如何优雅的使用Markdown

Lijie ZHOU (20809020)

CPU Weekly

Sharing Session

OVERVIEW

01

关于Markdown

Markdown 是一种轻量级标记语言

🤔 Why Markdown?

笔记干净整洁

排版简单

一次编写多处可用

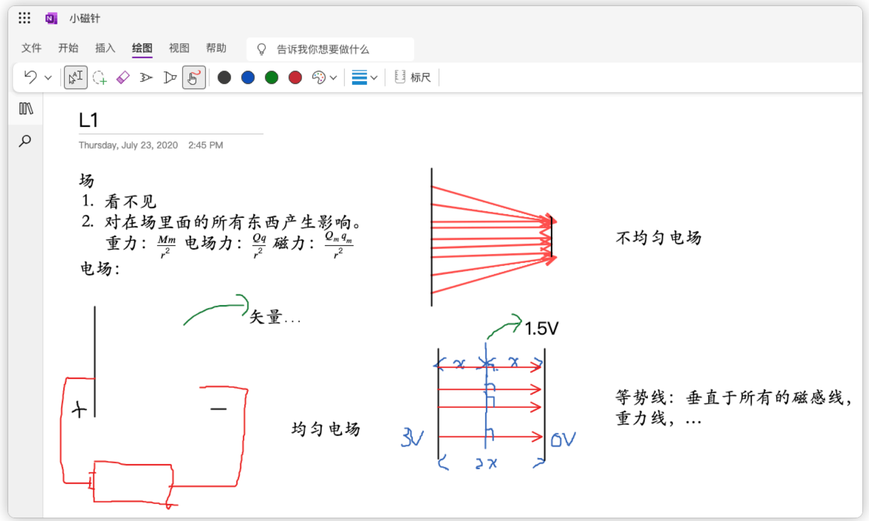
专注内容

版本控制友好

跨平台展示

🤔 Why Markdown?

笔记干净整洁



专注内容 排版简单 跨平台展示 版本控制友好 一次编写多处可用

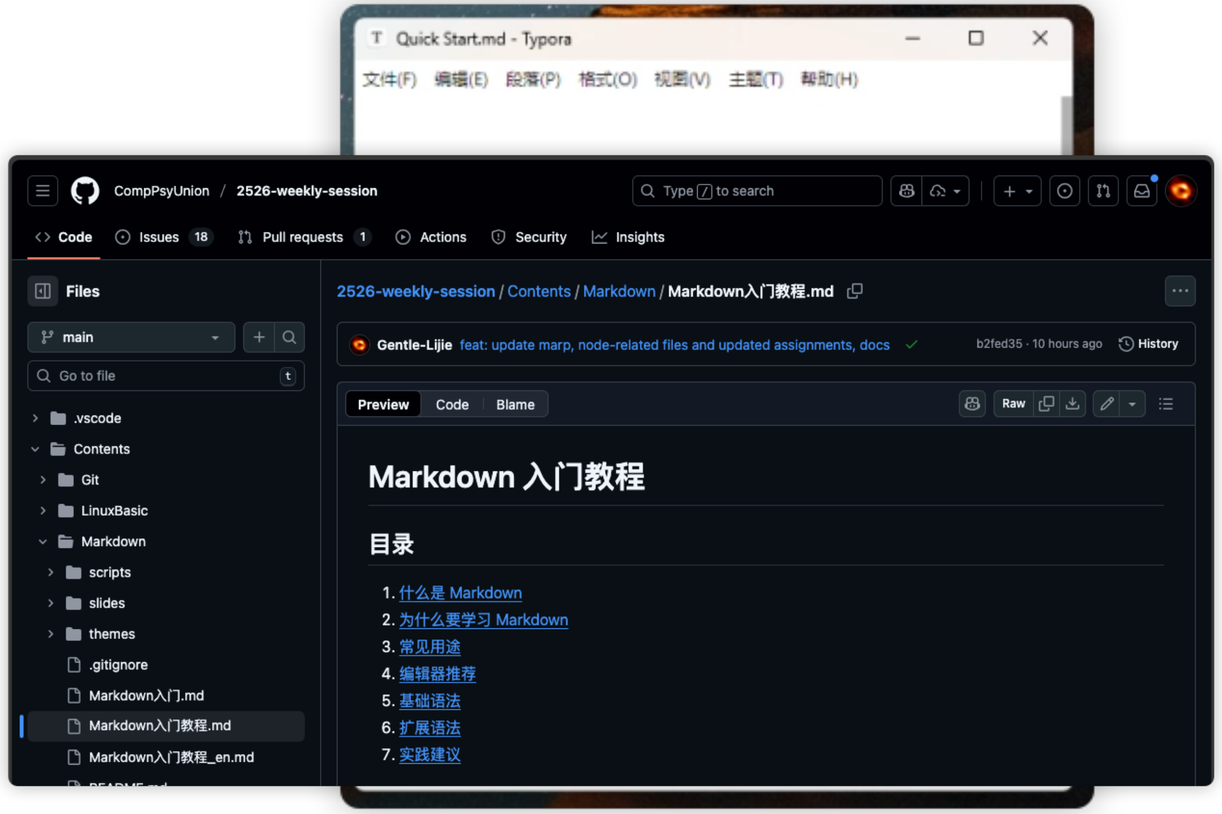
🤔 Why Markdown?

