

---

# 博客从 0 到 10

---

Hexo + GitHub, 或许是较为详细、小白的教程。

---

作者: jinghu-moon

时间: 2022.03.01

转载，注明来源即可。联系方式：[khi215215@163.com](mailto:khi215215@163.com)

# 目录

## 目录

---

### 目录

#### 第一章 预备知识

##### 1.1 为什么搭建博客

1.1.1 我真的有必要搭建博客吗?

1.1.2 我能做到持续输出文章吗?

##### 1.2 必看视频 ☆

##### 1.3 代码相关知识

1.3.1 Git 和 GitHub

1.3.2 前端三件套

1.3.2 Markdown 语法

##### 1.4 常见博客框架

1.4.1 动态博客

1.4.2 静态博客

##### 1.5 搜索与提问

1.5.1 提问

1.5.2 搜索

##### 1.6 文章排版

##### 1.7 版权意识

1.7.1 许可协议

1.7.2 无版权图片

1.7.3 图标库

1.7.4 可商用字体

#### 第二章 本地部署

##### 2.1 Node.js 下载

- 2.2 Node.js 安装流程
- 2.3 Git 下载
- 2.4 Git 安装流程
  - 2.4.1 双击运行程序
  - 2.4.2 选择安装目录
  - 2.4.3 选择组件
  - 2.4.4 选择开始菜单文件夹
  - 2.4.5 选择编辑器
  - 2.4.6 选择默认分支名
  - 2.4.7 环境变量、SSH、HTTPS 设置
  - 2.4.8 行结束转换、终端模拟器设置
  - 2.4.9 Git Pull 风格、凭据管理设置
  - 2.4.10 额外选项与实验选项
  - 2.4.11 安装完成
- 2.5 Git Bash 界面美化
  - 2.5.1 进入设置
  - 2.5.2 切换语言
  - 2.5.3 外观设置
  - 2.5.4 文本设置
  - 2.5.5 美化完成
- 2.6 检查是否安装成功
- 2.7 安装 Hexo
  - 2.7.1 建立博客文件夹
  - 2.7.2 下载 Hexo
  - 2.7.3 Hexo 初始化
  - 2.7.4 构建 Hexo
  - 2.7.5 启动 Hexo 本地服务器
  - 2.7.6 查看网站

## 第三章 远程部署

- 3.1 创建 GitHub 仓库
  - 3.1.1 注册 GitHub 账号

- 3.1.2 创建仓库
- 3.2 使用远程部署插件
  - 3.2.1 安装插件
  - 3.2.2 配置插件
- 3.3 Github SSH 配置
  - 3.3.1 生成密钥
  - 3.3.2 复制密钥到 Github
  - 2.3.3 验证是否配置成功
- 3.4 GitHub 账户验证
- 3.5 远程访问博客

## **第四章 网站配置详解**

- 4.1 博客文件夹
- 4.2 常用命令
  - 4.2.1 基本命令
  - 4.2.2 推荐命令
  - 4.2.3 其他命令
- 4.3 网站属性
- 4.4 发布文章

## **第五章 主题美化**

- 5.1 主题下载
- 5.2 主题推荐
- 5.3 主题配置
  - 5.3.1 图床
  - 5.3.2 开启二级菜单
  - 5.3.3 全局字体
  - 5.3.4 展示 PV 与 UV 统计
  - 5.3.3 自定义社交图标
  - 5.3.5 首页展示一言 / 今日诗词
  - 5.3.6 快速在页脚使用一言、今日诗词
  - 5.3.7 Fluid 页脚增加网站运行时长
  - 5.3.8 版权声明优化

- 5.3.9 首页图片动画
- 5.3.10 自定义 404 页面
- 5.3.10 调整Twikoo 评论框大小
- 5.3.11 网页标题栏设置
- 5.3.11 代码块折叠
- 5.3.12 默认在文章末显示结束语
- 5.3.13 添加谷歌翻译
- 5.3.14 博客添加书签页
- 5.3.15 博客添加公告栏

## **第六章 开启评论**

- 6.1 使用
- 6.2 配置
  - 6.2.1 即时通知
  - 6.2.2 邮件通知
- 6.3 美化

## **第七章 自定义域名**

- 7.1 什么是域名
- 7.2 常见域名注册商
  - 7.2.1 国内
  - 7.2.2 国外
- 7.3 购买流程
- 7.4 域名解析 (DNS 解析)
- 7.4 访问网站
- 7.5 网站备案 (可选)

## **第八章 常用插件**

- 8.1 下载与卸载
- 8.2 hexo-abbrlink
- 8.2 hexo-neat
- 8.3 Artitalk
- 8.4 hexo-browsersync
- 8.5 hexo-generator-random

- 8.6 hexo-tag-hint
  - 8.7 hexo-filter-nofollow
  - 8.8 hexo-recommended-posts
  - 8.9 hexo-wordcount
  - 8.10 hexo-auto-category
  - 8.11 hexo-githubcalendar
- 参考

## 第九章 自动化部署

- 9.1 自动化部署解释
  - 9.2 GitHub Actions 简介
    - 9.2.1 什么是 CI/CD
    - 9.2.2 使用教程
    - 9.2.3 费用计算
  - 1.3 配置 Github SSH
  - 1.4 新建仓库
  - 1.5 配置私钥
  - 1.6 编写 GitHub Actions
  - 1.7 设置 .gitignore
  - 1.8 推送代码
  - 1.9 简化推送命令
    - 1.9.1 多条命令合一
    - 1.9.2 Git 命令设置简写 (别名)
    - 1.9.3 使用 Quicker 动作
    - 1.9.4 使用脚本
  - 1.10 注意
- 参考

## 第十章 网站 SEO

- 10.1 什么是 SEO
- 10.2 搜索引擎收录
  - 10.2.1 百度收录
  - 10.2.2 必应收录
  - 10.2.3 谷歌收录

## 10.3 SEO 优化

### 10.3.1 网站信息设置

### 10.3.2 优化文章链接

### 10.3.3 nofollow 标签

### 10.3.4 Front-matter

## 第十一章 提高网站加载速度

### 11.1 常用测速网站

### 11.2 压缩静态资源

#### 11.2.1 Gulp 压缩文件

#### 11.2.2 常用字字体子集化

#### 11.2.3 使用 woff 字体

#### 11.2.4 图片处理

### 11.3 建立镜像站

#### 11.3.1 Vercel

#### 11.3.2 Netlify

#### 11.3.3 总结

### 11.4 CDN 加速

#### 11.4.1 CDN 简单介绍

# 第一章 预备知识

## 1.1 为什么搭建博客

在你想要搭建一个自己的博客前，你需要问自己两个问题。

### 1.1.1 我真的有必要搭建博客吗？

已经 2022 年了，如果你只是想简单的在网上发表文章、想法，知乎、简书、掘金、[博客园](#) 等等有太多的平台供你选择。在这些平台上，你只需要写作就行。别的事情，你不需要关注。

如果是一个博客的话，你需要本地部署、远程部署，有时候部署还会失败，你需要上网搜索解决办法，很可能一时半会解决不了。你还要考虑是否需要购买域名、服务器。这对于一个刚刚接触搭建博客的人来说，很头大。此外，百度、谷歌等搜索引擎不会主动爬取你博客里的内容。你需要让百度、谷歌收录你的文章，这又需要时间、精力。如果你的博客文章不打算让别人看或者你是想主动分享给别人看，请当我没说。

### 1.1.2 我能做到持续输出文章吗？

博客搭建完成意味着一个新的开始。没有装修好的房子不是一个真正的家。同理，没有文章的博客，不算一个合格的网站。持续输出文章是必然的。我想大多数人搭建博客，是为了在网上有一个『好好说话』的地方。但是你要知道，在中国大陆地区使用中国大陆内的网络服务，你需要进行中国工业化信息部的 ICP [备案](#)，在部分地区公安备案也是必须的（如：上海市），你的个人网站的内容和言论会收到网络审查和监督。似乎『好好说话』也不太可能。但注意网络审查并非独我国一份，各国都存在。**不存在绝对的自由。自由与否，与个人主观意识有很大关系。**也不用太担心，只要你的文章不涉及反党、反社会、反人类、黄赌毒，不会有事。😊

一个刚刚建站没有多久的博客，访问量接近于无。很可能辛辛苦苦写的文章，没人看。长久下去，你还能坚持吗？或许你的本意不是让别人看你的文章，但看着访问量为 0 的博文，心里应该不好受吧。若非需要，大多数人都更希望看视频，而不是看文字。或许你可以考虑做自媒体？也挺好。

在我搭建博客的时候，我访问了不少个人博主的博客网站，有些博客已经不能访问，有些博客长时间没有更新。这就回到了本小节的标题：**我能做到持续输出文章吗？**如果不能保证能否坚持下去，却又想试试的话，用 Hexo 吧。这是成本最低的搭建博客的方式。我的博客访问量都是我点出来的 🤖。我都不知道我是否能坚持下去。

可以看看这篇文章：[个人网站存在的意义 - JoyQi](#) 

在考虑了以上两个问题后，你仍然决定搭建博客的话。那就往下面看吧。我会讲一些搭建博客需要了解的知识。

## 1.2 必看视频 ☆

[【合集完结】想上线网站？通俗易懂的网站上线发布教程 | 个人网站如何托管建站 | 服务器 IP DNS CDN 备案 工作原理 | 腾讯云开发静态网站托管 - bilibili](#) 

这个合集视频解释了搭建一个博客网站，遇到的各种「名词」和解决方法。

## 1.3 代码相关知识

### 1.3.1 Git 和 GitHub

Git 是一个命令行工具，一个分布式版本控制系统，可以管理和跟踪你的代码历史版本。而 GitHub 是一个代码托管云服务网站，帮助开发者存储和管理其项目源代码，且能够追踪、记录并控制用户对其代码的修改。

- **Git 教程**

- [猴子都能懂的Git入门](#) 

内容全面，图文并茂，涵盖初级、中级、高级。

- [廖雪峰Git教程](#)

浅显易懂，适合初学者。

- [Git官网教程](#)

官方教程，可以当「新华字典」使用。

- [Learn Git Branching](#)

一个在线的 Git 学习网站，通过可视化交互的方式学习 Git 相关操作。适合有一定基础的同学。

- **GitHub 教程**

- [GitHub 官方 10 分钟入门教程](#)

GitHub 官方出品，教学内容包含：新建仓库、新建分支、修改提交（Commit）以及请求拉取（Pull Request）。原文为英文版，中文翻译可以点击 [此处](#)。

- [GitHub 秘籍](#)

主要包含 `git` 和 `github` 常用技巧，适合有一定基础的同学想要进一步掌握一些高级技巧。

- **其他教程**

- [Git及Github的基本使用 | Eulaの小破站](#)

参考网站： [GitHub中文社区 \(githubs.cn\)](#)

## 1.3.2 前端三件套

如果你是一个对代码零基础的人，建议学习 HTML、CSS、JavaScript，这对你搭建博客有很大帮助。有了一定前端基础后，修改博客主题更加方便，主题文档也能看懂。

- **学习资料**

- [Web 开发技术 | MDN \(mozilla.org\)](#)
- [黑马程序员pink老师前端入门教程, h5\(html5\)+css3+移动端前端](#)
- [JavaScript基础语法-dom-bom-js-es6新语法-jQuery-数据可视化echarts  
黑马pink老师前端入门基础视频教程\(500多集\)持续](#)

## 1.3.2 Markdown 语法

什么是 Markdown 语法，以下是维基百科对 Markdown 介绍的截取。

详情请看：[Markdown - 维基百科，自由的百科全书 \(wikipedia.org\)](#)

**Markdown** 是一种 [轻量级标记语言](#)，创始人为 [约翰·格鲁伯](#)。它允许人们使用易读易写的纯文本格式编写文档，然后转换成有效的 [XHTML](#)（或者 [HTML](#)）文档。这种语言吸收了很多在电子邮件中已有的纯文本标记的特性。

由于 Markdown 的轻量化、易读易写特性，并且对于图片，图表、数学式都有支持，目前许多网站都广泛使用 Markdown 来撰写帮助文档或是用于 [论坛](#) 上发表消息。如 [GitHub](#)、[Reddit](#)、[Diaspora](#)、[Stack Exchange](#)、[OpenStreetMap](#)、[SourceForge](#)、[简书](#) 等，甚至还能被用来撰写 [电子书](#)。

我们博客的文章基本采用 **MD** 格式。下面介绍 Markdown 基本语法。由于我不想污染整篇文章的排版，所以我只给出相应的语法，不给出示例。且 Markdown 文章渲染出来的样式与使用的 CSS 样式文件有很大关系。给出示例，容易引起误导。

### 1、标题

```
# 一级标题
## 二级标题
### 三级标题
#### 四级标题
##### 五级标题
##### 六级标题
```

标题从一级到六级，N 个 # 对应 N 级标题。

## 2、文字样式

```
2.1 加粗      **镜湖月**
2.2 倾斜      *镜湖月*
2.3 删除线    我被删除啦
```

## 3、块引用

```
3.1 基本用法
> 大家好，我是镜湖月。
> 大家好，我是镜湖月。

3.2 嵌套
> 大家好，我是父级块引用。
>> 大家好，我是子级块引用。

不推荐使用。
```

## 4、段落

在 Markdown 语法中，`enter`（硬换行）代表另起一段，`shift+enter`（软换行）代表在本段另起一行。如下：

这是一个自然段。

这也是一个自然。（硬换行）

这是软换行效果。

## 5、列表

```
Markdown
```

5.1 无序列表

- 哈哈
- 哈哈

5.2 有序列表

1. 哈哈
2. 哈哈

## 6、表格

```
Markdown
```

----	----	----	

----: 右对齐  
:---- 左对齐  
:----: 居中对齐

## 7、代码

## 7.1 行内代码

```
`xxx`
```

eg: `jinghu-moon`

## 7.2 代码块

```
```代码语言
```

```
xxx
```

```
xxx
```

```
```
```

```
//示例
```

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    printf("hello world\n");
```

```
    printf("你好, 镜湖月! \n");
```

```
    return 0;
```

```
}
```

## 8、超链接

```
[标题](网址)
```

示例:

```
[jinghu-moon 的个人博客](https://jinghumoon.top/)
```

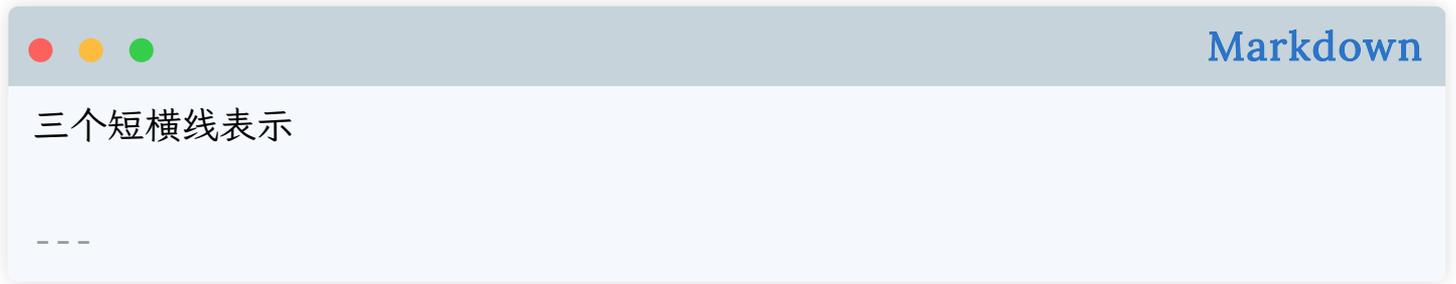
## 9、图片

```
![标题](图片网址)
```

示例:

