📌 GitHub 平台学习

GitHub 是一个面向开发者和内容创作者的协作平台,它基于版本控制工具 Git,提供了强大的 项目管理、团队协作、版本追踪、问题反馈和网站托管功能。它不仅适用于程序员,也逐渐成为研究者、数据分析师、作家等使用的工具。

GitHub 的核心价值:

- 集成版本控制工具 Git,跟踪每一次文件修改
- 支持多人协作开发
- 提供开源交流社区 (Stars、Fork、PR)
- 可以将 Markdown 内容发布为网页 (通过 GitHub Pages)

🃤 核心功能详解:

1. Repository (仓库)

- 类似于一个项目文件夹,包含所有相关资源(代码、图片、说明文档等)。
- 可以创建公开仓库供他人参考或私有仓库用于私人项目。
- 每个仓库首页的 README.md 可展示项目简介和使用说明。

2. (Clone & Fork (复制与分叉)

- Clone 是将别人的仓库下载到你本地电脑,通常与 Git 工具一起使用(如 Git Bash、VS Code)。
- Fork 是在线复制他人的仓库到你自己的账户中,适用于你想贡献别人的开源项目,或者基于别人的项目进行二次开发。

3. 🚣 Commit & Push (提交与上传)

- Commit 是对文件的每次更改做出记录,可以加备注说明改了什么。
- Push 是将你本地的更改推送到 GitHub 服务器上, 使他人可见。

4. 🌿 Branch (分支管理)

- GitHub 允许在一个项目中创建多个"平行宇宙",用于测试新功能或修复错误。
- 示例: 主分支 main、开发分支 dev、功能分支 feature/login。

5. Pull Request (合并请求)

- 他人 Fork 或你自己创建分支后,通过发起 Pull Request (PR),让项目负责人审核你所做的更改。
- PR 是 GitHub 协作开发最关键的功能,支持讨论、代码审查、修改建议。

★ Issues (问题追踪系统)

- 用于记录和讨论 Bug、功能建议、任务安排。
- 支持分配负责人、添加标签(如 bug、enhancement)、关联 PR。

7. Actions (自动化 CI/CD)

- GitHub Actions 可以在项目提交更新后自动执行操作,比如测试代码、构建应用、部署网站。
- 提升协作效率,常用于工程项目或静态博客自动发布。

8. (GitHub Pages (个人主页 & 静态网页托管)

- 利用仓库中的 HTML 或 Markdown 文件生成个人网站。
- 只需命名仓库为 yourusername.github.io,即可自动部署到网页上。

🔽 第2页:GitHub 使用流程与个人操作心得

🏹 实践操作总结(具体操作流程):

1. 创建仓库

- 注册账户后,点击 New 创建仓库,命名为 yourusername.github.io。
- o 勾选 Add a README file, 初始化文档。
- o 可设置 Public (公开) 或 Private (私密)。

2. 上传笔记

- o 点击 Add file → Upload files, 上传学习笔记 (PDF 或 PPT) 。
- 提交说明 commit message 用于记录变更理由,如 Add Week1 Learning Notes。

3. 开启 GitHub Pages

- 。 进入仓库 → Settings → Pages → 选择 main 分支。
- 系统将生成访问链接(通常几分钟内生效)。

4. 个性化主页

- 编辑 README.md 文件,加入自我介绍、项目链接、联系方式。
- o 使用 Markdown 语法美化布局。

⊕ 使用心得

- GitHub 提供的图形化界面操作简洁,即便初学者也能快速上手。
- Markdown 和仓库管理工具结合,极大提升了信息组织效率。
- 学会使用 GitHub 后,个人笔记和项目可以公开展示,适合日后求职或申请科研项目时展示作品集。

Markdown 语法

📗 Markdown 是什么?

Markdown 是一种专为"可读性 + 可转换性"而设计的纯文本标记语言。它可通过简洁的语法快

速生成格式化文档,并自动渲染为网页。适合技术文档、项目说明、笔记写作等多种场景。

🔠 语法分类与示例:

1. 👺 标题

#一级标题 ## 二级标题

三级标题

2. → 强调 **加粗文本**

斜体文本

~~删除线文本~~

3. 🔗 超链接与图片

[访问 GitHub](https://github.com)

![Logo 示例](https://example.com/logo.png)

4. 💬 引用与代码块

> 引用文字

`print("Hello")` # 行内代码

```python

#多行代码块

def hello():

print("Hello World")

## 5. 📊 表格与分割线

```markdown

|姓名|专业|

|----|----|

, |张三|经济学|

| 李四 | 金融学 |

🗿 Markdown 应用场景拓展

应用领域 示例

学术 课程讲义、会议纪要、实验记录 项目开发 技术文档、接口说明书、安装手册

自媒体 博客文章、公众号草稿 展示型网页简历、个人简介、项目作品集 团队协作 文档管理、任务分派、模板撰写

📌 总结反思

- 本周通过 GitHub 与 Markdown 的系统学习,不仅掌握了基本语法和操作流程,还初步建 立了"技术笔记结构化管理"意识。
- 更深刻理解了版本控制与协作的重要性。