专注内容　排版简单

跨平台展示 版本控制友好 一次编写多处可用　笔记干净整洁

🤔 Why Markdown?

跨平台展示

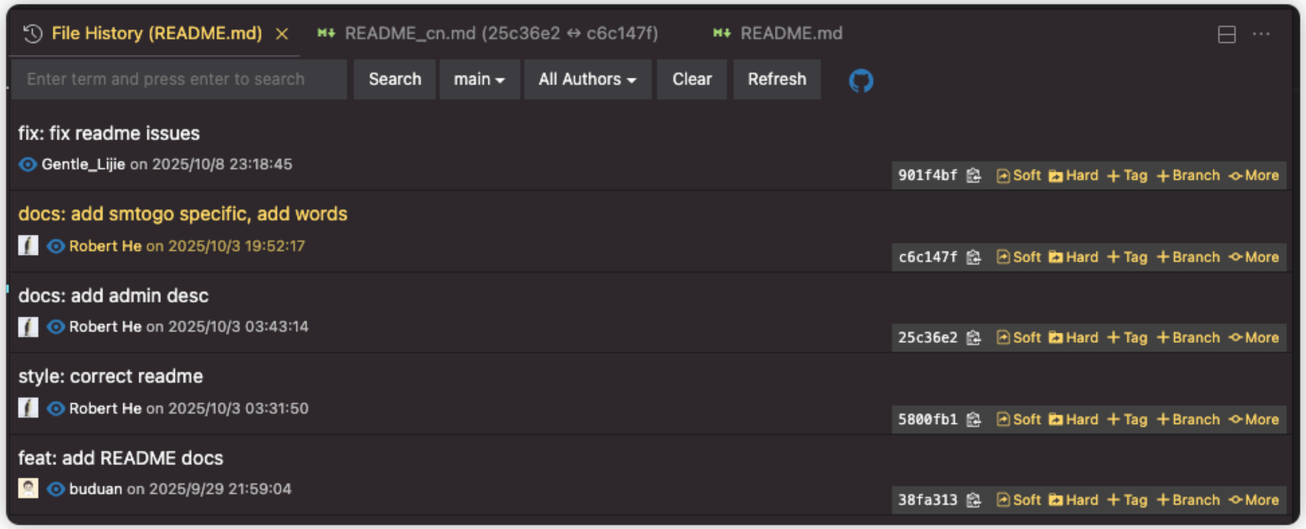


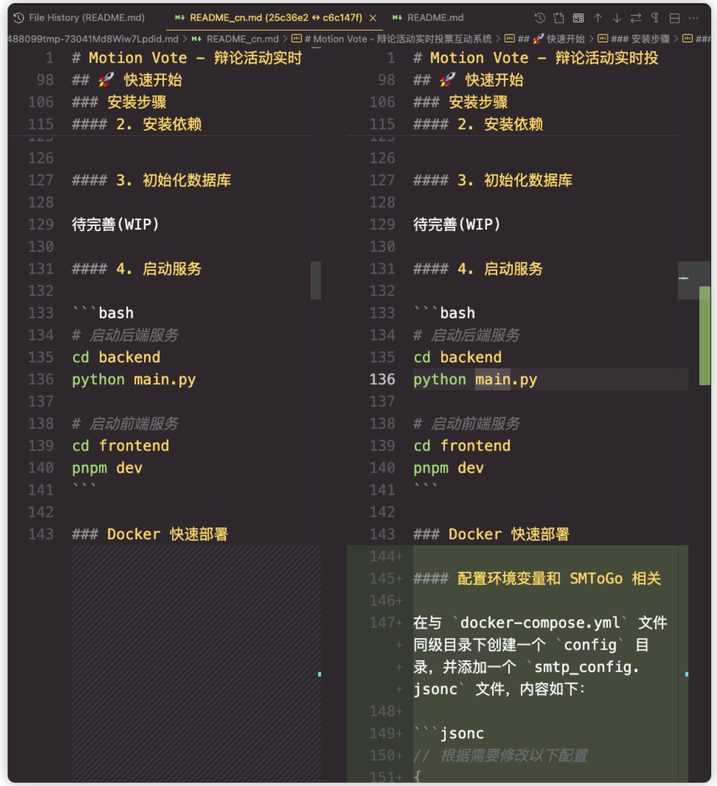
版本控制友好 一次编写多处可用

笔记干净整洁 专注内容 排版简单

🤔 Why Markdown?

版本控制友好





一次编写多处可用

笔记干净整洁　专注内容　排版简单　跨平台展示

🤔 Why Markdown?

一次编写多处可用

Visual Studio Code
https://code.visualstudio.com › languages · 翻译此页

Markdown and Visual Studio Code

Working with Markdown files in Visual Studio Code is simple, straightforward, and fun. Besides VS Code's basic editing, there are a several Markdown-specific ...

Google Help
https://support.google.com › docs › answer · 翻译此页

Use Markdown in Google Docs, Slides, & Drawings

You can use Markdown to quickly add formatting elements to your Google Docs, Slides, and Drawings. With Markdown, you can format text to add: Italics; Bold ...

CSDN博客
https://blog.csdn.net › meihualing › article › details

CSDN中使用Markdown文本编辑器原创

2023年2月8日 — 在Markdown编辑器中，用户可以方便地编写和预览Markdown格式的文本，而无需担心格式化问题。这些编辑器通常具有语法高亮、自动补全和HTML预览功能。一些高级 ...

Feishu