```
![博客头像](https://jinghumoon.top/img/avatar.png)
```

## 10、分隔线



## 11、转义字符

可使用反斜杠 `\` 来转义如下字符：

| 字符              | 中文名称     |
|-----------------|----------|
| <code>\</code>  | 反斜杠      |
| <code>`</code>  | 反引号      |
| <code>*</code>  | 星号       |
| <code>_</code>  | 下划线      |
| <code>{}</code> | 花括号      |
| <code>[]</code> | 方括号      |
| <code>()</code> | 圆括号      |
| <code>#</code>  | 井号       |
| <code>+</code>  | 加号       |
| <code>-</code>  | 短横线（减号）  |
| <code>.</code>  | 点        |
| <code>!</code>  | 感叹号      |
| <code> </code>  | 竖线（管道符号） |

## 12、待办事项

- [x] 待办事项一
- [ ] 待办事项二
- [ ] 待办事项三

### 13、脚注

你好<sup>[^脚注1]</sup>.

<sup>[^脚注1]</sup>: hello!

以上为 Markdown 最基本使用方法。总的来说，很简单。

#### 参考资料

- [Markdown 使用指南 - 基础语法 - 链滴 \(ld246.com\)](#) 
- [Markdown 使用指南 - 扩展语法 - 链滴 \(ld246.com\)](#) 
- [Markdown 使用指南 - 语法速查手册 - 链滴 \(ld246.com\)](#) 

详细介绍我会在 Typora 使用教程中出现（还没写）。

[Typora 中文站](#) 

## 1.4 常见博客框架

博客框架大致分为两种：动态博客与静态博客。两者最大的区别就是动态博客自带后端，可以随时登录后台写文章、发文章，很方便。而静态博客就必须在本地写作，再部署到远端上。

动态博客：WordPress、typecho、Halo；静态博客：Hexo、Hugo。

[搭博客到底选什么框架？Hexo还是Hugo？Typecho还是Wordpress？ - bilibili](#) ，可以看下。

## 1.4.1 动态博客

### 1、WordPress

- **介绍**：由 PHP 语言编写，43% 的网页在 WordPress 上构建。可以用来搭建个人博客、电商网站、论坛等等。
- **特点**：功能强大，插件丰富，主题多，你需要的功能，它都有。当然由于功能过于繁多，与另外两个动态博客框架相比，显得有些臃肿。需要用户具备一定的网站优化能力。需要购买服务器。

### 2、Typecho

- **介绍**：由 PHP 语言编写，极其轻量，仅 477KB。由国人开发。文档详细。
- **特点**：超低的 CPU 和内存使用率。主题多，插件不多，但满足使用需求。如果想要搭建动态博客，推荐使用。对于服务器的要求很低。
- **搭建参考**：[腾讯云轻量应用服务器搭建 Typecho 网站 !\[\]\(4b42b6ee6cea697ec73d43bbd0b91e24\_img.jpg\)](#)

### 3、Halo

- **介绍**：Halo 是一款现代化的个人独立博客系统，由 Java 语言编写。国人开发。文档十分详细。
- **特点**：初始界面看，操作简单，适合小白。对于服务器的要求高一点。目前不支持市面上的云虚拟主机，需要使用云服务器或者 VPS。

**总结**：学生用户不推荐搭建动态博客。你无法确定你搭建博客是不是一时兴起。购买服务器就有点浪费了。如果手里就有闲置的服务器汇或者不差钱，直接搭建动态博客，自带后端，用起来方便得多。

## 1.4.2 静态博客

### 1、Hexo

- **介绍**：Hexo 是一个快速、简洁且高效的博客框架。Hexo 使用 [Markdown !\[\]\(01d5914006c620a5d6bd40fdd52ce174\_img.jpg\)](#)（或其他渲染引擎）解析文章，在几秒内，即可利用靓丽的主题生成静态网

页。

- **特点**：主题超级多，插件多。比较热门的几个主题都有中文文档。配置起来很方便。网上的搭建教程也很多。无后端，只能在本地写文章，再部署到 GitHub 上。无评论系统。

## 2、Hugo

- **介绍**：由 Go 语言实现的静态网站生成器。简单、易用、高效、易扩展、快速部署。
- **特点**：可以快速渲染几百篇 Markdown 文章，而 Hexo 则可能会花费比较长时间。部署比 Hexo 麻烦一点。无后端，只能在本地写文章，再部署到 GitHub 上。无评论系统。使用人少，遇到问题，不好解决。

**总结**：对于初次接触搭建博客的同学来说，推荐使用 Hexo 搭建博客。简单易上手。等熟悉了，再考虑更换博客框架。

# 1.5 搜索与提问

## 1.5.1 提问

首先，我们要明白一个问题，我们在网上提出的问题，并不一定有人会回答。这时，你可能会生气，但这不能解决你的问题。为了提高问题被回答的几率，我建议你看看这篇文章：[《提问的智慧》精读注解版 - 链滴 \(ld246.com\) !\[\]\(b0ba827d2a487ec829cbd8e604d56e0e\_img.jpg\)](#)，再去提问。如果觉得这篇文章太长的话，看这篇文章：[《提问的智慧》思维导图 Xmind 版的 分享一下 - 链滴 \(ld246.com\) !\[\]\(a992687274647318f0230a378994f08c\_img.jpg\)](#)

## 1.5.2 搜索

维基百科镜像，内容有审核：[万维百科 \(wanweibaike.net\) !\[\]\(c7143b06b3915be2311cf128bb2424aa\_img.jpg\)](#)

现在国内主要的搜索引擎有百度、搜狗、必应等，最近出现了一个新的搜索引擎：[F 搜 \(fsou.cc\)](http://fsou.cc)，感觉还行，可以使用。百度目前是中国互联网上最大的中文搜索引擎，但它的表现却让人很不满意。用百度搜索，排在前面的很有可能是广告、盗版网站，一些转载了不知道多少次的文章。搜索效率较低。在国内，我推荐使用必应和百度一起使用。当你搜索软件下载时，最好使用必应。

个人感觉，谷歌对个人网站的收录优于百度。我搭建博客时，参照了不少谷歌上的教程。当然在中国大陆范围，谷歌是不能访问的。原因我们不需要深究，要相信我们的国家。在大陆，也可以使用谷歌搜索，这是谷歌镜像：[Google \(luciaz.me\)](http://luciaz.me)，第一次访问需要回答问题，答案如下：

## 需要验证您是否来自浙江大学

这样的验证只会出现一次，通过后您会被加入白名单，之后相同IP的访问不会再需要验证。  
提示：由于手机和宽带IP经常会发生改变，您可能会多次看到这一页面。

只需要回答出以下任意一个问题即可

本站仅允许浙江大学(含城院/宁波理工)师生和他们的小伙伴访问。  
如果您(或告诉您这个网站的小伙伴)也来自浙江大学，请您填写以下问题  
浙大内部欢迎帮忙宣传(特别是老硕博们·ω·)  
对外校允许小范围地在小伙伴间传播  
没有任何使用时间/流量的限制  
This site only allows people from Zhejiang University to access, please answer the following question(s).

### [For 逛内网论坛的小伙伴们]

请问浙江大学内网最大的论坛中，每天发帖量最大的版块是哪个？  
(全称，四个汉字)

(感谢陈豪\*同学提供本问题)

### [For 浙大本部本科生]

请根据以下指引找到并输入三个汉字以验证你是浙大学生(ps我跟求是潮一点关系都没):

Android: 最新版求是潮手机app左边菜单栏中, "其他"一栏的第一个项目是什么(点左上角的选项按钮调出左边栏)(三个汉字,第一个字是"水")

iOS:最新版求是潮手机app中, 从左往右划动主界面调出左边栏,其中有一栏是三个汉字,图标是一个"水"字,如果你没有自定义它的顺序的话是倒数第三个,它是什么  
windows Phone或者老师或者其余无法找到的: 请问问小伙伴

### [For 浙大城院]

请问浙江大学内网城市学院端统一密码是什么(即城院内网宽带连接 (PPPOE) 拨号的密码)?

提示: 四个字母@三个数字

(感谢\*\*伦同学提供本问题)

如果不知道, 请问问小伙伴

另附两篇文章：

[iMyShare - Google国内镜像网站](http://imyshare.com)；

[掌握这些相见恨晚的搜索小技巧，从此找资源不求人! \(imysshare.com\)](http://imyshare.com)。

但这是不够的。有时候学习、工作、娱乐需要访问一些国外的网站，却无法访问。就比如我常访问的 GitHub，时而打开时而打不开。这时候就需要『科学上网』。这是参考文章：[分类: 越狱 | 小帅同学的博客 \(xs2021.github.io\)](http://xs2021.github.io)，文章访问密码：[9527](http://9527)。另外 GitHub 上也有相关内容，请自行搜索。

**注意：**浏览外网时，要时刻保持清醒，不要有『二极管思维』，在不清楚事情全貌时，不轻易发表言论。要相信我们的党与国家。

另外附两篇文章：

[Halo 社区运行的基本准则 - Halo 社区](#) 

[互联网信息服务管理办法 - 2011.1 增刊 - 中国政府网 \(www.gov.cn\)](#) 

第一篇是 Halo 社区对社员在社区发言的要求。我觉得也挺适合我们这些在网上冲浪的人，就贴了出来。第二篇就不用多说了吧。

## 1.6 文章排版

---

以下为我个人意见。如果你觉得没有必要，请跳过。

- 1、中英文之间需要增加空格。
- 2、中文与数字之间需要增加空格。
- 3、英语专有词汇，按官方格式书写。例如：GitHub。
- 4、数字与单位之间需要增加空格。例如：20 TB。
- 5、行内代码与前后文字（包括符号）各空一格。
- 6、链接前后增加空格。

参考文章：[中文文案排版指北](#) 

推荐看下 [ruanyf/document-style-guide: 中文技术文档的写作规范](#) 。我认为很有用。如果看不了，可以看这篇：[中文技术文档写作规范 - 镜湖月的小世界 \(jinghumoon.top\)](#) 。我整理了下文章排版。

## 1.7 版权意识

---

当你开始写博文的时候，难免会使用图片，引用他人的文章。此时你需要注意图片是否为无版权图片。引用文章采用的是什么许可协议。我认为，作为博主需要具备一定的版权意识。下面我会简单介绍大多数博客文章采用的许可协议，同时分享一些无版权图片网站。

## 1.7.1 许可协议

**参考链接：** [关于许可协议 - Creative Commons](#)、[知识共享许可协议\\_百度百科 \(baidu.com\)](#)、[知识共享许可协议 - 维基百科，自由的百科全书 \(wikipedia.org\)](#)。

一般来说，你会在博客文章最后面看到这样一句话：

**版权声明：** 本博客所有文章除特别声明外，均采用 [CC BY-SA 4.0 协议](#) / [CC BY-SA 4.0 协议](#) / [CC BY 4.0 协议](#)，转载请注明出处！

其中的字母代表什么意思，下面来解释。

### 1、CC 协议

全称为 Creative Commons license，译为**知识共享许可协议**，简称 CC 协议。是一种允许他人分发作品的公共版权许可。2002 年 12 月 16 日，美国非盈利性组织知识共享（Creative Commons）首次发布 CC 协议。

CC 4.0 版本于 2013 年 11 月 25 日发布。此后，CC 4.0 被鼓励在全球范围内适用。除了放弃版权将作品完全公布到公共领域（即 CC0 协议）之外，CC 4.0 版本中一共有 **6** 种常用的版权规定组合。

CC 协议主要用于书籍、戏剧、电影、音乐、文章、照片、博客、网站。

### 2、BY

- 英语：Attribution（归属），简称 BY。译为**署名**。
- 使用条件：您可以自由复制、散布、展示及演出本作品；您必须按照作者或授权人所指定的方式，保留其姓名标示。

### 3、SA

- 英语：ShareAlike（一模一样），简称 SA。译为**相同方式共享**。
- 使用条件：您可以自由复制、散布、展示及演出本作品；如果您改变、转变或改作本作品，仅在遵守与本著作相同的授权条款下，您才能散布由本作品产生的衍生作品。

### 4、NC

- 英语：Noncommercial（非商业性的），简称 NC。译为**非商业性使用**。
- 使用条件：您可以自由复制、散布、展示及演出本作品；您不得为商业目的而使用本作品。

### 5、ND

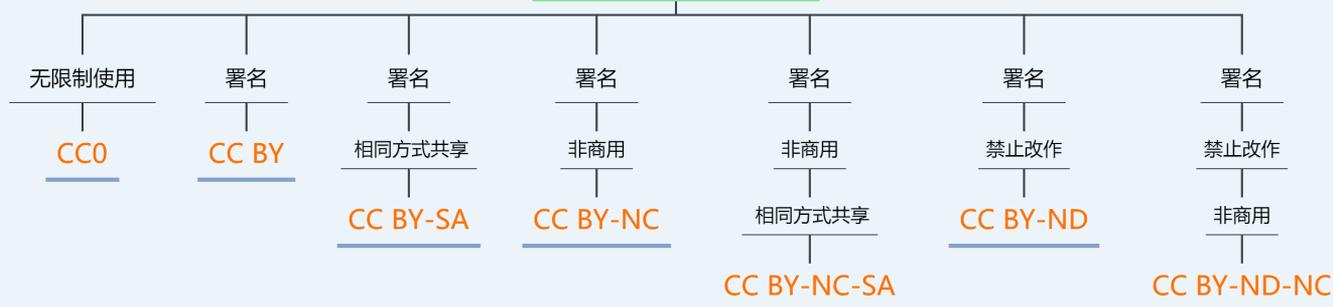
- 英语：No Derivative Works（没有衍生作品），简称 ND。译为**禁止改作 / 禁止演绎**
- 使用条件：您可以自由复制、散布、展示及演出本作品；您不得改变、转变或改作本作品。

以上四种条件组合共有 16 种可能。4 种组合是同时包括「ND」和「SA」的条件，相互排斥；还有 1 种没有包括任何条件（CC0 协议），因此有 5 个组合无效。11 种有效组合中，因为 98% 的特许人要求署名，5 种缺乏「BY」的条件不再使用。所以只有 **6** 种组合可以使用。

**问题：**如何选择适合自己的许可协议？

看图！

## CC 许可协议



附上 6 种 CC 许可协议的链接。

- [Creative Commons – 署名 4.0 国际 – CC BY 4.0](#)
- [Creative Commons – 署名-相同方式共享 4.0 国际 – CC BY-SA 4.0](#)
- [Creative Commons – 署名-非商业性使用 4.0 国际 – CC BY-NC 4.0](#)
- [Creative Commons – 署名-非商业性使用-相同方式共享 4.0 国际 – CC BY-NC-SA 4.0](#)
- [Creative Commons – 署名-禁止演绎 4.0 国际 – CC BY-ND 4.0](#)
- [Creative Commons – 署名-非商业性使用-禁止演绎 4.0 国际 – CC BY-NC-ND 4.0](#)

我的选择： **BY-NC-SA 4.0**。

## 1.7.2 无版权图片

### 1、[Unsplash](#)

互联网上最大的免费分享图片的平台之一。强推！上面的图片都很好看。注意这是一个英文网站。搜索只能用英语。

国内镜像：[Beautiful Free Images & Pictures | Unsplash \(dogedoge.com\)](#)

### 2、[pixabay](#)

免费正版高清图片素材库。Pixabay 拥有超过 2.5 百万张优质图片和视频素材，让你轻松应对各种设计场景。支持中文搜索。

### 3、[咸柠檬壁纸\(ZL Wallpaper\)](#)

图片源自 [Wallpaper Abyss - 高清壁纸, 桌面背景 \(alphacoders.com\)](#) ，支持中文搜索。图片质量，我觉得还行。

上面三个是我主用的图库。如果觉得不够，看下面的链接。

### [5 个免版权高清视频素材下载网站（一） - Topbook](#)

### [5 个免版权高清视频素材下载网站（二） - Topbook](#)

### [10 个免版权高清图片素材下载网站 - Topbook](#)

## 1.7.3 图标库

### 1、[iconfont-阿里巴巴矢量图标库](#)

目前国内最大的图标库，基本上你需要的图标，它都有，而且还有不同的样式。

注意：该素材仅供广大用户交流学习使用，未经 iconfont 或其关联公司书面授权许可，不得用于任何商业用途。

### 2、[ByteDance IconPark](#)

2400+ 基础图标，29 种图标分类。IconPark 图标库是一个通过技术驱动矢量图标样式的开源图标库。可以对图标进行较大的编辑。

## 1.7.4 可商用字体

### 免费商用中文字体下载网站

[猫啃网，最新最全的可免费商用中文字体下载网站！猫啃~ \(maoken.com\)](#) 

### 1、霞鹜文楷

一款楷体字体，很不错。[lxgw/LxgwWenKai: 一款基于 FONTWORKS 出品字体 Klee One 改造的开源中文字体](#) ，猫啃网下载链接：[霞鹜文楷 - 猫啃网](#) 

。

## 2、思源宋体、思源黑体

[思源宋体 \(github.com\)](#) 、[思源黑体 \(github.com\)](#) 

GitHub 可能会下载失败，推荐这个网站：[字体天下 - 提供字体免费下载和在线预览](#) 。

# 第二章 本地部署

## 2.1 Node.js 下载

官网：[Node.js \(Nodejs.org\)](https://nodejs.org) 。如下：



如果官网打不开，备用链接：[下载 | Node.js 中文网 \(Nodejs.cn\)](https://nodejs.cn) ，如下：

## 下载

最新版本: 16.13.1

|  |  |   |
|--|--|---|
| <b>v16.13.1</b><br>最新版本  | <b>v14.18.2</b><br>旧版本                                   |   |
| <br><b>Windows 安装包</b><br><small>node-v16.13.1-x64.msi</small> | <br><b>macOS 安装包</b><br><small>node-v16.13.1.pkg</small> | <br><b>源代码</b><br><small>node-v16.13.1.tar.gz</small> |

- Windows 安装包 (.msi)
- Windows 二进制文件 (.zip)
- macOS 安装包 (.pkg)
- macOS 二进制文件 (.tar.gz)
- Linux 二进制文件 (x64)
- Linux 二进制文件 (ARM)
- Docker 镜像
- 全部安装包

|              |       |
|--------------|-------|
| 32 位         | 64 位  |
| 32 位         | 64 位  |
| 64 位 / ARM64 |       |
| 64 位         | ARM64 |
| 64 位         |       |
| ARMv7        | ARMv8 |
| 官方镜像         |       |
| 全部安装包        |       |

下载这个

Node.js 是什么，与本教程内容无关，不解释（其实我也不知道 🙄）。如果你想了解 Node.js，建议阅读官方文档或者看看这个回答：[Node.js是用来做什么的? - 知乎 \(zhihu.com\)](#)。

## 2.2 Node.js 安装流程

一路 **next** 就可以了。安装目录由你决定。

参考教程：[Node.js安装及环境配置之Windows篇 - 周瑜周 - 博客园 \(cnblogs.com\)](#)

## 2.3 Git 下载

下载地址：[Git - Download for Windows\(git-scm.com\)](#)，如下：

About

Documentation

Downloads

- GUI Clients
- Logos

Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read [online for free](#). Dead tree versions are available on [Amazon.com](#).

## Download for Windows

[Click here to download](#) the latest (**2.34.1**) **64-bit** version of **Git for Windows**. This is the most recent [maintained build](#). It was released **about 1 month ago**, on 2021-11-25.

### Other Git for Windows downloads

Standalone Installer [安装版本](#)

[32-bit Git for Windows Setup](#).

[64-bit Git for Windows Setup](#). [→ 下载它](#)

Portable ("thumbdrive edition") [绿色版本](#)

[32-bit Git for Windows Portable](#).

[64-bit Git for Windows Portable](#).

### Using winget tool

Install [winget tool](#) if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

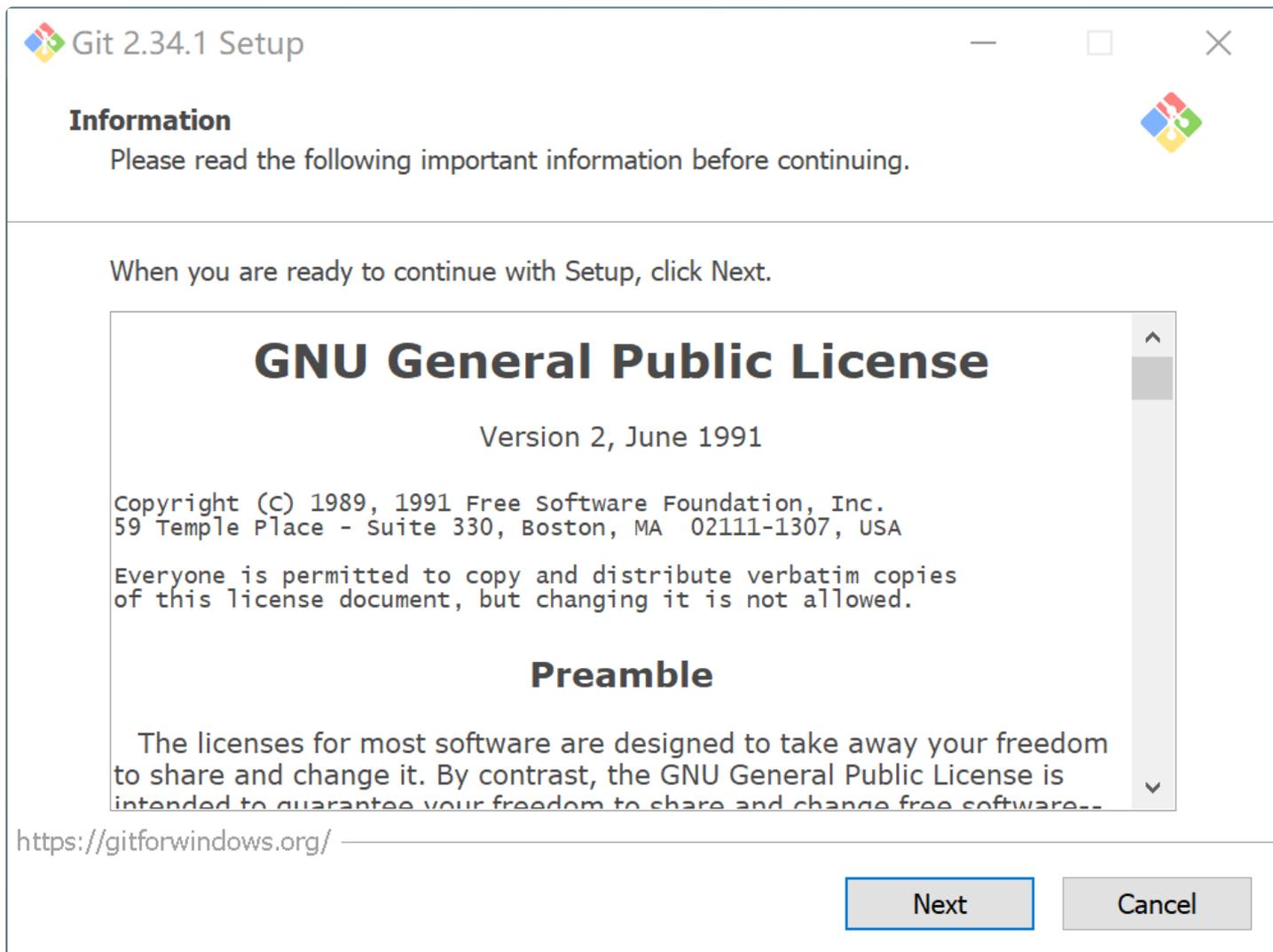
The current source code release is version **2.34.1**. If you want the newer version, you can build it from [the source code](#).

如果官网打不开，国内镜像网站：[git-for-windows Mirror \(taobao.org\)](#) [↗](#)。进入网站后，划到最下面，点击最新版本：[v2.34.1.windows.1/](#)，进入，选择下载 [Git-2.34.1-64-bit.exe](#)。

## 2.4 Git 安装流程

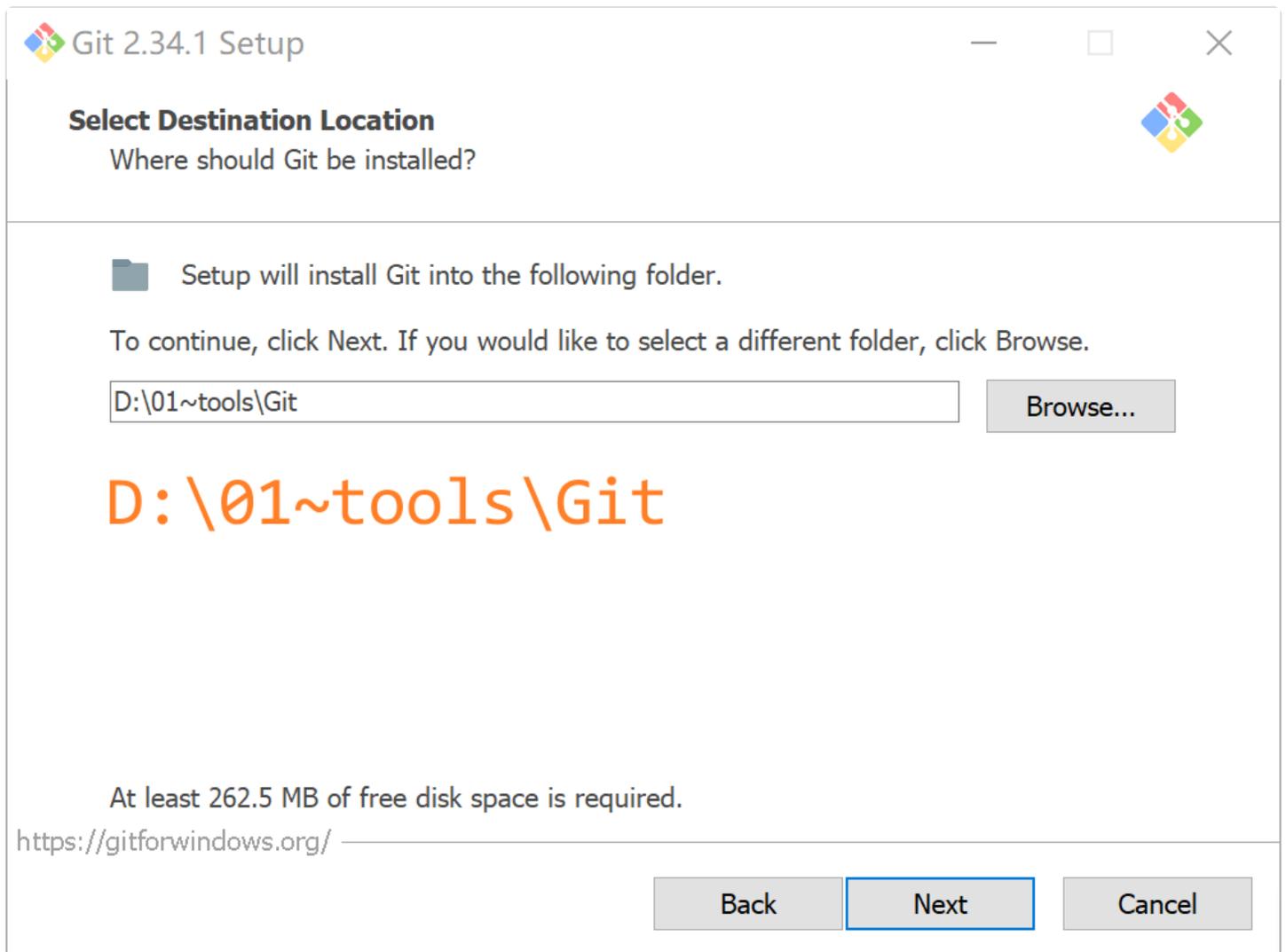
### 2.4.1 双击运行程序

双击运行 [Git-2.34.1-64-bit.exe](#)，进入 Git 的安装界面。



## 2.4.2 选择安装目录

点击 **next** ，选择安装目录。默认安装在 **C:\Program Files\Git** 。建议安装在 **D** 盘。如下：



## 2.4.3 选择组件

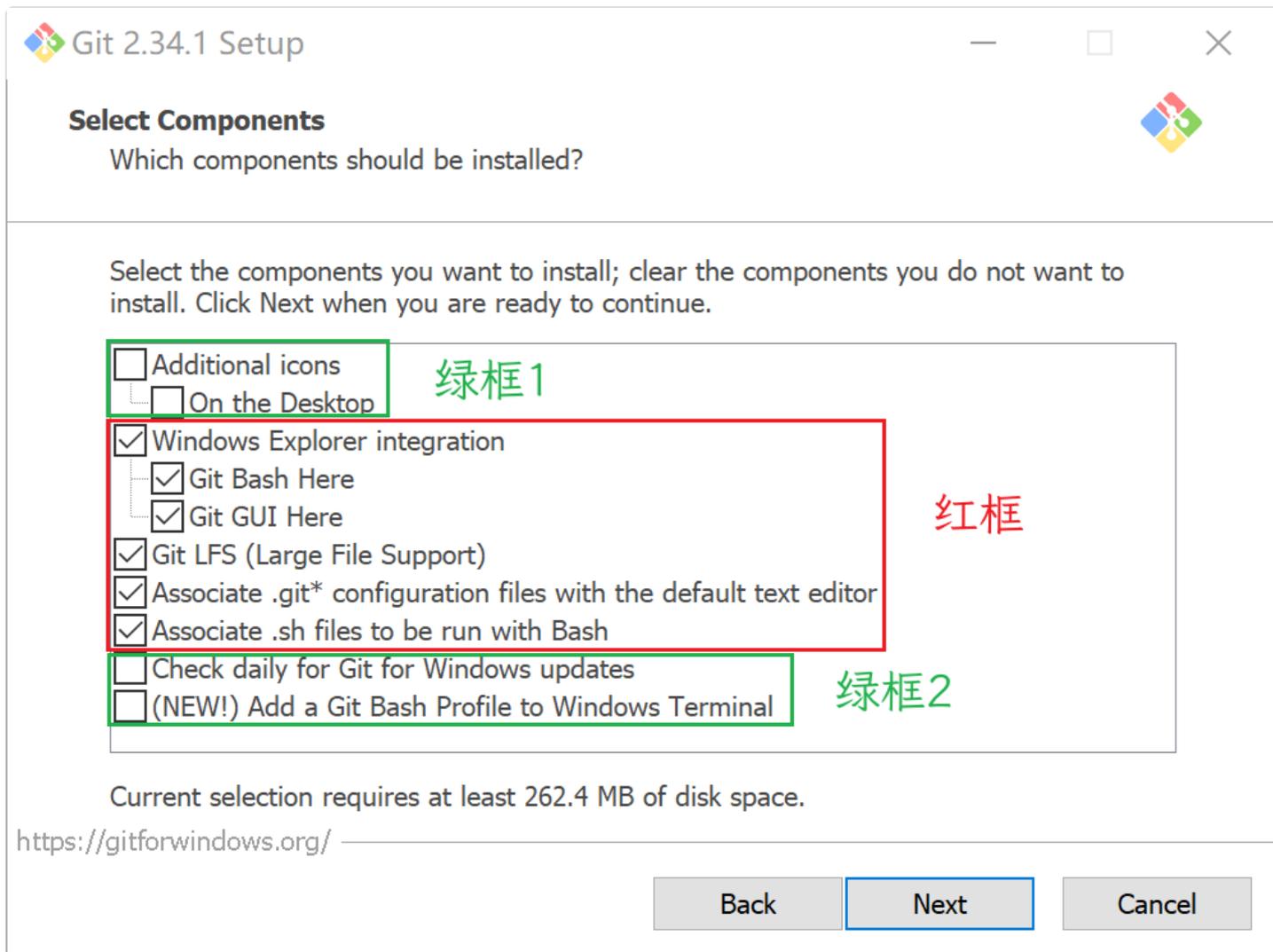
点击 **next** ，选择组件（Select Components）。

**红框**：默认选项，不要动。

**绿框1**：是否在桌面添加快捷方式。

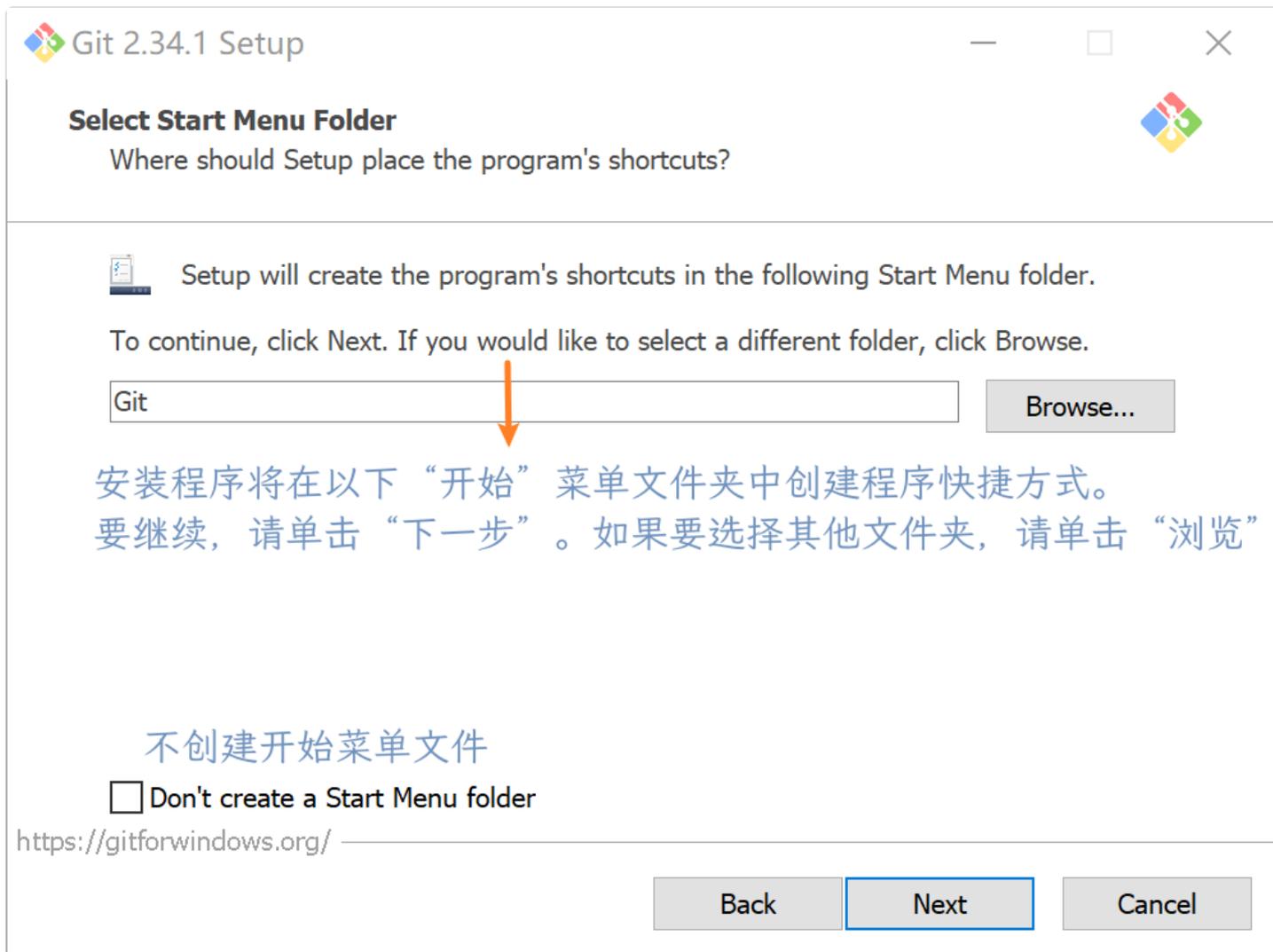
**绿框2**：是否每天检查 Git 有无 Windows 更新和将 Git Bash 配置文件添加到 Windows 终端。这些根据自己需要选择。

**我的选择**：保持默认就好。



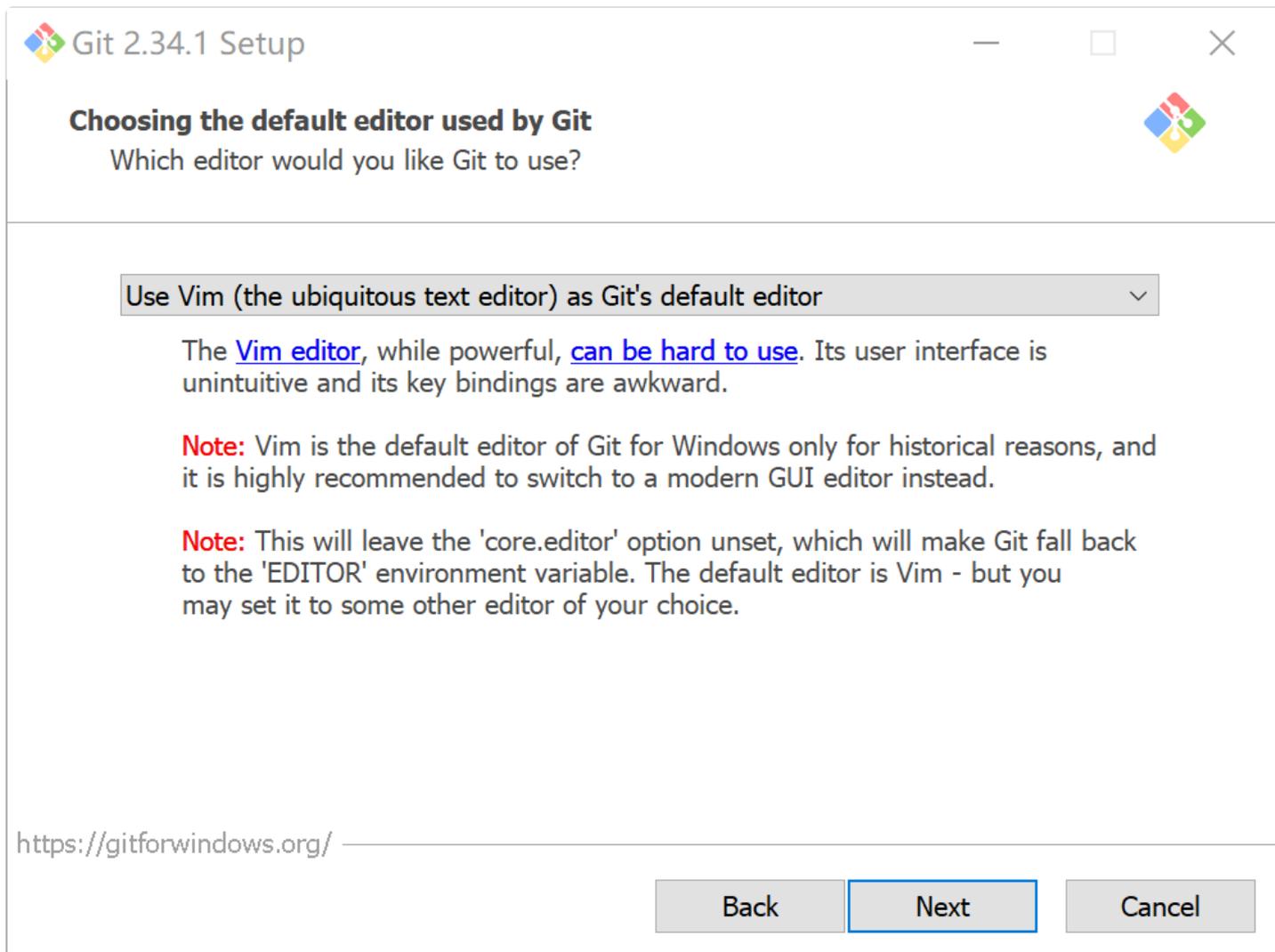
## 2.4.4 选择开始菜单文件夹

点击 **next** ，选择开始菜单文件夹（Select Start Menu Folder），按需选择。



## 2.4.5 选择编辑器

默认即可。点击 **next** ，进入下一步。[Vim \(Unix及类Unix系统文本编辑器\)](#) — [百度百科 \(baidu.com\)](#)  [vim教程网 \(vimjc.com\)](#) 



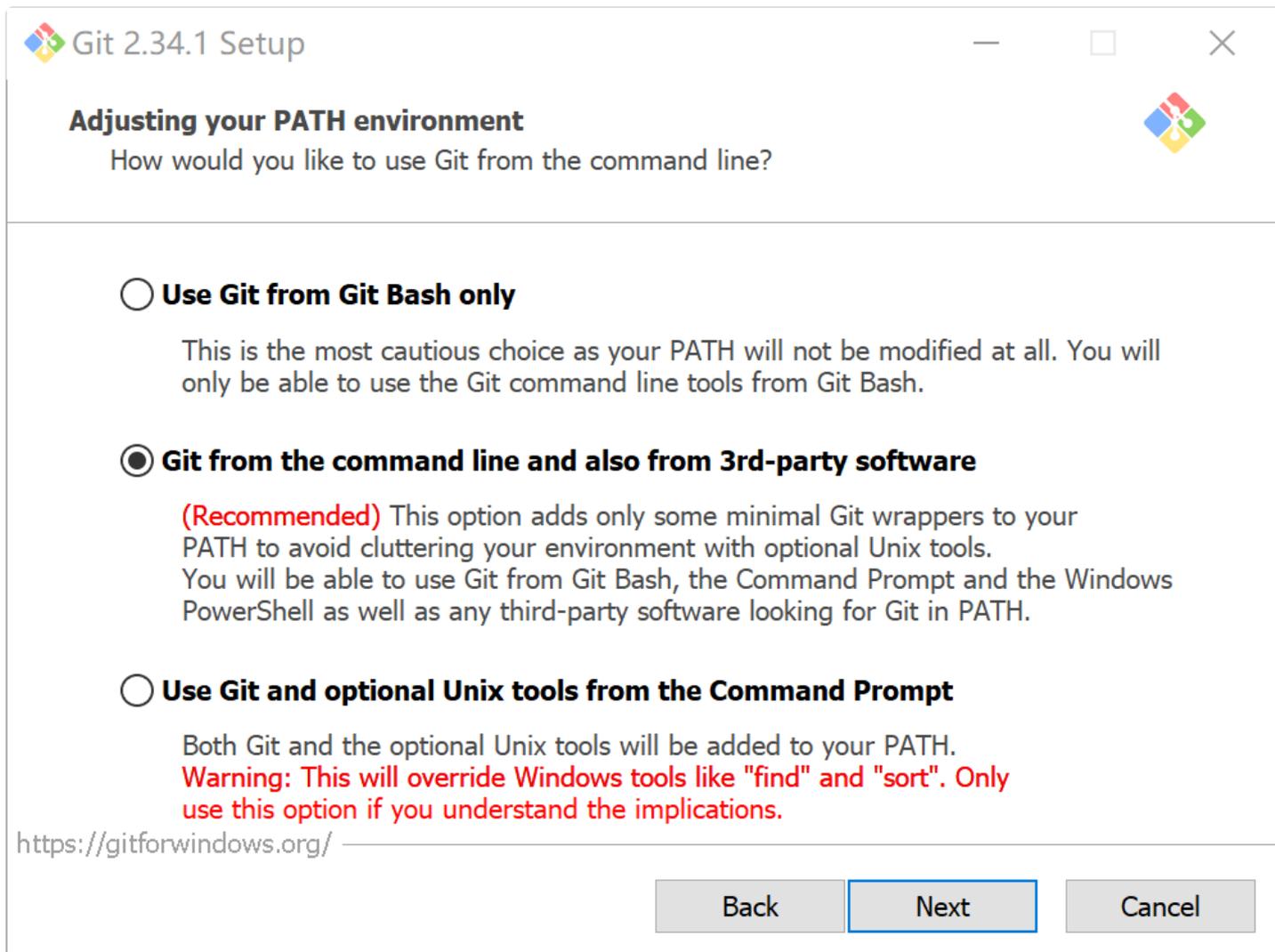
## 2.4.6 选择默认分支名

默认就好，当然你也可以自定义分支为 `main`。点击 `next`，进入下一步。

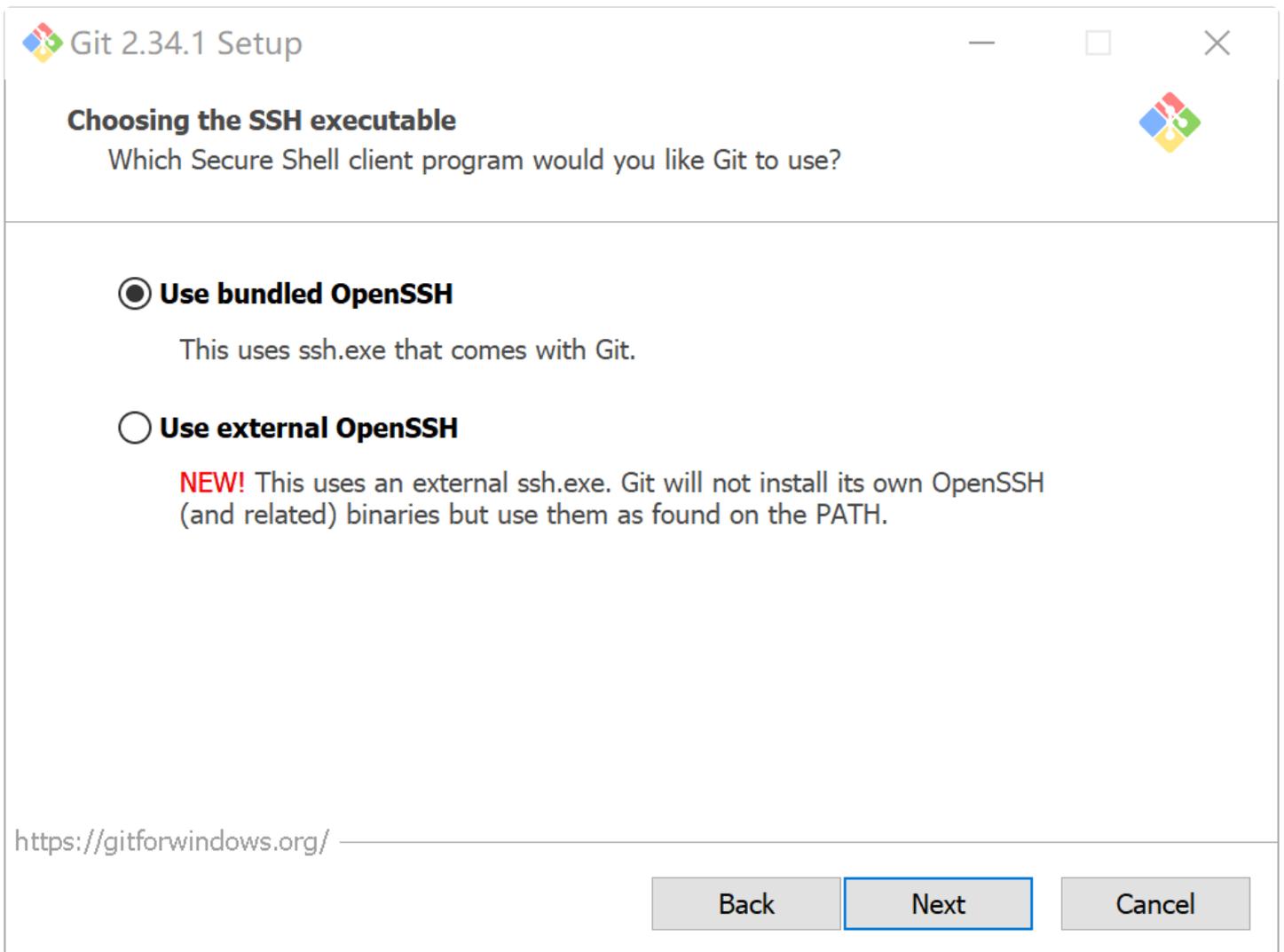


## 2.4.7 环境变量、SSH、HTTPS 设置

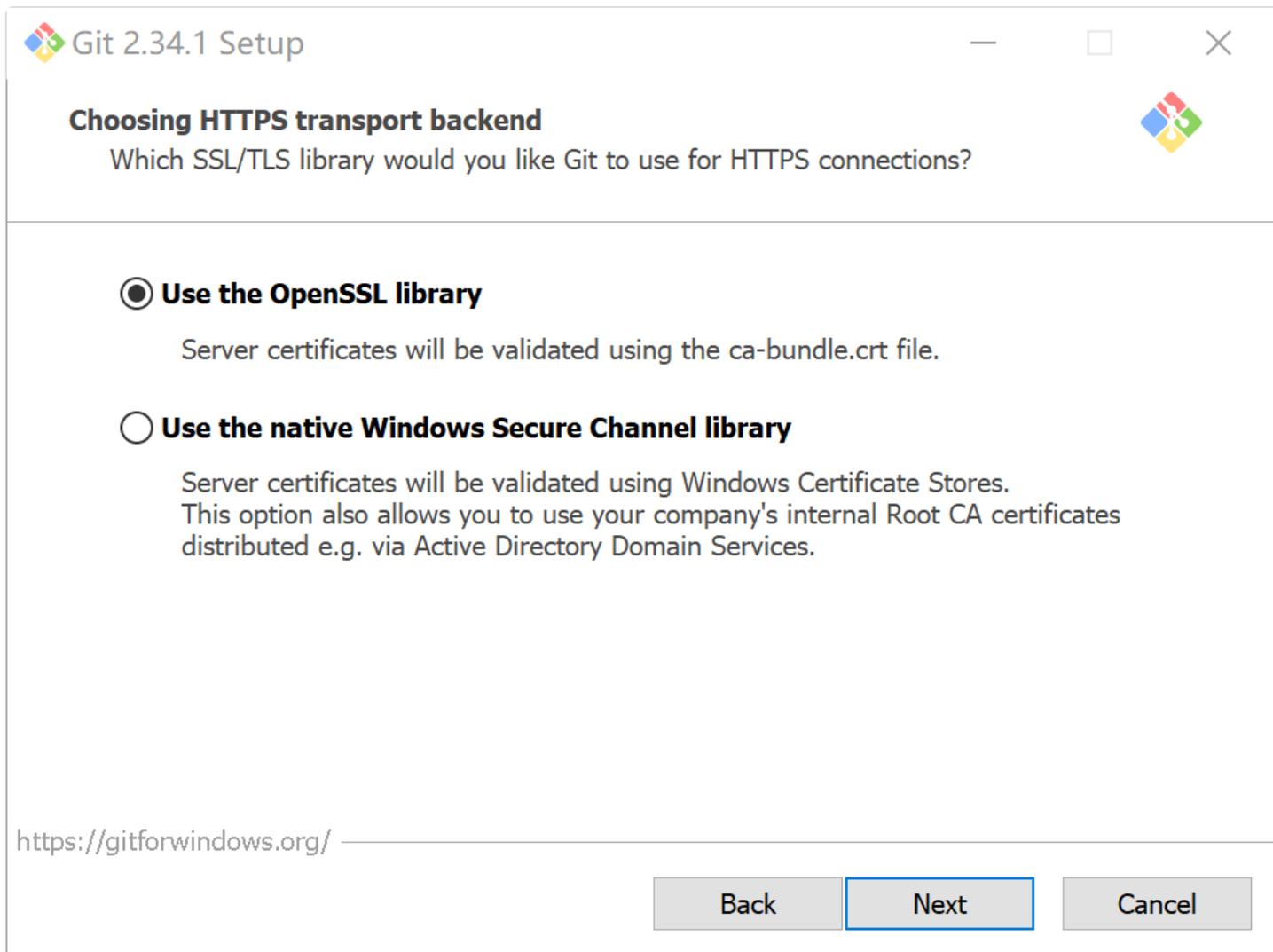
01. 调整 PATH 环境，默认即可。点击 **next** ，进入下一步。



02. 选择 SSH 可执行文件（Choosing the SSH executable），默认即可。点击 `next`，进入下一步。

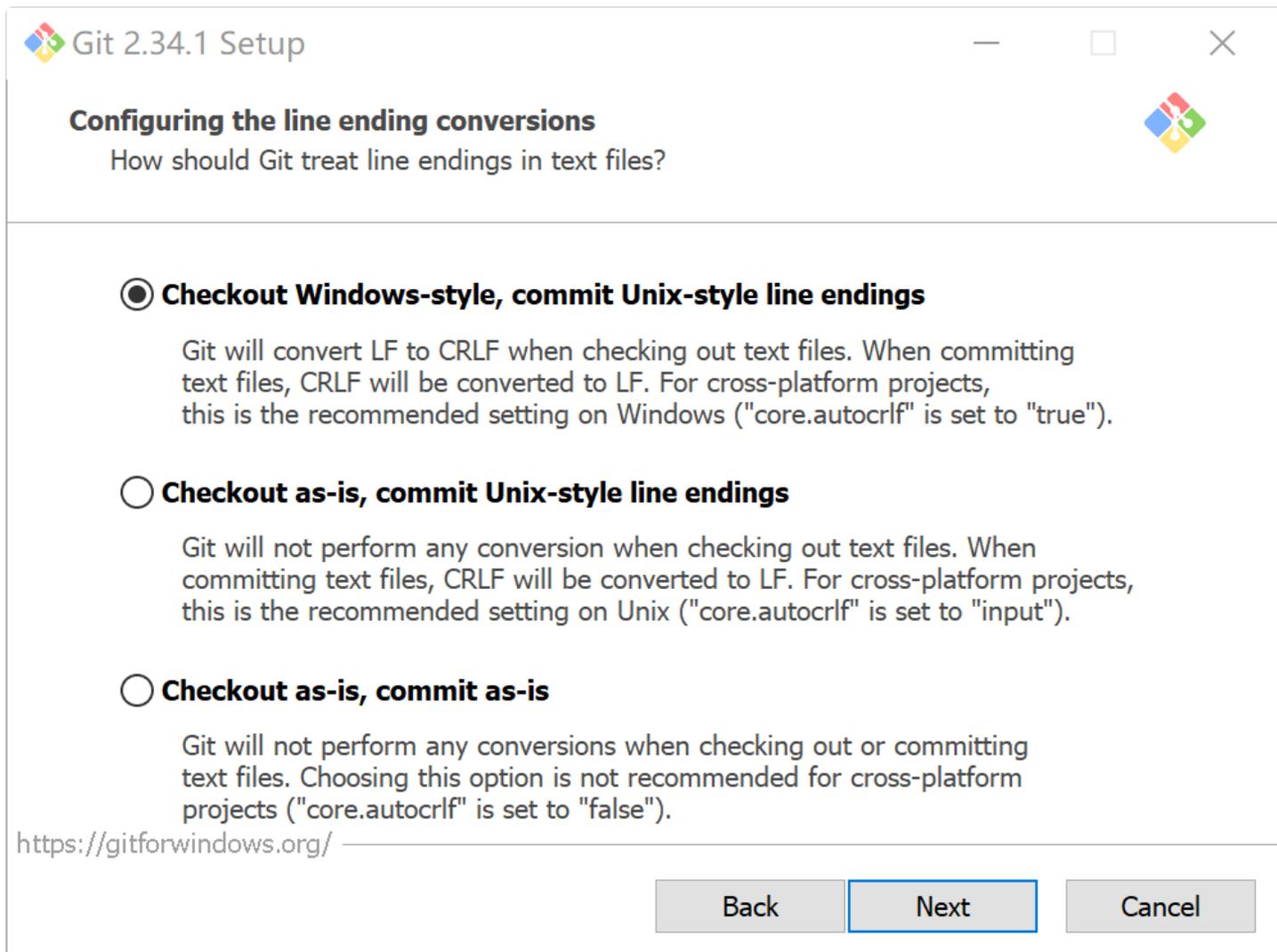


03. 选择 HTTPS 后端传输 (Choosing HTTPS transport backend) , 默认即可。点击 **next** , 进入下一步。

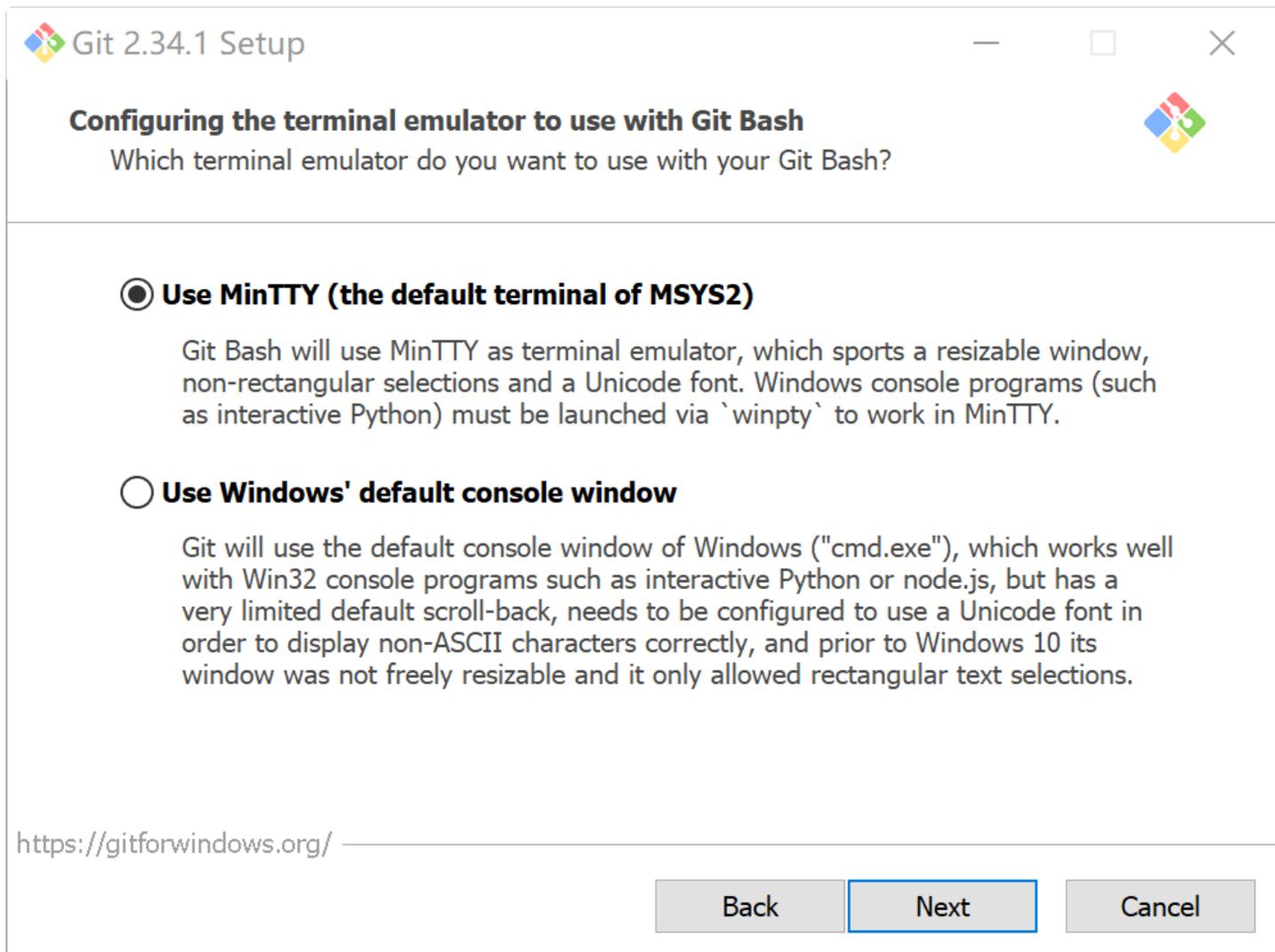


## 2.4.8 行结束转换、终端模拟器设置

01. 配置行结束转换（Configuring the line ending conversions），默认即可。  
点击 **next**，进入下一步。

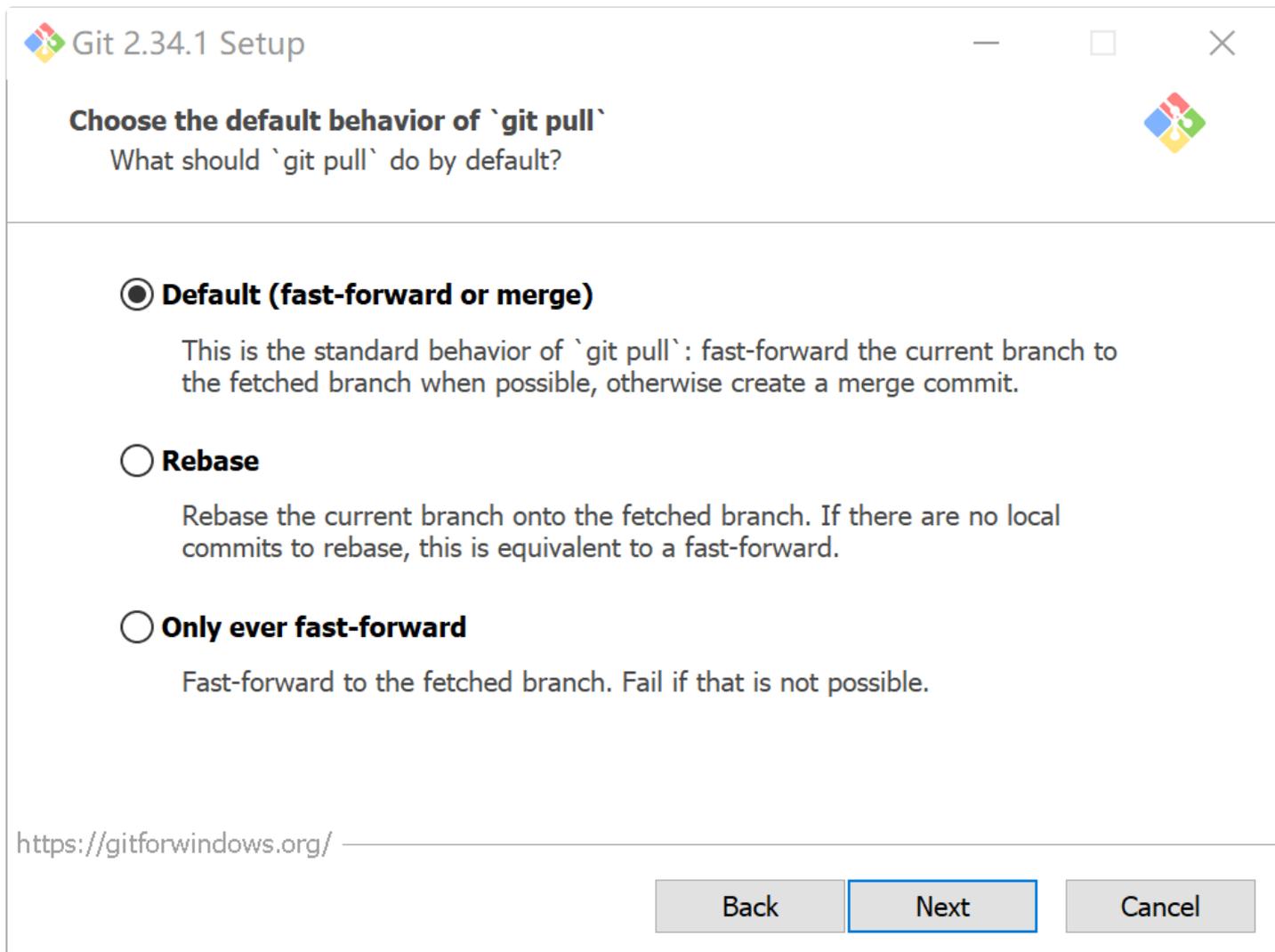


02. 配置终端模拟器与 Git Bash 一起使用（Configuring the terminal emulator to use with Git Bash），默认即可。点击 **next**，进入下一步。

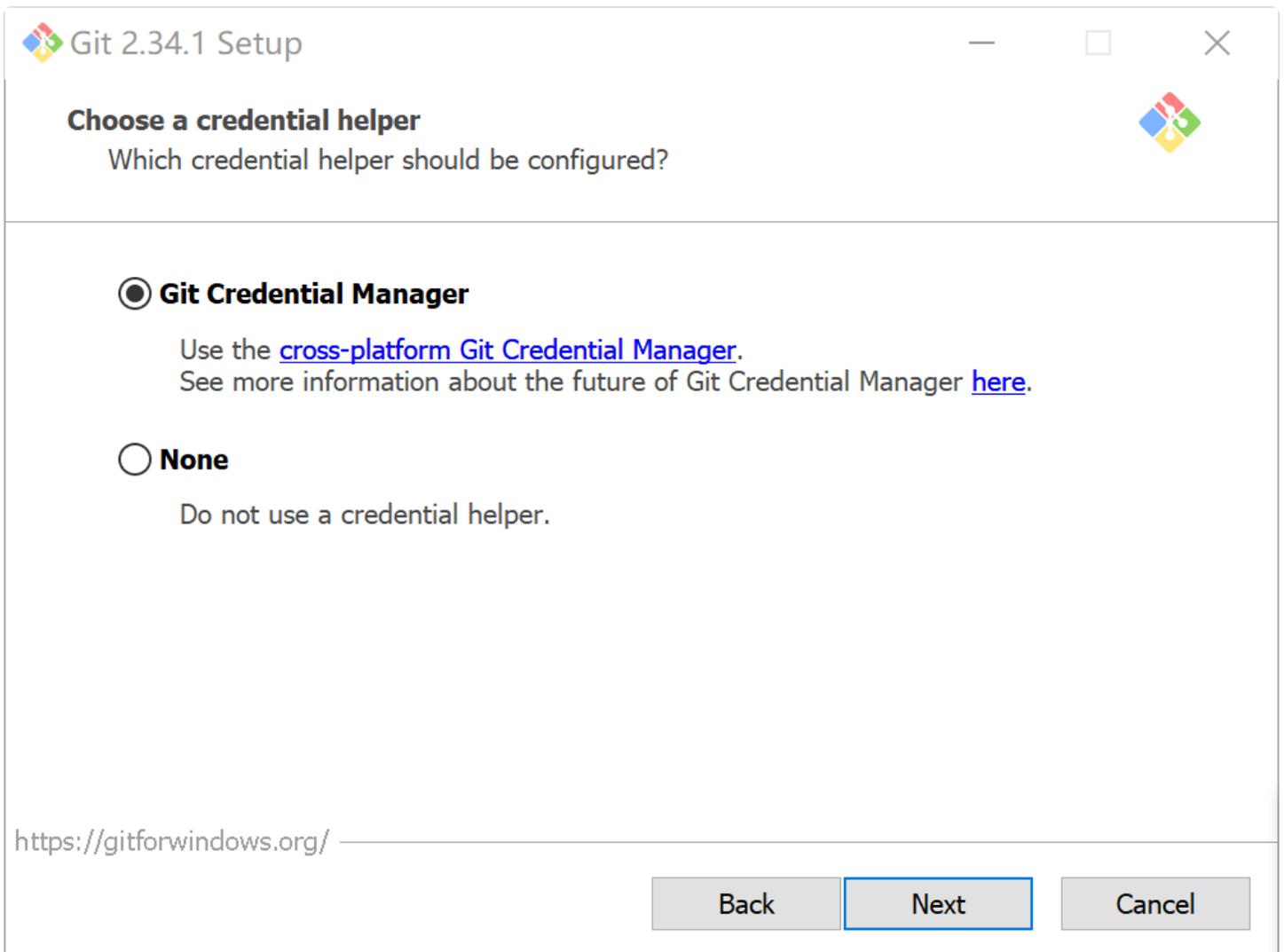


## 2.4.9 Git Pull 风格、凭据管理设置

01. 选择 Git Pull 风格，默认即可。点击 `next` ，进入下一步。

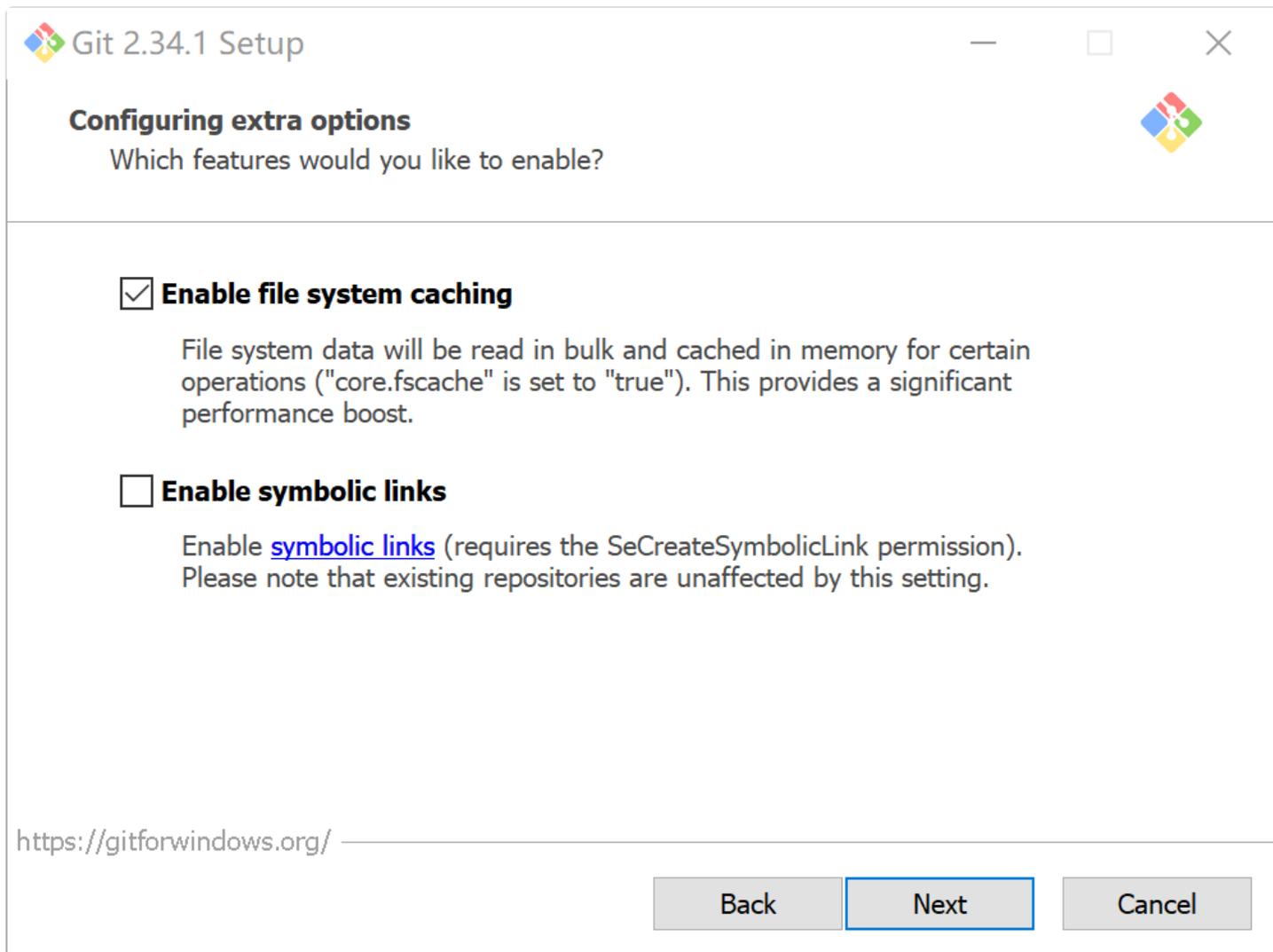


02. 选择凭据管理（Choose a credential helper），默认即可。点击 **next**，进入下一步。

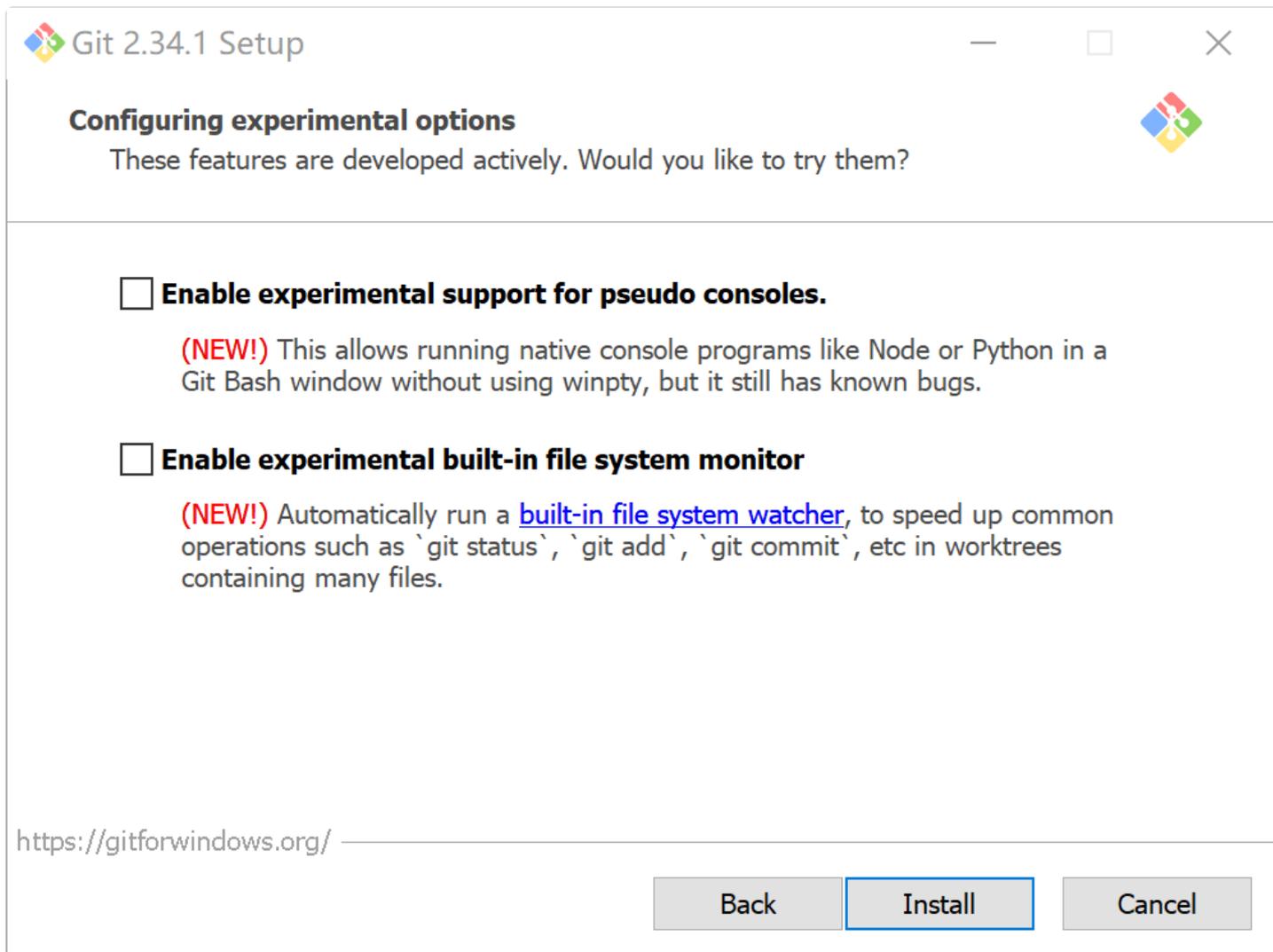


## 2.4.10 额外选项与实验选项

01. 配置额外选项 (Configuring extra options) ，默认即可。点击 **next** ，进入下一步。

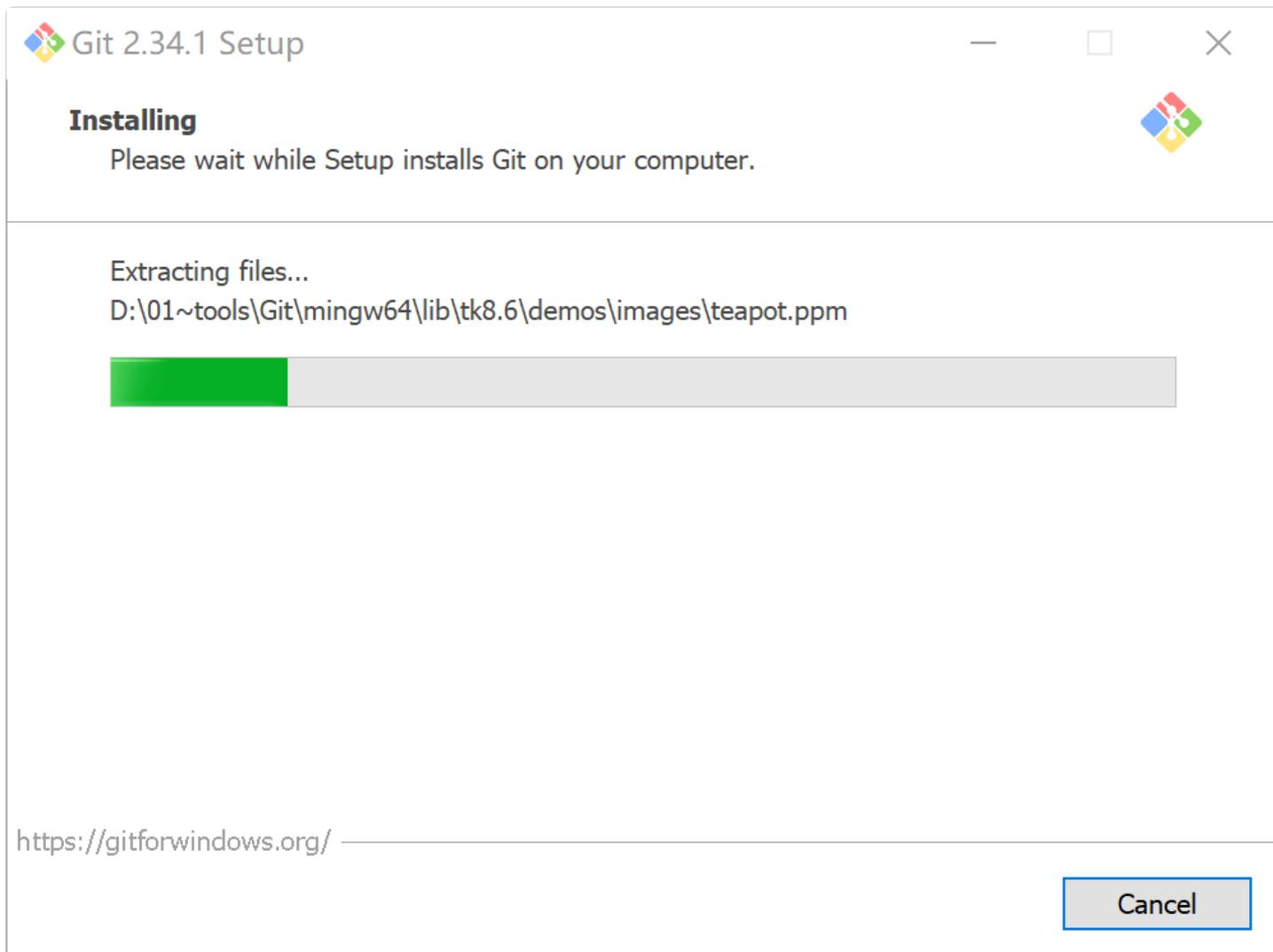


02. 配置实验选项（Configuring experimental options），默认即可。点击 **next**，进入下一步。



## 2.4.11 安装完成

01. 安装中。



02. 去掉勾选，点击 **Finish** 。安装完成！

## Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

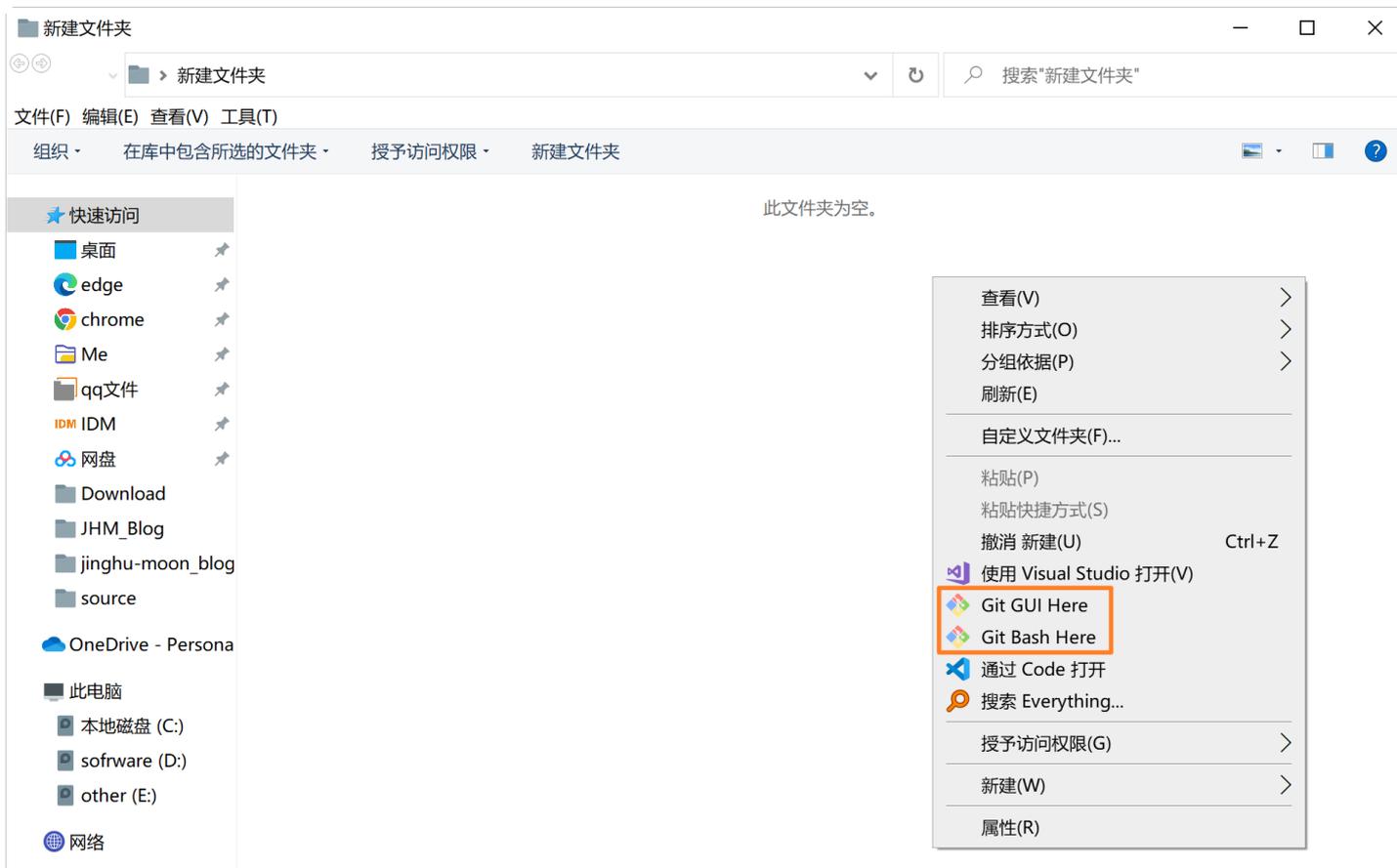


- Launch Git Bash
- View Release Notes

Finish

参考资料: [Windows系统Git安装教程 \(详解Git安装过程\) - 学为所用 - 博客园 \(cnblogs.com\)](https://www.cnblogs.com) 

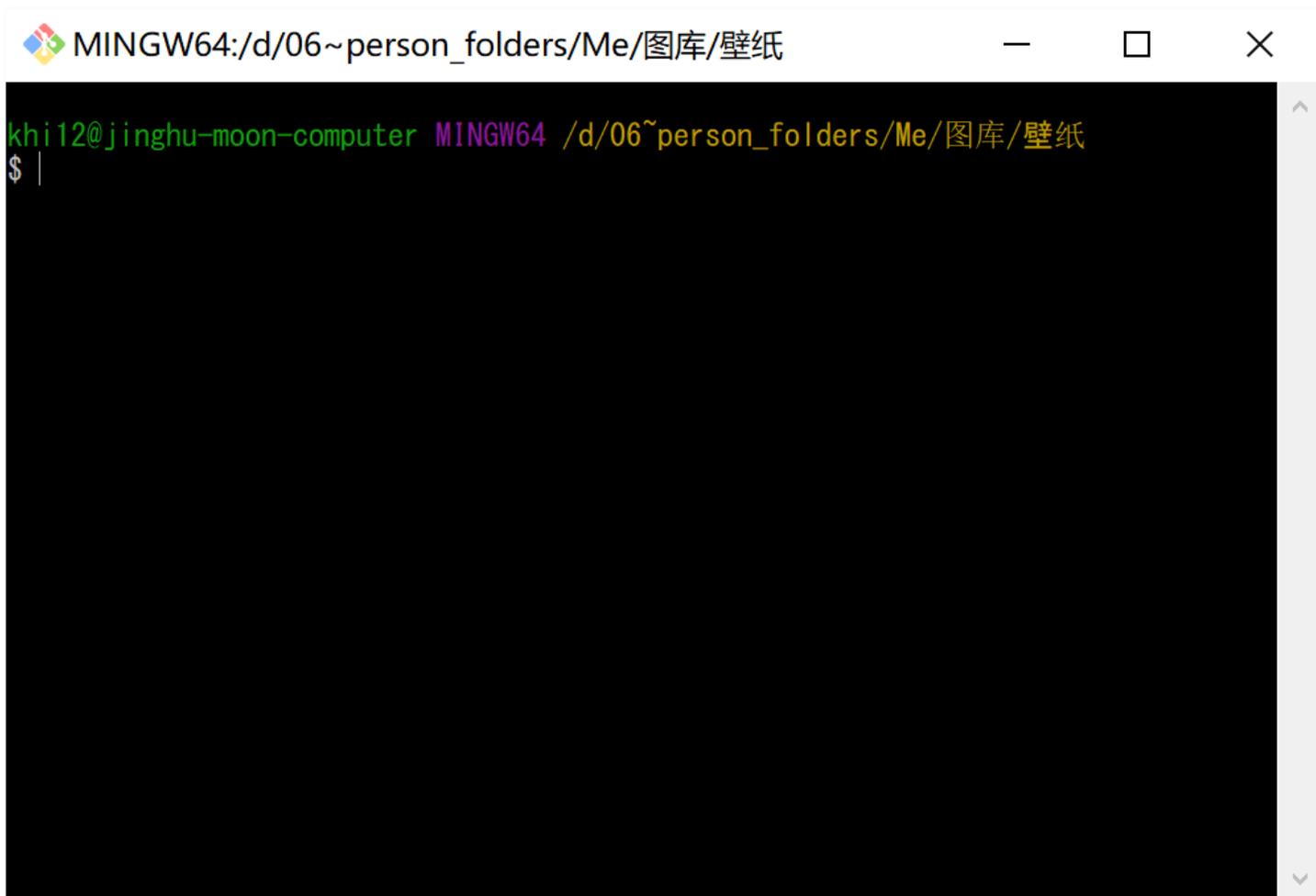
现在，在任意文件夹，右键鼠标，就可以看到 **Git GUI 选项**（git图形界面工具）、**Git Bash 选项**（命令行工具），如下：



在搭建博客的过程中，我们一般使用 `Git Bash` 命令行工具。

## 2.5 Git Bash 界面美化

当我们在任意文件夹，鼠标右键选择 `Git Bash Here` 时，会弹出以下界面。

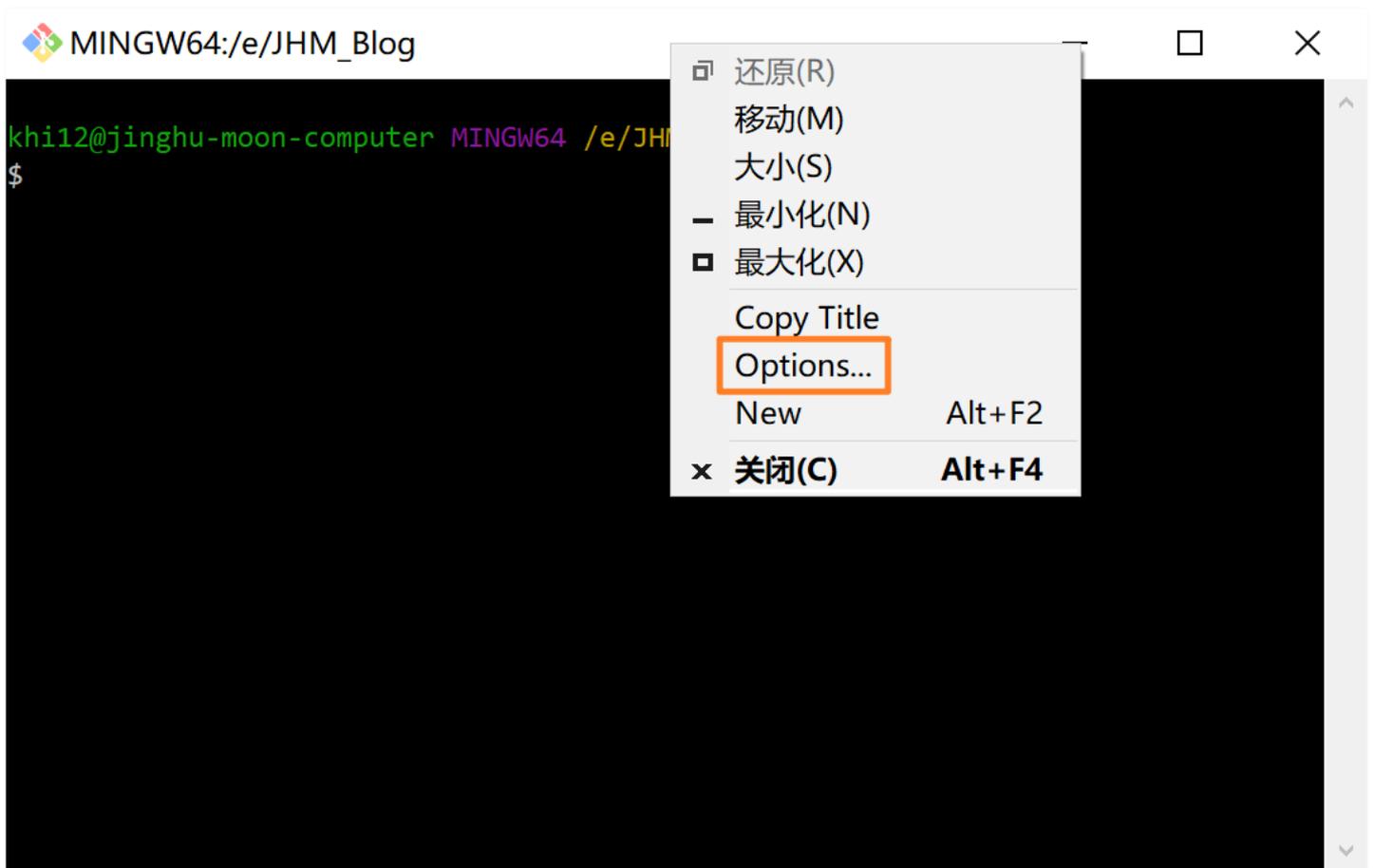


