

用 Octopress 搭建 Github Pages 博客

作者: [宇宙意识觉醒](#) | [心無極](#)

用 Octopress 搭建 Github Pages 博客

1. 准备工作
2. 下载和安装 Octopress
3. 创建博客网站
4. 使用主题
5. 使用插件
6. 修改配置文件
 - 以下是一个具体的 Octopress 配置示例
7. 编写博客文章和页面
8. 本地预览博客
9. 部署博客到 GitHub Pages

1. 准备工作

- 在 GitHub 上创建新的存储库，并将其命名为 `<username>.github.io`。
 - 如果您想要创建一个博客网站而不是个人或组织页面，请将存储库名称改为其他名称（例如 `blog`）。
- 安装 Git 并设置 Git 配置。
 - 如果您已经安装过 Git，则可以跳过此步骤。
 - 否则，您可以从 Git 的官方网站上下载适合您操作系统的版本，并按照安装向导进行安装。安装完成后，请运行以下命令配置全局 Git 配置：

```
git config --global user.name "Your Name"
git config --global user.email "your_email@example.com"
```

- 这些命令将设置您的用户名和电子邮件地址，并在将来的 Git 操作中使用它们。
- 安装必要软件和依赖项
 - 安装 Git 和 Ruby，并确保它们都设置为系统路径。
 - 如果您已经安装过 Ruby 和 Bundler，则可以跳过此步骤。

- 否则，您可以从 Ruby 的官方网站上下载适合您操作系统的版本，并按照安装向导进行安装。安装完成后，请运行以下命令安装 Bundler：

```
gem install bundler
```

2. 下载和安装 Octopress

- 通过 Github 安装 Octopress：
 - 打开终端并切换到您想要安装 Octopress 的目录。
 - 运行以下命令从 GitHub 上克隆 Octopress 存储库：

```
git clone https://github.com/octopress/octopress.git
```

- 或者，通过 gem 安装 Octopress：
 - 打开终端并运行以下命令安装 Octopress gem：

```
gem install octopress
```

3. 创建博客网站

- 运行 `octopress new <blog-name>` 命令创建新的博客网站。
- 修改 `_config.yml` 文件，配置网站信息、主题和插件等内容。

4. 使用主题

- Octopress 默认使用的主题是 Classic，如果您希望更改主题，可以在官方主题库 <https://github.com/octopress/themes> 中选择一个主题，并按照其文档说明进行安装和配置。
- 从官方主题库中选择一个主题，并通过 `octopress install [theme-name]` 命令将其安装到您的博客中。
- 如果您想要使用 Git 主题，则可以将其作为 Git 子模块添加到 `_themes` 目录中。具体操作方法如下：
 - 打开终端并切换到您的 Octopress 存储库所在的目录。
 - 运行以下命令将 Git 主题作为子模块添加到 `_themes` 目录中：

```
git submodule add https://github.com/octopress/[theme-name].git
.themes/[theme-name]

rake install[theme-name]
```

- 进入 `_config.yml` 文件并将 `theme` 字段设置为新添加的主题名称。

- 运行以下命令更新子模块并拉取主题代码：

```
git submodule update --init --recursive
```

- 重新生成博客网站即可看到新主题。

5. 使用插件

- Octopress 有许多可用的插件可供使用，您可以在官方插件库 <https://github.com/octopress/plugins> 中找到并下载它们。在使用插件之前，请确保先阅读该插件的文档并按照说明进行安装和配置。
- 在 `_config.yml` 文件的 `gems:` 列表中添加需要使用的插件名称。
- 如果您想要使用 Git 插件，则可以将其作为 Git 子模块添加到 `_plugins` 目录中。具体操作方法如下：
 - 打开终端并切换到您的 Octopress 存储库所在的目录。
 - 运行以下命令将 Git 插件作为子模块添加到 `_plugins` 目录中：

```
git submodule add https://github.com/octopress/[plugin-name].git
_plugins/[plugin-name]

rake install[plugin-name]
```

- 运行以下命令更新子模块并拉取插件代码：

```
git submodule update --init --recursive
```

- 在 `_config.yml` 文件中将 `gems:` 字段添加新插件的名称。
- 重新生成博客网站即可使用新插件。

6. 修改配置文件

Octopress 的配置文件是 `_config.yml`，它是一个 YAML 格式的文本文件。您可以通过编辑这个文件来修改博客的基本设置。例如，您可以更改博客的名称、作者信息、默认分类等。