https://www.feishu.cn › zh-CN › articles › 1188045747...

在文档中使用Markdown

2024年11月11日 — 一、功能简介 · Markdown 语法是一种用于写作的轻量级标记语言，能够用简洁的语法生成标题、加粗、列表等排版样式。例如，输入 # + 空格 即可输入一级标题。

Evernote
https://help.evernote.com › en-us › articles · 翻译此页

Markdown support and copy-paste behavior

2025年7月2日 — Evernote supports a selection of Markdown syntax elements, allowing you to format notes quickly and efficiently. Below are the supported Markdown elements.

Medium · 即期品女子
8 次赞 · 2年前

Notion 從零開始學習Markdown語法，輕鬆提升內容編輯技巧

另外，雖然支援Markdown 語法，但Notion也為不會用Markdown 的人設計可以直接用滑鼠點擊，這也是為什麼Notion好上手的因素。

幕布
https://mubu.com › explore

支持节点内部使用markdown语法的思维导图工具—MarkMind

MarkMind 是一款思维导图软件，与市面上的思维导图如Xmind、Mindmaster不同的是，该软件支持节点内部使用Markdown语法MarkMind 思维导图具有以下功能外框概要标注标签 ...

Hexo **原生支持 Markdown**，并且默认通过 `hexo-renderer-marked` 插件来渲染 Markdown 文件。你只需将文章以 `.md` 扩展名保存在 `source/_posts` 文件夹下，即可使用 Markdown 语法进行写作。

- **默认支持**：Hexo 是一个基于 Node.js 的静态博客框架，设计之初就支持 Markdown 格式。

笔记干净整洁　专注内容　排版简单　跨平台展示　版本控制友好

🤔 Why NOT Markdown?