```
MINGW64:/d/06~person_folders/Me/图库/壁纸
khi12@jinghu-moon-computer MINGW64 /d/06~person_folders/Me/图库/壁纸
$ |
```

这个界面很丑，字体太小，配色也丑。所以我们需要美化一下。

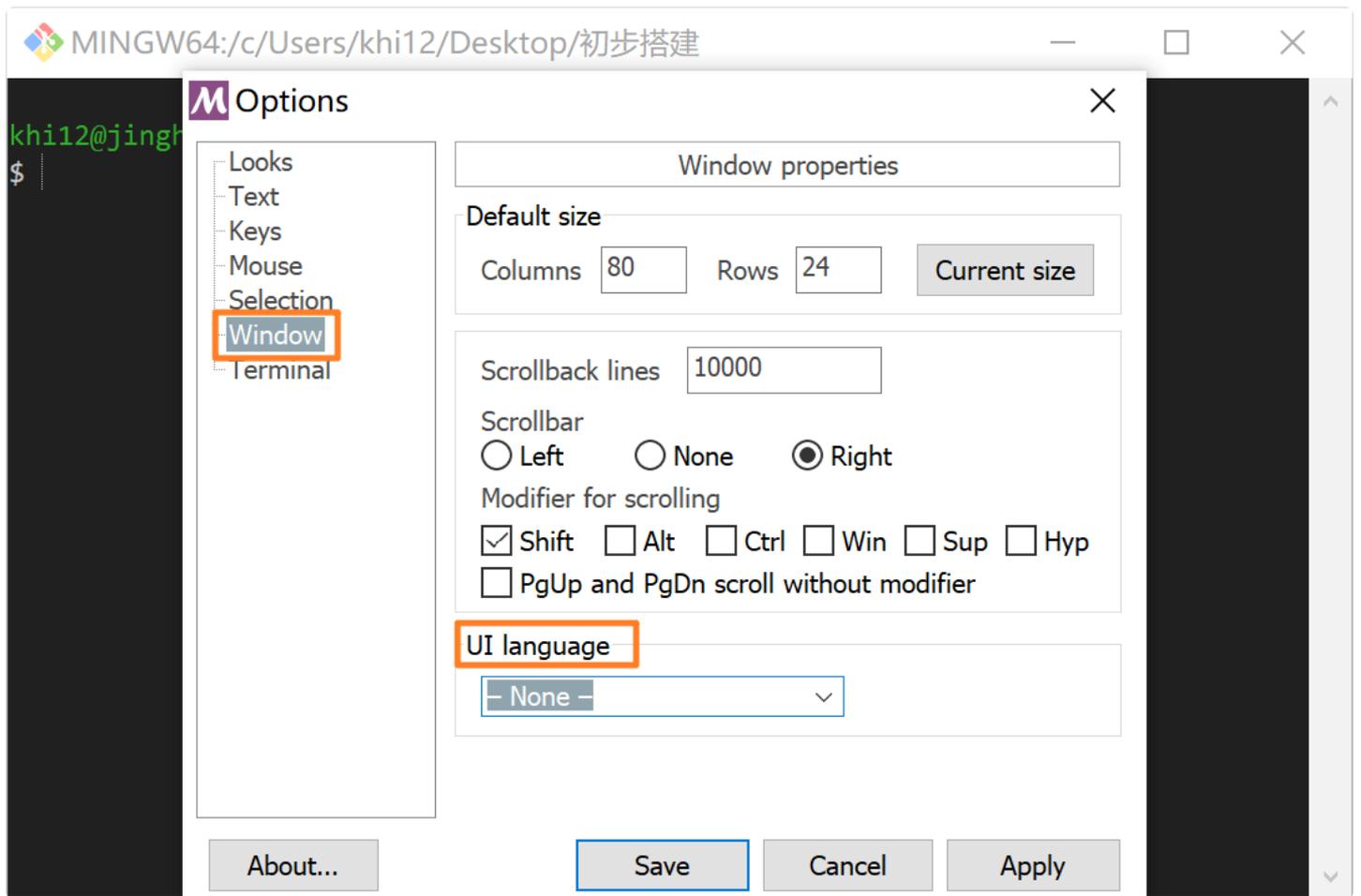
## 2.5.1 进入设置

鼠标右键界面的任务栏，点击 `options` ，进入配置界面。



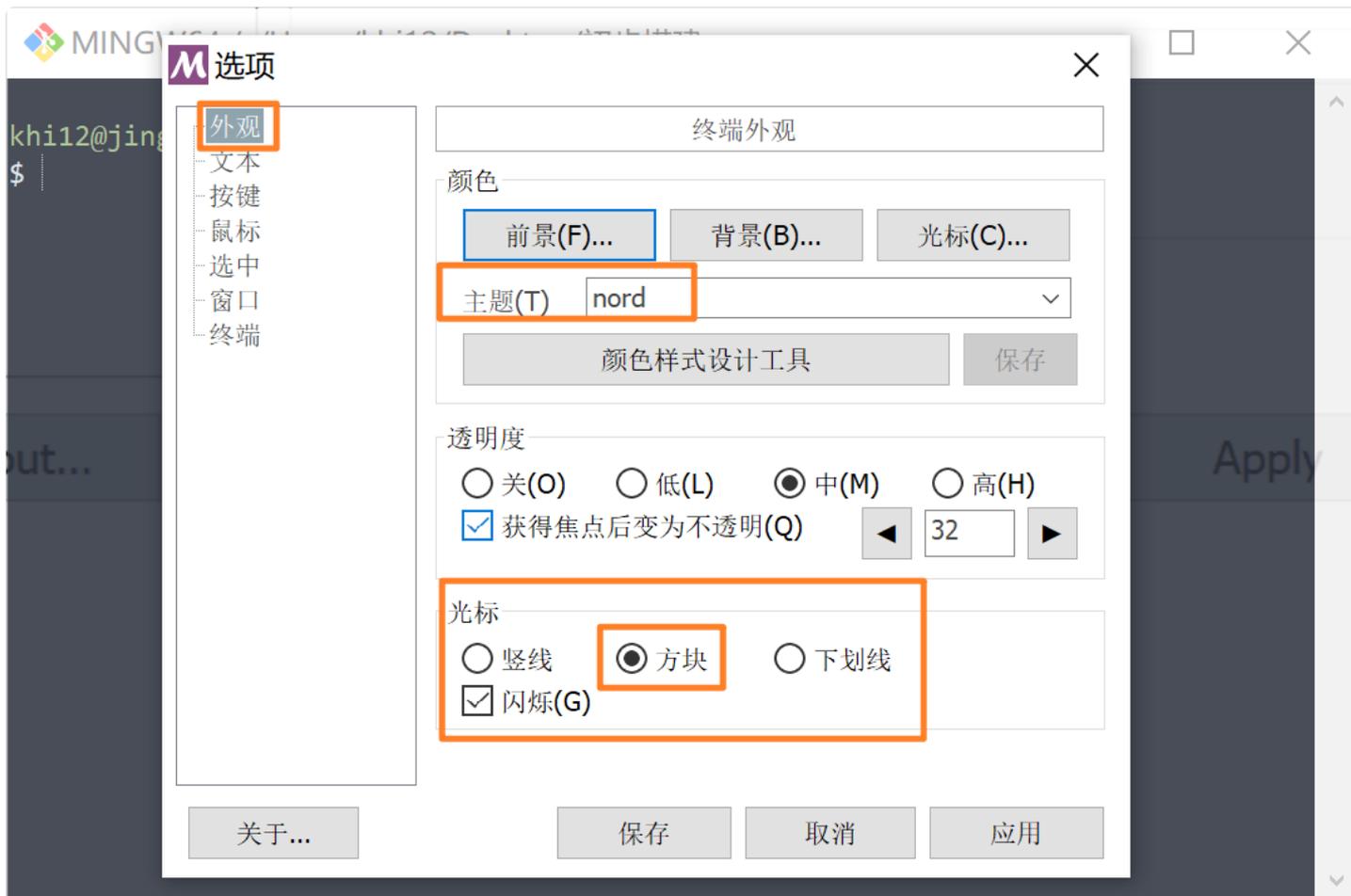
## 2.5.2 切换语言

切换语言为：zh\_CN 或者 @ Windows 语言 @。保存，再次打开配置界面，已经切换为中文。



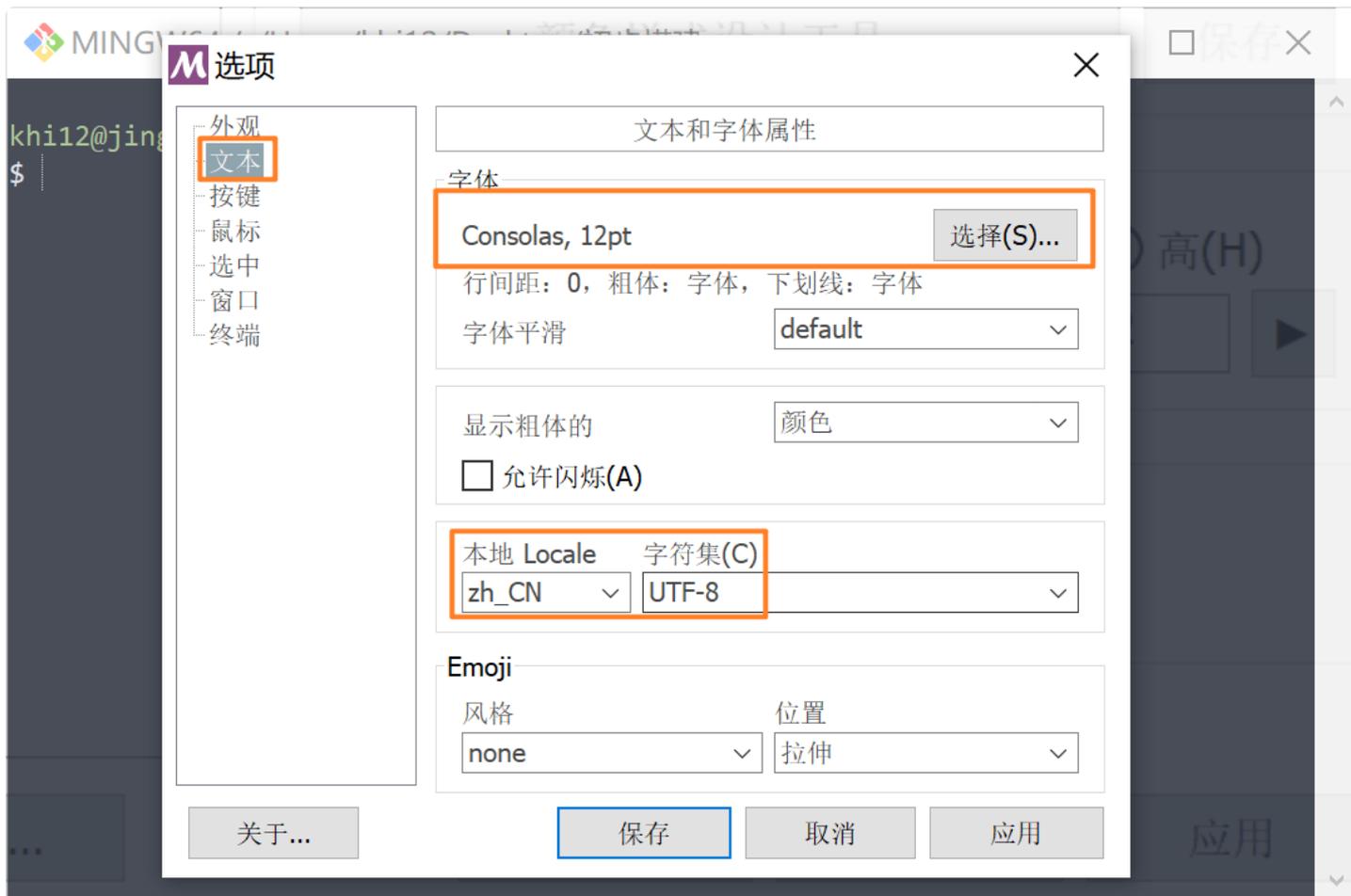
## 2.5.3 外观设置

外观设置，如下：



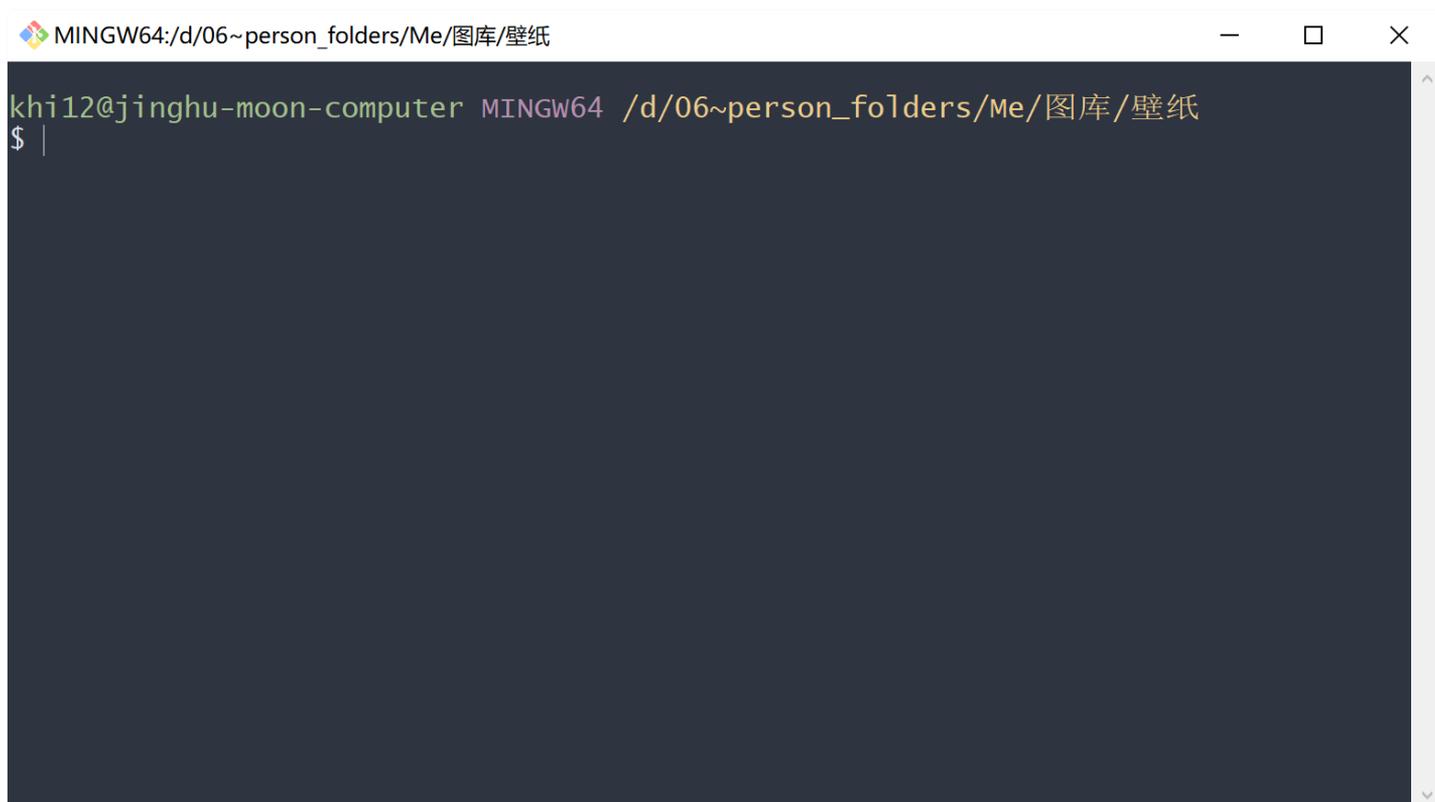
## 2.5.4 文本设置

文本设置，如下



## 2.5 5 美化完成

大功告成！现在的 **Git Bash Here** 如下：



是不是好看了许多。😊

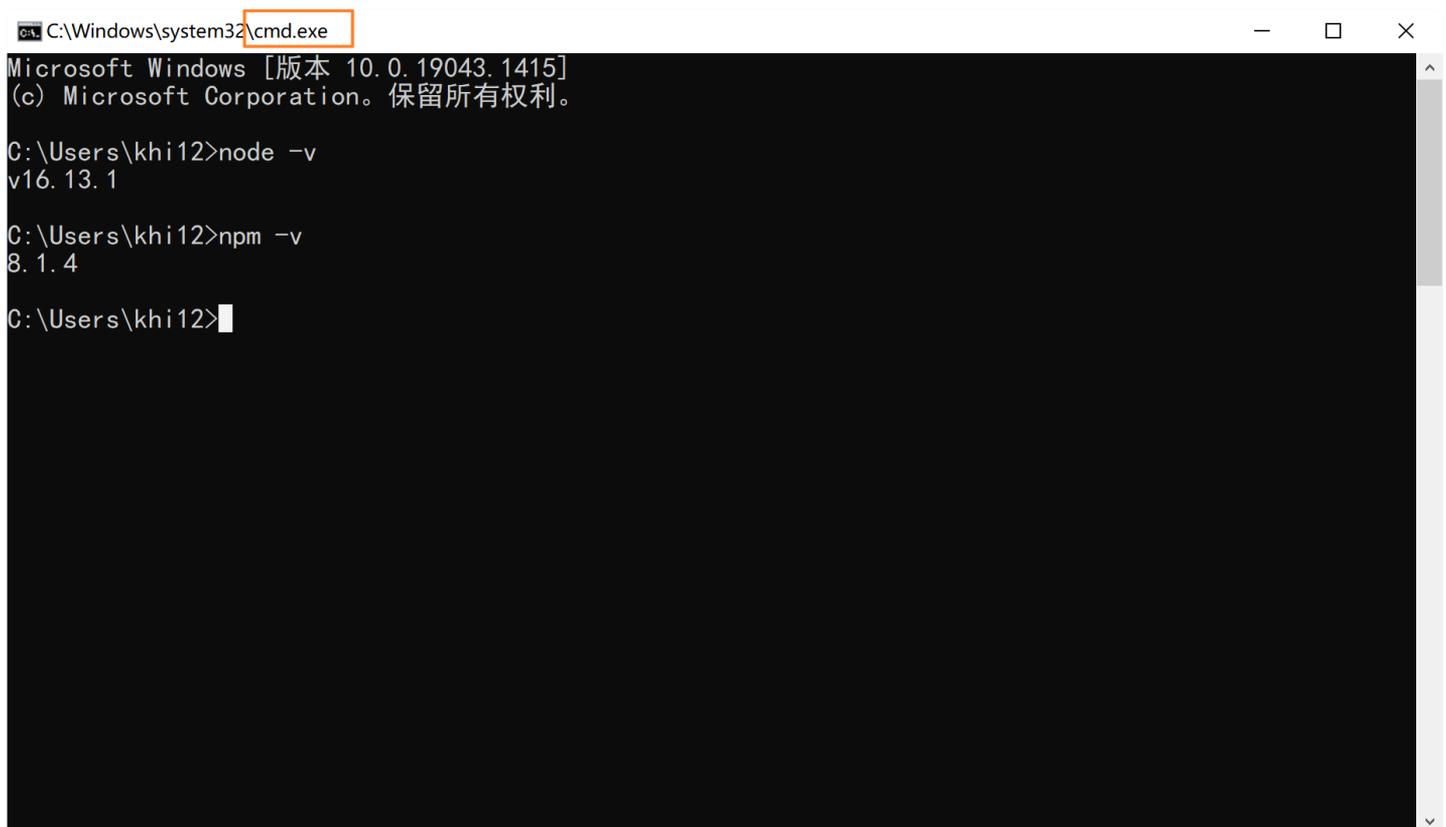
## 2.6 检查是否安装成功

**Win+R** ，输入 **cmd** ，回车，进入以下界面。

输入 **node -v** ，回车；输入 **npm -v** ，回车。如果如下图所示，则代表 **Node.js** 安装成功。

若报错，复制错误代码，百度。

**注意：**在 **Terminal**、**cmd**、**powershell** 界面时，复制、粘贴均不能使用 **Ctrl+C** 和 **Ctrl+V** ，只能使用 **Ctrl+Ins** 和 **Shift+Ins** 或者直接右键复制、粘贴。建议使用后者。笔记本电脑的键盘的 **Ins** 不太好用。



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.19043.1415]
(c) Microsoft Corporation。保留所有权利。

C:\Users\khi12>node -v
v16.13.1

C:\Users\khi12>npm -v
8.1.4

C:\Users\khi12>
```

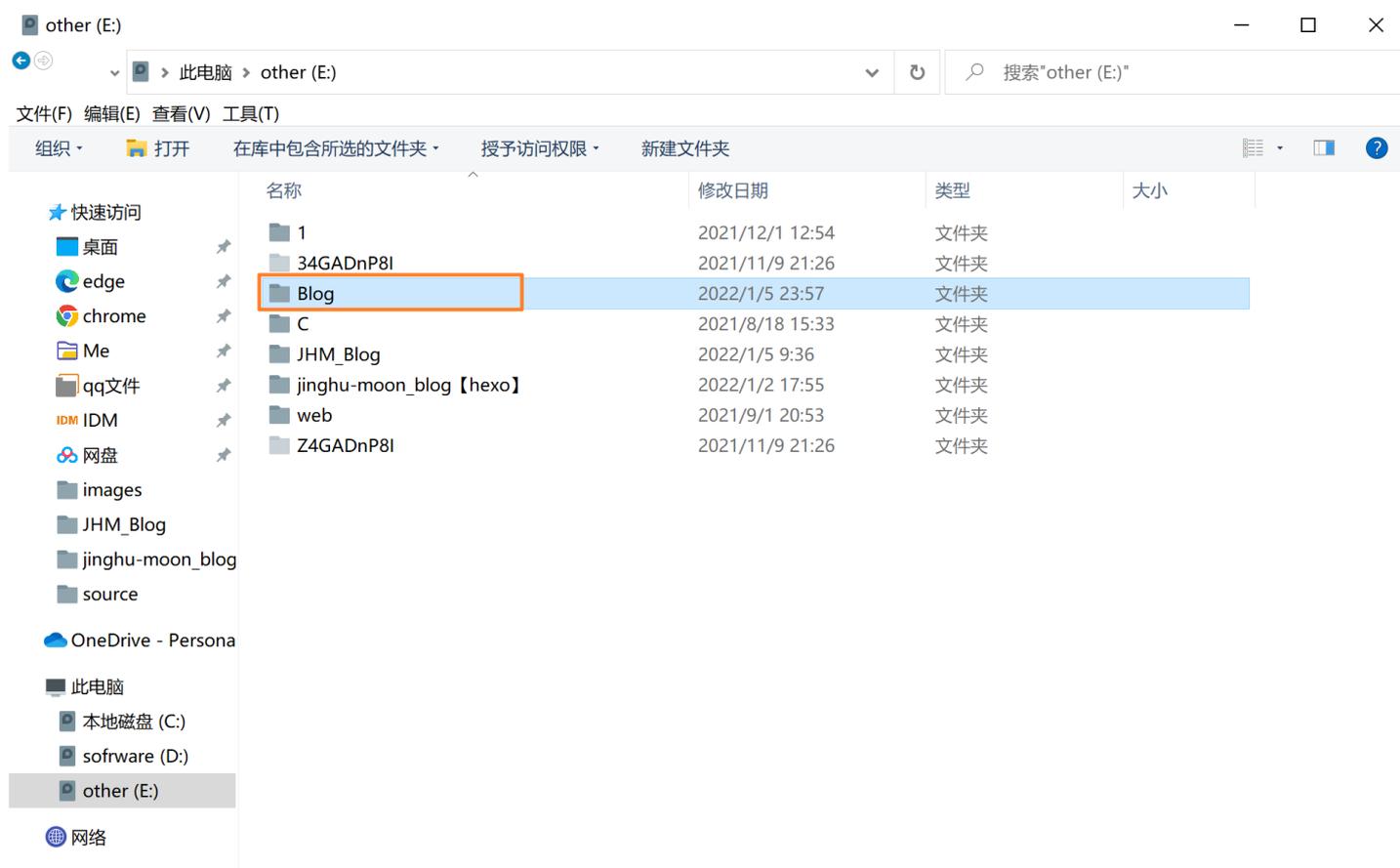
## 2.7 安装 Hexo

Hexo 是我们以后用于搭建个人博客的框架，所以最好新建一个 **Blog** 文件夹用于博客的管理，同时Hexo 框架与以后写作发布的网页都会保存在这个文件夹当中。