在命令行中进入 Octopress 目录，然后输入以下命令来编辑 `_config.yml` 文件：

```
nano _config.yml
```

以下是 `_config.yml` 文件中一些常见的字段及其说明：

```
# 博客名称
title: Octopress Blog

# 作者信息
```

```
author:
  name: Your Name
  email: your@email.com
  url: http://yourwebsite.com

# 基本 URL 路径, 用于生成文章和页面的永久链接
url: http://example.com

# GitHub 用户名 (用于部署到 GitHub Pages)
github_user: yourusername

# 默认语言
language: en

# 默认分类
default_category: General

# 主题名称
theme: classic

# 插件
gems:
  - octopress-sitemap
  - octopress-tag-cloud
  - octopress-paginate

# 静态文件目录
source: .

# 输出目录
destination: ./_site
```

您可以像编辑任何文本文件一样来编辑 `_config.yml` 文件, 修改其中的值即可更改相关设置。

除了上面提到的常见配置字段之外, Octopress 还有很多其他可以配置的选项。以下是一些常用的配置示例:

1. 设置 Disqus 评论系统

```
disqus_shortname: your_disqus_shortname
```

2. 配置 Google Analytics

```
google_analytics_id: UA-XXXXXXXX-X
```

3. 配置 RSS 订阅

```
# 是否生成 RSS Feed, 默认为 true
rss: true

# RSS Feed 文件名, 默认为 "atom.xml"
rss_name: atom.xml

# 网站 Feed URL
rss_url: /atom.xml
```

4. 配置社交媒体链接

```
# 社交媒体链接
social:
  twitter: https://twitter.com/your_twitter_username
  github: https://github.com/your_github_username
  facebook: https://www.facebook.com/your_facebook_username
```

5. 配置代码高亮

Octopress 默认使用 Pygments 来实现代码高亮。您可以在 `_config.yml` 文件中配置 Pygments 的样式:

```
# Pygments 样式
pygments_style: monokai
```

6. 设置文章摘要的长度

```
# 摘要长度
excerpt_length: 250
```

7. 配置图片存储路径

```
# 图片路径
img_dir: /assets/images/
```

8. 配置缩略图生成

```
# 是否启用自动生成缩略图, 默认为 true
auto_thumbnail_image: true

# 缩略图尺寸
thumbnail_size: 400x300^

# 缩略图质量
thumbnail_quality: 80

# 缩略图文件名后缀
thumbnail_suffix: _thumb
```

9. 配置分类和标签

```
# 启用分类
categories:
  - General
  - Technology
  - Lifestyle
```

```
# 启用标签
tags:
  - Octopress
  - Blogging
  - Tips
```

10. 配置时间格式

```
# 时间格式
time_format: "%Y-%m-%d %H:%M:%S %Z"
```

11. 设置默认的布局

```
# 默认布局，可以设置为 post、page 或其他自定义布局名称
default_layout: post
```

12. 配置文章和页面 URL 格式

```
# 文章 URL
permalink: /blog/:year/:month/:day/:title.html

# 页面 URL
page_permalink: /:title/
```

13. 配置 CDN

```
# 启用 CDN
cdn:
  enable: true

# CDN 域名
cdn_url: http://cdn.yourdomain.com/
```

14. 配置 RSS 订阅

```
# 是否生成 RSS Feed，默认为 true
rss: true

# RSS Feed 文件名，默认为 "atom.xml"
rss_name: atom.xml

# 网站 Feed URL
rss_url: /atom.xml
```

15. 配置多语言支持

```
# 支持的语言列表
languages:
  - en
  - zh-CN

# 默认语言
language: en

# 翻译文件路径
translations_dir: _i18n
```

保存修改后，重新生成博客网站即可生效。

Octopress 支持将博客部署到 Github Pages 上。以下是一些常见的 Github Pages 部署配置：

1. 设置 GitHub 用户名

在 `_config.yml` 文件中，将 `github_user` 字段设置为您的 GitHub 用户名：

```
github_user: your_github_username
```

2. 配置 GitHub Pages 分支

Github Pages 有两种发布方式：`master` 分支和 `gh-pages` 分支。如果您想使用 `gh-pages` 分支来部署博客，需要进行如下设置：

在 `_config.yml` 文件中，将 `deploy_branch` 字段设置为 `gh-pages`，并将 `url` 字段设置为 `http://your_github_username.github.io`。

```
# 部署目标
deploy:
  type: git
  deploy_branch: gh-pages

# 基本 URL 路径，用于生成文章和页面的永久链接
url: http://your_github_username.github.io
```

