 来分隔表头和其他行。例如：

```
| 列 1 | 列 2 | 列 3 |  
| --- | --- | --- |  
| 内容 1 | 内容 2 | 内容 3 |
```

- 对齐方式
 - 在表头分隔线中，可以使用 `:` 来设置列的对齐方式。
 - 左对齐：在表头分隔线左侧添加 `:`，如 `| 列名 | 列名 | 列名 |` 变为 `| :--- | :--- | :--- |`。
 - 右对齐：在表头分隔线右侧添加 `:`，如 `| 列名 | 列名 | 列名 |` 变为 `| ---: | ---: | ---: |`。
 - 居中对齐：在表头分隔线两侧都添加 `:`，如 `| 列名 | 列名 | 列名 |` 变为 `| :---: | :---: | :---: |`。

- 添加内容

在表格的单元格中直接输入文本内容即可。如果内容较长，可以手动换行，或者使用 Obsidian 的软换行（Shift + Enter）来保持表格的整洁。

- 表格的扩展
 - 合并单元格

- Markdown 本身不支持直接合并单元格，但可以通过一些技巧来实现类似的效果。比如，使用空单元格和适当的对齐方式来模拟合并单元格的外观。
- 添加格式
 - 可以在表格单元格中使用 Markdown 的其他语法，如加粗、斜体、链接等，来丰富表格内容的显示效果。
- 一些例子

一级表头 1	一级表头 2
二级表头 A	二级表头 B
内容 1A	内容 1B
内容 2A	内容 2B

描述	链接
这是一个加粗的描述 重要内容	另一个链接

步骤	说明
1.	第一步的说明，这里可以有 <ul style="list-style-type: none">- 子步骤 1- 子步骤 2
2.	第二步的说明

特别说明	列 1	列 2
这是对两列的共同说明	内容 1	内容 2

特性	项目 A	项目 B
优点	快速	稳定
缺点	资源占用高	功能较少

列表中使用代码块

在 Markdown 中，可以在列表中使用代码块。以下是示例：

- 无序列表中使用代码块：
 - 这里是一个任务描述。
 - 具体步骤如下：

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

这段代码可以输出 “Hello, World!” 。

- 另一个任务描述。
 - 有序列表中使用代码块：
 1. 第一步的说明。
 - 相关代码示例：

JS

```
const x = 10; console.log(x);
```

2. 第二步的说明。这样可以在列表中清晰地展示代码片段，方便进行技术文档的编写或任务说明中涉及代码的情况。

Vim 文本编辑器

Vim 简介

Vim (Vi IMproved) 是一个高度配置的文本编辑器，它是基于著名的 **vi** 编辑器的一个增强版本。Vim非常流行，特别是在Unix和Linux用户中，但它也适用于Windows系统。以下是Vim的一些关键特点：

1. **模式化操作**：Vim拥有多种模式，包括正常模式、插入模式、命令行模式等。用户在这些模式之间切换以执行不同的任务，例如编辑文本、执行命令或浏览文件。
2. **键盘驱动**：Vim设计为尽可能使用键盘而非鼠标，这使得熟练的用户可以非常快速地编辑文本。
3. **脚本和扩展**：Vim可以通过Vim脚本（Vimscript）进行扩展，用户可以编写脚本来自定义编辑器的行为。
4. **插件支持**：Vim支持各种插件，这些插件可以添加新的功能，如代码高亮、自动补全、语法检查等。
5. **跨平台**：Vim可以在多种操作系统上运行，包括Linux、Unix、Windows、Mac OS X等。

6. **强大的搜索和替换功能**：Vim提供了强大的搜索和替换功能，支持正则表达式，可以快速定位和修改文本。
7. **分割窗口**：Vim允许用户在同一编辑器窗口中垂直或水平分割多个视图，这样可以同时查看和编辑多个文件。
8. **命令行接口**：Vim可以通过命令行启动和操作，这对于远程编辑或脚本自动化非常有用。
9. **轻量级**：与其他一些图形界面的文本编辑器相比，Vim相对轻量级，占用的系统资源较少。
10. **可定制性**：用户可以根据自己的喜好和需求定制Vim的键位映射、颜色方案、插件等。
11. **适用于编程**：Vim特别适合编程，它支持多种编程语言的语法高亮和代码折叠。
12. **社区支持**：Vim有一个活跃的社区，用户可以从社区中获得帮助、插件和脚本。

Vim的学习曲线相对较陡，但它的强大功能和灵活性使得许多用户愿意投入时间学习它。一旦熟练掌握，Vim可以显著提高文本编辑的效率。

下载与安装

- 下载： [download : vim online](#)，建议下载32位版本。
- 添加路径到环境变量：
 1. 用Everything搜索 **gvim.exe**，复制路径(**D:\Program Files (x86)\Vim\vim91\gvim.exe**)
 2. 任务栏搜索框搜索 **系统环境变量**，点击 **环境变量**，定位到 **path**，选择编辑，添加 **D:\Program Files (x86)\Vim\vim91** 即可
 3. 重新打开CMD，运行 **gvim**，查看是否能够打开gvim
 4. 运行 **vimtutor** 打开vim教程

也可以使用

SHELL

```
vim --clean -c 'e D:\Program Files (x86)\Vim\vim91\tutor\tutor.zh_cn.utf-8' -c  
'w! TUTORCOPY' -c 'q'  
vim --clean TUTORCOPY
```