注：以下内容，我将会用 **Blog** 文件夹指代博客文件夹。

## 2.7.1 建立博客文件夹

我把 **Blog** 文件建立在 E 盘。注意文件名尽量不使用中文、空格。以免引起未知错误。如下：

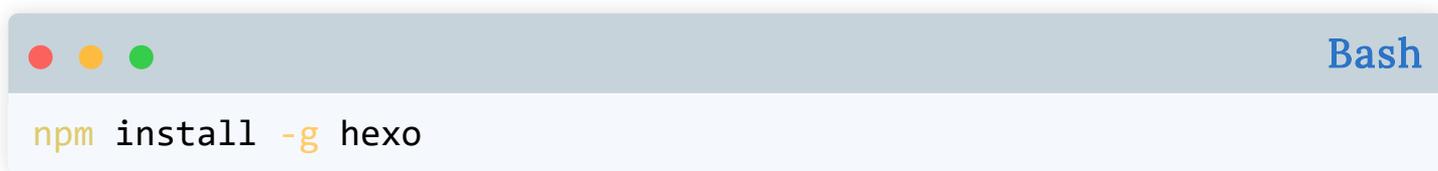


## 2.7.2 下载 Hexo

左键选中 **Blog** 文件夹，右键点击 **Git Bash Here** 。进入以下界面：

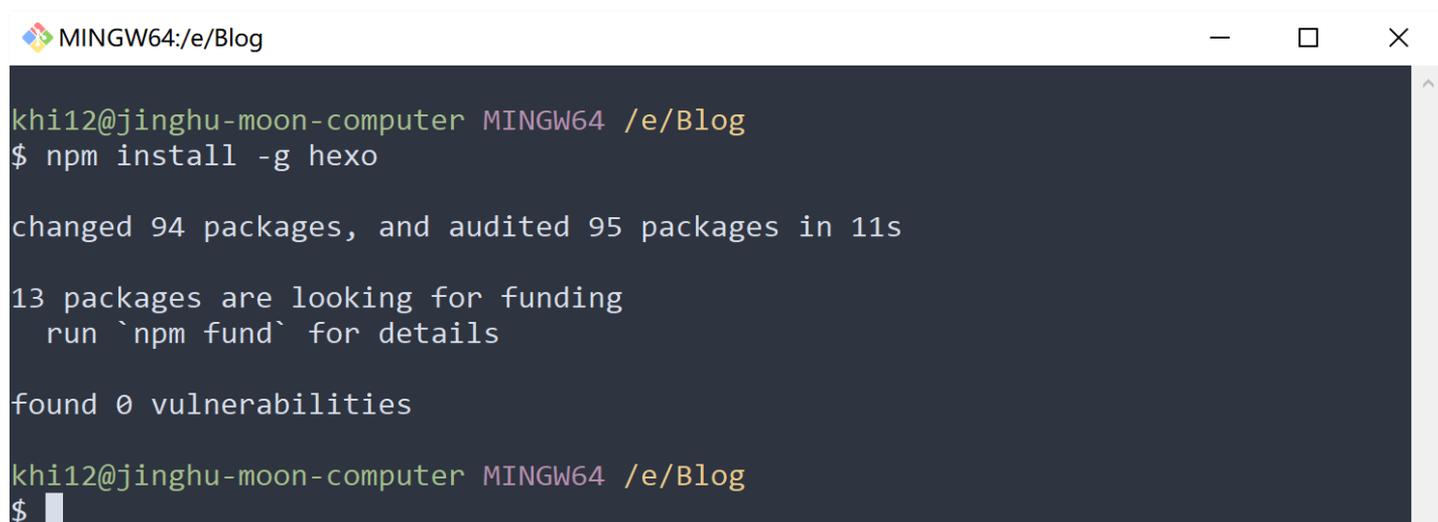


输入命令，下载 Hexo 。



```
Bash
npm install -g hexo
```

结果如下，则下载成功。



```
MINGW64:/e/Blog
khi12@jinghu-moon-computer MINGW64 /e/Blog
$ npm install -g hexo

changed 94 packages, and audited 95 packages in 11s

13 packages are looking for funding
  run `npm fund` for details

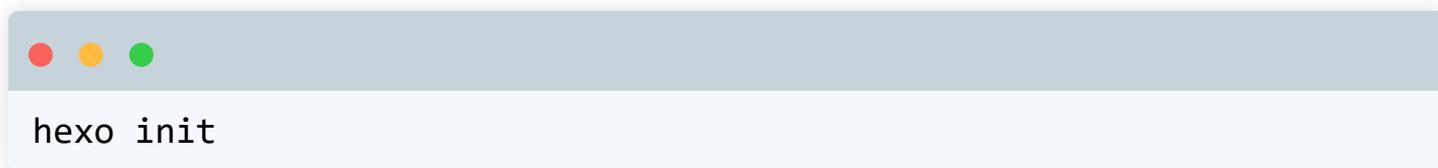
found 0 vulnerabilities

khi12@jinghu-moon-computer MINGW64 /e/Blog
$
```

## 2.7.3 Hexo 初始化

进入 `Blog` 文件夹，右键点击 `Git Bash Here` ，此时， `Blog` 文件夹内部为空白。

输入命令：



```
hexo init
```

结果如下，则初始化成功。



```
MINGW64:/e/Blog
khi12@jinghu-moon-computer MINGW64 /e/Blog
$ hexo init
INFO Cloning hexo-starter https://github.com/hexojs/hexo-starter.git
INFO Install dependencies
INFO Start blogging with Hexo!

khi12@jinghu-moon-computer MINGW64 /e/Blog
$
```

## 2.7.4 构建 Hexo

输入命令：

```
hexo g
```

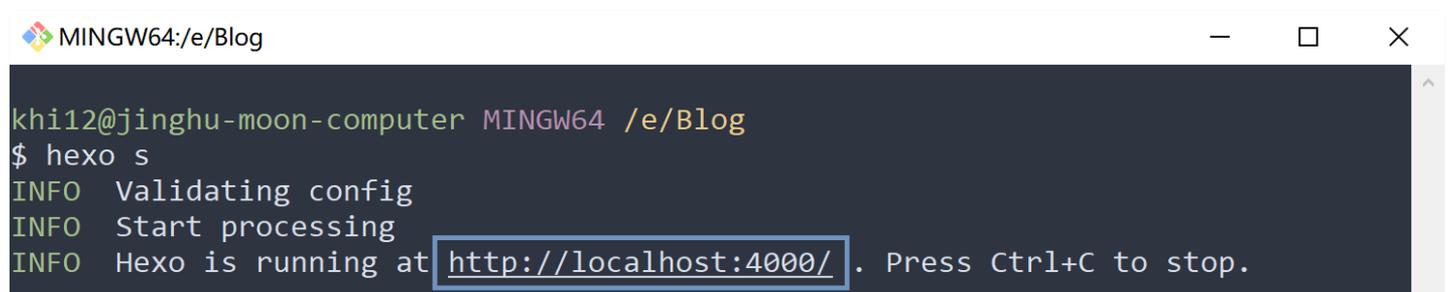
再次构建生成文件。会在当前目录下生成一个名为 `public` 的文件夹，这个文件夹就是根据你 Hexo 的配置，生成的 HTML + CSS + JS 的静态文件。

## 2.7.5 启动 Hexo 本地服务器

输入命令：

```
hexo s
```

结果如下，则成功启动本地服务器。

A terminal window titled 'MINGW64:/e/Blog' showing the execution of 'hexo s'. The output shows 'INFO Validating config', 'INFO Start processing', and 'INFO Hexo is running at http://localhost:4000/'. The URL is highlighted with a blue box.

```
MINGW64:/e/Blog
khi12@jinghu-moon-computer MINGW64 /e/Blog
$ hexo s
INFO Validating config
INFO Start processing
INFO Hexo is running at http://localhost:4000/. Press Ctrl+C to stop.
```

如果遇到报错，请输入以下命令。

```
npm install hexo-server --save
```

```
MINGW64:/e/A_Test_Blog
khi12@jinghu-moon-computer MINGW64 /e/A_Test_Blog (main)
$ npm install hexo-server --save

up to date, audited 979 packages in 11s

51 packages are looking for funding
  run `npm fund` for details

21 vulnerabilities (5 moderate, 13 high, 3 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues possible (including breaking changes), run:
  npm audit fix --force

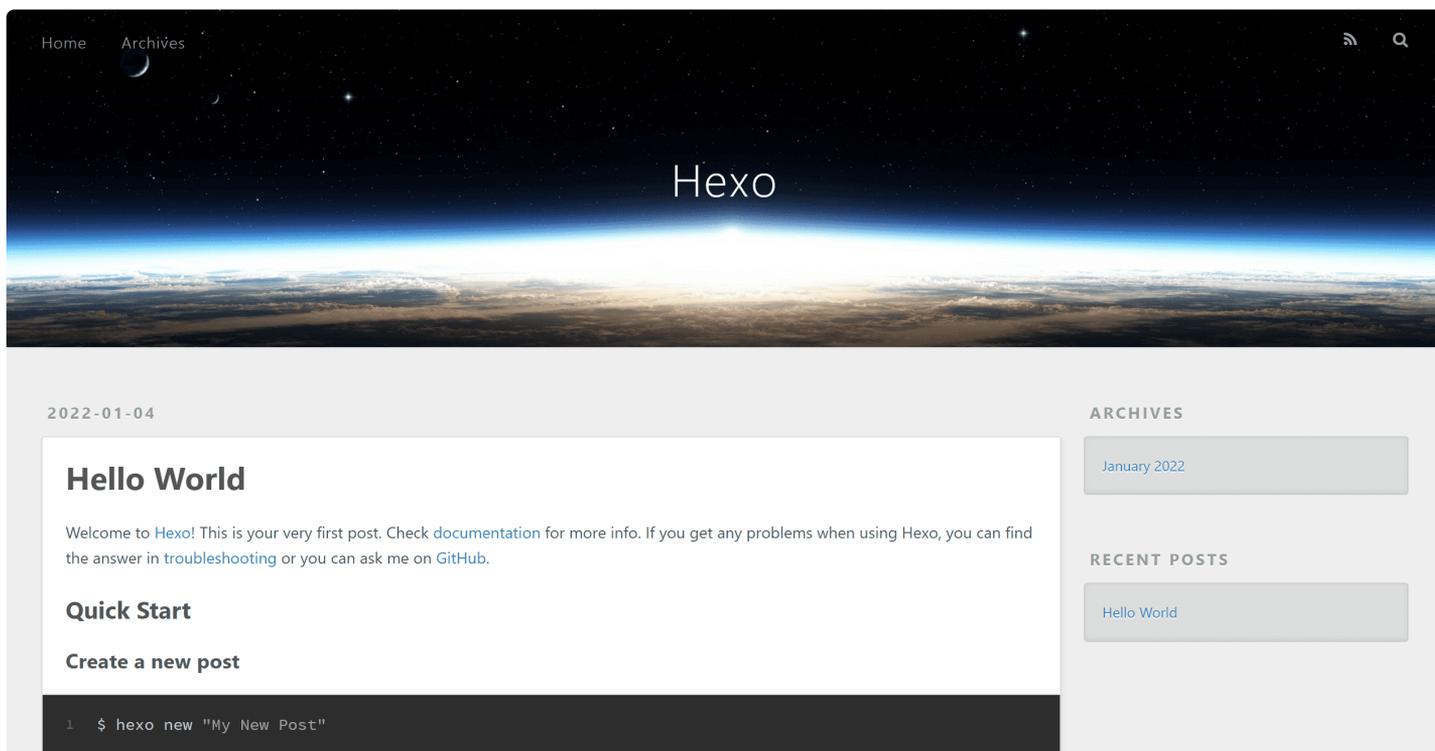
Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.

khi12@jinghu-moon-computer MINGW64 /e/A_Test_Blog (main)
$
```

## 2.7.6 查看网站

浏览器输入框起来的网址：`http://localhost:4000/`。网站如下。



当你刷新本地服务器打开的网页后，你会发现 `Git Bash Here` 界面如下：

```
MINGW64:/e/Blog
khi12@jinghu-moon-computer MINGW64 /e/Blog
$ hexo s
INFO Validating config
INFO Start processing
INFO Hexo is running at http://localhost:4000/ . Press Ctrl+C to stop.
(node:29728) Warning: Accessing non-existent property 'lineno' of module exports
inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:29728) Warning: Accessing non-existent property 'column' of module exports
inside circular dependency
(node:29728) Warning: Accessing non-existent property 'filename' of module exports
inside circular dependency
(node:29728) Warning: Accessing non-existent property 'lineno' of module exports
inside circular dependency
(node:29728) Warning: Accessing non-existent property 'column' of module exports
inside circular dependency
(node:29728) Warning: Accessing non-existent property 'filename' of module exports
inside circular dependency
```

这些 **Warnings** 对 Hexo 程序运行没啥影响，只是看起来不舒服罢了。

如果你想处理它们，可以看看这两篇博文：

- [解决 Hexo 在使用 Node.js 14 时的 Accessing non-existent property 'xxx' of module exports inside circular dependency 问题 - 好一则博 \(haoyizebo.com\)](#)
- [解决Hexo在 Node.js 14 下出现的 Accessing non-existent property 'xxx' of module exports inside circular dependency 问题 | GSGUNDAM砍柴工](#)

关闭 Hexo 本地服务器，在 **Git Bash Here** 界面按 **Ctrl + C**。

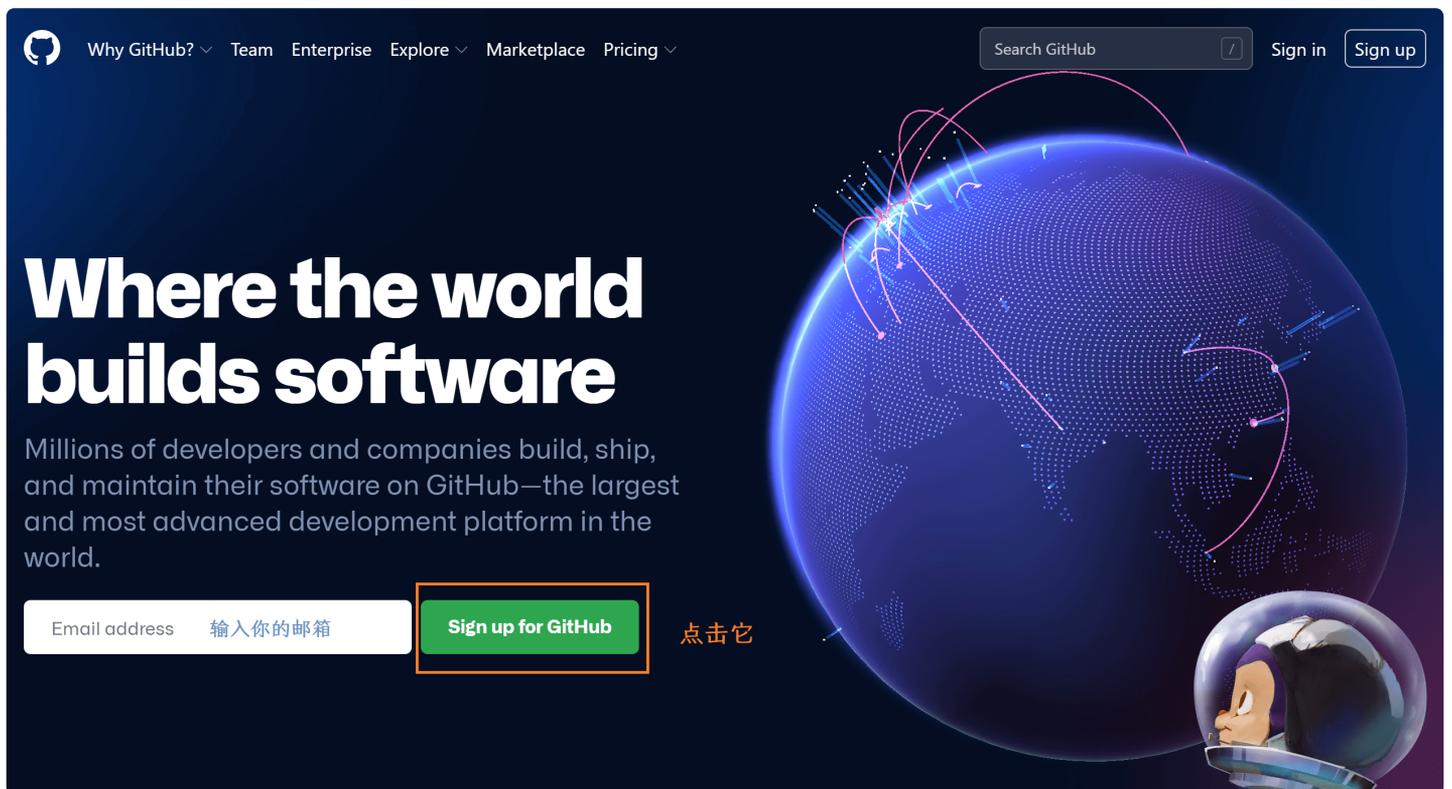
**至此，本地部署完成!**

# 第三章 远程部署

## 3.1 创建 GitHub 仓库

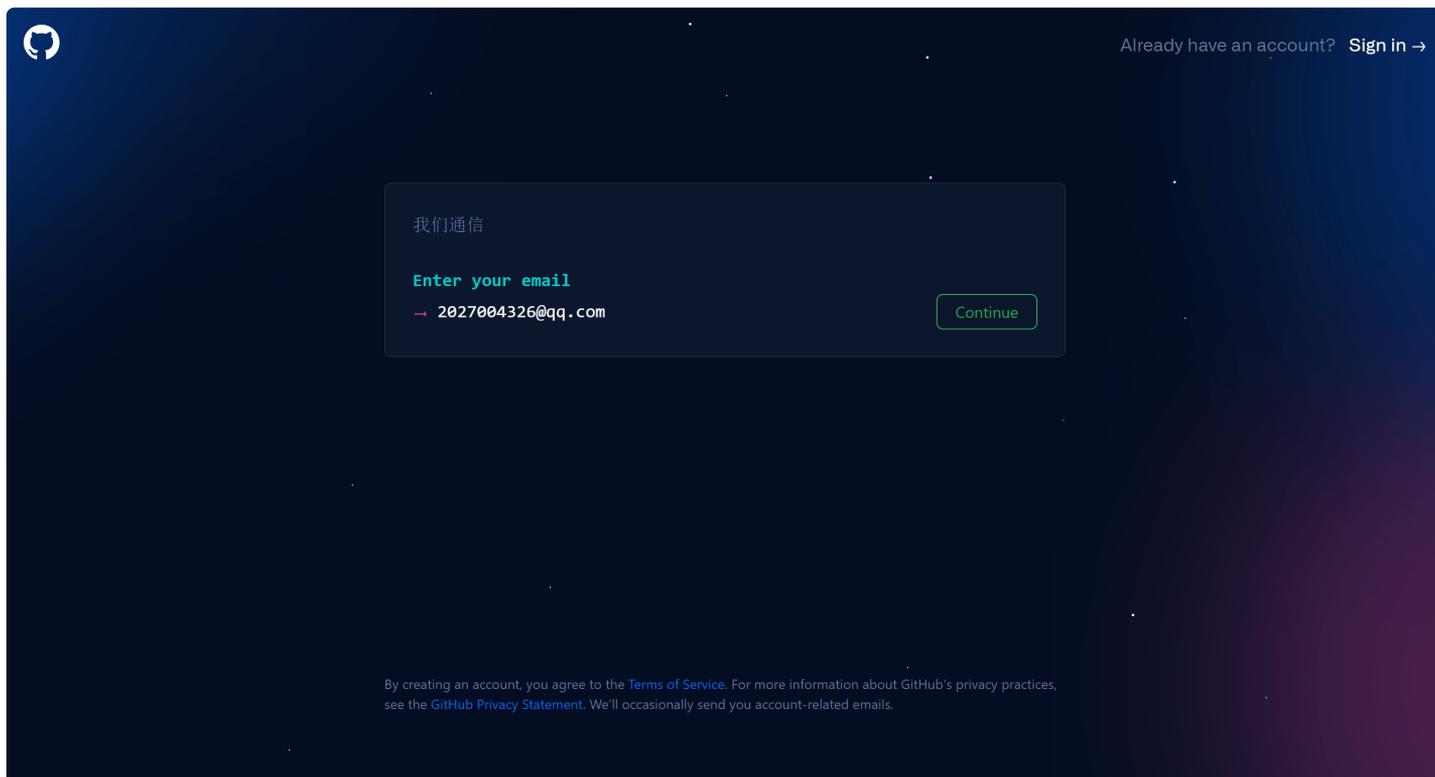
### 3.1.1 注册 GitHub 账号

输入网址：<https://github.com/>，进入，如下：



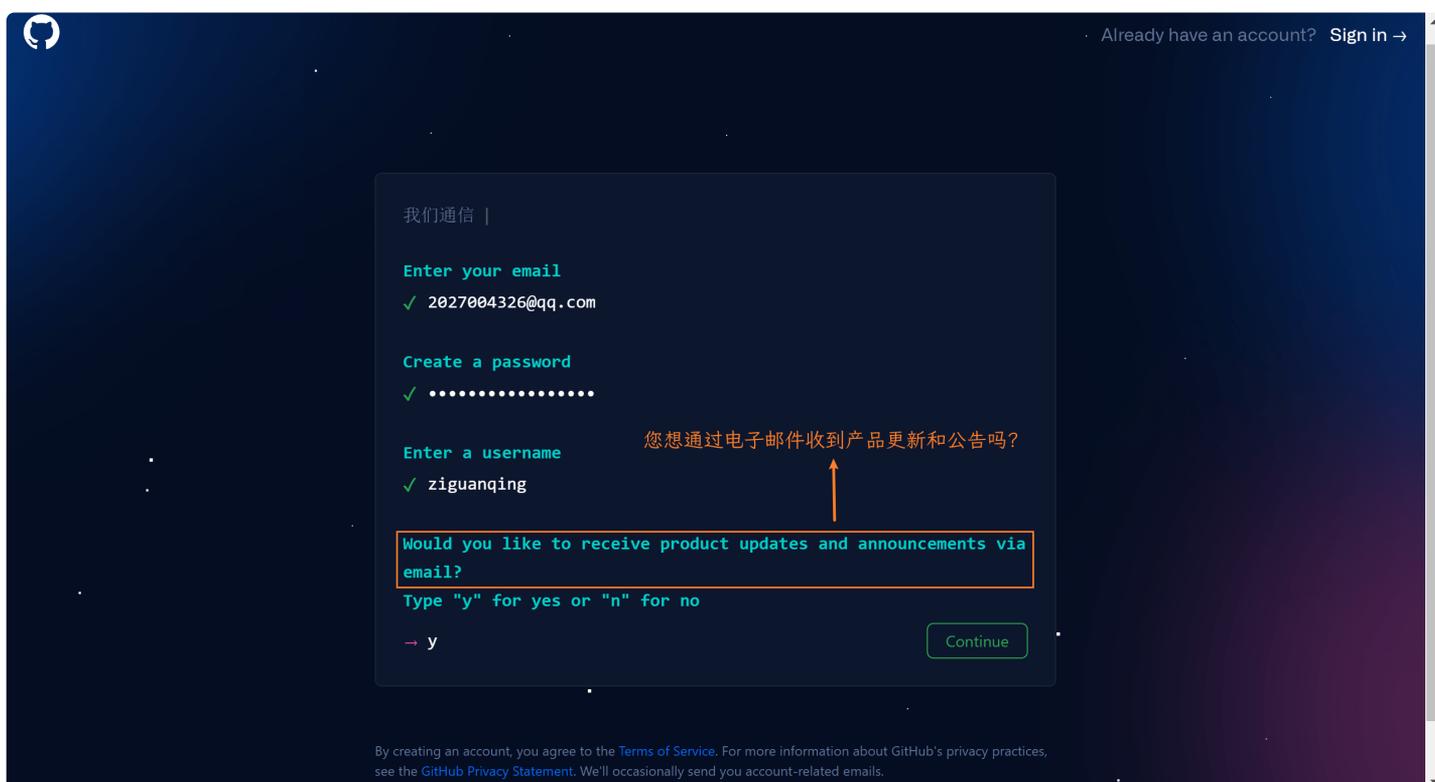
可以使用 QQ 邮箱，示例：[2027004326@qq.com](mailto:2027004326@qq.com)

进入下一个页面，如下：

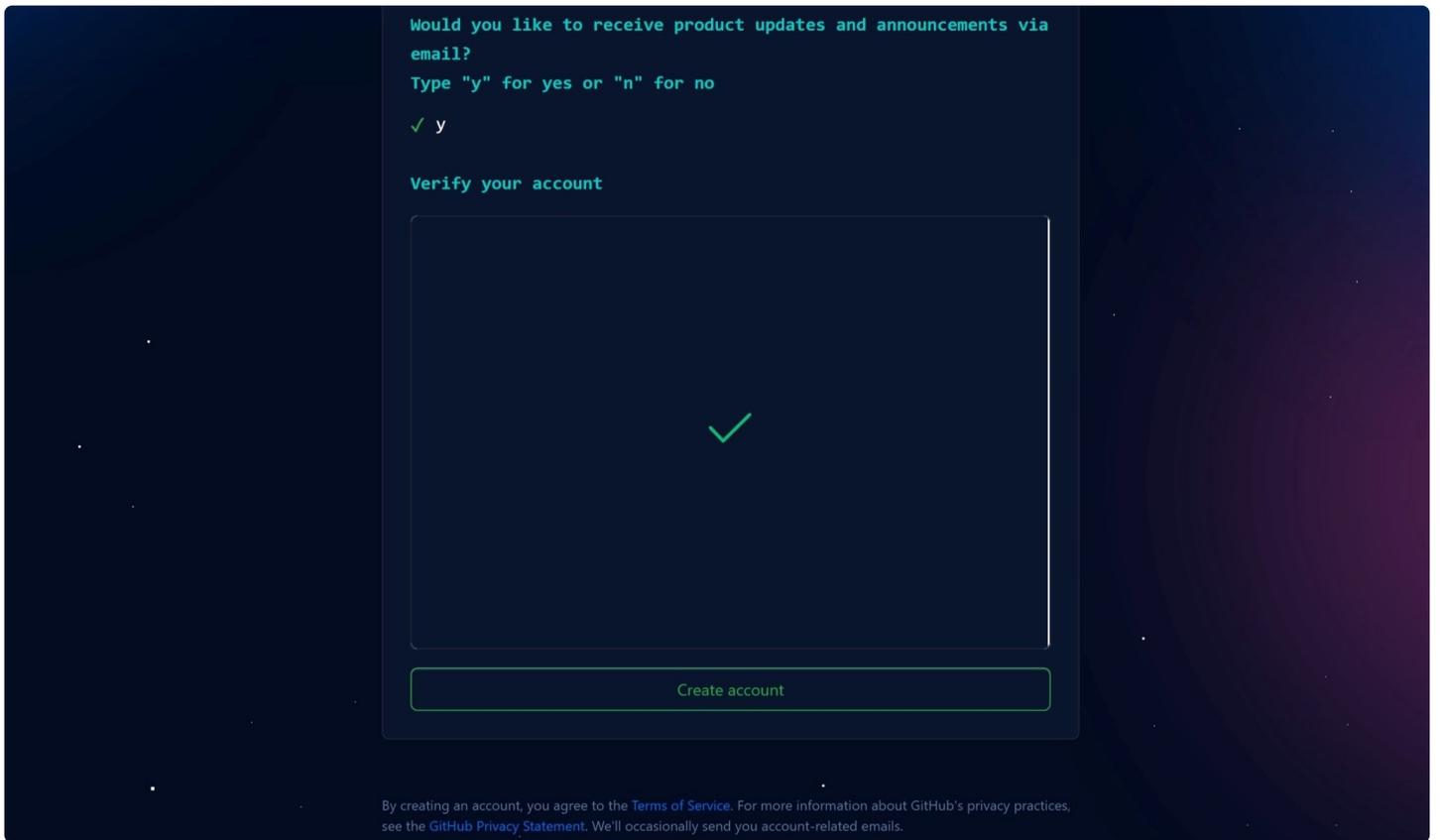


依次输入密码，用户名。点击 **Continue** 。如下：

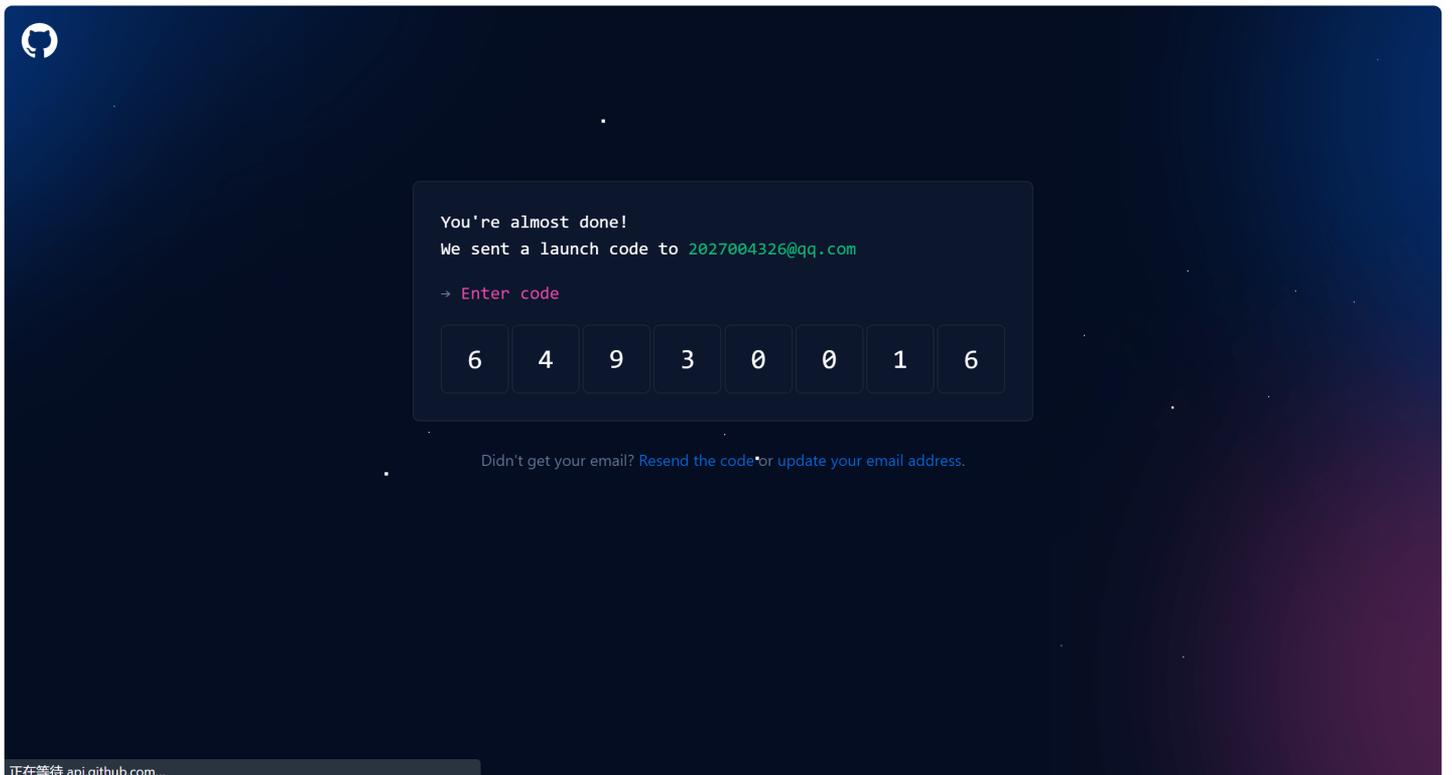
注意：用户名与你的博客网址有很大关系。请认真填写。



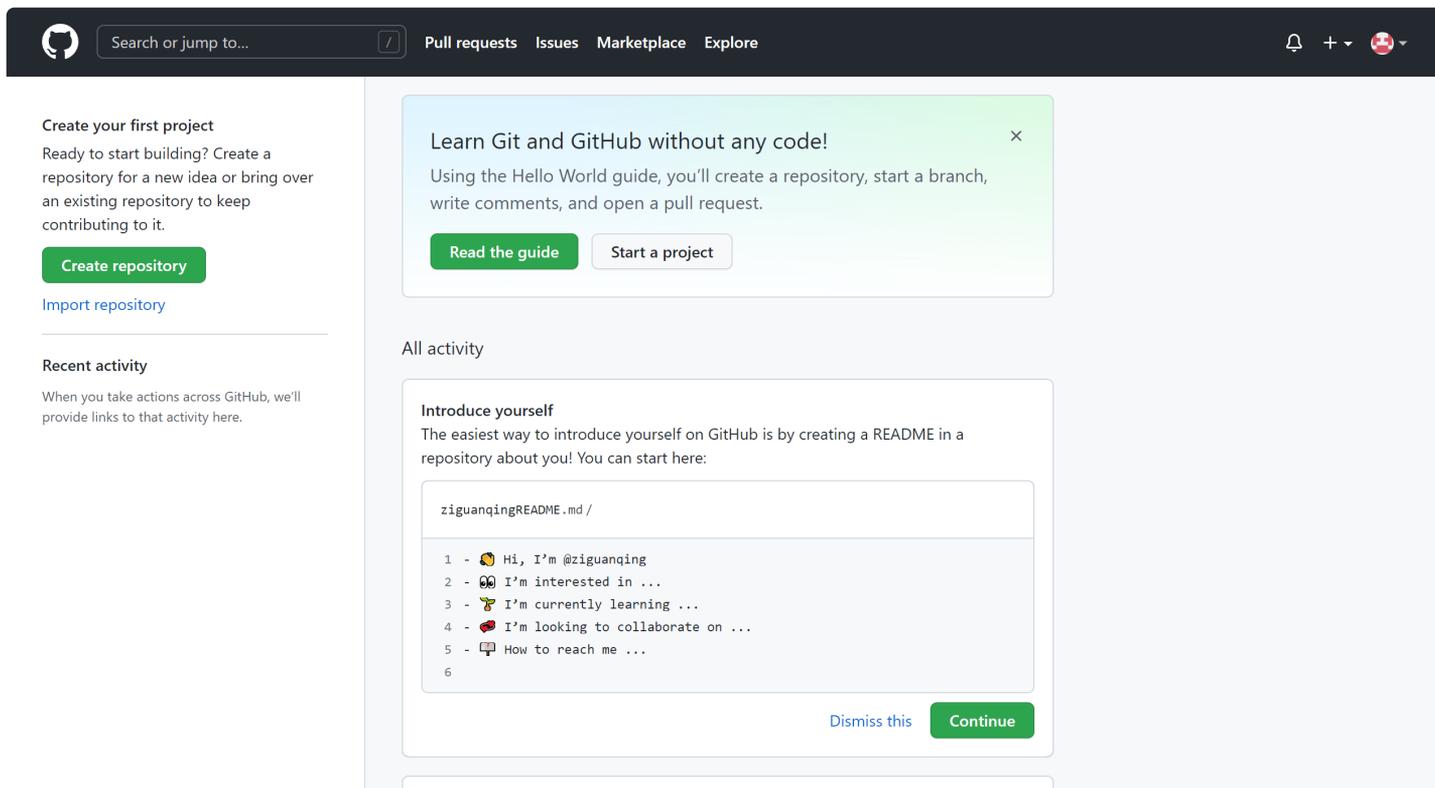
验证是否为真人，选取正确的图片验证码。点击 **Create account** ，如下：



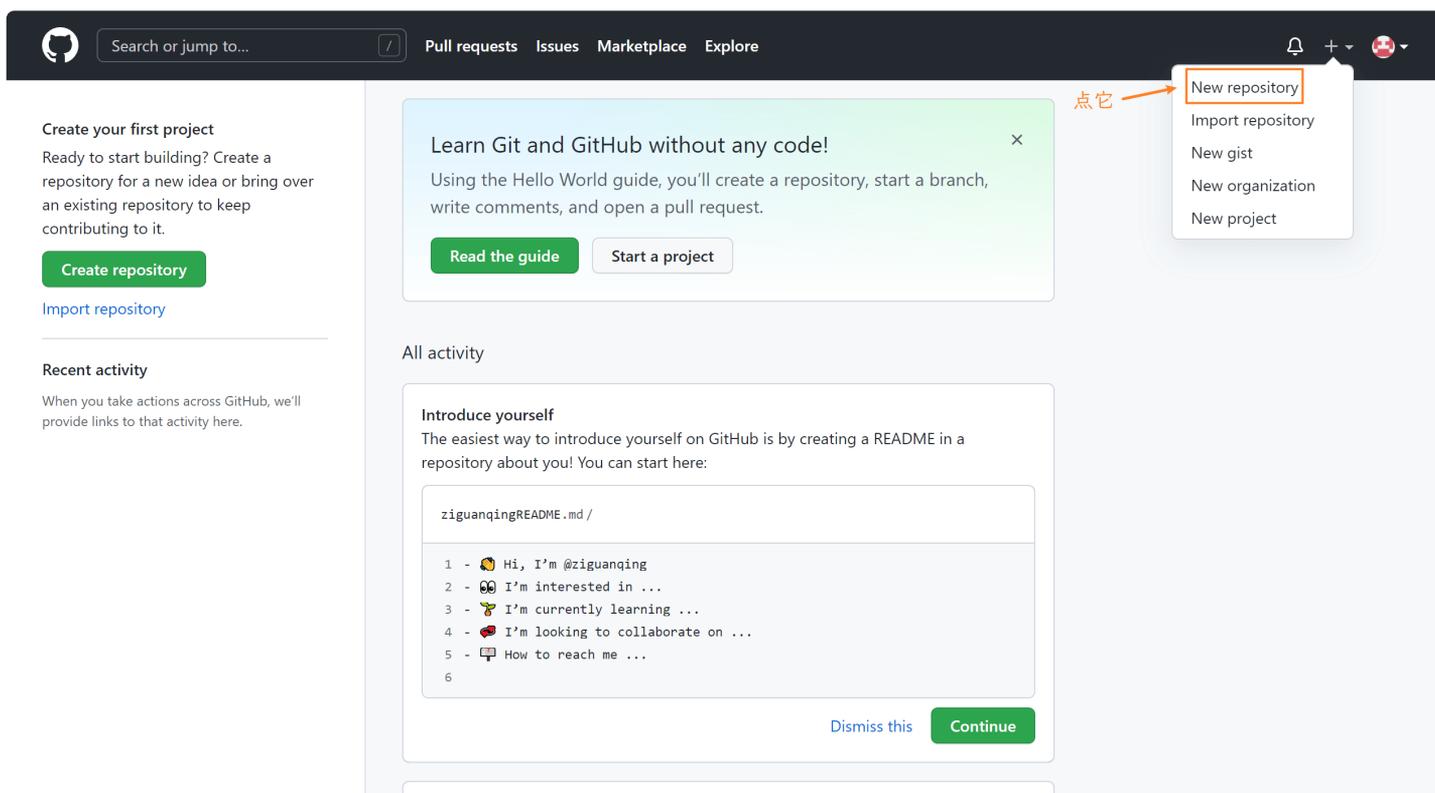
登录邮箱，填写验证码，注册成功！如下：



GitHub 界面如下：



## 3.1.2 创建仓库



进入以下界面：

为了能通过 `GitHub用户名.github.io` 访问博客，仓库（repository）命名为：`GitHub用户名.github.io`。

eg: 我的用户名: **ziguangqing**; 仓库名: **ziguangqing.github.io**

新仓库如下:

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*  / Repository name \*

Great repository names are short and memorable. Need inspiration? How about **musical-lamp**?

Description (optional)

我的 Hexo 博客

- Public**  
Anyone on the internet can see this repository. You choose who can commit.
- Private**  
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

- Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)
- Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)
- Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

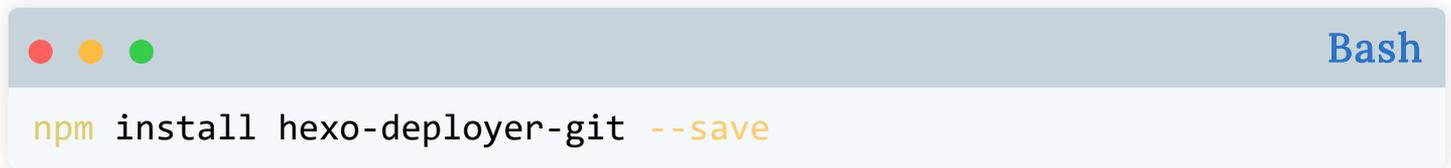
Create repository

**注意:** 2021.04.11, GitHub 新仓库分支命名为 **main**, 而 Hexo 默认用 **master** 分支。所以创建好后进入仓库, 可以看下是什么分支, 我的是 **main** 分支

## 3.2 使用远程部署插件

### 3.2.1 安装插件

在 `Blog` 目录下，使用 Git 终端输入命令：

A terminal window titled "Blog" with a "Bash" prompt. The command `npm install hexo-deployer-git --save` is entered and highlighted in yellow.

```
npm install hexo-deployer-git --save
```

结果如下，则插件安装成功。

A terminal window titled "MINGW64:/e/Blog" showing the output of the installation command. The output indicates that 1 package was added and 258 packages were audited in 2 seconds. It also shows that 16 packages are looking for funding and that no vulnerabilities were found.

```
MINGW64:/e/Blog
khi12@jinghu-moon-computer MINGW64 /e/Blog
$ npm install hexo-deployer-git --save
added 1 package, and audited 258 packages in 2s
16 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
khi12@jinghu-moon-computer MINGW64 /e/Blog
$
```

### 3.2.2 配置插件

打开 `Blog` 目录下的 `_config.yml` 文件，编辑（记事本、Sublime、VS code），在文件末尾补上以下内容：

```
deploy:
  type: git
  repo: git@github.com:GitHub用户名/GitHub用户名.github.io.git
  branch: main
```

# 示例

```
deploy:
  type: git
  repo: git@github.com:ziguanqing/ziguanqing.github.io.git
  branch: main
```

- type: git
- repo: GitHub 仓库链接
- branch: 你仓库的默认分支，可能为: main、master

\_config.yml ×

E: > Blog > \_config.yml

```
96
97 # Extensions
98 ## Plugins: https://hexo.io/plugins/
99 ## Themes: https://hexo.io/themes/
100 theme: landscape
101
102 # Deployment
103 ## Docs: https://hexo.io/docs/one-command-deployment
104 deploy:
105   type: git
106   repo: git@github.com:ziguanqing/ziguanqing.github.io.git
107   branch: main
```

## 3.3 Github SSH 配置

## 3.3.1 生成密钥

当我们频繁操作 GitHub 时，需要输入密码来验证账户，为了简化我们的 pull/push 过程，我们需要设置 SSH Key。

1、查看 Git 环境所有配置，输入以下命令：

```
git config --list
```

2、如果没有配置 `user.name` 和 `user.email`，则需要先配置。

3、配置用户名，随意。

```
git config --global user.name "用户名"
```

# 示例

```
git config --global user.name "jinghu-moon"
```

4、配置邮箱，必须为 GitHub 邮箱。

```
git config --global user.email "123456@qq.com"
```

# 示例

```
git config --global user.email "2027004326@qq.com"
```

配置完，如下：

```
khi12@jinghu-moon-computer MINGW64 /e/Blog
$ git config --global user.name "ziguanqing"

khi12@jinghu-moon-computer MINGW64 /e/Blog
$ git config --global user.email "2027004326@qq.com"

khi12@jinghu-moon-computer MINGW64 /e/Blog
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=D:/01~tools/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=main
user.name=ziguanqing
user.email=2027004326@qq.com
core.autocrlf=false
credential.helper=store
http.sslverify=false

khi12@jinghu-moon-computer MINGW64 /e/Blog
$
```

## 5、进入 SSH Key 目录



```
cd ~/.ssh
ls
```

MINGW64:/c/Users/khi12/.ssh

```
khi12@jinghu-moon-computer MINGW64 /e/Blog
$ cd ~/.ssh

khi12@jinghu-moon-computer MINGW64 ~/.ssh
$
```

## 6、生成密钥

执行以下命令，生成公钥和私钥

```
ssh-keygen -t rsa -C "github邮箱"  
  
# 示例  
ssh-keygen -t rsa -C "2027004326@qq.com"
```

生成密钥，一共需要按四次 `enter` 键。

成功生成密钥，界面如下：

```
khi12@jinghu-moon-computer MINGW64 /e/Blog  
$ cd ~/.ssh  
  
khi12@jinghu-moon-computer MINGW64 ~/.ssh  
$ ssh-keygen -t rsa -C "2027004326@qq.com" → 第一次回车  
Generating public/private rsa key pair.  
Enter file in which to save the key (/c/Users/khi12/.ssh/id_rsa): → 第二次回车  
Enter passphrase (empty for no passphrase): → 第三次回车  
Enter same passphrase again: → 第四次回车  
Your identification has been saved in /c/Users/khi12/.ssh/id_rsa  
Your public key has been saved in /c/Users/khi12/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:P0VVDPYjp/Rgi8XyaxB3GZ5sQd66L8XgRe03/SfZmro 2027004326@qq.com  
The key's randomart image is:  
+---[RSA 3072]---+  
|         . .B=O|  
|        o O***O|  
|       X.O*++|  
|      o.+..OO=|  
|     S  ..O.*+|  
|    . .O +.*|  
|   O.  .=.|  
|   .  +.|  
|   Eo ..|  
+-----[SHA256]-----+  
  
khi12@jinghu-moon-computer MINGW64 ~/.ssh
```

## 7、查看 `.ssh` 目录下文件

输入以下命令：

```
ls
```

```
khi12@jinghu-moon-computer MINGW64 ~/.ssh
$ ls
id_rsa  id_rsa.pub  known_hosts  known_hosts.old

khi12@jinghu-moon-computer MINGW64 ~/.ssh
$
```

## 8、查看公钥并复制

输入以下命令：

```
cat id_rsa.pub
```

```
khi12@jinghu-moon-computer MINGW64 ~/.ssh
$ cat id_rsa.pub
ssh-rsa AAAAC3NzaC1lZDA4NTQ1MjMwMDEyM0Rlc0R5bWVudC51b290frWIOIws7wBO/Om3koQf
cmhc 71206... 038
...
2027004
326@qq.com

khi12@jinghu-moon-computer MINGW64 ~/.ssh
$
```

**复制下来**

## 3.3.2 复制密钥到 Github

进入 Github 网站，「Setting」→「SSH and GPG keys」→「New SSH key」，取个名字，粘贴密钥保存即可。

|                     |   |
|---------------------|---|
| Account settings    |   |
| Profile             |   |
| Account             |   |
| Appearance          |   |
| Accessibility       |   |
| Account security    |   |
| Billing & plans     |   |
| Security log        |   |
| Security & analysis |   |
| Sponsorship log     |   |
| Emails              |   |
| Notifications       |   |
| Scheduled reminders |   |
| SSH and GPG keys    | 1 |
| Repositories        |   |

## SSH keys

New SSH key 2

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

 **Blog**  
SHA256:P0VVDpYjP/Rg18XyaxB3GZ5sQd66L8XgRe03/SfZmro  
Added on 4 Jan 2022  
Never used — Read/write

SSH

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

## GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

## Vigilant mode

Flag unsigned commits as unverified

This will include any commit attributed to your account but not signed with your GPG or S/MIME key.  
Note that this will include your existing unsigned commits.

