

# Git、GitHub Pages、CMake 入门

---

## 任务概述

这些小任务旨在帮助你入门Git、GitHub Pages和CMake，这些技术将成为你未来学习和工作的重要一环。任务包含多个步骤，每个步骤都有不同的难度。完成任务后，你将学会如何使用Git来管理项目、使用GitHub Pages托管静态网站和使用CMake构建项目。

*Tips: 尽量做，做到哪提交到哪*

## 任务一：Git分支和协作

### git是什么？为什么要学git？

Git 是一个分布式版本控制系统，它可以追踪和管理项目的代码变更。每个开发者都可以克隆整个代码库到本地，进行修改，然后将更改推送回主仓库，以便其他人查看和合并。

- 版本控制：Git 允许开发者跟踪项目的每个版本和修改。这意味着你可以轻松地回溯到以前的版本，查看代码的历史，并了解代码是如何演化的。
- 协作：Git 是多人协作的关键工具。多个开发者可以同时在一个项目上工作，而不会相互干扰。他们可以在不同的分支上进行独立的开发，然后将更改合并到主分支。
- .....不啰嗦了，不知道的同学可以自己再去深入了解

### 步骤概述：

### 创建和初始化Git仓库

- 创建一个github账号（如果不会魔法，也可以创建gitee账号）
- fork这个仓库：[https://github.com/val213/IBMinSCUT\\_test\\_Round2](https://github.com/val213/IBMinSCUT_test_Round2)（如果你使用的是gitee，fork另外一个仓库：<https://gitee.com/Yipper0915/ibm2023task>，后面的操作一致）

### 本地提交远程仓库

- 安装并配置Git（你可以用git的纯命令行形式，也可以选择GitHub Desktop、gitkraken等GUI）。
- 在本地创建一个用来存放项目的新文件夹。
- 克隆你刚刚fork完的仓库到本地。
- 创建一个新的分支，以你的年级班级和姓名缩写命名，如 **222y1w**。
- 切换到你的新分支。并添加一个.txt文件，以你的年级班级和姓名命名，如 **222y1w.txt**。
- 添加文件到暂存区。
- 提交更改并添加提交信息。
- 把本地的新分支推送到你的远程仓库。
- 在原始仓库（IBMinSCUT\_test\_Round2或者Yipper0915/ibm2023task）提出并请求pull request。

### 撤销更改和回滚版本

如果有兴趣，你还可以尝试以下操作，记得截图~：

- 查看提交历史。

- 撤销本地的未提交更改。
- 创建新的提交以回滚到以前的版本。

## 任务二：GitHub Pages网站托管

### 为什么要使用 **GitHub Pages**？

GitHub Pages是一个免费的静态网站托管服务，它可以帮助你将自己的静态网站部署到互联网上，让你的项目、作品或个人博客在线可访问。以下是一些使用 GitHub Pages 的理由：

- 免费托管：GitHub Pages 提供免费的托管服务，不需要支付额外费用。
- 简便快捷：使用 GitHub Pages，你可以将静态网站文件存储在 GitHub 仓库中，无需购买独立的托管服务或配置服务器。
- 版本控制：你可以将网站的源代码与 GitHub 仓库一起管理，实现版本控制和协作。
- 自定义域名：你可以选择使用自己的域名来访问你的网站，提高专业性。

#### 步骤概述：

1. 准备网站文件 创建一个文件夹，用于存放你的网站文件。这些文件可以包括 HTML、CSS、JavaScript、图像等。
2. 创建 GitHub 仓库 在 GitHub 上创建一个新的仓库，仓库的名称可以是你想要的域名，例如 `your-username.github.io`。
3. 上传网站文件 将你的网站文件上传到你新创建的 GitHub 仓库中。确保你的网站的首页命名为 `index.html`。
4. 启用 GitHub Pages 打开你的 GitHub 仓库，在仓库的设置页面中，找到 GitHub Pages 部分。

在 GitHub Pages 部分，选择从主分支（通常是 "main" 或 "master"）构建你的网站。

保存设置后，GitHub 将为你的仓库创建一个网站，并提供一个访问网站的链接。

5. 访问你的网站 访问你的网站链接，查看你的网站是否正常运行。
6. 自定义域名（可选） 如果你想使用自定义域名访问你的网站，你可以在 GitHub Pages 设置中配置自己的域名。
7. 分享你的网站 分享你的网站链接，让其他人可以访问你的项目、作品集或个人博客。这些步骤将帮助你开始使用 GitHub Pages 托管你的静态网站。不仅能够快速将你的内容分享给他人，还可以通过版本控制轻松管理你的网站代码。

## 任务三：使用框架，构建和部署你的个人博客

### 为什么要搭建个人博客？

在这个任务中，你将学习如何搭建个人博客。个人博客是一个强大的工具，可以帮助你展示你的技能、经验和知识，与社区互动，记录你的成长，建立个人品牌，并与他人分享有价值的信息。