Pros

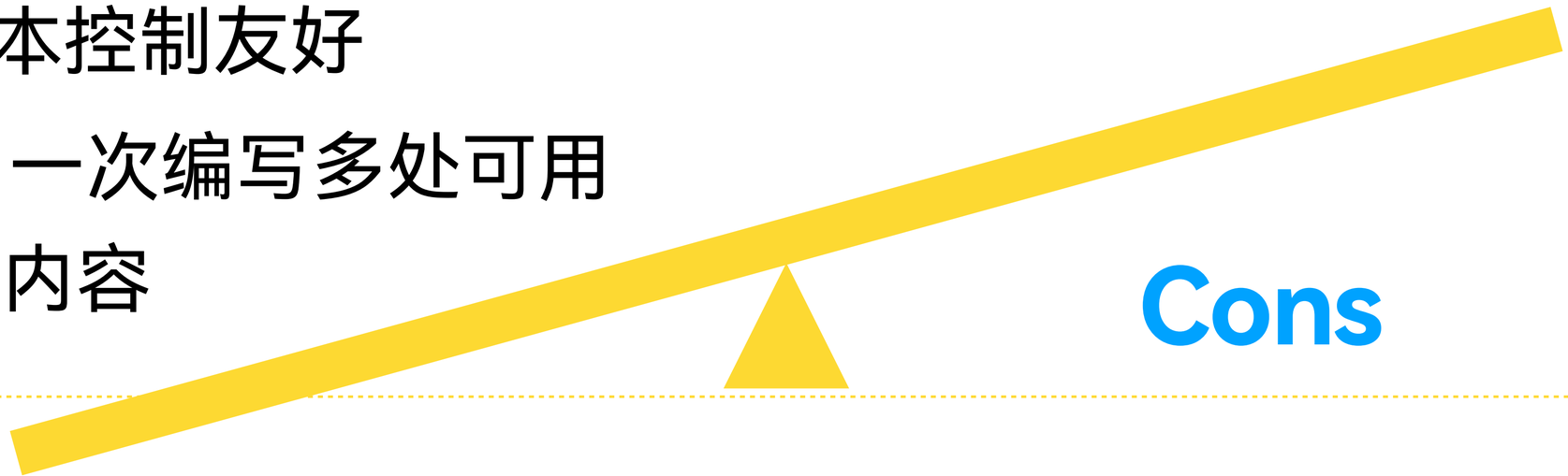
笔记干净整洁

排版简单　　跨平台展示

版本控制友好

一次编写多处可用

专注内容



Cons

对非相关行业
可读性差

不提供图片和公式
的原生支持

编辑器推荐

Typora

特点：所见即所得，实时预览，提供丰富的主题和扩展语法；
几年前是一个非常舒服的开源项目，但是商业化运作以后……
适合：初学者、注重写作体验的同学
优势：界面简洁，导出功能强大

Visual Studio Code

特点：常用 IDE，适合大型工程
适合：项目的文档撰写（以及不愿意买正版 Typora 的同学）
优势：与其他程序、项目和 Git 集成度高
快捷键：⌘+K, V / Control+K, V

SYNTAX

02

基础语法</>

Reference: Github Flavored Markdown

1 标题

语法说明

使用 `#` 符号来创建标题，`#` 的数量表示标题级别：

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

显示样例

一级标题

二级标题

三级标题

注意事项

- 保持标题层级的一致性，不要跳级
- 一级标题通常只使用一次（作为文档标题）

1 标题

语法说明

使用 `#` 符号来创建标题，`#` 的数量表示标题级别：

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

显示样例

一级标题

二级标题

三级标题

注意事项

- 保持标题层级的一致性，不要跳级
- 一级标题通常只使用一次（作为文档标题）

2 文本修饰

语法说明

```markdown
*斜体文本* 或 _斜体文本_

**加粗文本** 或 __加粗文本__

***粗斜体文本*** 或 ___粗斜体文本___

~~删除线文本~~

<u>下划线文本</u>（HTML 语法）

正常文本~下标~ 和 ^上标^

==高亮文本==（部分编辑器支持）
```

显示样例

斜体文本

加粗文本

粗斜体文本

~~删除线文本~~

<u>下划线文本</u>

==高亮文本==

注意事项

- 多种样式支持嵌套，一般不存在优先级
- 对于更高级的样式控制可以通过LaTeX 实现
- 高级的Markdown编辑器可以直接解析<font>甚至一些嵌套 的样式，但是我们非常不建议这么写