[Learn about vigilant mode.](#)

## 2.3.3 验证是否配置成功

1、打开 Git bash，输入以下命令。

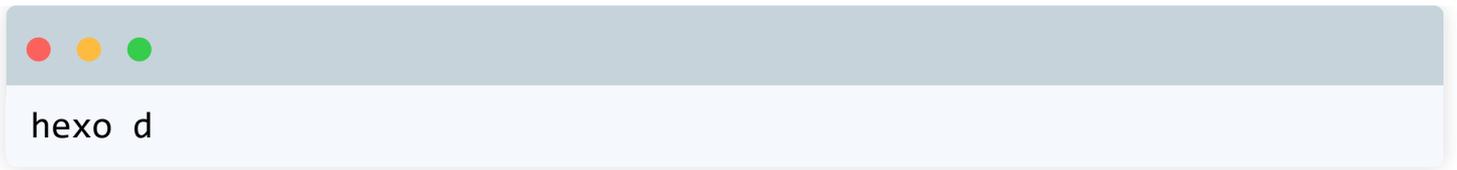
```
ssh -T git@github.com
```

显示 **successful** 表示配置成功!

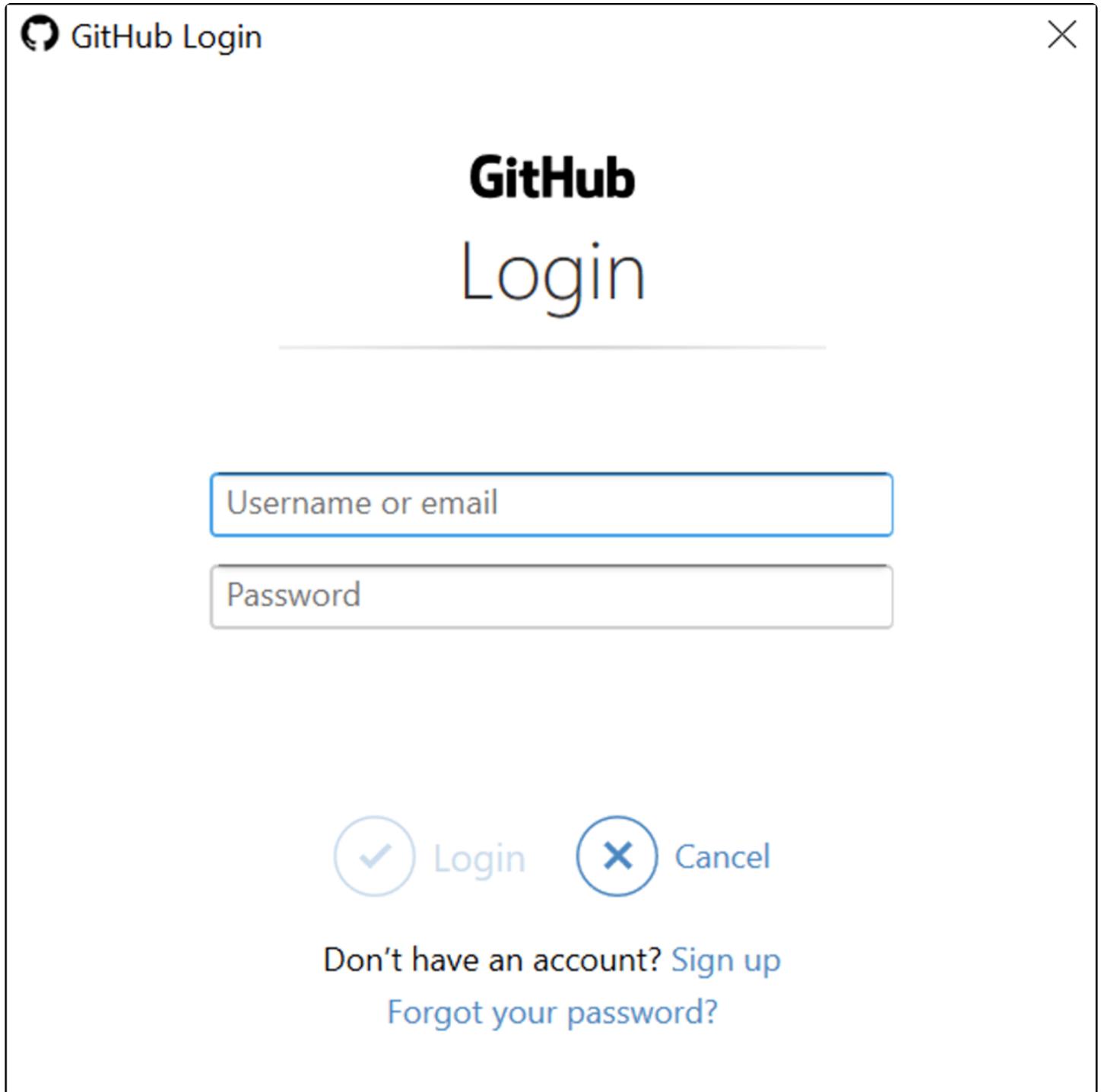
```
khi12@jinghu-moon-computer MINGW64 ~/.ssh  
$ ssh -T git@github.com  
Hi ziguangqing! You've successfully authenticated, but GitHub does not provide shell access.  
  
khi12@jinghu-moon-computer MINGW64 ~/.ssh  
$
```

## 3.4 GitHub 账户验证

在 **Blog** 文件目录下，打开 **Git Bash Here**。输入以下命令：



可能会出现以下页面，登录你的 GitHub 账户验证授权就行。



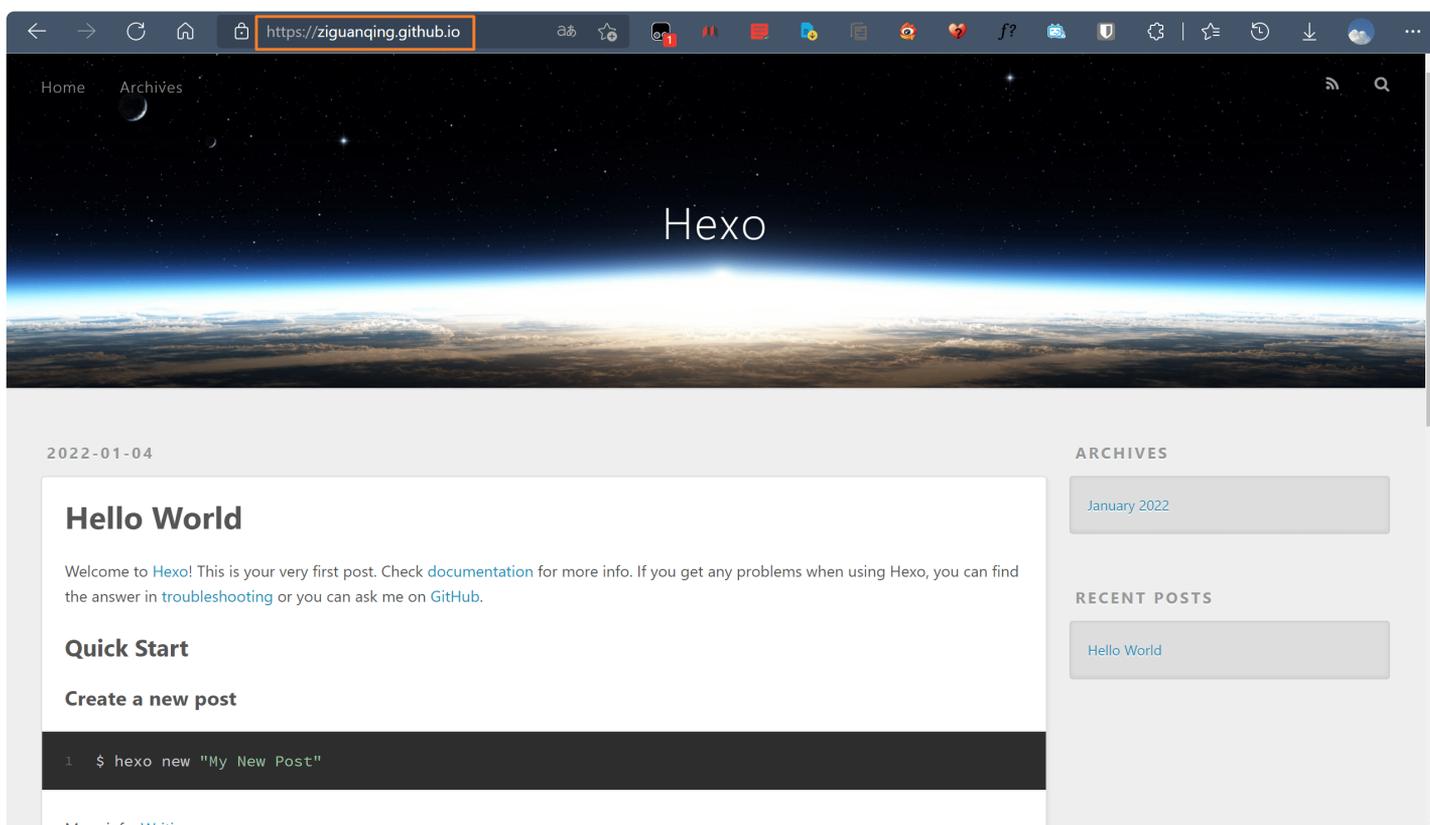
远程部署成功，界面如下。

```
MINGW64:/e/Blog
INFO  Generated: 2022/01/04/hello-world/index.html
INFO  Generated: css/fonts/fontawesome-webfont.svg
INFO  17 files generated in 446 ms
INFO  Deploying: git
INFO  Clearing .deploy_git folder...
INFO  Copying files from public folder...
INFO  Copying files from extend dirs...
[main 65f5dbc] site updated: 2022-01-06 11:53:58
 2 files changed, 2 insertions(+), 2 deletions(-)
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (8/8), 697 bytes | 348.00 KiB/s, done.
Total 8 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:ziguanqing/ziguanqing.github.io.git
 4cedf42..65f5dbc HEAD -> main
Branch 'main' set up to track remote branch 'main' from 'git@github.com:ziguanqing/ziguanqing.github.io.git'.
INFO  Deploy done: git

khi12@jinghu-moon-computer MINGW64 /e/Blog
$
```

## 3.5 远程访问博客

然后你就可以通过 GitHub 分配的域名访问你的博客了。如下：



# 第四章 网站配置详解

经过上面两章内容，一个网站框架就已经搭建好了。但是现在的博客网站就像是刚刚到手的新房，需要装修一番，才能成为真正的家！

## 4.1 博客文件夹

`themes`：主题文件夹

`package.json`：这是一些依赖的信息，json 格式，建议不要改动。

`scaffolds`：模板文件夹，新建文章的显示格式（下文的 Front-matter）都会以 scaffolds 中 post.md 的内容来加载。

`source`：这是一个资源文件夹。存放文章，CSS、JS、图片等各种页面。

`public`：公共文件夹，存放生成的站点文件,大部分来自 `source` 文件夹。

`node_modules`：Hexo 插件、依赖存放位置。

## 4.2 常用命令

参考资料：[指令 | Hexo](#) 

### 4.2.1 基本命令

| 命令                               | 简写                             | 功能   |
|----------------------------------|--------------------------------|--|
| <code>hexo clean</code>          | <code>hexo cl</code>           | 清除之前的缓存文件 ( <code>db.json</code> )、静态文件 ( <code>public</code> )。 |
| <code>hexo generate</code>       | <code>hexo g</code>            | 生成静态文件。  |
| <code>hexo server</code>         | <code>hexo s</code>            | 本地预览已经生成的静态文件。   |
| <code>hexo deploy</code>         | <code>hexo d</code>            | 部署已经生成的静态文件。   |
| <code>hexo new title</code>      | <code>hexo n xxx</code>        | 新建一篇文章，标题无空格。  |
| <code>hexo new 'xxx'</code>      | <code>hexo n 'xxx'</code>      | 新建一篇文章，标题有空格。  |
| <code>hexo new page 'xxx'</code> | <code>hexo n page 'xxx'</code> | 新建一个页面。  |
| <code>hexo version</code>        | 无                              | 显示 Hexo 版本。  |

## 4.2.2 推荐命令

### 1、远程部署

```
hexo cl && hexo g -d
```

### 2、本地预览

```
hexo clean && hexo g && hexo s
```

这两个动作我分享到了 [Quicker](#) ，有需要的朋友可以看下。

[一键向 Git Bash here 界面发送 Hexo 远端部署命令](#)

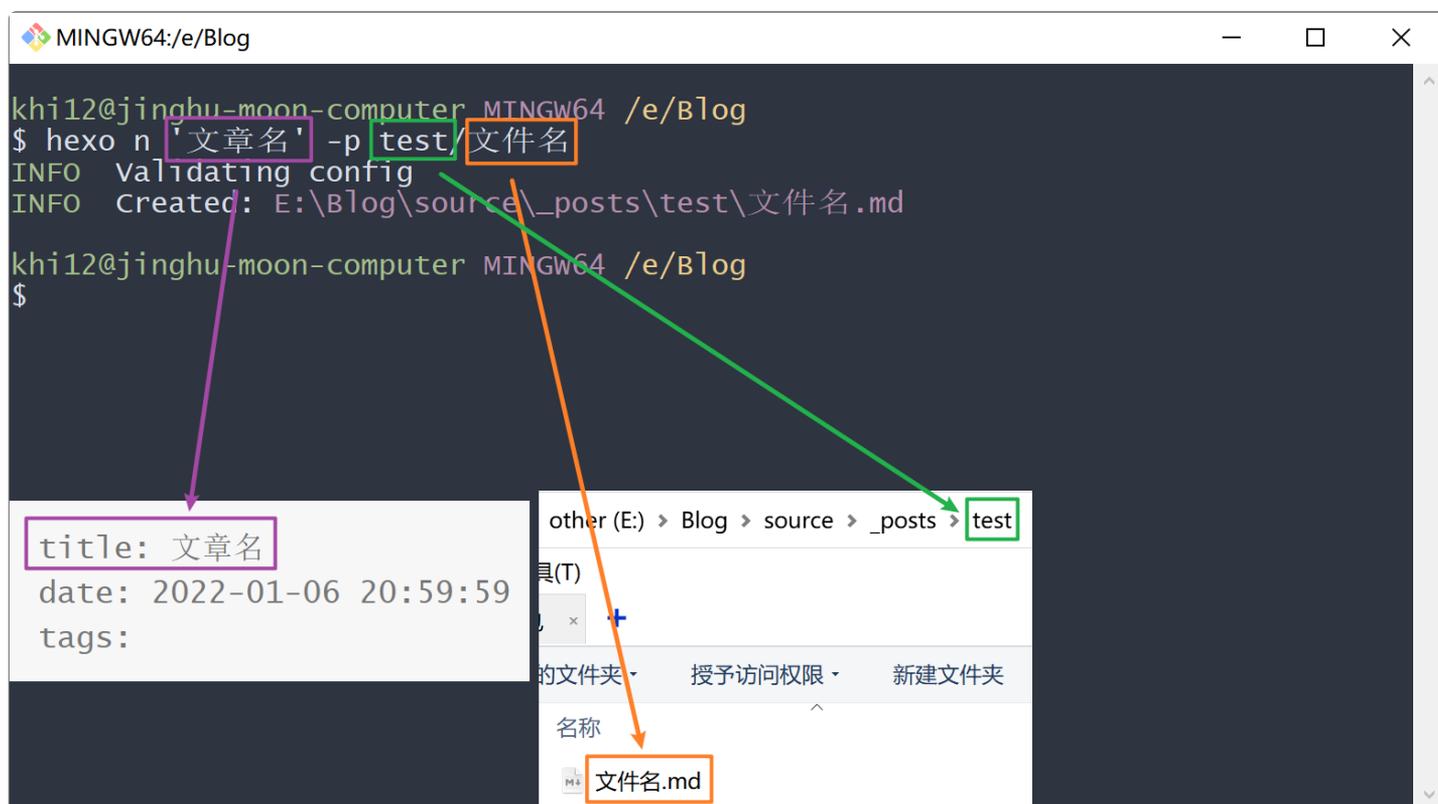
[一键向 Git Bash here 界面发送 Hexo 本地预览命令](#)

## 4.2.3 其他命令

### 1、创建文章，指定路径

命令如下：

```
hexo n '文章名' -p test/文件名
```



### 2、本地预览，指定端口

命令如下：

```
hexo s -p 5000
```

```
MINGW64:/e/Blog
khi12@jinghu-moon-computer MINGW64 /e/Blog
$ hexo s -p 5000
INFO validating config
INFO start processing
INFO Hexo is running at http://localhost:5000/. Press Ctrl+C to stop.
```

## 4.3 网站属性

| 参数          | 描述  |
|-------------|---|
| title       | 网站标题  |
| subtitle    | 网站副标题   |
| description | 网站描述  |
| keywords    | 网站的关键词。支持多个关键词。用英文逗号分开。   |
| author      | 您的名字  |
| language    | 网站使用的语言。对于简体中文用户来说，使用不同的主题可能需要设置成不同的值，请参考 <a href="#">你的主题文档</a> 自行设置。常见的有 zh-Hans 和 zh-CN。 |
| timezone    | 网站时区。Hexo 默认使用电脑时区。中国大陆地区使用 <code>Asia/Shanghai</code> 。                                    |

打开 `Blog` 目录下的 `_config.yml` 文件，修改如下部分：

E: > Blog >  \_config.yml

```
1 # Hexo Configuration
2 ## Docs: https://hexo.io/docs/configuration.html
3 ## Source: https://github.com/hexojs/hexo/
4
5 # Site
6 title: Hexo
7 subtitle: ''
8 description: ''
9 keywords:
10 author: John Doe
11 language: en
12 timezone: ''
13
14 # URL
15 ## Set your site url here. For example, if you use Github
16 url: http://ziguanqing.github.io
```

← 修改这个部分

改为你的网站地址

示例:

```
YAML
# Site
title: 'Guan Qing 的博客'
subtitle: '记录生活'
description: '仰天长啸出门去，我辈岂是蓬蒿人'
keywords: '观清,博客,Blog,Hexo'
author: '观清'
language: en # 根据主题要求更改，一般为 zh-CN、zh-hans。
timezone: 'Asia/Shanghai'
```

```
# Site
title: 'Guan Qing 的博客'
subtitle: '记录生活'
description: '仰天长啸出门去，我辈岂是蓬蒿人'
keywords: '观清, 博客, Blog, Hexo'
author: '观清'
language: en
timezone: 'Asia/Shanghai'

# URL
## Set your site url here. For example, if you use GitHub
url: http://ziguanqing.github.io
```

上面的参考链接里还有更多关于 Hexo 的配置，这里暂时不讲。想了解的，请自行查看。**参考资料：** [配置 | Hexo](#) 

## 4.4 发布文章

---

### 1、Front-matter

在基本命令部分，我已经讲了如何创建文章，但是过于简略，这里详细讲讲。下面是一篇文章应该具备的 Front-matter 参数。

| 参数         | 描述         | 默认值       |
|------------|------------|-----------|
| layout【可选】 | 布局         | post      |
| title【必需】  | 标题         | 文章的文件名    |
| date【必需】   | 建立日期       | 文件建立日期    |
| updated    | 更新日期       | 文件更新日期    |
| comments   | 开启文章的评论功能  | true      |
| tags       | 标签（不适用于分页） |           |
| categories | 分类（不适用于分页） |           |
| permalink  | 文章的网络链接    | 年/月/日/文章名 |
| excerpt    | 文章的简短描述    |           |

- layout

- 有三种布局方式：page、post、draft。
- page：新建页面。
- draft：新建草稿，命令：`hexo n draft 'xxx'`。会在 `source` 文件夹生成 `_drafts` 文件夹，存放草稿。在本地预览、远程部署，都不显示。文章移动到 `_post` 目录，才会显示。

- tags、categories：在下文的 `scaffolds` 文件夹讲如何使用。

- permalink：在下文的 `scaffolds` 文件夹讲如何使用。

## 2、scaffolds

`scaffolds` 文件夹里有三个文件：draft.md、page.md、post.md。我们主要使用 post.md。

post.md 打开如下：

```
title: {{ title }}
date: {{ date }}
tags:|
```

改为右图

```
title: {{ title }}
date: {{ date }}
author:
summary:
categories:
- ''
tags:
- ''
excerpt: ''
```

- categories
  - 父子分类: [xx,xx]
  - 并列分类: - 'xx' , 换行下一个。
- excerpt: 文章的简短描述, 显示在首页。

使用效果如下图:

```
title: 1234
categories:
- [A1,a1]
- A2
tags:
- AA
- BB
excerpt: 这是一个测试文档!
date: 2022-01-19 21:24:15
```

## 1234

这是一个测试文档!

📅 2022-01-19 🏷️ A1 a1 A2 🔖 AA BB

A1 ▾

a1 ▾

1234

A2 ▾

1234

📄 1

📄 1

📄 1

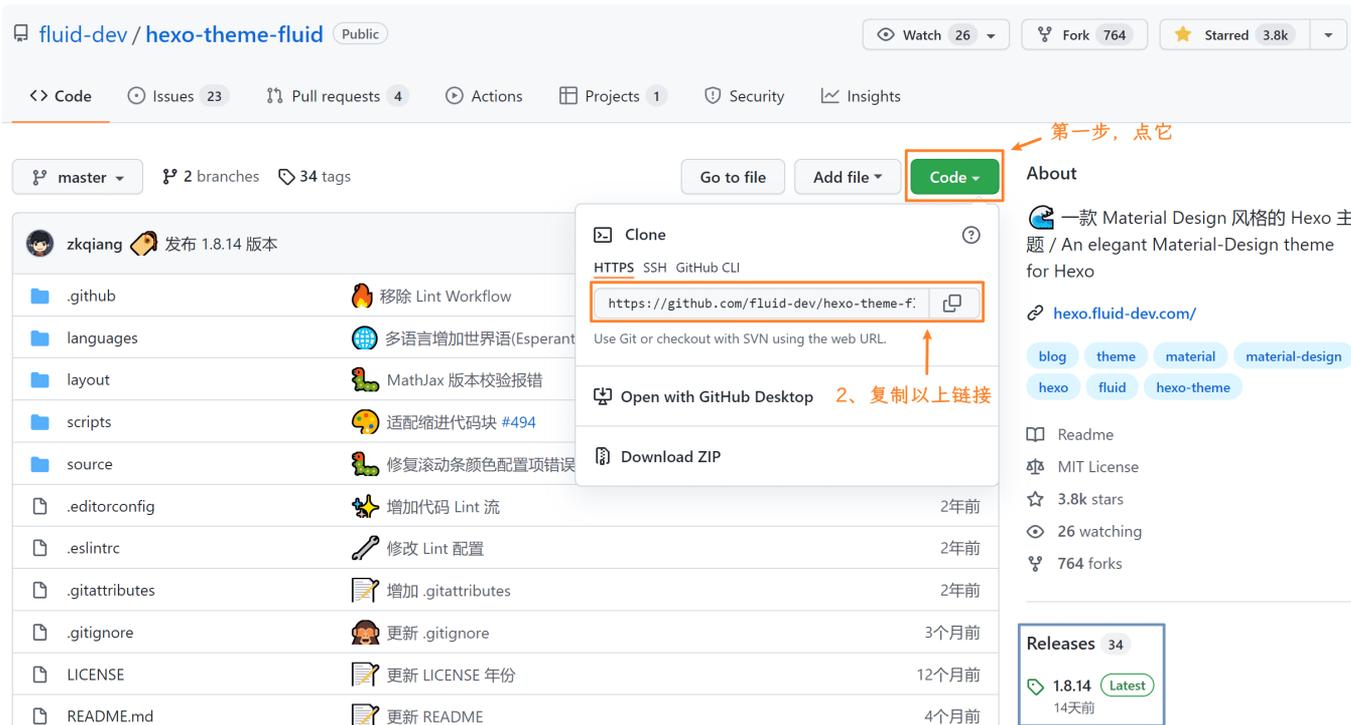
# 第五章 主题美化

上一章属于家装中的软装，让你对 `Blog` 文件夹内的文件和发布文章有了些许了解。这一章开始主题美化，属于家装中的硬装。

## 5.1 主题下载

以 Fluid 主题举例。

01. 进入 GitHub 官网，搜索 `hexo fluid`。点击第一个搜索结果。进入以下界面：



02. 下载主题，有两种方式。

A. 点击 `code` 绿框，复制第一个链接。然后任选一个文件夹，打开 Git Bash here，输入以下命令：

```
git clone https://github.com/fluid-dev/hexo-theme-fluid.git
```

回车，开始下载主题。用时可能比较久，根据你的网速而定。成功下载，界面如下：

```
MINGW64:/d/06~person_folders/Me
khi12@jinghu-moon-computer MINGW64 /d/06~person_folders/Me
$ git clone https://github.com/fluid-dev/hexo-theme-fluid.git
Cloning into 'hexo-theme-fluid'...
remote: Enumerating objects: 6216, done.
remote: Counting objects: 100% (653/653), done.
remote: Compressing objects: 100% (327/327), done.
remote: Total 6216 (delta 401), reused 531 (delta 319), pack-reused 5563
Receiving objects: 100% (6216/6216), 7.68 MiB | 6.98 MiB/s, done.
Resolving deltas: 100% (3960/3960), done.

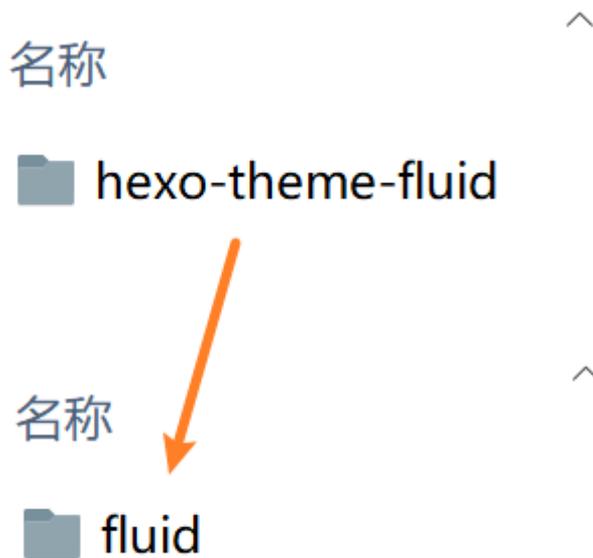
khi12@jinghu-moon-computer MINGW64 /d/06~person_folders/Me
$
```

B. 点击右下方的 **Releases**，你会看到如下界面。往下滑，点击 **Source code (zip)** 选项，开始下载主题。

**注意：**可能作者没有发布 **Releases** 版本，此时使用方法一。



03. 主题下载好后，改名。一般是去掉 **hexo-theme-**。例如：

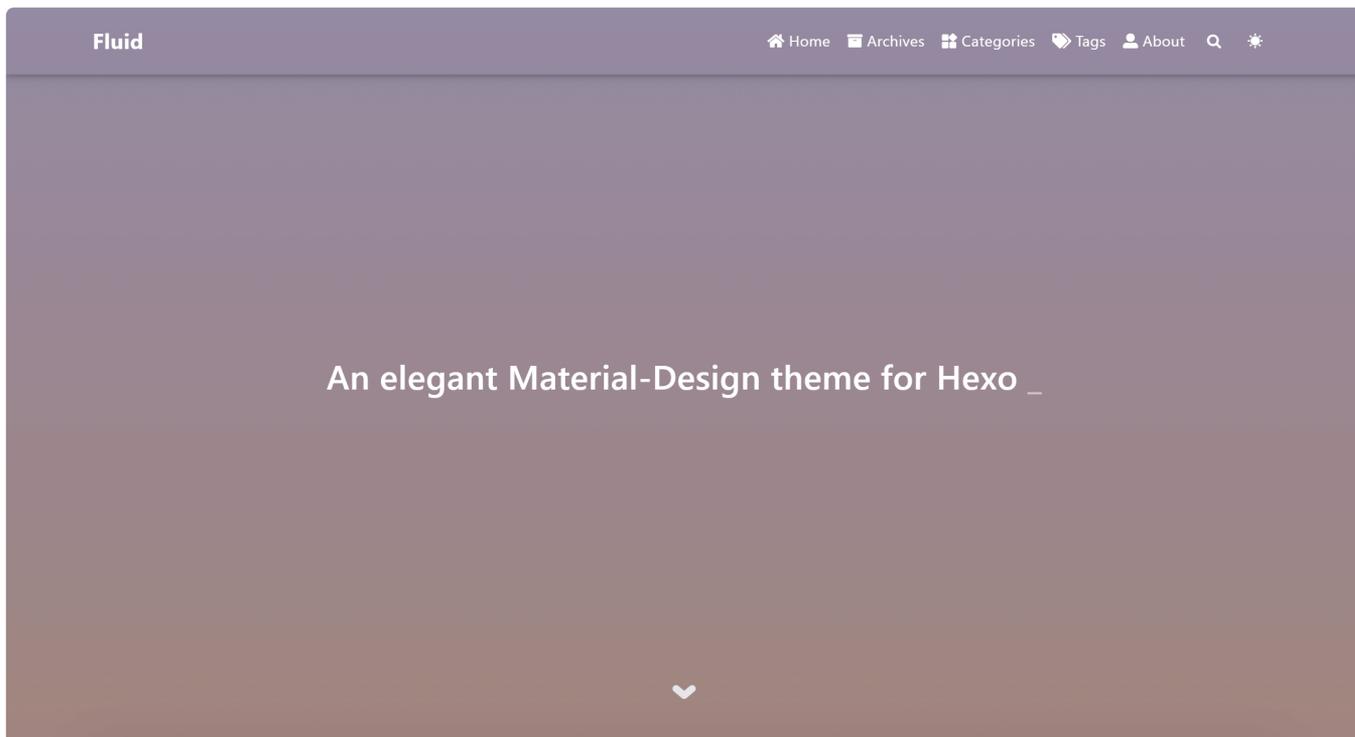


放入 `Blog` 文件夹的 `themes` 文件夹。同时在 `_config.yml` 中，把主题改为 `fluid`。

```
# Extensions
## Plugins: https://hexo.io/plugins/
## Themes: https://hexo.io/themes/
theme: fluid

# Deployment
## Docs: https://hexo.io/docs/one-command-deployment
deploy:
  type: git
  repo: git@github.com:ziguanqing/ziguanqing.github.io.git
  branch: main
```

04. 重新部署下，你会发现你的博客发生了改变。



## 5.2 主题推荐

进入 GitHub 官网，搜索 `hexo themes`。sort 选择 `Most stars`。开始选择心仪的主题。

我推荐的主题有三个要求：

- 1、有详细的中文配置文档。
- 2、功能全面，尽量不需要自己写代码。
- 3、更新较频繁。

以下为推荐主题表格，排序规则：`star` 数由多到少。当然还有很多主题由于时间原因，我没有推荐，见谅。

| 序号 | 主题名       | GitHub 链接                            | 主题文档                           |
|----|-----------|--------------------------------------|--------------------------------|
| 01 | NexT      | <a href="#">hexo-theme-next</a>      | <a href="#">NexT 使用文档</a>      |
| 02 | Butterfly | <a href="#">hexo-theme-butterfly</a> | <a href="#">Butterfly 使用文档</a> |
| 03 | Material  | <a href="#">hexo-theme-material</a>  | <a href="#">Material 使用文档</a>  |
| 04 | matery    | <a href="#">hexo-theme-matery</a>    | <a href="#">matery 使用文档</a>    |
| 05 | Fluid     | <a href="#">hexo-theme-fluid</a>     | <a href="#">Fluid 使用文档</a>     |
| 06 | Volantis  | <a href="#">hexo-theme-volantis</a>  | <a href="#">Volantis 使用文档</a>  |
| 07 | archer    | <a href="#">hexo-theme-archer</a>    | <a href="#">archer 使用文档</a>    |
| 08 | Ayer      | <a href="#">hexo-theme-ayer</a>      | <a href="#">Ayer 使用文档</a>      |
| 09 | Fexo      | <a href="#">fexo</a>                 | <a href="#">Fexo 使用文档</a>      |
| 10 | Yun       | <a href="#">hexo-theme-yun</a>       | <a href="#">Yun 使用文档</a>       |
| 12 | pure      | <a href="#">hexo-theme-pure</a>      | <a href="#">pure 使用文档</a>      |
| 13 | Keep      | <a href="#">hexo-theme-keep</a>      | <a href="#">Keep 使用文档</a>      |
| 14 | Anisina   | <a href="#">hexo-theme-Anisina</a>   | <a href="#">Anisina 使用文档</a>   |
| 15 | SPFK      | <a href="#">hexo-theme-spfk</a>      | <a href="#">SPFK 使用文档</a>      |
| 16 | Stun      | <a href="#">hexo-theme-stun</a>      | <a href="#">Stun 使用文档</a>      |
| 17 | Cards     | <a href="#">hexo-theme-cards</a>     | <a href="#">Cards 使用文档</a>     |

| 序号 | 主题名           | GitHub 链接                                     | 主题文档                               |
|----|---------------|---|------------------------------------|
| 18 | Aurora        | <a href="#">hexo-theme-aurora</a>             | <a href="#">Aurora 使用文档</a>        |
| 19 | Aomori        | <a href="#">hexo-theme-aomori</a>             | <a href="#">Aomori 使用文档</a>        |
| 20 | ZHAOO         | <a href="#">hexo-theme-zhaoo</a>              | <a href="#">ZHAOO 使用文档</a>         |
| 21 | yet-the-books | <a href="#">hexo-theme-yet-the-books</a>      | <a href="#">yet-the-books 使用文档</a> |
| 22 | Miracle       | <a href="#">hifun-team/hexo-theme-miracle</a> | <a href="#">Miracle 使用文档</a>       |

以上主题，我比较推荐 NexT、Butterfly、mtery、Fluid。这几个主题，使用的人比较多，遇到问题。网上搜到解决方法的可能性比较大。且网上关于这几个主题的魔改教程也很多，很适合小白。Aurora 主题很炫酷，可以试试。yet-the-books 主题是我见过最特别的主题，值得一试。

## 5.3 主题配置

我的博客网站使用的是 Fluid。个人认为 Fluid 的主题配置文档非常详细。但是编写者从业多年，他认为的小白用户可能与现实中的小白有所出入。所以下面，我会按照 Fluid 使用文档，讲解一些我认为比较难理解和操作的部分。

我不会把 Fluid 使用文档的所有内容都讲一遍。如果你在后续的主题美化过程中，发现你的博客界面和我的不一样。请查看 Fluid 使用文档。

参考文档：[Hexo Theme Fluid \(fluid-dev.com\)](#) 、[Hexo Fluid 用户手册 \(fluid-dev.com\)](#)

修改主题配置时，建议使用本地预览。若修改了 `_config.fluid.yml`、`_config.yml`，须关闭本地服务，再次预览，才能看到效果。

## 5.3.1 图床

在**页面顶部大图**部分，提到了图片外链链接。如果博客的图片文件过多，可能会减慢 Hexo 生成静态文件的速度。所以，你可以试试图床。可以看看这个软件：[PicGo](#)。支持多种图床，可以自动复制链接，支持插件系统。功能十分强大。

我用的是 gitee 作为图床。PicGo 没有内置 gitee 图床，需要下载插件 `gitee-smart 1.1.7`。

### 配置方法

01. 注册 gitee，并创建仓库（勾选公开）。
02. 左上角设置 → 下滑，找到私人令牌 → 创建新令牌。
03. 配置 gitee 插件。如下：



04. 配置好，测试上传图片。成功上传，PicGo 会有提示弹出。

注意：gitee 对图片大小有限制。图片大小超过 1MB，无法在 gitee 以外访问。如果开启了『科学上网』，图片可能无法加载。

所以超过 1MB 的图片，我建议使用微博图床。下载地址：[新浪微博图床 V2.9.4 - 极简插件](#)。使用时需要登录微博。

## 安装扩展

下载后的压缩包解压，**crx** 文件，直接拖入如下页面：



再推荐几个图床网站：

- [ImgURL 免费图床](#)
- [IMGBED \(图床\) - 公共图床, 免费图片上传, 专业快速分享图片外链](#)
- [免费图床 - 不死鸟 - 分享为王官网 \(iao.su\)](#)
- [图床导航 - 稳定, 速度快, 长期运营的图床推荐 \(imgdh.com\)](#)
- 还有一个：**百度百家号**，访问快。可以试试。[百家号 \(baidu.com\)](#)

目前，我的选择：主用 gitee，大图片用微博图床。

下面推荐一些图片处理网站。

- 图片压缩
  - [在线图片压缩 - docsmall 在线图片压缩工具, 在线图片压缩软件](#)，推荐。

- [Picdiet - 极速在线压缩80%图片体积](#) ，访问有点慢。
- 图片放大
  - [AI Image Upscaler - Upscale Photo, Cartoons in Batch Free \(imgupscaler.com\)](#) 
  - [waifu2x \(udp.jp\)](#) 
  - [Bigjpg - AI人工智能图片无损放大](#) ，国内网站，访问快。
- 图片编辑
  - [iLoveIMG | 图像文件在线编辑工具](#) 

## 5.3.2 开启二级菜单

在**导航菜单**一节，提到了开启二级菜单。在 `_config.fluid.yml` 文件添加如下代码：

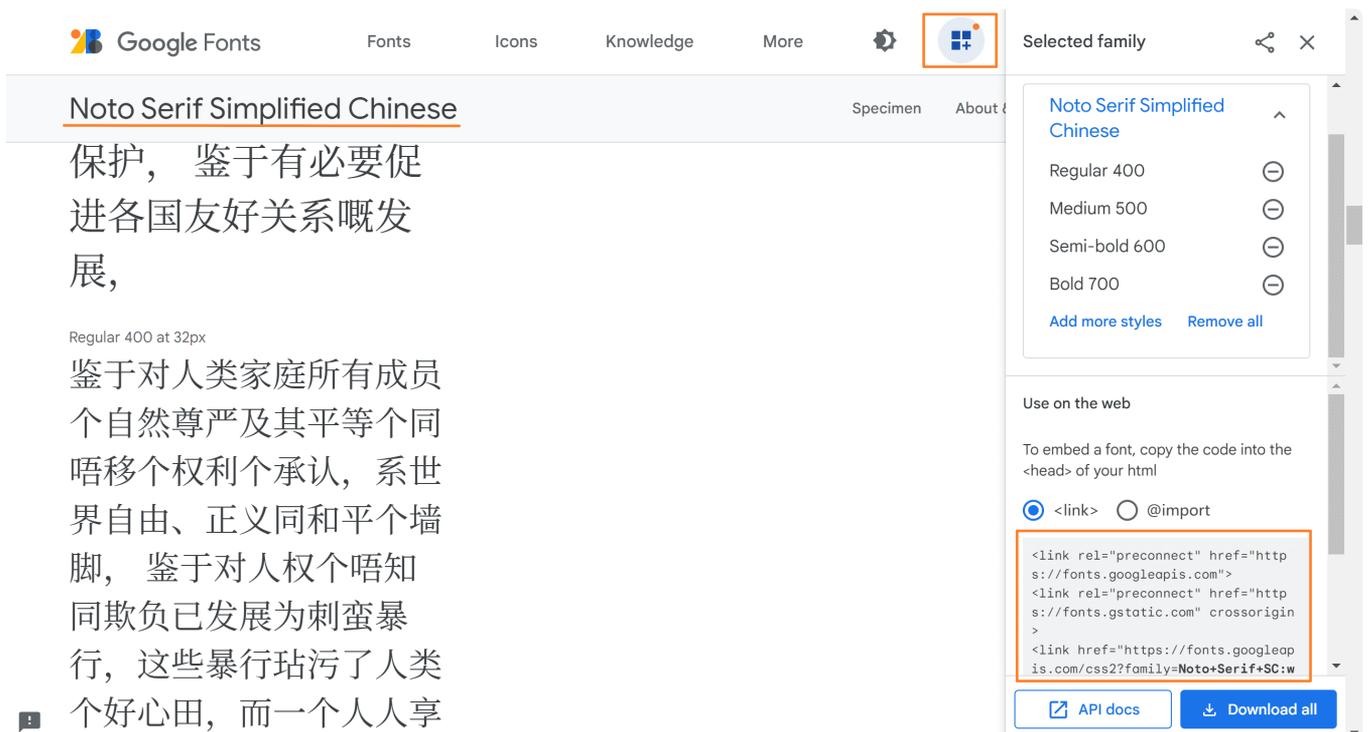
```
YAML
# 导航栏菜单，可自行增减，key 用来关联 languages/*.yml，如不存在关联则
# 显示 key 本身的值；icon 是 css class，可以省略；增加 name 可以强制显示指
# 定名称
menu:
  - { key: "home", link: "/", icon: "iconfont icon-home-fill" }
  - {
    key: '文章',
    icon: 'iconfont icon-books',
    submenu: [
      { key: "archive", link: "/archives/", icon: "iconfont icon-
archive-fill" },
      { key: "category", link: "/categories/", icon: "iconfont
icon-category-fill" },
      { key: "tag", link: "/tags/", icon: "iconfont icon-
biaoqian1" }
    ]
  }
  - { key: "about", link: "/about/", icon: "iconfont icon-user-
fill" }
```

```
- { key: "links", link: "/links/", icon: "iconfont icon-link-fill" }
```