3. 配置 CNAME

如果您想将自己的域名绑定到 Github Pages，可以在项目根目录下创建一个名为 `CNAME` 的文件，并将您要绑定的域名写入该文件中。例如：

```
example.com
```

4. 配置 Travis CI

Travis CI 是一个持续集成工具，可以帮助我们在每次代码更新后自动构建并部署网站。如果您想使用 Travis CI 来自动化部署您的博客，可以按照以下步骤进行配置：

- 在 Github 上创建一个名为 `.travis.yml` 的文件，将下面的内容复制进去：

```
language: ruby
cache: bundler
```

```

branches:
  only:
    - source

rvm:
  - 2.6
  - 2.7

before_script:
  - chmod +x ./script/cibuild

script: ./script/cibuild

```

- 在项目根目录下创建一个名为 `script` 的文件夹。
- 在 `script` 文件夹中创建一个名为 `cibuild` 的脚本文件，并将下面的内容复制进去：

```

#!/usr/bin/env bash

set -e # Exit with nonzero exit code if anything fails

echo "Building site..."
bundle exec jekyll build --trace

echo "Deploying site..."

# 如果您使用 master 分支发布，则更改以下命令：
# git push -f "https://${GH_TOKEN}@github.com/${TRAVIS_REPO_SLUG}.git"
master:gh-pages
git clone https://github.com/${TRAVIS_REPO_SLUG}.git deploy
cd deploy
git checkout --orphan gh-pages
git rm -rf .
cp -r ../_site/* .
touch .nojekyll
git add -A .
git -c user.name='Travis CI' -c user.email='travis@travis-ci.org' commit -m
'Deploy site'
git push -f origin gh-pages
cd ..

```

- 创建一个 Github token，并将其添加到 Travis CI 中作为环境变量。在 Travis CI 控制台中找到你的项目，然后在 "Settings" 页面中添加一个名为 `GH_TOKEN` 的环境变量，将您的 Github token 添加到该环境变量中。
- 将您的博客源代码推送到 Github，并在 Travis CI 中启用自动构建。现在，每当您向 Github 推送新的更改时，Travis CI 就会自动构建和部署您的网站。

以下是一个具体的 Octopress 配置示例

假设您需要创建一个名为 "myblog" 的博客，使用 "default" 主题，并且希望启用 Disqus 评论系统和 Google Analytics。

首先，在命令行中进入您的 Octopress 安装目录，然后打开 `_config.yml` 文件进行编辑。将其修改如下：

```
# 博客名称
title: My Blog

# 作者信息
author:
  name: Your Name
  email: your@email.com
  url: http://yourwebsite.com

# 基本 URL 路径，用于生成文章和页面的永久链接
url: http://example.com/myblog

# 默认语言
language: en

# 主题名称
theme: default

# 插件
gems:
  - octopress-disqus
  - octopress-analytics

# Disqus 评论系统设置
disqus_shortcode: myblog

# Google Analytics 设置
google_analytics_id: UA-XXXXXXXXXX-X
```

在上述配置中，我们将博客名称设置为 "My Blog"，并将主题名称设置为 "default"。同时，我们还启用了 Disqus 评论系统和 Google Analytics，分别使用了 `octopress-disqus` 和 `octopress-analytics` 插件实现。

注意，这里的 `disqus_shortcode` 和 `google_analytics_id` 应该替换为您自己的 ID。

保存 `_config.yml` 文件，然后重新生成博客网站即可看到新的配置生效。可以使用以下命令重新生成博客网站：

```
octopress build
```

或者，您可以在本地启动一个服务来实时预览变更。在命令行中，输入以下命令运行本地服务：

```
octopress serve
```

7. 编写博客文章和页面

- 编写博客文章

- 在 `_posts` 目录中创建一个新的 Markdown 格式文件，文件名应该符合 `YYYY-MM-DD-title.md` 的格式（其中，“YYYY-MM-DD”表示文章发布日期，“title”是文章的标题）。
- 使用 Markdown 语法编写文章内容，并在文件开头添加 YAML 头信息（即文章元数据），如下所示：

```
---
layout: post
title: Your Title Here
date: YYYY-MM-DD HH:MM:SS +/-TTTT
categories: [category1, category2]
tags: [tag1, tag2]
author: Your Name
image: /path/to/featured/image.jpg
---
```