4 引用和代码段

语法说明

```markdown
> 这是一个引用文本示例
>
> 可以包含多个段落
>
> > 引用也可以嵌套
```

显示样例

> 这是一个引用文本示例
>
> 可以包含多个段落
>
> > 引用也可以嵌套

语法说明

使用反引号 `` ` `` 包裹行内代码：

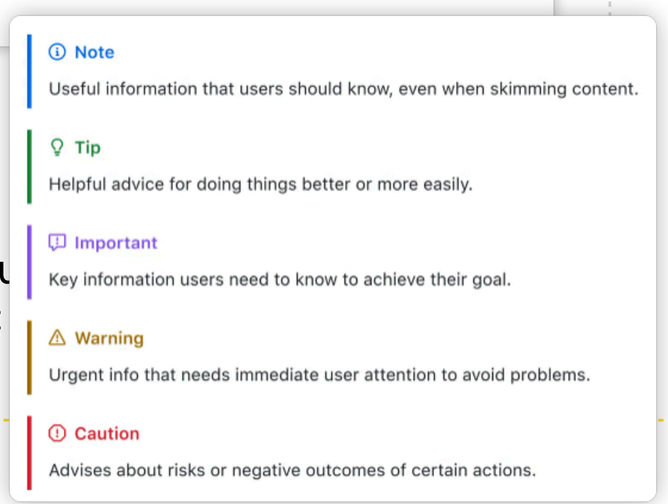
```markdown
这是一段包含 `inline code` 的文本。
```

显示样例

这是一段包含 `inline code` 的文本。

注意事项

- 在引用段的开头使用 GitHu
  的引用段，如 > [!Note] 等



4 引用和代码段

语法说明

使用三个反引号 ``` 创建代码块，可以指定语言以获得语法高亮：

markdown

````
```python
def hello_world():
    print("Hello, World!")

hello_world()
```


```javascript
function helloWorld() {
    console.log("Hello, World!");
}

helloWorld();
```
````

显示样例

python

```python
def hello_world():
    print("Hello, World!")

hello_world()
```

注意事项

- 代码高亮的方法取决于使用的编辑器/平台

  支持的常见语言标识：

  - `python`, `java`, `javascript`, `c`, `cpp`, `go`, `rust`
  - `html`, `css`, `json`, `xml`, `yaml`
  - `bash`, `shell`, `sql`, `markdown`

5 表格

语法说明

```markdown
| 表头1 | 表头2 | 表头3 |
|-------|-------|-------|
| 内容A | 内容B | 内容C |
| 内容D | 内容E | 内容F |



| 左对齐 | 居中对齐 | 右对齐 |
|:-------|:--------:|-------:|
| 左     | 中       | 右     |
| Left   | Center   | Right  |
```

显示样例

表头1	表头2	表头3
内容A	内容B	内容C
内容D	内容E	内容F

左对齐	居中对齐	右对齐
左	中	右
Left	Center	Right

注意事项

- 强烈建议使用 Typora 编辑表格，要不然还是贴截图算了

6 链接和图片

语法说明

```markdown
[链接文本](https://example.com)

[带标题的链接](https://example.com "鼠标悬停显示的标题")

<https://example.com>  （自动链接）
```

显示样例

链接文本

带标题的链接

鼠标悬停显示的标题

语法说明

```markdown
![图片描述](图片URL)

![图片描述](图片URL "图片标题")
```

显示样例

```
![UoN Logo](https://www.nottingham.edu.cn/SiteElements/Images/2023-Revamp/Logo/UoN-Logo.jpg)
```

6 链接和图片

注意事项

- Markdown 本身并没有储存图像的功能（与编辑器无关）
- Typora 在设置中提供了多种图片保存功能，如保存到子文件夹、自动上传图床等等，这些设置会同步到导出的 HTML 中，但是如果导出为 Word 或 PDF则会嵌入
- 不建议使用Github的图片服务，可能会因为网络环境造成一些不可知的错误
- Typora的图片提供更高级的编辑功能，本质上是将图片的渲染方式切换为 HTML 渲染，可能会造成一定的渲染问题，慎用
- 对于链接：Markdown 页面内支持内部跳转，通过例如[跳转到开头](#这是大标题的内容)实现，但请注意如果存在多个相同的标题只会返回第一个

7️⃣ 脚注

语法说明

```markdown
这是一段包含脚注的文本[^1]，还有另一个脚注[^note]。

[^1]: 这是第一个脚注的内容。
[^note]: 这是一个命名脚注。
```

显示样例

这是一段包含脚注的文本[1]，还有另一个脚注[note]。

[^1]: 这是第一个脚注的内容。 ↩

[^note]: 这是一个命名脚注。 ↩

TECHNICAL

03

更多语法

Reference: Github Flavored Markdown, Latex library, Mermaid

1 公式

markdown

```
$$
\LaTeX\mathrm{是一种很强大的排版工具，能够提供\color{red}{着色}
等功能}\\
\ce{Zn^2+  <=>[+ 2OH-][+ 2H+]
$\underset{\text{amphoteres Hydroxid}}{\ce{Zn(OH)2 v}}$
<=>[+ 2OH-][+ 2H+]  $\underset{\text{Hydroxozikat}}
{\ce{[Zn(OH)4]^2-}}$}
$$
```