## 5.3.3 全局字体

如果你觉得网页默认字体不好看，可以试试引入网络字体。

01. 进入 [Browse Fonts - Google Fonts](#) ，选择好看的字体。注意访问需要「科学上网」。配置好后，访问博客不需要「科学上网」，也可以显示自定义字体。
02. 选择好看的字体，比如 Noto Serif Simplified Chinese。
03. 点击该字体，进入具体展示界面，往下滑。点击 **+ Select this style** ，选择你喜欢的字重。
04. 点击右上角 **+** 号。弹出如下界面：



复制右下角框框里的链接：

```
<link href="https://fonts.googleapis.com/css2?family=Noto+Serif+SC:wght@400;500;600;700&display=swap" rel="stylesheet">
```

## 05. 第一种方法（不推荐）

在目录：E:\Blog\themes\fluid\layout\\_partial，打开 `head.ejs`。添加以上链接，如下：

```
E:\Blog > themes > fluid > layout > _partial > <% head.ejs > <? > <? > head > <? link
1 <%
2 var separator = theme.title_join_string || theme.tab_title_separator
3 var title = page.title ? [page.title, config.title].join(separator) : config.title
4 var keywords = page.keywords || config.keywords
5 if (keywords instanceof Array) {
6   keywords = keywords.join(',')
7 }
8 var description = page.description || page.excerpt || (is_post() && page.content) || config.description
9 if (description) {
10  description = escape_html(strip_html(description).substring(0, 200).trim()).replace(/\n/g, ' ')
11 }
12 var ogImage = page.og_img || page.index_img
13 var ogConfig = Object.assign({ image: ogImage && url_for(ogImage) }, theme.open_graph)
14 %>
15
16 <head>
17 <meta charset="UTF-8">
18 <link href="https://fonts.googleapis.com/css2?family=Noto+Serif+SC:wght@400;500;600;700&display=swap" rel="stylesheet">
19 <link rel="apple-touch-icon" sizes="76x76" href="<%= url_for(theme.apple_touch_icon) %>">
20 <link rel="icon" href="<%= url_for(theme.favicon) %>">
21 <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=5.0, shrink-to-fit=no">
```

在 `_config.fluid.yml` 中，修改如下：

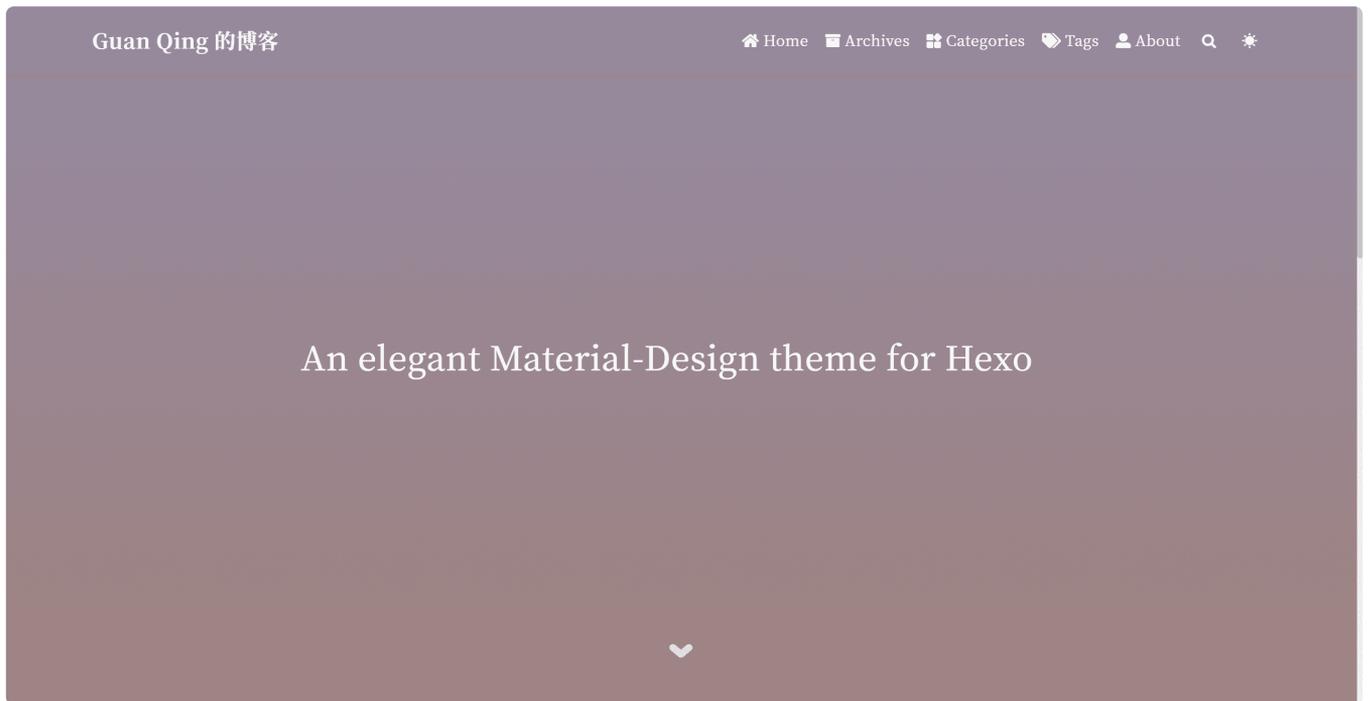
```
# 主题字体配置
# Font
font:
  font_size: 16px
  font_family: Noto Serif SC
  letter_spacing: 0.02em
  code_font_size: 85%
```

## 06. 第二种方法

- 在 `_config.fluid.yml` 中，修改如下：

```
E: > Blog > _config.fluid.yml
197 font_family: Noto Serif SC
198 letter_spacing: 0.02em
199 code_font_size: 85%
200
201 # 指定自定义 .js 文件路径, 支持列表: 路径是相对 source 目录, 如 /js/custom.js 对应存放目录 source/js/custom.js
202 # Specify the path of your custom js file, support List. The path is relative to the source directory, such as `js/
  custom.js` corresponding to the directory `source/js/custom.js`
203 custom_js:
204
205 # 指定自定义 .css 文件路径, 用法和 custom_js 相同
206 # The usage is the same as custom_js
207 custom_css:
208
209 # 自定义 <head> 节点中的 HTML 内容
210 # Customize <head> HTML content
211 custom_head: '<link href="https://fonts.googleapis.com/css2?family=Noto+Serif+SC:wght@400;500;600;700&display=swap"
  rel="stylesheet">'
```

## 07. 重新本地预览。博客界面字体已经改变。



## 补充

如果你觉得使用 Google Fonts, 降低了网站打开速度。可以使用镜像源。

- [fonts.googleapis.com](https://fonts.googleapis.com) -> [fonts.lug.ustc.edu.cn](https://fonts.lug.ustc.edu.cn)
- [fonts.googleapis.com](https://fonts.googleapis.com) -> [fonts.loli.net](https://fonts.loli.net)
- 第二个速度要好点。

## 参考

- [科大博客提供-Google-Fonts-加速 - 掘金 \(juejin.cn\)](https://juejin.cn)
- [Google Fonts字体国内镜像加速 \(支持https\) - Pop's Blog \(429006.com\)](https://429006.com)



## 5.3.4 展示 PV 与 UV 统计

介绍如何使用 [LeanCloud](#) 展示 PV 与 UV 统计。

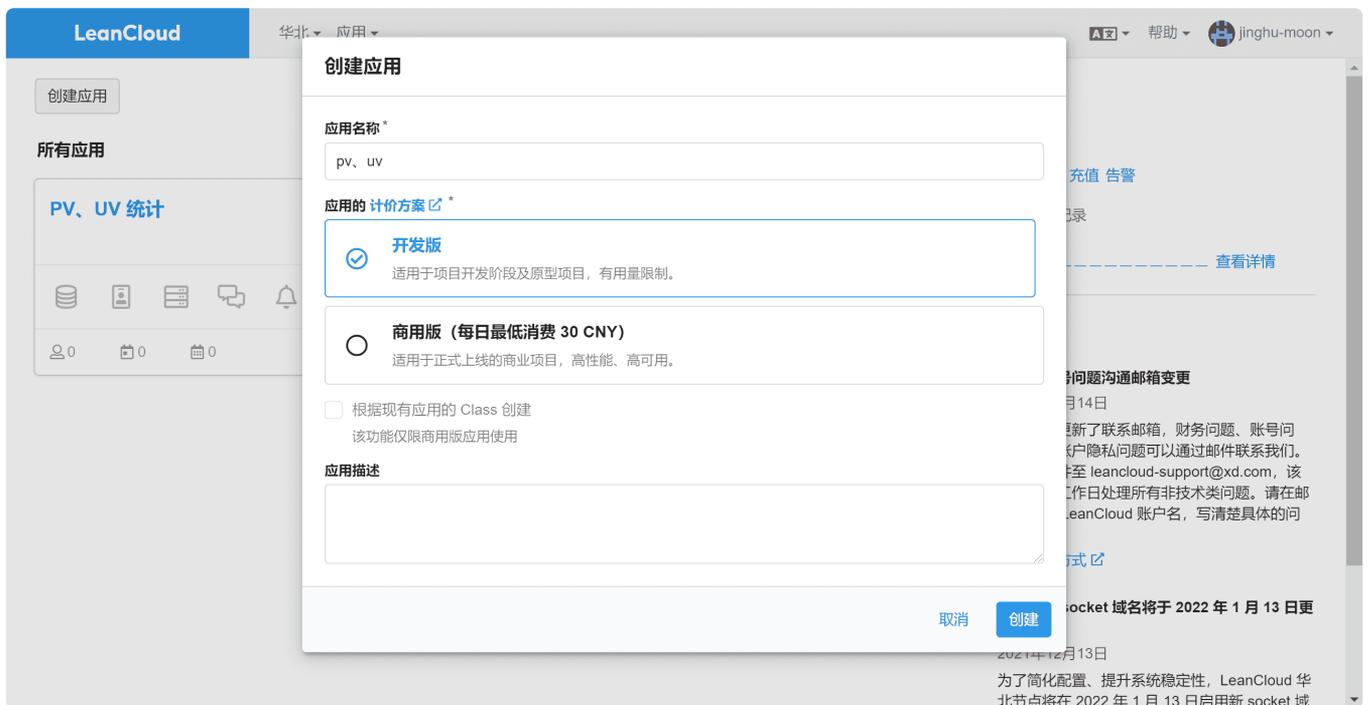
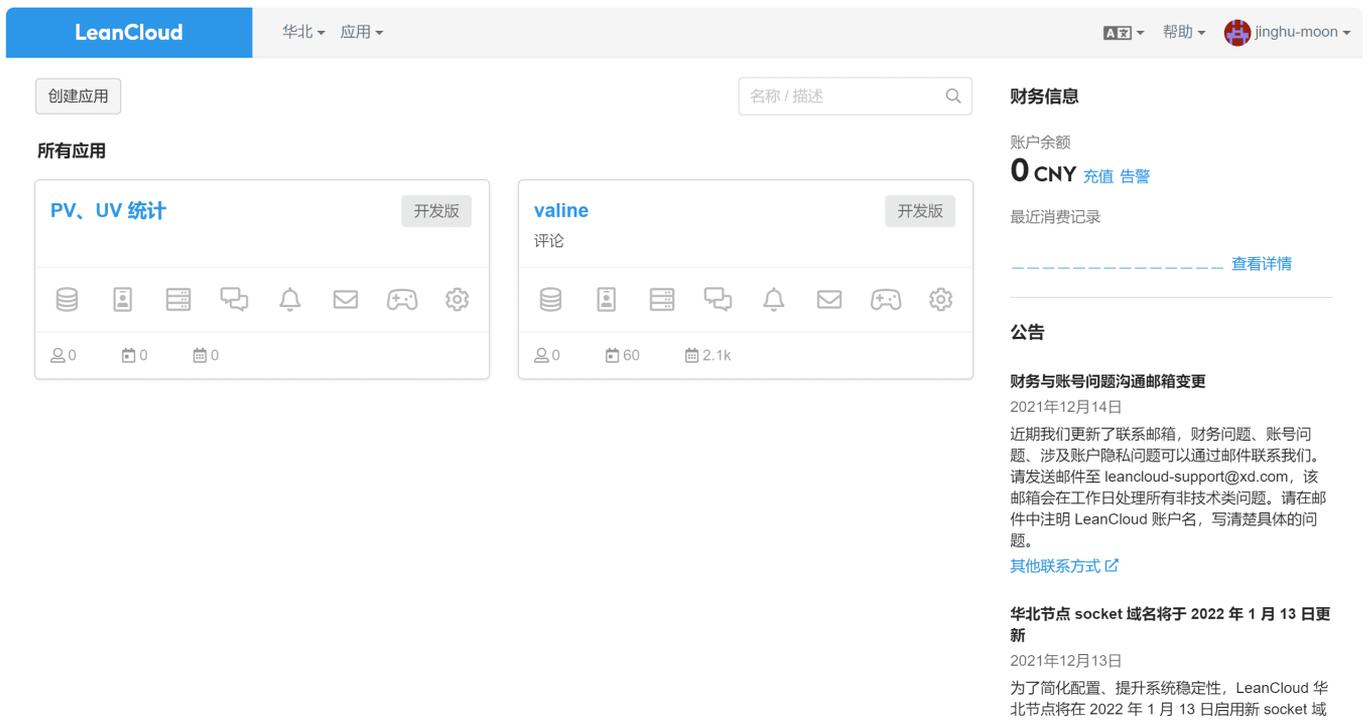
01. 注册 LeanCloud。注册后，进入如下界面：



进行实名认证（国内版需要）。不实名认证，无法进行后续操作。实名认证内容

02. 实名认证内容：姓名 + 身份证。

03. 进入如下界面，创建应用。



04. 点击 **小齿轮** ，进入设置界面。点击 **应用凭证** ，如下：



05. 复制 `AppID`、`AppKey`、`REST API 服务器地址`。在 `_config.fluid.yml` 中，修改如下：

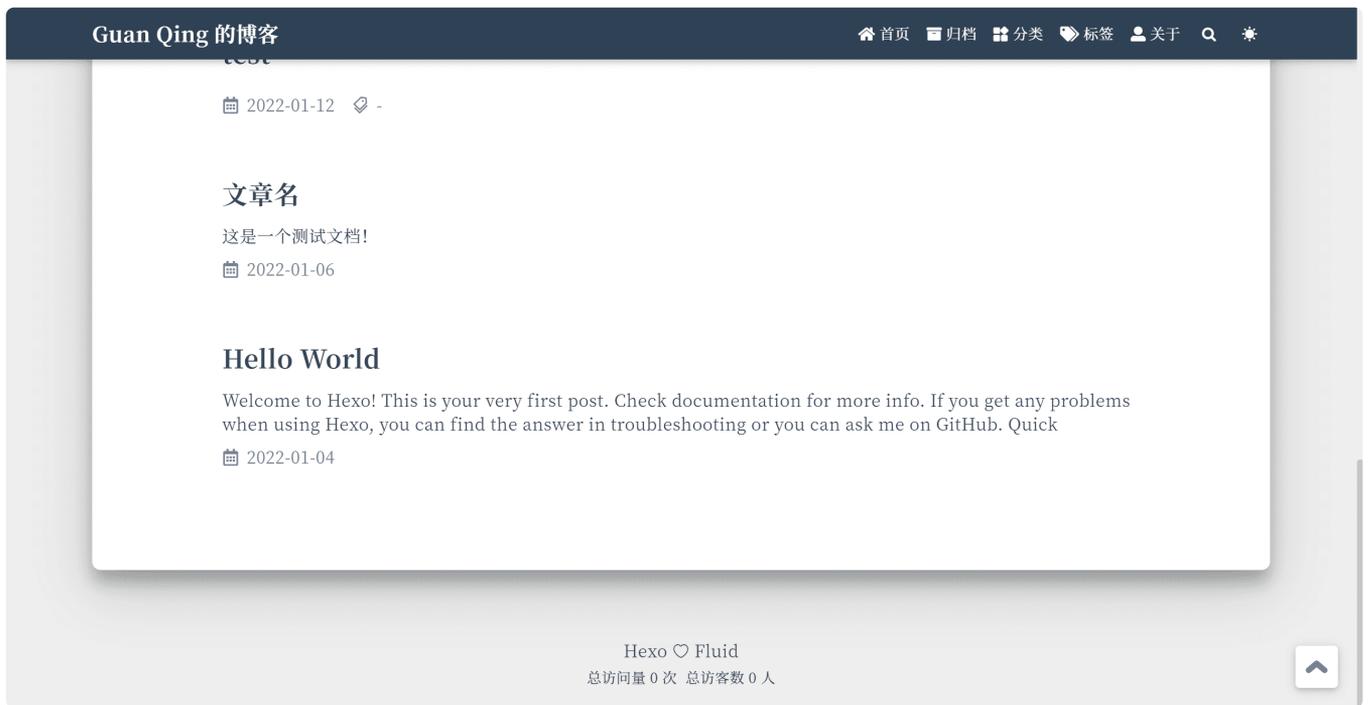
```
leancloud:
  app_id: Hj9XxRfFahtxNYYkBG87h1Lb-gzGzoHsz           AppID
  app_key: hp4zQdF2mPCqhbRM9b30412i                 AppKey
  # REST API 服务器地址，国际版不填
  # Only the Chinese mainland users need to set
  server_url: https://hj9xxrff.lc-cn-n1-shared.com     REST API 服务器地址
  # 统计页面时获取路径的属性
  # Get the attribute of the page path during statistics
  path: window.location.pathname
  # 开启后不统计本地路径(localhost 与 127.0.0.1)
  # If true, ignore localhost & 127.0.0.1
  ignore_local: true
```

```
# 展示网站的 PV、UV 统计数
# Display website PV and UV statistics
statistics:
  enable: true

# 统计数据来源，使用 leancloud 需要设置 `web_analytics: leancloud` 中的参数；使用 busuanzi 不需要额外设置，但是有时不稳定，另外本地运行时 busuanzi 显示统计数据很大属于正常现象，部署后会正常
# Data source. If use LeanCloud, you need to set the parameter in `web_analytics: LeanCloud`
# Options: busuanzi | LeanCloud
source: "leancloud"

# 页面显示的文本，{} 是数字的占位符 (必须包含)，下同
# Displayed text, {} is a placeholder for numbers (must be included), the same below
pv_format: "总访问量 {} 次"
uv_format: "总访客数 {} 人"
```

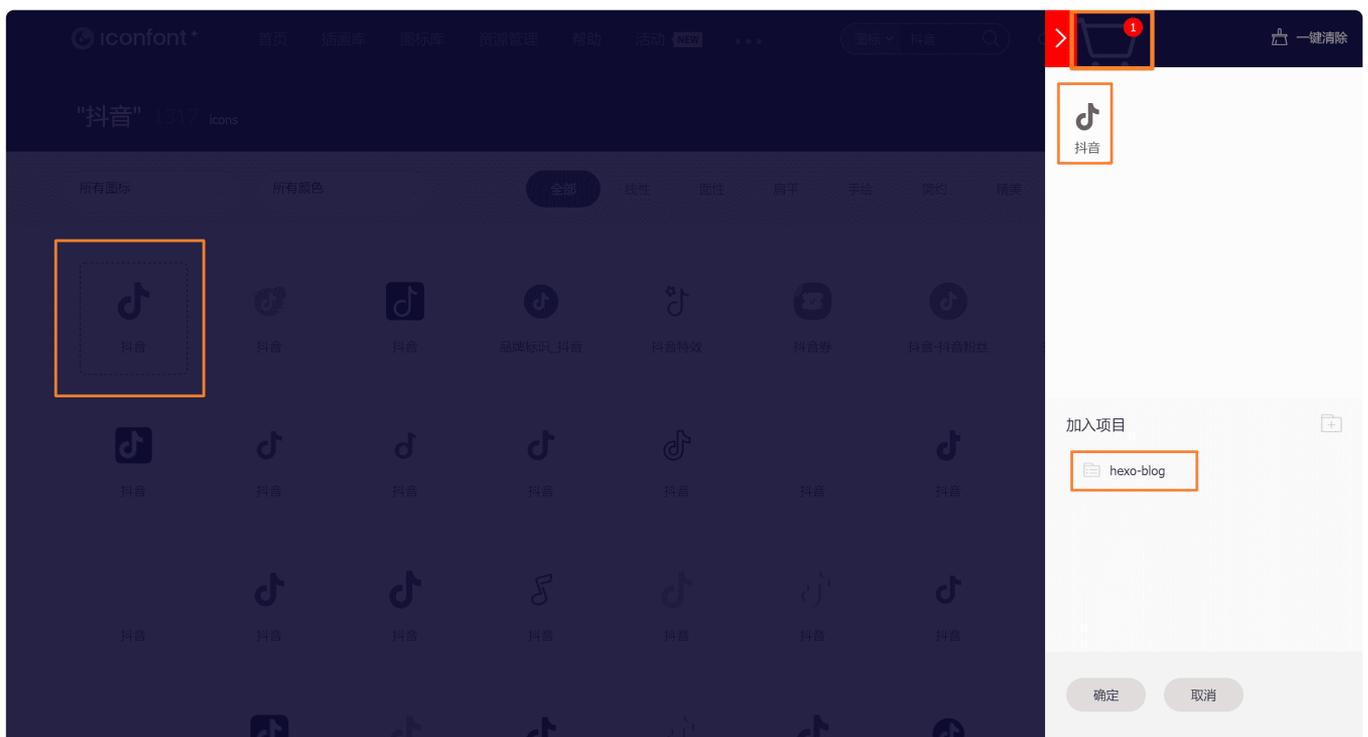
06. 重新本地预览。博客界面底部已经改变。



### 5.3.3 自定义社交图标

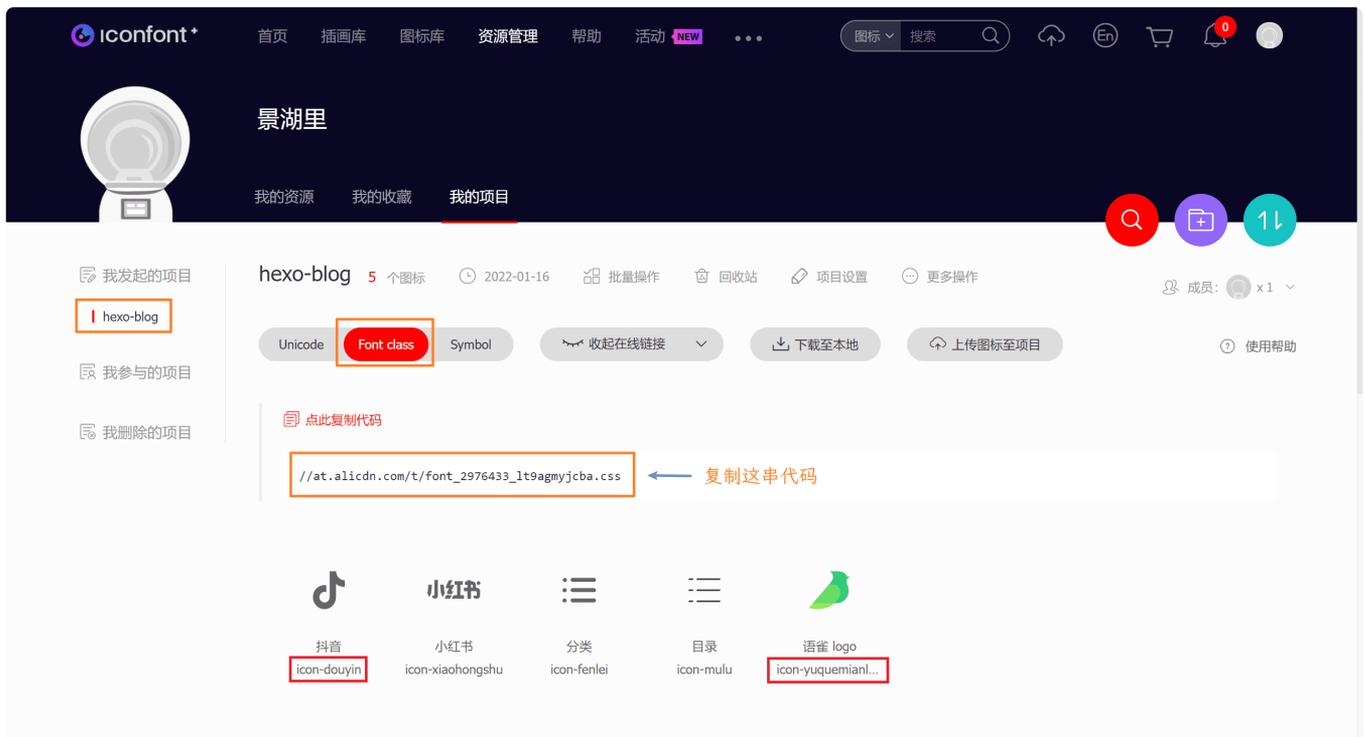
一句话，使用 [iconfont](#) 的图标，通过 CSS 样式文件引入，实现自定义社交图标。下面讲具体实现步骤。

01. 进入 [iconfont](#)，选择图标。
02. 鼠标悬浮到任意图标，点击第一个「添加入库」按钮。
03. 再点击右上角的购物车图标，弹出如下界面：



04. 新建项目，将图标加入项目。

05. 进入个人主页，操作如下：



06. 在 `_config.fluid.yml` 中，修改如下：

```
custom_css:
- //at.alicdn.com/t/font_2976433_lt9agmyjcba.css

icons:
- { class: "iconfont icon-github-fill", link: "https://github.com", tip: "GitHub" }
- { class: "iconfont icon-douban-fill", link: "https://douban.com", tip: "豆瓣" }
- { class: "iconfont icon-yuquemianlogo", link: "https://www.yuque.com/jinghuli", tip: "语雀", }
- { class: "iconfont icon-douyin", link: "https://www.douyin.com", tip: "豆瓣" }
- { class: "iconfont icon-wechat-fill", qrcode: "/img/favicon.png" }
- { class: "iconfont icon-qq-fill", qrcode: "/img/favicon.png" }
```

图标添加方式：`iconfont xxx-xxx`，`xxx-xxx` 为图标名。

07. 重新本地预览。博客关于页已经改变。

关于



Fluid

An elegant theme for Hexo



## 5.3.5 首页展示一言 / 今日诗词

法一

`config.yml` 文件配置如下:

```

slogan:
  enable: true

# 为空则按 hexo config.subtitle 显示
# If empty, text based on `subtitle` in hexo config
text: "仰天长啸出门去，我辈岂是蓬蒿人"
# hitokoto: false # 首页显示一言
# jinrishici: false # 首页显示今日诗词

# =====
api:
  enable: true
  url: "https://v1.hitokoto.cn/"
  method: "GET"
  headers: {}
  keys: ["hitokoto"]

# enable: true
# url: "https://v1.jinrishici.com/all.json"
# method: "GET"
# headers: {}
# keys: ["content"]
# =====

```

两个 api 任选一个开启，另一个必须关闭。

这个方法有缺陷，不能显示作者、出处，而且 Hitokoto 不能指定句子的类型。但是用起来很简单。

参考文章：[Hexo全过程记录 从Centos到Fluid - IRIDIUM-SUB - 博客园 \(cnblogs.com\)](https://cnblogs.com/IRIDIUM-SUB/)

## 法二

在首页展示一言 / 今日诗词 / text，可以指定 Hitokoto 的类型和显示作者、出处。

看这篇文章：[在首页 Slogan 处添加一言 / 今日诗词 - 镜湖月的小世界 \(jinghumoon.top\)](https://jinghumoon.top/)

## 5.3.6 快速在页脚使用一言、今日诗词

看这篇文章：[快速在页脚使用一言、今日诗词 - 镜湖月的小世界 \(jinghumoon.top\)](https://jinghumoon.top) 

## 5.3.7 Fluid 页脚增加网站运行时长

看这篇文章：[Fluid 页脚增加网站运行时长 - Hexo Theme Fluid \(fluid-dev.com\)](https://fluid-dev.com) 

## 5.3.8 版权声明优化

进入 `Blog/themes/fluid/layout` 路径，打开 `post.ejs` 文件，找到以下代码：

```
YAML

    <% if(theme.post.copyright.enable &&
theme.post.copyright.content && page.copyright !== false) { %>
      <p class="note note-warning">
        <% if (typeof page.copyright === 'string' &&
page.copyright !== '') { %>
          <%- page.copyright %>
        <% } else { %>
          <%- theme.post.copyright.content %>
        <% } %>
      </p>
    <% } %>
```

替换为以下代码：

```

    <% if(theme.post.copyright.enable &&
theme.post.copyright.content && page.copyright !== false) { %>
      <p class="note note-warning">
        <strong>本文作者: </strong><a href="<%- url_for()
%>"><%- theme.about.name || config.author || config.title %></a>
      <br>
        <strong>本文链接: </strong><a href="<%-
full_url_for(page.path) %>"><%- full_url_for(page.path) %></a> <br>
        <strong>版权声明: </strong><%-
theme.post.copyright.content %>
      </p>
    <% } %>

```

刷新本地预览界面，即可看到结果。作者名在 `_config.fluid.yml` 的 `about.name` 值修改。

还可以修改下样式，代码如下：

```

.note a {
  color: #7face9;
  border-bottom: 1px solid #298bd9;
  /* font-weight: bold; */
  font-family: '微软雅黑';
}

```

参考文章：[Hexo Fluid主题添加网站运行时间与版权声明优化 - 夕枫 | 笔绘人生 \(xifenghhh.top\)](#) 

## 5.3.9 首页图片动画

在自定义引入的 `ui.css` 文件中，加入以下代码：

```
.index-img {  
  /* 动画时间 */  
  transition: .4s;  
  /* 放大后，左右两侧有两条小白边，加 border 可以抵消 */  
  border: 5px solid transparent !important;  
}  
  
.index-card:hover .index-img {  
  transform: scale(1.05);  
  /* 放大倍数 */  
}
```

参考文章：[Hexo Fluid 瞎折腾 - Jalen's Blog \(jalenchuh.cn\)](https://jalenchuh.cn) 

## 5.3.10 自定义 404 页面

01. 寻找你想要的 404 页面。可以在 GitHub 上搜索，或者在 [404模板网\\_重新定义404页面](#)  上选择 404 页面。如果有 CSS、JS 文件，建议内嵌到 HTML 文件中。
02. HTML 文件命名为 `404.html`，放入 `source` 文件夹中。
03. 打开 `_config.yml` 文件，修改如下。

```
skip_render:  
  - 404.html
```

04. 查看效果。

## 5.3.10 调整 Twikoo 评论框大小

在自定义引入的 `ui.css` 文件中，加入以下代码：

```
.twikoo .tk-submit .el-textarea__inner {  
  min-height: 150px !important;  
}
```

参考文章: [为Hexo-Fluid自定义一些效果 - ForSure's Blog](#) 

## 5.3.11 网页标题栏设置

### 法一

进入 Blog\themes\fluid\layout 路径, 打开 `layout.ejs` 文件。添加以下代码。

```
<script type="text/javascript">  
  var OriginTitile = document.title;  
  var titleTime;  
  document.addEventListener('visibilitychange', function () {  
    if (document.hidden) {  
      //离开当前页面时标签显示内容  
      document.title = 'w(°Д°)w 不要走! 再看看嘛! ';  
      clearTimeout(titleTime);  
    }  
    else {  
      //返回当前页面时标签显示内容  
      document.title = '♪(^▽^*)欢迎回来! ' + OriginTitile;  
      //两秒后变回正常标题  
      titleTime = setTimeout(function () {  
        document.title = OriginTitile;  
      }, 2000);  
    }  
  });  
</script>
```

## 示例

```
JavaScript

<script type="text/javascript">
  var OriginTitile = document.title,
    st;
  document.addEventListener("visibilitychange", function () {
    document.hidden ? (document.title = "Guan Qing 的个人博客",
clearTimeout(st)) : (document.title = OriginTitile)
  })
</script>
```

## 法二

进入 Blog\themes\fluid\layout\\_partial 路径，打开 `script.ejs` 文件，添加以下代码。

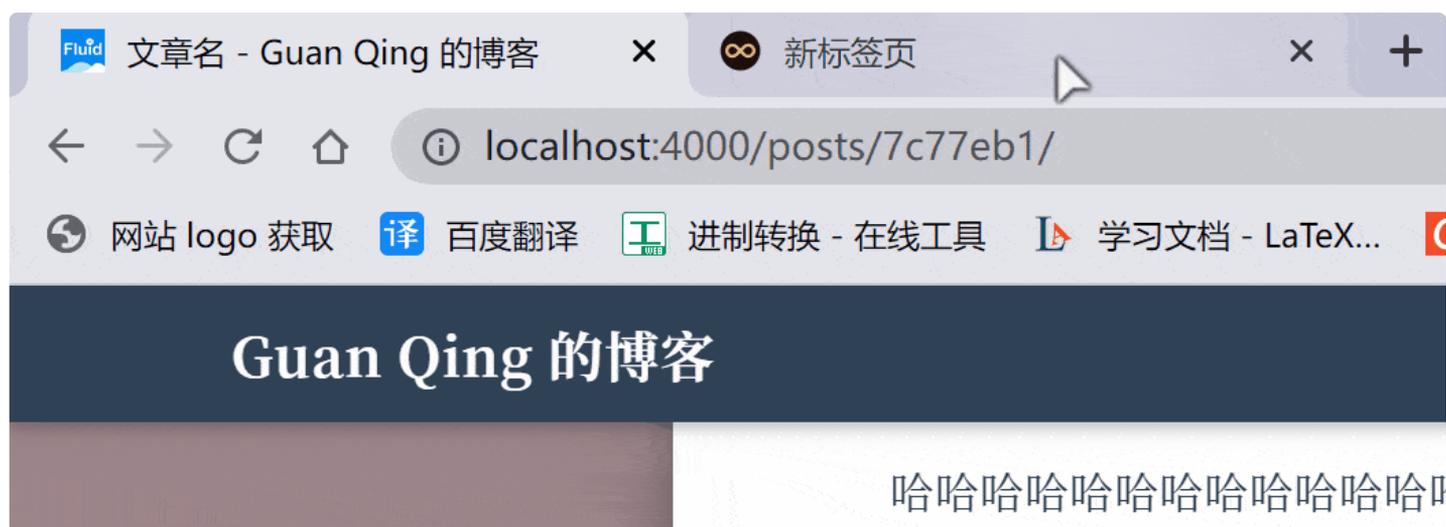
```
JavaScript

<% if(theme.fun_features.monitortext.enable) { %>
  <script type="text/javascript">
    /*窗口监视*/
    var originalTitle = document.title;
    window.onblur = function(){document.title = "<%-
theme.fun_features.monitortext.text %>"};
    window.onfocus = function(){document.title = originalTitle};
  </script>
<% } %>
```

打开 `_config.fluid.yml` 文件，在 `fun_features:` 下面，加入以下代码。

```
monitortext:  
  enable: true  
  # 离开网页，显示的标题内容。  
  text: Guan Qing 的个人博客
```

效果如下：



参考文章

- [Custom Beautify | Akilar の糖果屋](#)
- [Fluid -16- 设置页面焦点监控文字 - 明眸如初 \(zywvvd.com\)](#)

## 5.3.11 代码块折叠

目前 Fluid 主题还没有代码块折叠功能，但已经提上日程了。开发进度在这里看：[hexo-theme-fluid - TODO \(github.com\)](#)。

不过我在 **Pull requests** 看到有老哥提供了实现代码块折叠的代码。在向老哥请教一番后，我搞定了代码块折叠的功能。这是老哥的博客：[KiyangYang's Blog](#)。

### 1、代码块开启行号

01. 在 Blog\source\js 路径，创建 `code-folding-linenumbers.js` 文件，代码如下：

```
// KiyanYang

// 代码语言标准形式（这里应该有更合理的解决方式，但是我不会写）
var code_language = {
  txt: "",
  md: "Markdown",
  markdwon: "Markdown",

  yml: "YAML",
  yaml: "YAML",
  json: "JSON",

  xml: "XML",
  html: "HTML",
  css: "CSS",
  styl: "Stylus",
  stylus: "Stylus",

  js: "JavaScript",
  javascript: "JavaScript",
  ts: "TypeScript",
  typescript: "TypeScript",
  py: "Python",
  python: "Python",

  c: "C",
  "c++": "C++",
  cpp: "C++",
  cxx: "C++",
  "c#": "C#",
  cs: "C#",
  csharp: "C#",
  java: "Java",
};
```

```

// 代码语言格式化
function getLanguage(lang) {
    var lang_lower = lang.toLowerCase();
    return lang_lower in code_language ? code_language[lang_lower] :
lang;
}

// 获取唯一 ID
function getUuiD() {
    return Math.random().toString(36).substring(2, 8) +
Date.now().toString(36);
}

function addLanguage() {
    // 获取所有 figure.highlight 元素
    var hs = $("figure.highlight");
    for (var i = 0; i < hs.length; i++) {
        // 获取代码语言
        var lang =
hs[i].firstChild.firstChild.firstChild.lastChild.firstChild.firstChild.classList[1];
        // 折叠块的 id
        var id = `kiyan-collapse-${getUuiD()}`;
        // 前面折叠按钮，这里我使用了 FAS 图标
        // 注意：这里是 bootstrap 的 collapse 功能的简略代码，使用时会自动补
        全，补全后的 HTML 可以通过浏览器 F12 查看，此外也可查看官方文档
        var btn = `${getLanguage(lang)}</span>`;
        // 折叠块包裹原来的内容
        var div = `

No. 107 / 204


```

```
$(document).ready(addLanguage);
```

02. 在 Blog\source\css 路径，创建 `code-folding-linenumbers.css` 文件，代码如下：

```
figure.highlight {
  background: #e6ebf1;
  border-radius: 0.5rem;
}

figure.highlight table {
  border-radius: 0 0 0.5rem 0.5rem;
}

figure.highlight .fas.fa-angle-down {
  transform: none;
  transition: transform 0.2s ease-in-out;
}

figure.highlight .fas.fa-angle-down.collapsed {
  transform: rotate(-90deg);
}

figure.highlight>i {
  color: #777;
  margin-left: 10px;
  line-height: 2rem;
}

figure.highlight>span {
  color: #777;
  margin-left: 10px;
  font-weight: bold;
}
```

```
figure.highlight td:first-child {
  position: sticky;
  left: 0;
  z-index: 1;
}

.copy-btn {
  top: 5px;
  font-size: 1rem;
}

.copy-btn>i {
  font-size: 0.875rem;
  font-weight: bold;
}

.copy-btn-dark {
  color: #2f4f4f;
}
```

03. 在 `_config.fluid.yml` 文件引入上面创建的两个文件。刷新本地预览，即可看到效果。

## 2、代码块未开启行号

01. 在 `Blog\source\js` 路径，创建 `code-folding-not-linenumbers.js` 文件，代码如下：

```
// KiyangYang

// 代码语言标准形式
var code_language = {
  txt: "",
  md: "Markdown",
```

JavaScript

```

markdwon: "Markdown",
md_txt: "Markdown",

yaml: "YAML",
yam1: "YAML",
json: "JSON",

xml: "XML",
html: "HTML",
css: "CSS",
styl: "Stylus",
stylus: "Stylus",

js: "JavaScript",
javascript: "JavaScript",
ts: "TypeScript",
typescript: "TypeScript",
py: "Python",
python: "Python",

c: "C",
"c++": "C++",
cpp: "C++",
cxx: "C++",
"c#": "C#",
cs: "C#",
csharp: "C#",
java: "Java",
};

// 代码语言格式化
function getLanguage(lang) {
    var lang_lower = lang.toLowerCase();
    return lang_lower in code_language ? code_language[lang_lower] :
lang;
}

```

```

// 获取唯一 ID
function getUiID() {
    return Math.random().toString(36).substring(2, 8) +
Date.now().toString(36);
}

function addLanguage() {
    // 获取所有 .code-wrapper 元素
    var hs = $(".code-wrapper"); // 此处改动
    for (var i = 0; i < hs.length; i++) {
        // 获取代码语言
        var lang = hs[i].firstChild.firstChild.classList[1]; // 此处
改动
        // 折叠块的 id
        var id = `kiyan-collapse-${getUiID()}`;
        // 前面折叠按钮
        var btn = `${getLanguage(lang)}</span>`;
        // 折叠块包裹原来的内容
        var div = `

02. 在 Blog\source\css 路径，创建 code-folding-not-linenumbers.css 文件，代码如下：





```

.css
.code-wrapper {
    background: #e6ebf1;
}

```



No. 111 / 204


```

```
}

.code-wrapper .fas.fa-angle-down {
  transform: none;
  transition: transform 0.2s ease-in-out;
  background: transparent;
}

.code-wrapper .fas.fa-angle-down.collapsed {
  transform: rotate(-90deg);
}

.collapseing,
.collapsed {
  background: transparent;
}

.code-wrapper>i {
  color: #777;
  margin-left: 10px;
  line-height: 2rem;
}

.code-wrapper>span {
  color: #777;
  margin-left: 10px;
  font-weight: bold;
}

.code-wrapper {
  margin-bottom: 16px;
}

.markdown-body pre {
  margin-bottom: 0;
}
```

03. 在 `_config.fluid.yml` 文件引入上面创建的两个文件。刷新本地预览，即可看到效果。

## 5.3.12 默认在文章末显示结束语

进入 `E:\Blog\themes\fluid\layout` 路径，打开 `post.ejs` 文件，添加以下代码。

```
JavaScript
<div class="markdown-body">
  <%- page.content %>
</div>
<hr>
<!-- 添加的代码 -->
<div class="note note-success">本文到这就结束了，如果你对文章内容有什么疑问的话，可以在下面的评论区留言，我会及时地回复你！ </div>
<div>
  <div class="post-metas mb-3">
    <% if (page.categories && page.categories.length >
0) { %>
      <div class="post-meta mr-3">
        <i class="iconfont icon-category"></i>
        <% page.categories.each(function(cate) { %>
          <a class="hover-with-bg" href="<%=
url_for(cate.path) %>"><%- cate.name %></a>
          <% }) %>
        </div>

```

本文到这就结束了，如果你对文章内容有什么疑问的话，可以在下面的评论区留言，我会及时地回复你！

更多样式请看 [配置指南 | Hexo Fluid \(fluid-dev.com\)](#)。

## 5.3.13 添加谷歌翻译

[为你的博客添加谷歌翻译 - 镜湖月的小世界 \(jinghumoon.top\)](#) 

## 5.3.14 博客添加书签页

[为你的博客添加书签页 - 镜湖月的小世界 \(jinghumoon.top\)](#) 

## 5.3.15 博客添加公告栏

[博客首页添加公告栏 - 镜湖月的小世界 \(jinghumoon.top\)](#) 

# 第六章 开启评论

对于一个博客网站来说，评论系统是很重要的。没有评论，就不能与来访者互动。你不知道你的文章写得好不好，有没有错误。对我来说，写文章的热情减少大半。

## 6.1 使用

[Twikoo 中文文档](#) 

我使用的评论插件是 **Twikoo**，一个简洁、安全、免费的静态网站评论系统，基于 [腾讯云开发 \(opens new window\)](#) 。配置起来十分简单，功能也非常多。作者文档写得也比较详细，还录了视频，发在 B 站上。

Vercel 部署教程：[Twikoo Vercel 部署教程 - bilibili](#) 

Vercel 部署教程完成后，你会得到 Vercel 分发给你的一个域名，在 `_config.fluid.yml` 中填写该域名。

```
twikoo:
  envId: "https://xxxxxx.vercel.app/"
  region: ap-shanghai
  path: window.location.pathname
```

