

## 简介

GitHub Pages 是一项静态网站托管服务，它直接从 GitHub 仓库获取 HTML、CSS 和 JavaScript 文件，通过构建过程运行文件，然后发布网站。

## 网站的类型

有三种类型的 GitHub Pages 网站：项目、用户和组织。每个 GitHub 帐户只能创建一个用户或组织网站。项目网站没有限制。

### 1. 用户网站

[用户网站](#)连接到特定的 GitHub 帐户，该地址中的三级域与 GitHub 的登录帐户相同。

### 2. 项目网站

[项目网站](#)是用户网站的子目录。

## 在 GitHub 上构建网站

### 1. 发布用户网站

要发布用户网站，必须创建名为“user.github.io”的用户帐户所拥有的仓库。

#### 步骤 1：创建仓库

- 1) 前往 [GitHub](#) 创建一个名为“user.github.io”的仓库，其中“user”是 GitHub 的登录帐户。
- 2) 前往“<https://github.com/user/user.github.io>”，在仓库名称下，点击“Settings”。

或前往“<https://github.com/user/user.github.io/settings>”。

- 3) 在“GitHub Pages”下，点击“Choose a theme”，然后选择主题并点击“Select theme”。

现在“<https://user.github.io>”已经自动发布，稍等片刻便可访问。

#### 步骤 2：使用 GitHub Desktop 部署网站

GitHub Desktop 是 Git 的图形化工具。

- 1) 前往“<https://desktop.github.com/>”下载并安装 GitHub Desktop 桌面程序。
- 2) 在本地磁盘上创建一个用于与 GitHub 仓库同步的文件夹“E:\GitHub”。
- 3) 启动 GitHub Desktop 程序，登录帐户。点击上方菜单栏的“File” — “Clone repository”，从仓库“user/user.github.io/”克隆到本地的“E:\GitHub\”目录下。
- 4) 删除“E:\GitHub\user.github.io\”目录下所有的文件。只保留名为“.git”的隐藏文件。
- 5) 前往“<http://jekyllthemes.org/>”下载主题，并复制文件至“E:\GitHub\user.github.io\”目录下。
- 6) 回到 GitHub Desktop，在“summary”一栏里标注，然后点击“Commit to main”，再点击“Push origin”，从本地磁盘的“E:\GitHub\user.github.io\”推送至“dangiaon.github.io”仓库。

### 2. 发布项目网站

#### 步骤 1：创建仓库

- 1) 前往 [GitHub](#) 创建一个名为“blog”的仓库。
- 2) 页面自动跳转至“<https://github.com/user/blog>”，然后新建分支（Branches）“gh-pages”。
- 3) 点击“settings”，在“GitHub Pages”中找到生成的项目网站：“<https://user.github.io/blog/>”。点击“Choose a theme”，然后选择主题并点击“Select theme”。
- 4) 返回“blog”仓库的设置页面：“<https://github.com/user/blog/settings>”。点击左侧的“Branches”。

5) 展开“Default branch”（默认分支）下拉框，选择“gh-pages”并点击“Update”。

6) 在弹出的对话框中点击“I understand, update the default branch.”。

现在“<https://user.github.io/blog/>”已经发布，稍等片刻便可访问。

步骤 2：使用 GitHub Desktop 部署网站

GitHub Desktop 是 Git 的图形化工具。

- 1) 打开 GitHub Desktop，点击上方菜单栏的“File” — “Clone repository”，从仓库“user/blog”克隆到本地的“E:\GitHub\”目录下。
- 2) 删除“E:\GitHub\blog\”目录下所有的文件。只保留名为“.git”的隐藏文件。
- 3) 前往“<http://jekyllthemes.org/>”，下载主题模板并解压至“E:\GitHub\blog”目录下。
- 4) 回到 GitHub Desktop，在“summary”一栏里标注，然后点击“Commit to main”，再点击“Push origin”，从本地磁盘“E:\GitHub\blog\”推送至“blog”仓库。

## Fork 与 Clone、Pull 与 Fetch 之间的区别

Fork：复制他人的 GitHub 仓库到自己的 GitHub 中。

Clone：从自己或他人的 GitHub 仓库克隆到本地的磁盘上。

Fork、Clone 和 Push origin 的操作步骤：

- 1) Fork。复制他人的 GitHub 仓库到自己的 GitHub 中。自己 GitHub 中的仓库是 Fork 源的一个分支。
- 2) Clone。从自己的 GitHub 仓库克隆到本地的磁盘上。
- 3) Push origin。从本地磁盘推送到自己的 GitHub 仓库中。

pull 等效于 fetch + merge

获取更新，然后合并。

```
1. git fetch origin master:tmp
2. git diff tmp
3. git merge tmp
```

上述过程等效于：

```
1. git pull origin master
```

## 目录结构（Jekyll）

- index.html 网站的主页。
- \_config.yml 配置文件。
- \_layouts 页面模板。
- \_includes 复用代码块。
- \_posts 博文。必须使用 markdown 编写。