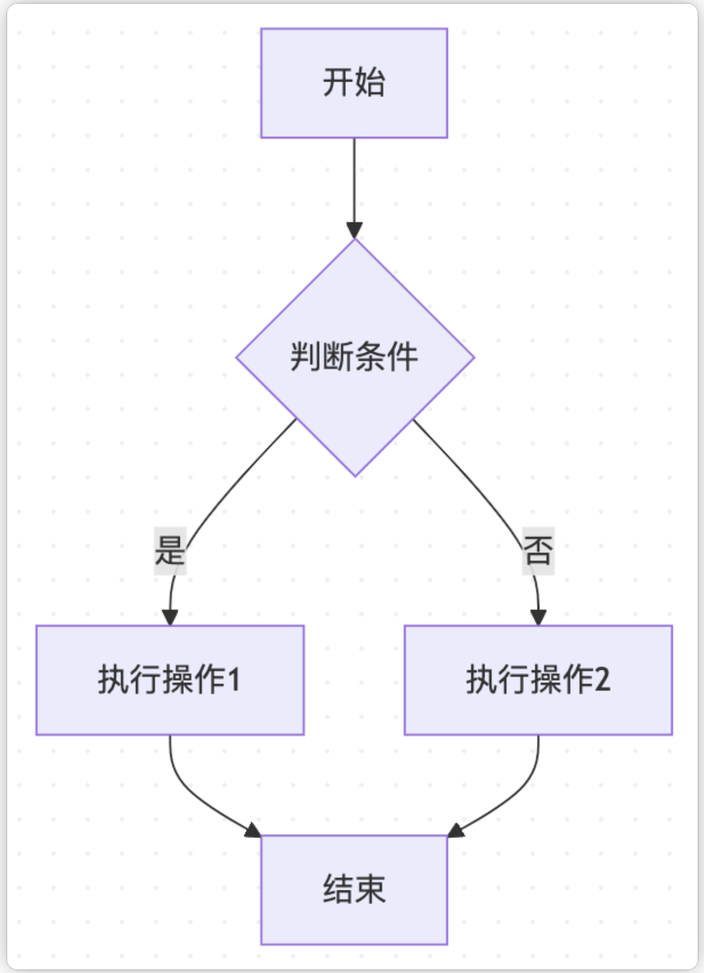
$$\LaTeX\mathrm{是一种很强大的排版工具，能够提供\color{red}{着色}等功能}$$

$$\ce{Zn^2+ <=>[+2 OH^-][+2 H^+] \underset{\text{amphoteres Hydroxid}}{Zn(OH)2 v} <=>[+2 OH^-][+2 H^+] \underset{\text{Hydroxozikat}}{[Zn(OH)4]^2-}} \tag{2}$$

2 Mermaid

markdown

```mermaid
flowchart TD
    A[开始] --> B{判断条件}
    B -->|是| C[执行操作1]
    B -->|否| D[执行操作2]
    C --> E[结束]
    D --> E
```



3 目录（ToC）

```markdown
[toc]
```

PRACTICE

04

小试牛刀

Try Markdown

尝试模仿这个文档 (Document accessible on Moodle&Github)

Solution: Implement a function to determine whether a string is a palindrome.

> A palindrome is a string that reads the same forward and backward, for example: `level`, `noon`.

Solution idea:

- **Remove spaces and ignore case** ↓

  Use `str.lower()` and `str.replace()` to preprocess the input.

- **Two-pointer method** ↓

  Use two pointers starting from both ends of the string moving toward the center and compare characters.

Code implementation:

```python
def is_palindrome(s):
    s = s.lower().replace(' ', '')
    return s == s[::-1]
```

Test cases

```python
print(is_palindrome("Level"))      # True
print(is_palindrome("Hello"))      # False
print(is_palindrome("A man a plan a canal Panama"))  # True
```

Task list:

- [x] ~~Handle case sensitivity~~
- [x] ~~Remove spaces~~
- [x] ~~Use slicing to reverse the string~~
- [ ] Handle punctuation (optional extension, not completed)

Summary: ↓

Palindrome checking is a common string-processing problem. It's important to master text preprocessing and slicing techniques. ↓

~~Don't forget to consider edge cases such as an empty string or a single-character string.~~ ↓

Practicing multiple implementations will improve coding skills!

Q & A

05

有疑问？欢迎提问

Lijie ZHOU (20809020) scylz12@nottingham.edu.cn