YAML

# twikoo-comment

View Git Repository

Visit

## Production Deployment

View Build Logs

View Function Logs

View Domains

The deployment that is available to your visitors.

```

{
  "code": 100,
  "message": "Twikoo 云函数运行正常, 请参考 https://twikoo.com",
  "version": "1.4.17"
}

```

DEPLOYMENT  
twikoo-comment-ozbl937hn-ziguanqing.vercel.app

DOMAINS  
twikoo-comr[redacted]app +2

这个

STATE CREATED  
● Ready 4d

参考文章: [Twikoo Vercel 部署 | OYの博客 \(oy6090.top\)](#)

腾讯云云开发部署教程: [Twikoo 腾讯云云开发部署视频教程 - bilibili](#)

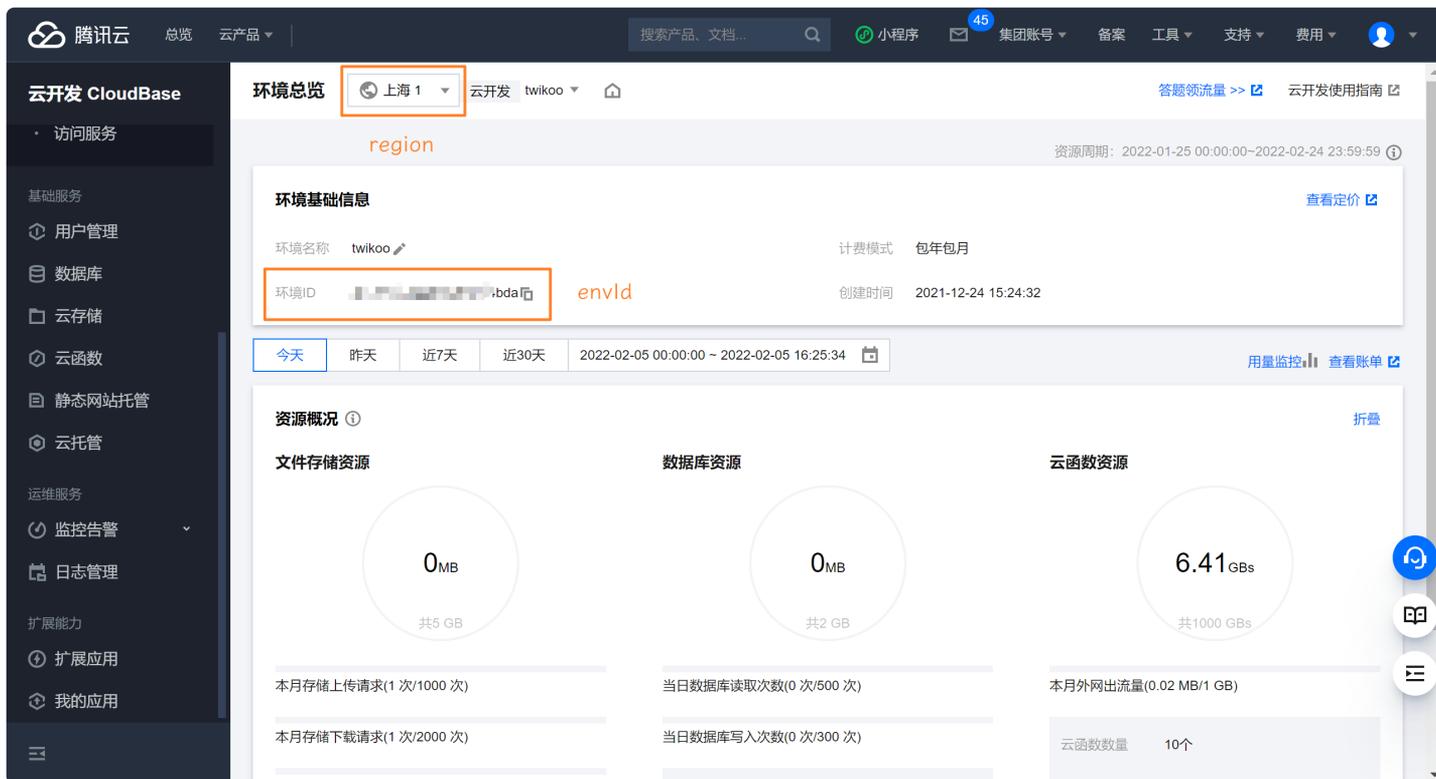
视频中有一步是**添加 WEB 安全域名**。填自己的域名，要确认自己的域名有没有 `www`。建议设置两个域名，一个是 `www.xxx.com` 一个是 `xxx.com`。或者直接使用泛域名 `*.xxx.com`。

腾讯云云开发部署教程后，在 `_config.fluid.yml` 中修改如下。

```

twikoo:
  envId: "xxxxxxx"
  region: ap-shanghai
  path: window.location.pathname

```



个人认为部署在腾讯云云开发上，加载速度更快（中国大陆地区）。文档说付费购买环境才能部署。但是还可以免费部署在云开发的。就是额度比较少。付费也不贵，特惠基础版（1年）只要 54 元。

目前我使用的 Twikoo 版本为 1.4.17。

## 6.2 配置

### 6.2.1 即时通知

即时通知：来访者评论，通知博主。

这个版本，Twikoo 提供了 Server 酱、Qmsg 酱、推送加、PushDeer、企业微信通知 API、钉钉 WebHook API，共 7 种即时通知方式。下面简单介绍下每种推送方式。

- Server 酱（不推荐）
  - 推荐指数：★
  - 微信公众号推送消息，仅显示标题。
  - 非订阅用户，每天最多推送 5 条消息。

- Qmsg 酱（不推荐）
  - 推荐指数：★
  - QQ 用户或者 QQ 群机器人推送消息。
  - 非捐赠用户不能用，腾讯官方有限制。
- [推送加](#) （推荐）
  - 推荐指数：★★★
  - 微信公众号推送消息，仅显示标题。
  - 免费，使用简单。
- [PushDeer](#) 
  - 推荐指数：★★★
  - APP 推送，界面美观。
  - 免费，使用简单。
  - [Android 测试版 · Easy/pushdeer - Gitee.com](#) 
- 企业微信通知 API
  - 推荐指数：★★★★
  - 使用腾讯云云开发环境搭建，访问极快。
  - 基本不会失效。
  - 搭建比较费工夫。请看这篇文章 [微信通知 API 实现 Twikoo 新消息提醒 - 镜湖月的小世界 \(jinghumoon.top\)](#) 。
- 邮件通知
  - 推荐指数：★★★★
  - 不会失效。相当简单。

下面四种方式，除了**微信 API**，其余的使用很简单 — 获取 `key` ，填入相应位置。



钉钉 WebHook API 用不来。感兴趣的朋友请看这篇文章 [Twikoo 的 DingTalkWebHook 推送 | Ljcbaby 的网络小屋](#)。

对了，使用博主的账号发评论，是不会通知你的。

## 6.2.2 邮件通知

邮件通知：博主回复来访者评论，通知来访者。配置比较简单。但使用 **163 邮箱**，貌似不行。推荐使用 **QQ 邮箱**。

如何生成授权码？

进入网页版 QQ 邮箱 → 设置 → 账户 → 开启 POP3/SMTP 服务、IMAP/SMTP 服务 → 生成授权码。

## ▶ 通用

## ▶ 插件

## ▶ 反垃圾

## ▶ 即时通知

## ▼ 邮件通知

SENDER\_EMAIL

2027004326@qq.com

邮件通知邮箱地址。对于大多数邮箱服务商，SENDER\_EMAIL 必须和 SMTP\_USER 保持一致，否则无法发送邮件。

SENDER\_NAME

博客新评论提醒

邮件通知标题。

SMTP\_SERVICE

QQ

邮件通知邮箱服务商。支持: "126", "163", "1und1", "AOL", "DebugMail", "DynectEmail", "FastMail", "GandiMail", "Gmail", "Godaddy", "GodaddyAsia", "GodaddyEurope", "Hotmail", "Mail.ru", "Maildev", "Mailgun", "Mailjet", "Mailosaur", "Mandrill", "Naver", "OpenMailBox", "Outlook365", "Postmark", "QQ", "QQex", "SES", "SES-EU-WEST-1", "SES-US-EAST-1", "SES-US-WEST-2", "SendCloud", "SendGrid", "SendPulse", "SendinBlue", "Sparkpost", "Yahoo", "Yandex", "Zoho", "hot.ee", "iCloud", "mail.ee", "qiye.aliyun"

SMTP\_HOST

示例: smtp.qq.com

自定义 SMTP 服务器地址。如您已配置 SMTP\_SERVICE，此项请留空。

SMTP\_PORT

示例: 465

自定义 SMTP 端口。如您已配置 SMTP\_SERVICE，此项请留空。

SMTP\_SECURE

示例: true

自定义 SMTP 是否使用 TLS, 请填写 true 或 false。如您已配置 SMTP\_SERVICE, 此项请留空。

SMTP\_USER

2027004326@qq.com

邮件通知邮箱用户名。

SMTP\_PASS

.....

邮件通知邮箱密码, QQ 邮箱请填写授权码。

MAIL\_SUBJECT

你在博客上的评论收到了回复!

自定义通知邮件主题, 留空则使用默认主题。

MAIL\_TEMPLATE

自定义通知邮件模板, 留空则使用默认模板。可包含的字段:  
\${SITE\_URL}, \${SITE\_NAME}, \${PARENT\_NICK},  
\${PARENT\_COMMENT}, \${NICK}, \${COMMENT}, \${POST\_URL},  
\${IMG}, \${PARENT\_IMG}

MAIL\_SUBJECT...

博客有新评论啦!

自定义博主通知邮件主题, 留空则使用默认主题。

MAIL\_TEMPLAT...

自定义博主通知邮件模板, 留空则使用默认模板。可包含的字段:  
\${SITE\_URL}, \${SITE\_NAME}, \${NICK}, \${COMMENT},  
\${POST\_URL}, \${IP}, \${MAIL}, \${IMG}

## ▶ 邮件通知测试

保存

重置

邮件通知测试如下, 则邮件通知配置成功。



如果您收到这封邮件，说明 Twikoo 邮件功能配置正确

## 6.3 美化

[Twikoo美化教程 | 呆逼の博客 \(keepdai.cn\)](#)

[自定义Twikoo访客通知邮件模板 | Guo Le's Blog](#)

有兴趣的朋友可以看下。

# 第七章 自定义域名

## 7.1 什么是域名

请看视频 [【白话科普】域名？网址？原来域名是这么个玩意 啊！终于听懂了 | 互联网的运作原理 | 网站上线发布分享教程系列内容第三期 - bilibili](#) 

视频讲得非常清楚。

## 7.2 常见域名注册商

### 7.2.1 国内

[阿里云](#) 、[腾讯云](#) 、[西部数码](#) 

购买域名，我比较推荐「腾讯云」。腾讯云价格在三家中，最低。到期续费价格也不会涨太多。据工信部要求，在中国大陆地区开展网站服务，必须在服务器提供商处完成网站备案。就是说，购买国内的域名，需要**备案**。备案需要服务器，而腾讯云提供轻量应用服务器。3 个月只要 27 元（在校学生）。恰好，备案要求服务器购买时长在 3 个月及以上（包含续费）且备案期间剩余有效期需大于（含）1 个月。具体要求看 [网站备案 备案云服务 - 备案准备 - 文档中心 - 腾讯云 \(tencent.com\)](#) 。这不就是刚刚好! 😊

### 7.2.2 国外

- [namesilo](#) 

- 网站介绍和购买流程看这篇文章 [Namesilo 域名购买及使用教程（附 Namesilo 优惠码） - 知乎 \(zhihu.com\)](#) 、[如何购买境外域名并迁移到 CloudFlare 托管 - 麦子尖的个人小站 \(hash070.top\)](#) 

- 网站不支持中文。付款可用支付宝、信用卡等。并提供永久的免费隐私保护。

- [GoDaddy官方中文站](#) 

- 网站支持中文。购买有坑，有诱导消费。不建议在该网站购买域名。

我最推荐在 [namesilo](#)  上购买域名。购买国外的域名，不需要备案。

参考文章: [6个最好的域名注册商推荐 - 知乎 \(zhihu.com\)](#) 

## 7.3 购买流程

以腾讯云为例。

01. 进入腾讯云购买域名界面 [域名注册 - 腾讯云官网](#) 。
02. 按照自己的喜好确定想要的「一级域名」，并搜索。
03. 选择「顶级域名」。
  - A. 域名尽量太随机，如果想以后有人能访问的话。
  - B. 有些域名首年很便宜，但是续费很贵。
  - C. 推荐 `top` 后缀。比较便宜。
04. 付款。需要填写个人信息模板。添写就好。需要注意的地方，腾讯云都标注了。

## 7.4 域名解析 (DNS 解析)

### 什么是域名解析?

在互联网，每个网站都有自己的地址，这个地址是数字和点的组合，称作 [IP 地址](#) 。例如 `127.0.0.1`。为方便记忆，我们采用域名代替网站的 IP 地址。我们普通人使用的计算机无法识别未解析的域名，所以我们需要解析域名。解析域名由 DNS 服务器（一种特殊的计算机）完成，它将域名解析成 IP 地址，然后在

此 IP 地址的主机上将一个子目录与域名绑定。这样一来，我们便可以通过域名访问我们的网站。

更多内容请看以下链接。

- [域名解析\\_百度百科 \(baidu.com\)](#)
- [DNS 解析的过程是什么，求详细的？ - 知乎 \(zhihu.com\)](#)

解析域名以腾讯云为例。点击 [我的域名 - DNSPod-免费智能DNS解析服务商](#)，进入 DNS 解析界面。

腾讯云 | DNSPOD

域名	状态	记录数	套餐	最后操作时间	操作
<a href="#">jinghumoon.top</a>	正常	7 条	免费版	2022-01-31 16:17:16	
<a href="#">jinghu-moon.top</a>	正常	3 条	免费版	2022-01-28 09:47:26	

第 1-2 行, 共 2 行

每页显示行 20

1 / 1 页

© 2006-2022 DNSPod, Inc. All rights reserved. © 2006-2022 烟台帝思普网络科技有限公司

系统状态 | 帮助支持 | 服务协议 | 隐私

01. 添加域名。进入以下界面。



- **主机记录**相当于域名的前缀，比如 `www`。 [查看详细指引](#)
- 不同的**记录类型**有不同的用途，一般选择 A 记录。 [查看详细指引](#)
- 我们这主要使用 A 类型和 CNAME 类型。

记录类型	用途
A	A 记录是最常用类型，将域名指向一个 IPv4 地址，如 8.8.8.8
CNAME	将域名指向另一个域名地址，与其保持相同解析，例如 <a href="https://www.dnspod.cn">https://www.dnspod.cn</a> 。
MX	用于邮件服务器，相关参数一般由邮件注册商提供。
TXT	可填写附加文本信息，常用于域名验证。
NS	域名服务器记录，可将指定域名交由其他 DNS 服务商解析管理。
AAAA	将域名指向一个 IPv6 地址，如 ff06:0:0:0:0:0:0:c3。
SPF	用于指定发送邮件的服务器，是一种高效的反垃圾邮件解决方案。
SRV	用于标识某台服务器使用了某个服务，常见于微软系统的目录管理。格式为「服务名字.协议类型」，如 _sip._tcp。
CAA	用于指定域名的证书颁发机构（CA），减少证书颁发风险。
显性 URL	将一个域名重定向至某个具体网页，且显示实际 URL。仅支持 301 重定向，该记录要求双方域名均已完成备案。
隐性 URL	将一个域名重定向至某个具体网页，但隐藏实际 URL。仅支持 301 重定向，该记录要求双方域名均已完成备案。

- DNSPod 支持按多种方式划分线路，让指定用户访问该记录，升级套餐还可增加线路选项。在后面讲到 GitHub+coding 双线部署时，会使用该选项。 [查看详细指引](#)

线路类型	解释
默认	默认线路，一般情况下保留默认即可，每一条记录必须保证已有默认线路才可正常解析。
联通	运营商线路，可为联通用户单独指定一个记录值，其他用户则仍然访问默认线路。
搜索引擎	可指定一个服务器 IP 专用于搜索引擎抓取。

02. 查看自己的仓库 IP 地址，`win+r` 打开 cmd，输入以下代码。

```

bash
ping xxx.github.io
# 示例
ping ziguanqing.github.io

Microsoft Windows [版本 10.0.19043.1469]
(c) Microsoft Corporation。保留所有权利。

C:\Users\khi12>ping ziguanqing.github.io

正在 Ping ziguanqing.github.io [185.199.111.153] 具有 32 字节的数据:
来自 185.199.111.153 的回复: 字节=32 时间=52ms TTL=46
来自 185.199.111.153 的回复: 字节=32 时间=66ms TTL=46
请求超时。
来自 185.199.111.153 的回复: 字节=32 时间=53ms TTL=46

185.199.111.153 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 3, 丢失 = 1 (25% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 52ms, 最长 = 66ms, 平均 = 57ms

C:\Users\khi12>

```

03. 添加解析记录。如下。

jinghu-moon.top ▾

! 套餐服务 域名设置 权限管理 数据统计 线路管理 操作日志

升级正式版套餐，获得更极致的 DNS 解析服务，最低仅需 8 元/月

+ 记录

快速添加解析

更多操作 ▾

全部记录 ▾

🔽 筛选器

主机记录	记录类型	线路类型	记录值	权重	MX	TTL	最后操作时间	
@	A	默认	185.199.111.153	-	-	600	2022-01-31 22:13:26	
www	CNAME	默认	ziguanqing.github.io.	-	-	600	2022-01-31 22:16:20	

理论上来说，A 记录有四条。

```
185.199.108.153
185.199.109.153
185.199.110.153
185.199.111.153
```

但是，免费版只允许添加两条 A 记录。另外，如果你想你的博客域名是 `blog.xxxx.xxxx`，可以把 CNAME 的主机记录改为 **blog**。

- 在 `Blog\source` 路径下，创建 `CNAME` 文件。推荐使用命令行创建。打开 `CNAME` 文件，输入购买的域名，不要 `http//:`。我的是 `jinghu-moon.top`。保存文件。

about	2022/1/22 19:17	文件夹
categories	2022/1/20 22:34	文件夹
css	2022/1/29 19:11	文件夹
icon	2022/1/20 22:34	文件夹
js	2022/1/29 19:11	文件夹
links	2022/1/20 22:34	文件夹
random	2022/1/29 19:11	文件夹
tags	2022/1/20 22:34	文件夹
talking	2022/1/22 17:57	文件夹
manifest.json	2022/1/20 21:21	JSON 源文件

MINGW64:/e/Blog/source

```
khi12@jinghu-moon-computer MINGW64 /e/Blog/source (main)
$
```

05. 打开 GitHub，进入 `用户名.github.io` 仓库。点击 `Settings`，选择 `Pages`。填写域名，并开启强制 HTTPS。

06. 打开 `config.yml` 文件，将 `url` 改为自己的域名。

```
url: https://jinghu-moon.top
```

## 7.4 访问网站

`hexo cl && hexo g -d`，部署到 GitHub 上。等待一会，你就可以通过自己的域名访问博客了。一会是多久，我也不清楚。24 小时后再访问网站吧。

成功通过新域名访问网站! 😊



## 7.5 网站备案 (可选)

[腾讯云备案\\_个人域名备案\\_网站备案\\_企业备案 \(tencent.com\)](#) 

备案，按照腾讯云的流程走，就行。备案通过时间是未知数。我等了 17 天。🤔

### 备案注意事项

- 各省管局备案要求不同。具体看下面链接。

- 备案人需年满 16 周岁。
- 备案网站 DNS 解析时间超过 3 天。
- 备案网站时，网站必须暂停 DNS 解析。
- 备案流程有一步是「视频核身」，背景必须为**白色**。
- 备案订单提交后，保持手机通讯正常。客服小姐姐会打电话告诉你，你提交的备案信息哪里有问题、该如何修改。
- ~~网站备案期间，尽量不要做修改。（我新增备案时，网站时不时修改，也备案成功子）~~
- 在备案订单提交几天后，工信部会发一条核验信息，记得去验证。



# 验证码



为保证安全,请谨慎使用



工业和信息化部公共服务电... 2021-12-6 16:04 2

083330

工业和信息化部 | 验证码

复制

【工业和信息化部】验证码: 083330。您的新增备案已提交, 核验链接: [beian.miit.gov.cn](http://beian.miit.gov.cn), 有效期1天【工信部ICP备案】



腾讯云

2021-12-6 11:12 2

823690

腾讯云 | 验证码

复制

【腾讯云】域名备案验证码: 823690, 请在30分钟完成输入

备案成功后，工信部会发信息提醒你。

2 2021-12-20 14:16

【工业和信息化部】尊敬的用户[REDACTED]，您的ICP备案申请已通过审核，备案/许可证编号为：蜀ICP备[2021032128](#)号，审核通过日期：[2021-12-20](#)。特此通知！【工信部ICP备案】

2 1-24 9:45

【工业和信息化部】尊敬的用户[REDACTED]，您的备案信息已被变更，详情请咨询您的接入服务提供商。【工信部ICP备案】



短信 2



在网站页脚添加备案号，备案就完成了。

# 第八章 常用插件

## 8.1 下载与卸载

在 `Blog` 文件夹打开 `Git Bash here` 。输出以下命令。

下载插件：

```
Bash
npm install --save 插件名1 插件名2 插件名3
示例
npm install --save hexo-abbrlink
```

卸载插件：

```
Bash
npm uninstall 插件名1 插件名2 插件名3
示例
npm uninstall hexo-abbrlink
```

```
MINGW64:/e/Blog
khi12@jinghu-moon-computer MINGW64 /e/Blog (main)
$ npm install --save hexo-abbrlink

added 5 packages, and audited 263 packages in 4s
16 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

khi12@jinghu-moon-computer MINGW64 /e/Blog (main)
$ npm uninstall hexo-abbrlink

removed 5 packages, and audited 258 packages in 2s
16 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

khi12@jinghu-moon-computer MINGW64 /e/Blog (main)
$
```

## 8.2 hexo-abbrlink

### 1、功能

为你的文章生成永久链接。不因为你更改了文章标题，导致原来的链接失效。

### 2、GitHub 链接

[rozbo/hexo-abbrlink: create one and only link for every post for hexo \(github.com\)](https://github.com/rozbo/hexo-abbrlink)

### 3、下载

```
Bash
npm install hexo-abbrlink --save
```

### 4、使用

打开 `_config.yml` 文件，添加以下代码。

```
Bash
```

```
# hexo-abbrlink
abbrlink:
  alg: crc32      #支持crc16和crc32算法 (默认crc16)
  rep: hex        #支持dec和hex值 (默认dec)
  drafts: false   #(true)Process draft,(false)Do not process draft.
false(default)
  # Generate categories from directory-tree
  # depth: the max_depth of directory-tree you want to generate,
should > 0
  auto_category:
    enable: true  #true(default)
    depth:        #3(default)
    over_write: false
  auto_title: false #enable auto title, it can auto fill the title
by path.(启用自动标题, 可按路径自动填充标题)
  auto_date: false #enable auto date, it can auto fill the date by
time today.(启用自动日期功能, 可按今天的时间自动填写日期)
  force: false #enable force mode,in this mode, the plugin will
ignore the cache, and calc the abbrlink for every post even it
already had abbrlink.(启用强制模式, 在此模式下, 插件将忽略缓存, 并为每个
帖子计算 abbrlink, 即使它已经有了 abbrlink。)
```

同时将 `permalink: :year/:month/:day/:title/` 改为 `posts/:abbrlink/`。在文章的 Front-matter 中加入 `abbrlink:`。即可在网页查看文章时, 自动生成文章链接。`abbrlink` 属性值可以自定义, 但尽量不设置为中文。

**注意:** 使用此插件, `categories`、`tags` 不能为空。若不想添加分类、标签, 删除即可。

参考资料: [优化 Hexo 的永久链接 \(dejavu.moe\)](https://dejavu.moe/) 

## 8.2 hexo-neat

## 1、功能

压缩静态页面，提高网站响应速度。

## 2、GitHub 链接

[rozbo/hexo-neat: auto Minify html、js、css and make it neat \(github.com\)](https://github.com/rozbo/hexo-neat)



## 3、下载

```
Bash
npm install hexo-neat --save
```

## 4、使用

在 `config.yml` 文件中添加以下代码。

```
YAML
# hexo-neat
neat_enable: true

# 压缩html
neat_html:
  enable: true
  exclude:

# 压缩css
neat_css:
  enable: true
  exclude:
    - '**/*.min.css'

# 压缩js
neat_js:
  enable: true
```

```
mangle: true
output:
compress:
exclude:
  - '**/*.min.js'
  - '**/jquery.fancybox.pack.js'
  - '**/index.js'
```

更多配置请看文章：[使用Hexo-neat插件对网页进行压缩 | Akilarの糖果屋](#) 

## 8.3 Artitalk

### 1、功能

为你的博客添加说说。

### 2、官网

<https://artitalk.js.org/> 

### 3、使用

01. 前往 [LeanCloud 国际版](#)  或者 [LeanCloud 国内版](#) , 注册账号。
02. 创建应用, 应用名随便取。例如: talking。
03. 进入设置。
04. 左侧边栏, 数据存储 → 结构化数据。创建 `class`, 名为 `shuoshuo`。
05. 左侧边栏, 内建账户 → 用户管理。添加用户, 一定记住密码。
06. 回到**结构化数据**中, 点击 `class` 下的 `shuoshuo`。找到权限, 在 `Class 访问权限` 中将 `add_fields` 以及 `create` 权限设置为指定用户, 输入你刚才输入的用户名会自动匹配。为安全起见, 将 `delete` 和 `update` 也设置为指定用户。
07. 然后新建一个名为 `atComment` 的 `class`, 权限默认即可。

08. 点击 `class` 下的 `_User` 添加列，列名称为 `img`，列类型为 `String`，**默认值**填上想要用来发布说说的头像 url，这一项不进行配置，说说头像会显示为默认头像 —— Artitalk 的 logo。
09. 左侧边栏，设置 → 应用凭证。复制 `AppID`、`AppKey`、`REST API 服务器地址`。（serverURL）
10. 在 `Blog` 目录下，创建新页面。

```
Bash
hexo n page talking
```

11. 进入 `Blog\source\talking` 路径，打开 `index.md`。输入以下内容。

```
HTML
<script type="text/javascript" src="https://unpkg.com/artitalk">
</script>
<div id="artitalk_main"></div>
<script>
new Artitalk({
  appId: 'xxxxxxxxxxxxxxxxxxxx', // Your LeanCloud appId
  appKey: 'xxxxxxxxxxxxxxxxxxxx', // Your LeanCloud appKey
  serverURL: 'https://xxxxx.xxxxx.com' // REST API 服务器地址
})
</script>
```

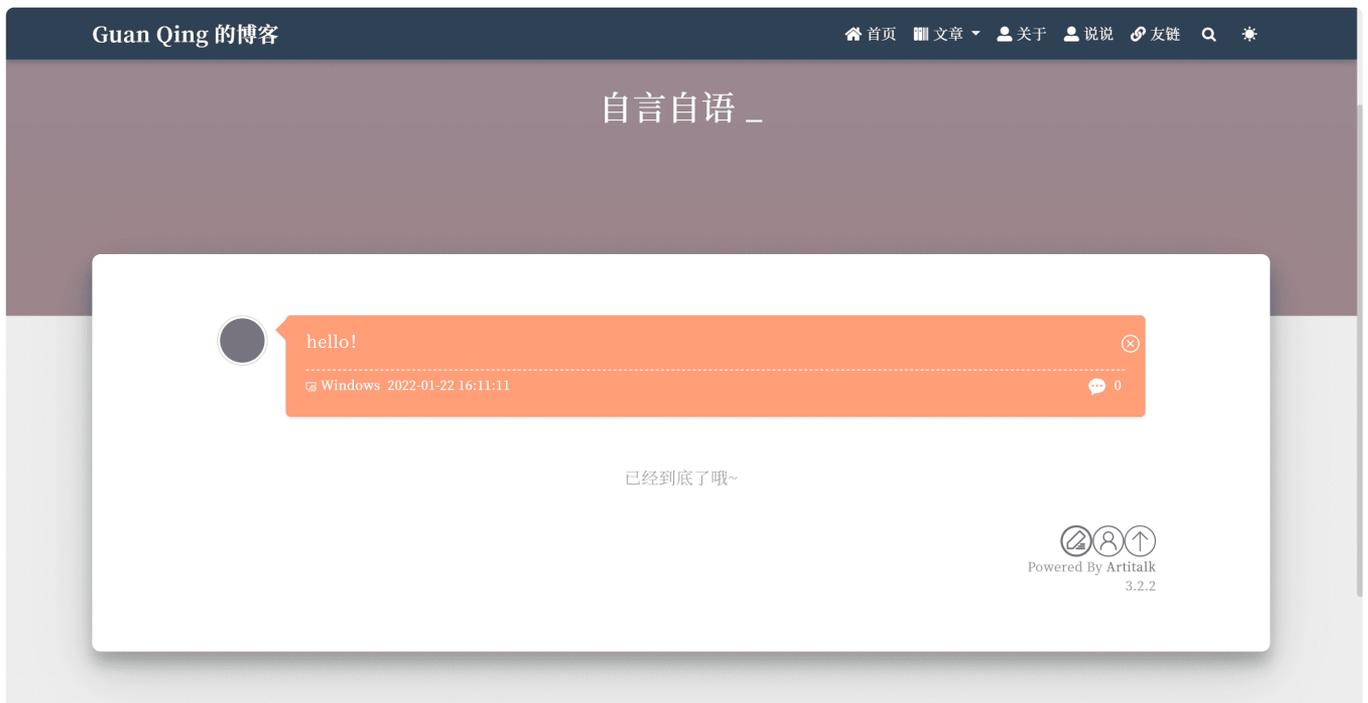
行与行之间不要有空行。

12. 在 `_config.fluid.yml` 文件中，在 `navbar.menu` 处添加如下代码。

```
YAML
- { key: "说说", link: "/talking/", icon: "iconfont icon-liaotian" }
```

`icon` 可以自行修改。

13. 本地预览，即可看到效果。

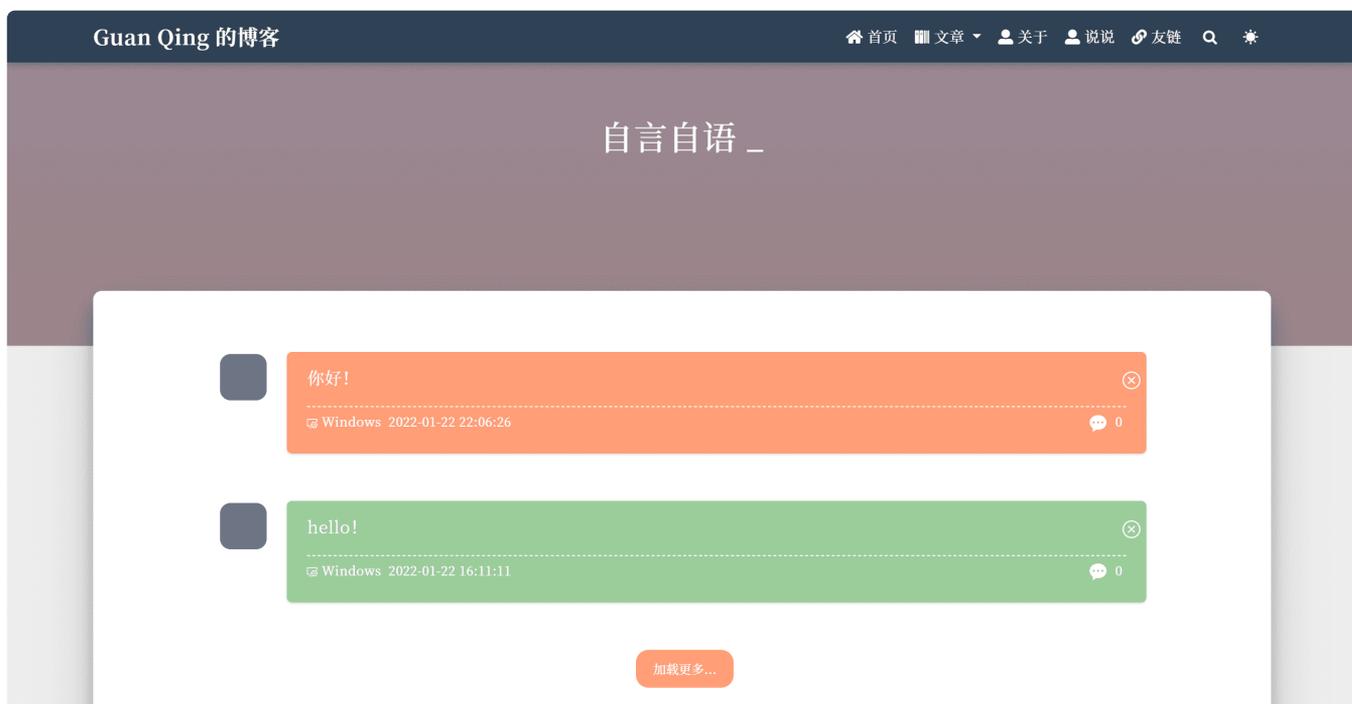


#### 14. 样式修改。（参考）

```
/* talking */
#artitalk_main .cbp_tmtimeline>li:nth-child(odd)
.cbp_tmlabel:after,
#artitalk_main .cbp_tmtimeline>li .cbp_tmlabel:after {
    content: none !important;
}

#artitalk_main .shuoshuo_author_img img {
    border: none;
    padding: 2px;
    float: left;
    border-radius: 12px !important;
    transition: all 1s;
}

#artitalk_main .shuoshuo_author_img img {
    border: none !important;
}
```



## 8.4 hexo-browsersync

### 1、功能

在 `hexo server` 预览博客时，改动 js, css, md 等文件，<http://localhost:4000/> 会实时同步刷新网页。

### 2、GitHub 链接

[hexojs/hexo-browsersync: BrowserSync plugin for Hexo. \(github.com\)](https://github.com/hexojs/hexo-browsersync)

### 3、下载

```
Bash
npm install hexo-browsersync --save
```

### 4、使用

`hexo server` 后，修改文件，即可看到效果。

**注意：**改动 `_config.fluid.yml` 或者 `_config.yml` 文件，无法实时同步，需要手动 `hexo cl && hexo g && hexo s`。

## 8.5 hexo-generator-random

不推荐使用此插件，请看这篇文章 [博客实现文章随机漫游功能 - 镜湖月的小世界 \(jinghumoon.top\)](http://jinghumoon.top) 。

### 1、功能

生成一个 random.html 文件，点击可跳转到博客下所有文章中的一篇。

### 2、GitHub 链接

[Drew233/hexo-generator-random \(github.com\)](https://github.com/Drew233/hexo-generator-random)

### 3、下载

首先安装依赖。

```
Bash
npm install object-assign
```

然后下载插件。

```
Bash
npm install hexo-generator-random --save
```

### 4、使用

`hexo g` 之后，在 public 文件夹下会生成一个 random.html。直接访问 `yoursite/random.html` 就可以感受这个插件效果。

为更方便使用该插件。在 `_config.fluid.yml` 文件中添加以下代码。

```
menu:  
  - { key: "漫游", link: "/random.html", icon: "iconfont icon-manyou3" }
```

**icon** 请自己设置。

## 5、Bug

使用非 [sakura](#) 主题时，随机访问的内容不一定是文章，还可能混入其他页面，比如关于页、分类页等。

参考：[hexo-generator-random | Hexo插件 | Uncle\\_drew \(cndrew.cn\)](#)

## 8.6 hexo-tag-hint

使用 fluid 主题和插件会出现 **Bug**，引入的 `js` 文件可能会失效。不建议使用。

### 1、功能

给一句话添加悬浮注释。

### 2、GitHub 链接

[etigerstudio/hexo-tag-hint: A Hexo tag plugin to display text hint/spoiler tooltip. \(github.com\)](#)

### 3、下载

```
npm install hexo-tag-hint
```

### 4、使用方法

#### 01. 单行注释

```
{% hint '这里是正文' '这里是注释' %}
```

## 02. 多行注释

```
{% hint '《岳阳楼记》' '【作者】范仲淹,' '【朝代】宋' %}
```

## 5、示例

```
#include <stdio.h>
int main()
{
    printf("hello,world!");
    return
}
```

呵呵哈哈 《岳阳楼记》 哈哈

哈哈哈哈哈 这里是正文 哈哈

本文到这也就结束了，如果你对文章内容有什么疑问的话，可以在下面的评论区留言，我会及时地回复你！

哈哈 1 哈哈 啊

本文作者: [Guan Qing](#)

本文链接: <http://ziguanqing.github.io/posts/3b8ba7c7/>

版权声明: 本博客所有文章除特别声明外，均采用 [CC BY-SA 4.0 协议](#)，转载请注明出处！

## 6、注意

- 连续使用两段悬浮文本（之间没有普通文字），必须使用 `<br>`，才能换行。
- 使用单引号 `'` 需要反斜杠 `\` 转义。

## 7、修改

使用该插件，关于页的社交图标下会出现一条横线，需要修改。代码如下：

```
CSS
.about-icons > a:not(:last-child) {
  margin-right: 0.5rem;
  border: none;
}
```

## 8.7 hexo-filter-nofollow

### 1、功能

一个自动为博客中的外链加 `rel="external nofollow noreferrer"` 的插件。

[nofollow标签的使用方法及作用说明 - 萌新SEO \(yrsafe.cn\)](#) 

### 2、GitHub 链接

[hexojs/hexo-filter-nofollow: Add nofollow attribute to all external links automatically. \(github.com\)](#) 

### 3、下载

```
Bash
npm i hexo-filter-nofollow --save
```

### 4、使用

```
nofollow:
  enable: true
  field: site
  exclude: '你的域名'
```

- **field** : 插件的处理范围，默认值为 **site**，可选 **post** 或 **site**。
  - **post** : 仅处理文章内容。
  - **site** : 处理全站所有页面。
- **exclude** : 域名白名单，不同的子域名视为不同的域名。（如 www）
  - **exclude1.com** 不包括 [www.exclude1.com](#) 或 [en.exclude1.com](#)。

参考: [hexo-filter-nofollow - Hexo 官方的 nofollow 插件 | Sukka's Blog \(skk.moe\)](#)



## 8.8 hexo-recommended-posts

### 1、功能

在文章底部，根据分类、标签推荐相关的文章。

### 2、GitHub 链接

[huiwang/hexo-recommended-posts: Keyword based cross blog posts recommendation / Hexo跨博客文章互推插件，共享空间，提升流量 \(github.com\)](#)

### 3、下载

下载插件。

```
npm install hexo-recommended-posts --save
```

下载推荐文章列表。

```
hexo recommend
```

## 4、使用

在 `config.yml` 文件添加以下代码。

```
recommended_posts:
  server: https://api.truelaurel.com           # 后端推荐服务器地址
  timeoutInMillis: 10000                       # 服务时长，超过此时长，则使
用离线推荐模式
  internalLinks: 3                             # 内部文章数量
  externalLinks: 1                             # 外部文章数量
  fixedNumber: false                           # 控制是否返回固定数量的推荐
文章，如果默认推荐文章不够的话会填充当前文章的前后文章作为推荐文章。
  autoDisplay: true                            # 自动在文章底部显示推荐文章
  excludePattern: []                           # 添加想要被过滤的链接的正则
表达式，如配置为 ["example.com"]，则所有包含 example.com 的链接都会从推
荐文章中过滤掉。
  titleHtml: <h2>推荐文章<span style="font-size:0.45em; color:gray">
(由<a href="https://github.com/huiwang/hexo-recommended-posts">
Hexo 文章推荐插件</a>生成) </span></h2>      #
自定义标题
```

参考文章：[魔改hexo博客 | GeekOcean \(godliuyang.wang\)](https://www.geekoc.com/2018/05/20/magic-hexo-blog/)

# 8.9 hexo-wordcount

## 1、功能

文章字数、阅读时长、全站总字数统计。

## 2、GitHub 链接

[willin/hexo-wordcount: A Word Count Plugin for Hexo \(github.com\)](https://github.com/willin/hexo-wordcount) 

## 3、下载

```
Bash
npm i --save hexo-wordcount
```

## 4、使用

01. 在 `config.yml` 文件添加以下代码。

```
YAML
post_wordcount:
  item_text: true
  wordcount: false      # 单篇 字数统计
  min2read: false      # 单篇 阅读时长
  totalcount: true     # 网站 字数统计
```

02. 进入 `Blog\themes\fluid\layout\_partial` 路径，在 `foot.ejs` 文件中加入以下代码。

```
JavaScript
<span class="post-count">本站已有 <%= totalcount(site) %> 字。
</span>
```

03. 修改显示样式，添加以下代码。

```
.post-count {  
  font-size: 85%;  
}
```

## 5、注意

使用 fluid 主题和此插件，字数统计好像只统计**中文**。

## 8.10 hexo-auto-category

看这篇文章 [【Hexo插件系列】日志的自动分类插件 hexo-auto-category | ESON](#)。

个人认为该插件使用前提是一篇文章最好仅有一个分类。

## 8.11 hexo-githubcalendar

### 1、功能

在博客中添加 Github 的 commit 日历。

### 2、GitHub 链接

[Zfour/hexo-github-calendar: 基于hexo的github calendar贡献插件](#)

### 3、下载

```
npm i hexo-githubcalendar --save
```

### 3、使用

01. 在 `config.yml` 文件添加以下代码。

第一种在关于页展示。

YAML

```
# Ice Kano Plus_in
# Hexo Github Canlendar
# Author: Ice Kano
# Modify: Lete乐特
githubcalendar:
  enable: true # true/false,是否开启插件
  enable_page: /about/ # 地址, /: 网站首页, /about/: 关于
  页。
  user: ziguangqing # github 或者 gitee 用户名
  layout:
    type: id
    name: recent-posts
    index: 0
  githubcalendar_html: '<div class="recent-post-item"
style="width:100%;height:auto;padding:10px;"><div
id="github_loading" style="width:10%;height:100%;margin:0
auto;display: block"><svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" viewBox="0 0 50 50"
style="enable-background:new 0 0 50 50" xml:space="preserve">
<path fill="#d0d0d0" d="M25.251,6.461c-10.318,0-18.683,8.365-
18.683,18.683h4.068c0-8.071,6.543-14.615,14.615-14.615V6.461z"
transform="rotate(275.098 25 25)"><animateTransform
attributeType="xml" attributeName="transform" type="rotate"
from="0 25 25" to="360 25 25" dur="0.6s"
repeatCount="indefinite"></animateTransform></path></svg></div>
<div id="github_container"></div></div>'
  pc_minheight: 280px
  mobile_minheight: 0px
  color:
    # - "['#ebedf0', '#fdcdcd', '#fc9bd9', '#fa6ac5', '#f838b2',
'#f5089f', '#c4067e', '#92055e', '#540336', '#48022f',
'#30021f']" #紫红色
```

```

    - "[ '#e4dfd7', '#f9f4dc', '#f7e8aa', '#f7e8aa', '#f8df72',
'#fcd217', '#fcc515', '#f28e16', '#fb8b05', '#d85916',
'#f43e06' ]" #橘黄色
    # - "[ '#ebedf0', '#fdcdec', '#fc9bd9', '#fa6ac5', '#f838b2',
'#f5089f', '#c4067e', '#92055e', '#540336', '#48022f',
'#30021f' ]" #浅紫色
    # - "[ '#ebedf0', '#f0fff4', '#dcffe4', '#bef5cb', '#85e89d',
'#34d058', '#28a745', '#22863a', '#176f2c', '#165c26',
'#144620' ]" #翠绿色
    # - "[ '#ebedf0', '#f1f8ff', '#dbedff', '#c8e1ff', '#79b8ff',
'#2188ff', '#0366d6', '#005cc5', '#044289', '#032f62',
'#05264c' ]" #天青色
    api: https://python-github-calendar-api.vercel.app/api
    # api: https://python-gitee-calendar-api.vercel.app/api
    calendar_js: https://cdn.jsdelivr.net/gh/Zfour/hexo-github-
calendar@1.21/hexo_githubcalendar.js
    plus_style: ""