其中，`layout` 表示使用的布局，可以是主题自带的布局或自定义的布局；`title` 表示文章标题；`date` 表示文章发布日期和时间；`categories` 和 `tags` 表示文章所属的分类和标签；`author` 表示文章作者；`image` 表示显示在文章列表和分享链接中的特色图片（可选）。

- 保存文件并提交到 Git 存储库中：

```
git add _posts/YYYY-MM-DD-title.md
git commit -m "Add new blog post"
```

- 创建新的页面

- 在 `_pages` 目录中创建一个新的 Markdown 格式文件，文件名可以是任意名称，但必须以 `.md` 结尾。
- 使用 Markdown 语法编写页面内容，并在文件开头添加 YAML 头信息，如下所示：

```
---
layout: page
title: Your Title Here
permalink: /your-permalink/
---
```

其中，`layout` 表示使用的布局；`title` 表示页面标题；`permalink` 表示页面的永久链接地址。

- 保存文件并提交到 Git 存储库中：

```
git add _pages/your-page.md
git commit -m "Add new page"
```

- 编辑博客文章和页面

- 在 `_posts` 目录中找到要编辑的博客文章，使用 Markdown 语法对文章内容进行修改。
- 在 `_pages` 目录中找到要编辑的页面，使用 Markdown 语法对页面内容进行修改。

- 保存文件并提交到 Git 存储库中：

```
git add _posts/YYYY-MM-DD-title.md
git commit -m "Edit blog post"

git add _pages/your-page.md
git commit -m "Edit page"
```

- 删除博客文章和页面

- 在 `_posts` 目录中找到要删除的博客文章，将其从目录中删除。

```
rm _posts/YYYY-MM-DD-title.md
git rm _posts/YYYY-MM-DD-title.md
git commit -m "Delete blog post"
```

- 在 `_pages` 目录中找到要删除的页面，将其从目录中删除。

```
rm _pages/your-page.md
git rm _pages/your-page.md
git commit -m "Delete page"
```

- 添加图片和其他资源

- 将图片和其他资源添加到 `assets` 目录下的适当子目录中。
- 在博客文章或页面中使用以下 Markdown 语法引用图片和资源：

```
![alt text](/path/to/image.jpg)
[link text](/path/to/file.pdf)
```

- 保存文件并提交到 Git 存储库中：

```
git add assets/
git add _posts/YYYY-MM-DD-title.md
git commit -m "Add new image and resource"
```

8. 本地预览博客

- 运行 `octopress serve` 命令启动本地服务器，预览博客效果。
- 打开浏览器，访问 `http://localhost:4000/`，查看博客文章和页面的效果。
- 当您修改了博客内容后，只需要按下 `Ctrl+C` 组合键来停止服务，然后再次输入 `octopress serve` 命令即可重新启动本地服务器并预览博客。

9. 部署博客到 GitHub Pages

- 在 GitHub 上创建一个用于部署博客的分支，并将其命名为 `gh-pages`。
 - 如果您想要创建一个博客网站而不是个人或组织页面，请将分支名称改为 `master`。
- 在完成博客文章和页面的编辑后，将其提交到 Git 存储库中：

```
git add _posts/YYYY-MM-DD-title.md
git add _pages/your-page.md
git commit -m "Edit blog post and page"
```

- 运行 `octopress deploy` 命令将静态 HTML 文件和其他资源推送到 GitHub 上的 `gh-pages` 分支上：

```
octopress deploy
```

- 访问 `<username>.github.io` 网址查看博客文章和页面。

这个流程应该可以帮助您快速地创建并部署自己的 GitHub Pages 博客，并添加 Git 主题和 Git 插件。

如果您选择通过 Github 安装 Octopress，请注意在运行 `bundle install` 命令之前，先进入 `octopress` 目录并运行 `gem install bundler` 命令安装 Bundler。然后再运行 `bundle install` 安装其他依赖项。

无论您是通过 Github 还是 gem 安装 Octopress，都应该注意 Octopress 的版本和相关插件的兼容性。建议在安装和升级时查看 Octopress 和插件的最新版本信息，并根据需要进行更新。

此外，如果您发现某些插件无法正常工作或导致错误，请检查其文档和代码库中是否有已知问题和解决方法。如果没有，请尝试联系插件作者或社区寻求帮助。

希望这个完整的流程大纲能够对您有所帮助，祝您搭建成功并创作出优秀的博客！

文件由 [宇宙意识觉醒](#) | [心無極](#) 免费分享，引用请注明来源。