#### 步骤概述：

1. 选择博客框架 首先，选择一个博客框架，例如 Jekyll、Hugo、hexo 等。这些框架都是用来构建静态博客的工具，它们具有易于使用和自定义的特点。

2. 设置开发环境 安装所选框架所需的开发环境。这通常包括安装一个支持框架的编程语言（如Ruby、Node.js等）以及其他必要的工具。
3. 创建博客项目 使用框架提供的命令或模板，创建一个新的博客项目。这将包括博客的基本结构、页面模板等。
4. 编写博客文章 开始编写你的第一篇博客文章。使用Markdown或其他标记语言，为你的博客编写内容。博客框架通常提供了一个方便的地方来存放你的文章。
5. 自定义博客 根据你的喜好和需求，自定义博客的外观和布局。这包括选择主题、设计页面、添加个人信息等。
6. 添加功能和插件 根据需要，添加博客所需的功能和插件。这可能包括评论系统、社交分享按钮、标签云等。
7. 测试和优化 在本地测试你的博客，确保一切正常工作。修复任何问题，优化性能。
8. 部署博客 选择一个托管平台（如GitHub Pages、Netlify、Vercel等），将你的博客部署到互联网上。这通常涉及将你的代码上传到托管平台，并配置域名和HTTPS等。
9. 发布第一篇博客 一旦博客成功部署，发布你的第一篇博客文章。分享博客链接，让你的朋友和社区了解你的博客。
10. 持续更新 持续编写和发布新的博客文章，与读者互动，建立博客的受众群体。这些步骤将帮助你搭建一个个人博客，并为你提供一个展示和分享你的技能和知识的平台。博客的内容和风格完全取决于你的兴趣和目标，所以不断尝试和改进是一个重要的部分。祝你在任务三中取得成功！

## 任务四：使用CMake构建项目

### 什么是CMake？为什么要用Cmake？：

CMake是一个开源的构建工具，用于管理跨平台的项目的构建过程。它的主要目的是简化项目的构建和配置，使开发者能够轻松地将其代码在不同的操作系统和编译器上构建，而不需要为每个平台编写不同的构建脚本。

### 步骤概述：

1. 创建一个名为 "CMakeDemo" 的文件夹。
2. 编写一个简单的C++程序。
3. 创建名为 "CMakeLists.txt" 的CMake配置文件。
4. 在终端中构建项目并生成可执行文件。
5. 提供包含项目文件的压缩包。

## 相关参考文档和博客

- GitHub官方文档：<https://docs.github.com/en>
- GitHub Pages官方文档：<https://docs.github.com/en/pages>
- CMake官方文档：<https://cmake.org/documentation/>
- 【关于Git这一篇就够了】<http://t.csdn.cn/N28ee>
- 【Git + GitHub 10分钟完全入门】[https://www.bilibili.com/video/BV1KD4y1S7FL/?share\\_source=copy\\_web&vd\\_source=589ae6273d7e5597533c35b38ecfdb77](https://www.bilibili.com/video/BV1KD4y1S7FL/?share_source=copy_web&vd_source=589ae6273d7e5597533c35b38ecfdb77)
- 【GitHub Pages 个人网站构建与发布】[https://www.bilibili.com/video/BV1hL4y1w72r/?share\\_source=copy\\_web&vd\\_source=589ae6273d7e5597533c35b38ecfdb77](https://www.bilibili.com/video/BV1hL4y1w72r/?share_source=copy_web&vd_source=589ae6273d7e5597533c35b38ecfdb77)
- 【hexo框架官方文档】<https://hexo.io/zh-cn/docs/index.html>
- 【软件构建: CMake 快速入门】[https://www.bilibili.com/video/BV1rR4y1E7n9/?share\\_source=copy\\_web&vd\\_source=589ae6273d7e5597533c35b38ecfdb77](https://www.bilibili.com/video/BV1rR4y1E7n9/?share_source=copy_web&vd_source=589ae6273d7e5597533c35b38ecfdb77)

这些参考文档和博客将为你提供更多关于Git、CMake和GitHub Pages的详细信息和教程。

## 注意事项

- 这个任务旨在培养你的Git、CMake和GitHub Pages技能，同时也是一个学习的机会。不要害怕犯错误，因为它们是学习的一部分。
- 任务三和任务四比较难，可能需要花费你不少时间。你可以根据自己的时间、能力或者兴趣选择是否完成。
- 如果你选择完成任务三和任务四，请确保提供相应的证据，例如博客的链接、压缩包以及过程中的相关截图等。

祝你好运，完成这些任务后，你将拥有更广泛的技术知识和实践经验。

如果有任何问题欢迎联系负责人。

## 联系与提交方式

添加负责人微信（微信号：[val213](#)）发送相关文件或者打包压缩文件发送到邮箱：[val213666@gmail.com](mailto:val213666@gmail.com)

如果选择发送邮件，发送后可以在群里@我或者微信问我确定是否被收到。

最后，祝你好运。