**GitHub 学习笔记**

**1. GitHub 概述**

GitHub 是一个基于 Git 的代码托管平台，提供了版本控制和协作功能。它允许开发者将代码托管在远程仓库中，并与他人进行协作开发。GitHub 支持公共和私有仓库，并提供了大量的开源项目和社区支持。

**2. 基本概念**

**仓库（Repository）**：用于存储项目的地方，包括项目的所有文件和历史版本。

**分支（Branch）**：允许开发者在独立的环境中进行开发，避免对主分支（通常是 main 或 master）造成直接影响。

**提交（Commit）**：记录对文件的更改，每次提交都包含了对文件修改的快照和描述。

**合并（Merge）**：将一个分支的更改合并到另一个分支中。

**拉取请求（Pull Request）**：在将分支的更改合并到主分支之前，开发者可以创建拉取请求以进行代码审查和讨论。

**克隆（Clone）**：将远程仓库复制到本地计算机中进行开发。

**推送（Push）**：将本地的更改上传到远程仓库中。

**拉取（Pull）**：从远程仓库获取最新的更改并合并到本地仓库。

**3. GitHub 的基本操作**

**创建仓库**：

登录 GitHub 账户。

点击右上角的 "+" 图标，选择 "New repository"。

输入仓库名称、描述、选择公开或私有，然后点击 "Create repository"。

**克隆仓库**：

打开需要克隆的仓库页面。

点击绿色的 "Code" 按钮，复制仓库的 URL。

在终端或命令行中输入 git clone <URL>，将仓库克隆到本地。

**提交更改**：

在本地仓库中进行更改。

使用 git add <文件名> 将更改的文件添加到暂存区。

使用 git commit -m "提交信息" 提交更改。

使用 git push 将更改推送到远程仓库。

**创建分支**：

使用 git checkout -b <分支名> 创建并切换到新分支。

在新分支中进行开发和提交更改。

使用 git push origin <分支名> 将分支推送到远程仓库。

**合并分支**：

切换到需要合并到的主分支：git checkout main。

使用 git merge <分支名> 将指定的分支合并到主分支。

处理可能出现的冲突，完成合并。

**创建拉取请求**：

在 GitHub 上打开要合并的分支。

点击 "Pull request" 按钮。

添加标题和描述，选择要合并的目标分支，点击 "Create pull request"。

请求审查，讨论更改，直到合并。

**4. 实用技巧**

**使用 Issues 管理任务和Bug**：

在仓库页面的 "Issues" 标签下，创建和管理任务、Bug、功能请求等。

**配置 README 文件**：

在仓库根目录中创建 README.md 文件，用于描述项目的基本信息、使用方法和贡献指南。

**利用 GitHub Actions 自动化工作流**：

使用 GitHub Actions 设置自动化工作流，如测试、构建和部署。

**5. 常见问题**

**如何解决冲突**：

合并分支时，如果出现冲突，Git 会标记冲突的文件。需要手动编辑文件，解决冲突后重新提交。

**如何撤销更改**：

使用 git checkout -- <文件名> 撤销未提交的更改。

使用 git reset --hard 撤销所有未提交的更改（谨慎使用）。