```

第二种在**首页**展示。

```

# Fluid
githubcalendar:
  enable: true
  priority: 0
  enable_page: /
  user: ziguangqing
  layout:
    type: class
    name: container
    index: 1

```

YAML

```
githubcalendar_html: '<div id="github-calendar"
style="width:100%;height:auto;padding:10px;margin-bottom:20px">
<div id="github_loading" style="width:10%;height:100%;margin:0
auto;display: block"><svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" viewBox="0 0 50 50"
style="enable-background:new 0 0 50 50" xml:space="preserve">
<path fill="#d0d0d0" d="M25.251,6.461c-10.318,0-18.683,8.365-
18.683,18.683h4.068c0-8.071,6.543-14.615,14.615-14.615V6.461z"
transform="rotate(275.098 25 25)"><animateTransform
attributeType="xml" attributeName="transform" type="rotate"
from="0 25 25" to="360 25 25" dur="0.6s"
repeatCount="indefinite"></animateTransform></path></svg></div>
<div id="github_container"></div></div>'

pc_minheight: 200px
mobile_minheight: 0px
color:
  # - "[ '#ebedf0', '#fdcddec', '#fc9bd9', '#fa6ac5', '#f838b2',
'#f5089f', '#c4067e', '#92055e', '#540336', '#48022f',
'#30021f' ]" #紫红色
  - "[ '#e4dfd7', '#f9f4dc', '#f7e8aa', '#f7e8aa', '#f8df72',
'#fcd217', '#fcc515', '#f28e16', '#fb8b05', '#d85916',
'#f43e06' ]" #橘黄色
  # - "[ '#ebedf0', '#fdcddec', '#fc9bd9', '#fa6ac5', '#f838b2',
'#f5089f', '#c4067e', '#92055e', '#540336', '#48022f',
'#30021f' ]" #浅紫色
  # - "[ '#ebedf0', '#f0fff4', '#dcffe4', '#bef5cb', '#85e89d',
'#34d058', '#28a745', '#22863a', '#176f2c', '#165c26',
'#144620' ]" #翠绿色
  # - "[ '#ebedf0', '#f1f8ff', '#dbedff', '#c8e1ff', '#79b8ff',
'#2188ff', '#0366d6', '#005cc5', '#044289', '#032f62',
'#05264c' ]" #天青色

api: https://python-github-calendar-api.vercel.app/api
# api: https://python-gitee-calendar-api.vercel.app/api
calendar_js: https://cdn.jsdelivr.net/gh/Zfour/hexo-github-
calendar@1.21/hexo_githubcalendar.js
```

```
plus_style: "#github_container > .position-relative >
.border{border:0!important}#github-calendar{position:
relative;margin-top: -2rem;background-color: var(--board-bg-
color);transition: background-color 0.2s ease-in-out;border-
radius: 0.5rem;z-index: 3;-webkit-box-shadow: 0 12px 15px 0
rgb(0 0 0 / 24%), 0 17px 50px 0 rgb(0 0 0 / 19%);box-shadow: 0
12px 15px 0 rgb(0 0 0 / 24%), 0 17px 50px 0 rgb(0 0 0 / 19%);}"
```

更多配置请看文章：[教程：hexo-githubcalendar 插件 1.0 | 小冰博客 \(zfe.space\)](#)

## 参考

- [「推荐」本站用到的 hexo 插件 | 若风 \(loafing.cn\)](#)
- [Hexo 插件推荐 | Merrier 说](#)
- [Hexo 常用插件推荐及使用方法 | 陈亮的个人博客 \(itchenliang.club\)](#)
- [Hexo 博客 | 插件推荐 | Justlovesmile's BLOG](#)
- [Hexo 博客优化 —— 推荐插件 | 爱爬虫与大数据 | 张凯恒的个人博客 \(zhangkh.top\)](#)
- [Hexo 博客配置升级集合 | 默 \(jasper1024.com\)](#)
- [魔改 hexo 博客 | GeekOcean \(godliuyang.wang\)](#)
- [Hexo 搭建个人博客系列：进阶设置篇 | Yearito's Blog](#)
- [分享两个实用的 Hexo 插件 | Gahotx's blog](#)

# 第九章 自动化部署

[使用 GitHub Actions 实现博客的自动化部署 - 镜湖月的小世界 \(jinghumoon.top\)](#) 

## 9.1 自动化部署解释

### 什么是自动化部署？

简而言之，就是远端部署不在本地执行，在 GitHub 上进行。在本地部署 Hexo，每次都需要部署指令三件套，比较麻烦。且随着文章数目的增多，在本地部署 Hexo 耗费的时间越来越长。不如将这一步骤交给 GitHub 进行。我们只需要在本地写好文章，上传代码到 GitHub 即可。

这两个指令，我分享到了 [Quicker](#) ，有需要的朋友可以使用。

[一键向 Git Bash here 界面发送 Hexo 远端部署命令](#) 

[一键向 Git Bash here 界面发送 Hexo 本地预览命令](#) 

## 9.2 GitHub Actions 简介

[Github Action](#)  是 GitHub 于 2018 年 10 月推出的一个 CI/CD 服务。GitHub Actions 让我们轻松实现所有软件工作流程的自动化，从 GitHub 上直接构建、测试和部署我们的代码。让代码审查、分支管理和问题处理以我们想要的方式进行。

## 9.2.1 什么是 CI/CD

CI/CD 是三件事情：持续集成 — **Continuous Integration**、持续交付 — **Continuous Delivery**、持续部署 — **Continuous Deployment**。因为「持续交付」和「持续部署」的英文缩写一致，所以这三件事情缩写成 **CI/CD**。

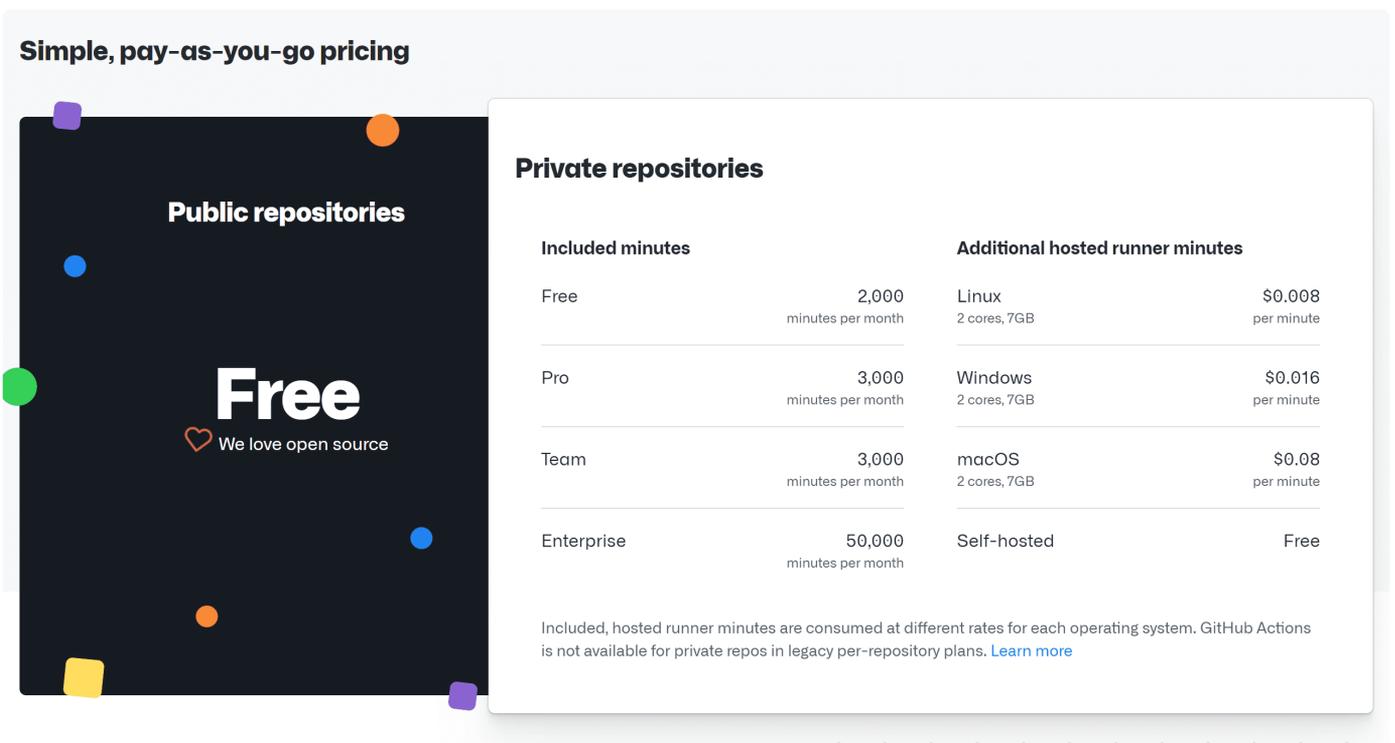
## 9.2.2 使用教程

- [GitHub Actions 入门教程 - 阮一峰的网络日志 \(ruanyifeng.com\)](#)
- [【CI/CD】github新功能actions全方位讲解！！ - bilibili](#)

## 9.2.3 费用计算

- 公开仓库，免费使用，无限制。
- 私有仓库，如下图。

Simple, pay-as-you-go pricing



Included minutes		Additional hosted runner minutes	
Free	2,000 minutes per month	Linux 2 cores, 7GB	\$0.008 per minute
Pro	3,000 minutes per month	Windows 2 cores, 7GB	\$0.016 per minute
Team	3,000 minutes per month	macOS 2 cores, 7GB	\$0.08 per minute
Enterprise	50,000 minutes per month	Self-hosted	Free

Included, hosted runner minutes are consumed at different rates for each operating system. GitHub Actions is not available for private repos in legacy per-repository plans. [Learn more](#)

简单解释下计费规则。以下内容均来自 [管理 GitHub Actions 的计费 - GitHub Docs](#)

- 在 GitHub 主机的 Windows 运行器上运行的作业，其消耗分钟数是在 Linux 运行器上运行的作业的 2 倍。即使用 1000 Windows 分钟将消耗帐

户中包含的 2,000 分钟。

- 我们的 GitHub Actions 运行在 Linux 运行器。经测试，一次 Actions 大约需要 1 分钟。即，一个月我们可以提交代码 2000 次。平均每天接近 70 次。
- 当然，随着博客内容的增多，Actions 时间也会增加。假设需要需要 5 分钟。一天也可以提交 10 来次代码。足够了。如果还不够，用公开仓库。  
(不推荐)
- 「GitHub 主页」→「Settings」→「点击 Billing and plans 一栏」→「右侧，下滑，Actions 处」，查看使用记录详情。注意这里的计费，有缓存机制。

下面开始介绍自动化部署流程。

## 1.3 配置 Github SSH

理论上来说，搭建过以 Hexo 为框架的博客。应该是配置过 Github SSH。如果没有，请点击下面的链接，配置 Github SSH。其中 `id_rsa` 是私钥，`id_rsa.pub` 是公钥。

[Github SSH 配置 - 镜湖月的小世界 \(jinghumoon.top\)](#) 

## 1.4 新建仓库

建立一个**私有**仓库，存放博客源码。

# Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*      Repository name \*

 jinghu-moon / Blog\_Source\_Files 

Great repository names are short and memorable. Need inspiration? How about [glowing-computing-machine?](#)

Description (optional)

我的博客源文件。

-  **Public**  
Anyone on the internet can see this repository. You choose who can commit.
-  **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

- Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)
- Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)
- Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  `main` as the default branch. Change the default name in your [settings](#).

[Create repository](#)

然后拉取到本地。

## 法一

- 新建文件夹，进入，右键打开 Git Bash here。

```
Bash
git clone git@github.com:用户名/仓库名.git
```

## 法二

- 点击绿色的 `code` 按钮。下载压缩包。

打开该文件。理论上里面应该有 `.git` 和 `README.md` 两个文件。全选，复制到本地博客文件夹里。

没有，则说明没有勾选显示隐藏文件。

显示隐藏文件：打开文件管理器 → 点击左上方的『查看』 → 右侧，勾选『显示隐藏的项目』

## 1.5 配置私钥

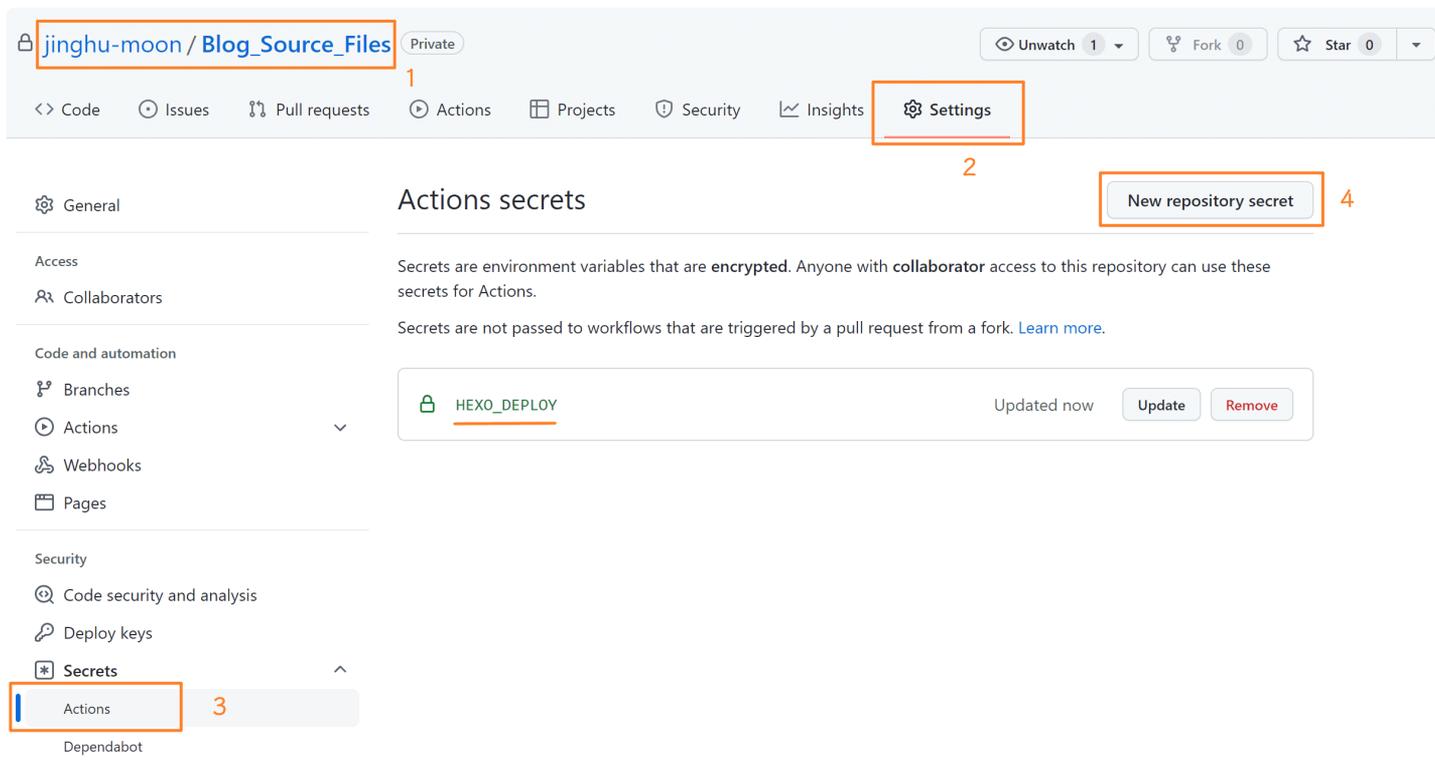
在 Github SSH 配置环节，我们生成了密钥，并使用了公钥。现在我们将使用私钥。

### 1、复制私钥

进入 `C:\Users\用户名\.ssh` 路径，用记事本打开 `id_rsa`，复制里面的内容。

### 2、新建私钥

「博客源文件仓库」 → 「Settings」 → 「Actions」 → 「New repository secret」。私钥命名为 `HEXO_DEPLOY`，把私钥内容复制进去。



## 1.6 编写 GitHub Actions

以下内容，不懂也没关系。我也不懂 🤔。直接复制，稍作修改，就行。如果你想了解 GitHub Actions 的使用，请看文章开头。我给了相应的链接。

在**本地**的博客文件夹内，进入 `\.github\workflows` 路径，没有的话，就建立该路径。新建一个 `auto_deploy.yml` 文件。打开 `Blog\.github\workflows\auto_deploy.yml`，添加以下内容。

### 注意

- 用户名，GitHub 邮箱记得修改成自己的。
- 必须安装 **hexo-deployer-git** 插件。

```
YAML

# workflow name
name: 博客自动化部署

# push 代码至 main 时，运行。
on:
```

push:

branches:

- main

jobs:

build:

# 运行环境

runs-on: ubuntu-latest

steps:

# from: <https://github.com/actions/checkout>

- name: 检查分支

uses: actions/checkout@master

# from: <https://github.com/actions/setup-node>

- name: 安装 Node.js

uses: actions/setup-node@master

with:

node-version: "12.x"

- name: 安装 Hexo

run: |

npm install hexo-cli -g

npm install

- name: 缓存 Hexo

uses: actions/cache@v2

id: cache

with:

path: node\_modules

key: \${{runner.OS}}-\${{hashFiles('\*\*/package-lock.json')}}

- name: 安装依赖

if: steps.cache.outputs.cache-hit != 'true'

run: |

npm install --save

```
- name: 生成私钥
  env:
    HEXO_DEPLOY_PRIVATE_KEY: ${{ secrets.HEXO_DEPLOY }}
  run: |
    mkdir -p ~/.ssh/
    echo "$HEXO_DEPLOY" > ~/.ssh/id_rsa
    chmod 600 ~/.ssh/id_rsa
    ssh-keyscan github.com >> ~/.ssh/known_hosts

- name: 配置 Git 信息
  run: |
    git config --global user.name '用户名'
    git config --global user.email 'GitHub 邮箱'

- name: 部署 Hexo
  run: |
    hexo clean
    hexo generate
    hexo deploy
```

## 1.7 设置 .gitignore

### 什么是 .gitignore?

.gitignore 文件的作用是，在上传代码到 GitHub 仓库时，指定某些文件不上传。

使用方法: [Git忽略文件.gitignore的使用 - 简书 \(jianshu.com\)](https://jianshu.com/p/1e1e1e1e) 

在本地博客源文件里，打开 `.gitignore` 文件，复制以下代码。

```
Bash
.DS_Store
Thumbs.db
db.json
*.log
node_modules/
public/
.deploy*/
.deploy_git*/
themes/主题名/.git
```

`node_modules` : Hexo 的一些缓存信息和插件的代码文件。占用空间很大。GitHub Actions 会生成，不上传。

`public` : 生成的 HTML 文件和一些静态资源文件。GitHub Actions 会生成，不上传。

`themes/主题名/.git` : 上传此文件，themes 下的文件不显示。GitHub 会默认这些文件在你的某一个仓库里。当然，你也可以直接删除这个 `.git` 文件。推荐后者。

Github 仓库容量限制在 1G，单个文件不能超过 100M。

「GitHub 主页」→「Settings」→「Repositories」，查看仓库已使用容量。

## 1.8 推送代码

1、添加所有文件至缓冲区

```
Bash
git add .
```

2、添加所有文件至归档区

```
Bash
git commit -m '代码提交相关信息'
```

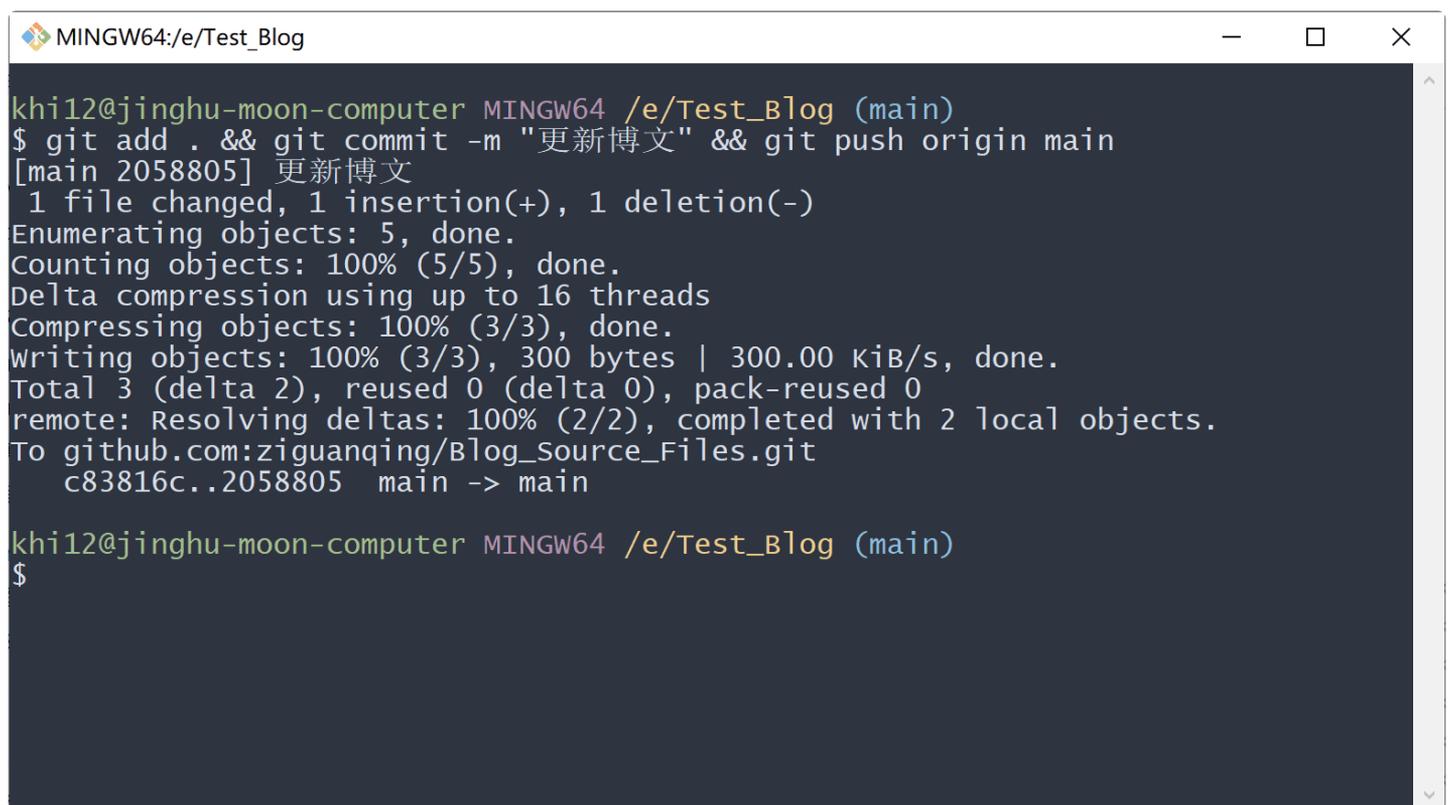
### 3、文件上传至远程仓库

```
Bash
git push origin -u main
```

第一次上传，需要使用以上代码。后面再次提交，使用以下代码。

```
Bash
git push / git push origin main
```

推送成功截图如下。



```
MINGW64:/e/Test_Blog
khi12@jinghu-moon-computer MINGW64 /e/Test_Blog (main)
$ git add . && git commit -m "更新博文" && git push origin main
[main 2058805] 更新博文
1 file changed, 1 insertion(+), 1 deletion(-)
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 300 bytes | 300.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:ziguanqing/Blog_Source_Files.git
 c83816c..2058805  main -> main
khi12@jinghu-moon-computer MINGW64 /e/Test_Blog (main)
$
```

在推送代码过程中，出现报错，请复制报错信息，百度搜索解决办法。

在如下图处查看 GitHub Actions 是否正常运行。

ziguangqing / Blog\_Source\_Files Private

Unwatch 1 Fork 0 Star

Code Issues Pull requests **Actions** Projects Security Insights Settings

Workflows New workflow

All workflows

博客自动化部署

All workflows

Showing runs from all workflows

Filter workflow runs

20 workflow runs

	Event	Status	Branch
<input checked="" type="checkbox"/> 更新博文 打勾, 则代表运行成功! 博客自动化部署 #20: Commit b6177c7 pushed by ziguangqing	main	Success	1 minute ago 41s
<input checked="" type="checkbox"/> 更新博文 博客自动化部署 #19: Commit 2058805 pushed by ziguangqing	main	Success	4 minutes ago 42s
<input checked="" type="checkbox"/> 更新博文 博客自动化部署 #18: Commit c83816c pushed by ziguangqing	main	Success	5 minutes ago 41s
<input checked="" type="checkbox"/> 更新博文 博客自动化部署 #17: Commit 8d6b764 pushed by ziguangqing	main	Success	1 hour ago 41s
<input checked="" type="checkbox"/> 更新博文			

以后再推送文件到 GitHub 上，使用以下代码。

```
Bash
git add .
git commit -m "更新博文"
git push origin main
```

## 1.9 简化推送命令

### 1.9.1 多条命令合一

```
Bash
git add . && git commit -m "更新博文" && git push origin main
```

### 1.9.2 Git 命令设置简写（别名）

参考资料: [Git - Git 别名 \(git-scm.com\)](https://git-scm.com/docs/git-alias)

进入 C:\Users\Username 路径，打开 `.gitconfig` 文件。添加以下代码。

```
Bash
[alias]
  ad = add .
  cmm = commit -m
  st = status
  pl = pull
  ps = push
```

找不到的话，在 **Git Base here** 界面，输入以下代码。

```
Bash
git config --global alias.别名 "命令全称"
```

```
Bash
git config --global alias.ad 'add .'
git config --global alias.br 'branch'
git config --global alias.cmm 'commit -m'
git config --global alias.pl 'pull'
git config --global alias.ps 'push'
```

现在推送文件，可以使用以下代码。

```
Bash
git ad && git cmm "更新博文" && git ps
```

## 1.9.3 使用 Quicker 动作

[Quicker](#) ，您的指尖工具箱。操作更少，收获更多！推荐使用。

[文档 - Quicker \(getquicker.net\)](#) 

下面是我的动作。

[一键向 Git Bash here 界面发送 Git Push 命令](#) 

## 1.9.4 使用脚本

### 1、bat 脚本

新建 `Push.txt` 文本文件，添加以下代码。

```
TXT

@echo off
echo ===== Begin
=====
echo=
set Default_msg=更新博客
set "Input_msg="
set "Submit_msg="
echo (1) 拉取远端文件
echo =====
echo 拉取中...
echo=
git pull
echo=
echo 拉取成功!
echo=
echo (2) 查看文件状态
echo =====
git status
echo=
echo [
echo | untrack: 新建文件   modified: 已修改文件 |
echo | staged: 暂存文件   committed: 已提交文件 |
echo |
echo ]
echo=
echo (3) 提交本地文件
echo =====
echo 输入 0，提交信息默认为更新博客。输入多条提交信息，请用逗号隔开。
echo=
set /p Input_msg=提交信息:
```

```

echo=
if %Input_msg% equ 0 (
    set Submit_msg=%Default_msg%
) else (
    set Submit_msg=%Input_msg%
)
git add .
git commit -m "%Submit_msg%"
git push
echo=
echo (4) 本次更新内容:
echo =====
set remain=%Submit_msg%
:loop
for /f "tokens=1* delims=" %%a in ("%remain%") do (
    echo | %%a
    set remain=%%b
)
if defined remain goto :loop
echo=
echo ===== End
=====
pause

```

保存时，以 **ANSI** 编码方式保存。然后，后缀改为 **bat**，放入本地博客文件夹。推送代码时，双击 **Push.bat** 文件，输入提交信息，回车即可。

## 2、使用 shell 脚本

[hexo.sh: 一个方便管理与发布hexo博客的脚本 - XAOXUU](#) 

### 1.10 注意

使用 GitHub Actions 进行博客的自动化部署，文章最后更新时间会变成 GitHub Actions 的执行时间。所以需要自己指定更新时间。

```
updated: xxxx-xx-xx xx:xx:xx
```

手动输入，好麻烦。虽然输入法可以实现输入 `shijian`，输入当前时间。但是格式需要修改，推荐使用我的 [Quicker](#) 动作。

[一键向当前页面输入当前时间](#)

## 参考

- [利用 GitHub+Actions 自动部署 Hexo 博客 | 小康博客 \(antmoe.com\)](#)
- [利用 Github Action 实现 hexo 博客的自动部署 - 黑石博客 - Hexo 博客 \(heson10.com\)](#)
- [利用 Travis-CI 实现在线更新 Hexo || 陈YFの博客\( - ▽ - \)" \(cyfan.top\)](#)
- [GitHub Action 实现 Hexo 博客持续集成 \(qingxu.live\)](#)
- [利用 GitHub Action 自动部署 Hexo 博客 - YFun's Blog](#)
- [使用 Github Action 实现全自动部署 | Akilarの糖果屋](#)

# 第十章 网站 SEO

## 10.1 什么是 SEO

SEO (Search Engine Optimization) ，汉译为搜索引擎优化。是一种利用 [搜索引擎](#) 的规则提高网站在有关搜索引擎内的 [自然排名](#) 的行为。很大程度上是网站经营者的一种商业行为，将自己或自己公司网站的排名前移。

现如今，个人博客式微，我想大部分搭建博客的个人博主，都应该有分享欲。博客是为了记录生活、知识，写的文章也希望可以分享给有需要的人。但是以 Hexo 为框架搭建的静态博客，做 SEO 不像动态网站那么容易。你不主动向搜索引擎提交收录，搜索引擎基本搜不到你写的文章。

下面讲一讲静态网站如何向搜索引擎提交收录和进行 SEO 优化。

## 10.2 搜索引擎收录

个人博主通常使用两种域名：[xxx.xxx.com](#) 和 [xxx.com](#) 。关于这两种域名，到底哪种域名的 SEO 更好，众说纷纭。可以看看下面的链接。

[为什么越来越多的网站域名不加「www」前缀？](#)

### 10.2.1 百度收录

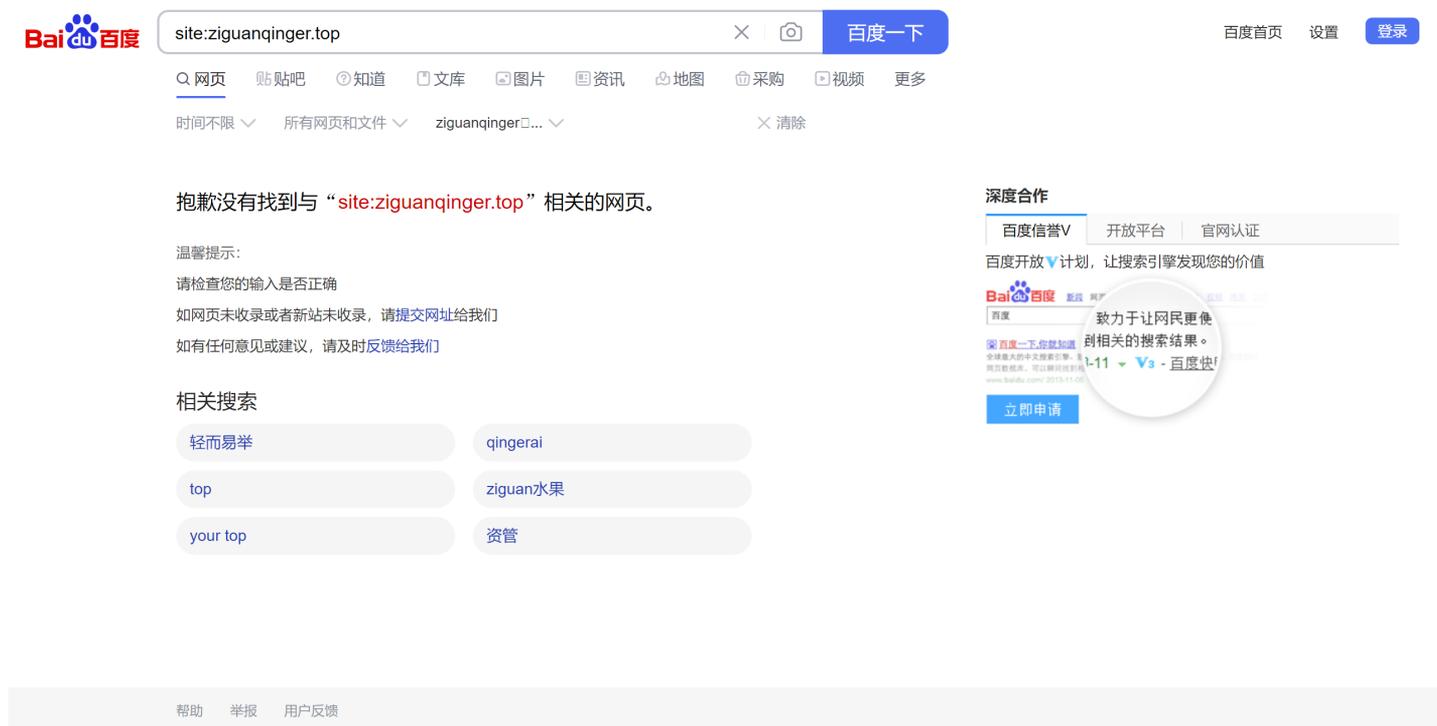
基于 GitHub Pages 的个人博客，Google 收录非常及时全面。然而，到目前为止，GitHub 还是拒绝百度爬虫的访问，直接返回 403。官方给出原因：百度爬虫爬得太狠，影响了 Github Pages 服务的正常使用。这就导致，在 Github Pages 上搭建的个人博客，都无法被百度收录。也就是 [xxxx.github.io](#) 的域名，百度不会收录。

当然，这是可以解决的 —— 建立 gitee、coding 镜像站。但是新版 Coding 被腾讯云收购，静态页面的部署依赖于腾讯云的对象存储功能，对于新用户会提供 **6 个月的免费流量包**，之后**开始收费**；而 gitee 开通 gitee pages 需要**手持身份证拍摄照片**。如果可以接受。请看下面的文章链接。

[使用 Gitee 提供镜像站支持 - Akilar.top](#)

[使用 Coding 和 Github 来实现双线部署 - Akilar.top](#)

进入 [百度](#)，搜索 **site:域名**，如下结果，则说明百度没有收录你的域名。



下面开始把你的网站提交给百度。

进入 [百度站长工具](#) 网站。按照网站提示，添加网站。验证网站可选【文件验证】、【HTML 标签验证】、【CNAME 验证】。

- 文件验证

下载「验证文件」，放入 Blog\source 路径。打开该文件，添加以下代码：

```
---
layout: false
---
```

然后 Hexo 部署指令三连，完成验证。

- HTML 标签验证

复制 HTML 标签验证，进入 `Blog\themes\fluid\layout\_partial` 路径，打开 `head.ejs` 文件，添加复制的 HTML 标签验证。

```
<!-- 添加位置 -->
<meta name="baidu-site-verification" content="code-???????" />
<title><%= title %></title>
```

然后 Hexo 部署指令三连，完成验证。

- CNAME 验证

不推荐使用，要占一个 CNAME 记录。

## 10.2.2 必应收录

进入 [必应](#)，搜索 `site:域名`，如下结果，则说明必应没有收录你的域名。

没有与此相关的结果: **site:jinghu-moon.top**

没有找到您要的结果  
但我们可以帮他找到亲人



姓名: 赵斌斌

性别: 男

出生日期: 1988-12-09

详情描述: 失踪时5岁(现在年21岁), 右侧眼眉上有一儿时碰痕(很轻微, 可能消失), 记忆最深刻词汇是一些地名和父母的小名, 有: “天水市”、“娘娘坝乡”、

联系方式: 13934556338

[\[详情\]](#)

[帮助更多儿童 404 公益](#)



@微博打拐

如果有人看到了这个小孩,  
请与他的家长联系, 非常感谢!

下面开始把你的网站提交给必应。

进入 [Bing Webmaster Tools](#) 网站, 按照网站提示, 添加网站。验证网站可选【文件验证】、【HTML 标签验证】、【CNAME 验证】。

验证方式与百度收录差不多, 【文件验证】不需要在验证文件添加代码。

## 10.2.3 谷歌收录

进入 [谷歌](#), 搜索 `site:域名`, 如下结果, 则说明必应没有收录你的域名。

### 尝试使用 Google Search Console

www.google.com/webmasters/

您对 **jinghumoon.top** 是否具有所有权? 请从 Google 获取索引和排名数据。

找不到和您查询的“**site:jinghumoon.top**”相符的内容或信息。

建议:

- 请检查输入字词有无错误。
- 请尝试其他查询词。
- 请改用较常见的字词。



下面开始把你的网站提交给谷歌。

进入[谷歌站长](#)  网站。

## 1、使用 sitemap 插件

这个插件可以生成适合搜索引擎优化的站点地图。

[ludoviclefevre/hexo-generator-seo-friendly-sitemap \(github.com\)](#) 



Bash

```
npm install hexo-generator-seo-friendly-sitemap --save
```

打开 `_config.yml` 文件，添加以下代码。

```
sitemap:  
  path: sitemap.xml  
  tag: false # 排除标签  
  category: false # 排除分类
```

hexo g 后，生成 sitemap.xml、page-sitemap.xml、category-sitemap.xml、post-sitemap.xml、sitemap.xml、tag-sitemap.xml。

## 10.3 SEO 优化

### 10.3.1 网站信息设置

各项参数按照 [配置 | Hexo](#) 说明如下：

参数	描述
title	网站标题。
subtitle	网站副标题。
description	网站描述。
keywords	网站的关键词。支持多个关键词。
author	您的名字。
language	网站使用的语言。对于简体中文用户来说，使用不同的主题可能需要设置成不同的值，请参考你的主题的文档自行设置，常见的有 zh-Hans 和 zh-CN。

参数	描述
<code>timezone</code>	网站时区。Hexo 默认使用您电脑的时区。请参考 <a href="#">时区列表</a> 进行设置，如 <code>America/New_York</code> ， <code>Japan</code> ，和 <code>UTC</code> 。一般的，对于中国大陆地区可以使用 <code>Asia/Shanghai</code> 。

## 示例

```
YAML
# Site
title: "Guan Qing 的博客" # 网站标题
subtitle: "记录生活" # 网站副标题
description: "Guan Qing's Blog,Guan Qing 的博客,观清的小世界" # 网站描述
keywords: "观清,博客,Blog,Hexo,日志,Guan Qing" # 网站关键词
author: "观清" # 作者
language: "zh-CN" # 语言
timezone: "Asia/Shanghai" # 时区
```

## 10.3.2 优化文章链接

Hexo 文章的默认链接是 `域名/年/月/日/文章名`，如果修改了文章名，原先的文章链接就失效了。这非常不利于 SEO。所以我们需要指定文章的**永久链接**。

使用方法：[hexo-abbrevlink](#)。

## 10.3.3 nofollow 标签

[nofollow标签的使用方法与作用说明 - 萌新SEO \(yrsafe.cn\)](#)

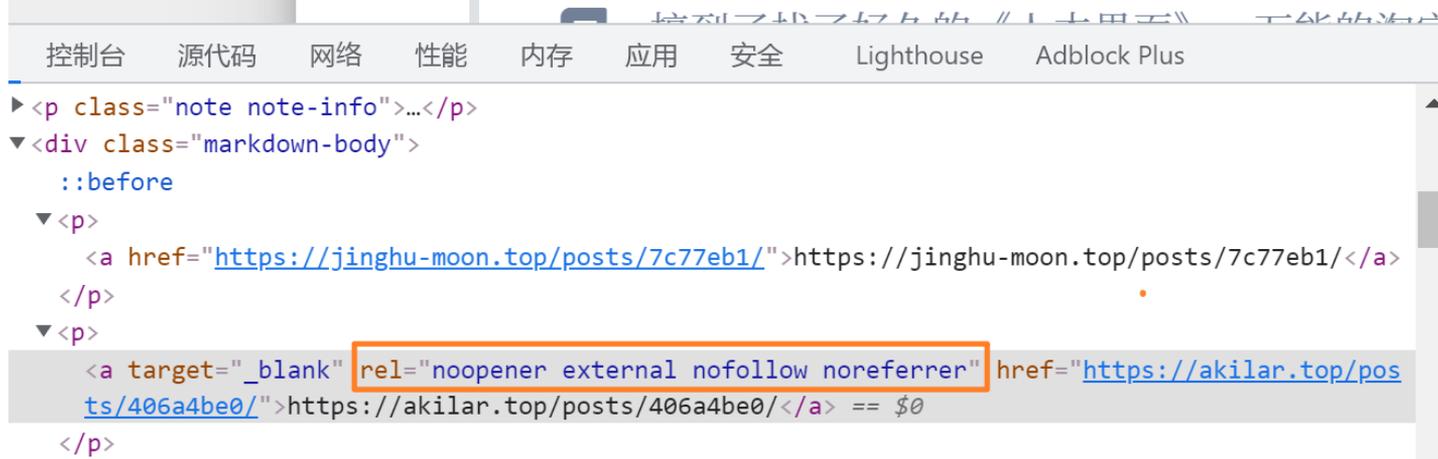
`hexo-filter-nofollow` 插件会为你的博客中的外链自动添加 `rel="external nofollow noreferrer"` 属性，从而改善网站的安全性和 SEO。链接类型可以查看 [链接类型 - HTML \(超文本标记语言\) | MDN \(mozilla.org\)](#)。

使用方法: [hexo-filter-nofollow](#)。使用后效果如下。

<https://jinghu-moon.top/posts/7c77eb1/>

<https://akilar.top/posts/406a4be0/>

以下是时间线测试。

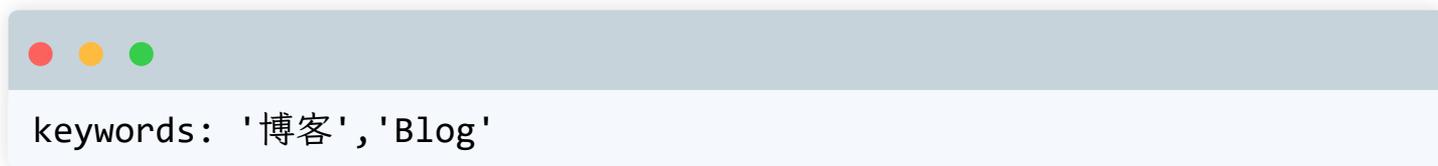


```
<p class="note note-info">...</p>
<div class="markdown-body">
  ::before
  <p>
    <a href="https://jinghu-moon.top/posts/7c77eb1/">https://jinghu-moon.top/posts/7c77eb1/</a>
  </p>
  <p>
    <a target="_blank" rel="noopener external nofollow noreferrer" href="https://akilar.top/posts/406a4be0/">https://akilar.top/posts/406a4be0/</a> == $0
  </p>
```

[关于a标签target\\_blank使用rel=noopener - 简书 \(jianshu.com\)](#) 

## 10.3.4 Front-matter

Front-matter 里面添加 `keywords` 属性, 填写文章的关键词。



```
keywords: '博客', 'Blog'
```

# 第十一章 提高网站加载速度

GitHub 在国内访问，时好时坏，基于 GitHub 搭建的博客自然也是如此。所以下面讲一些方法提高网站的加载速度。

## 11.1 常用测速网站

### • 测速网站

- [多个地点Ping服务器,网站测速 - 站长工具 \(chinaz.com\)](#) 
- [免费域名检测-网站测速-ping检测-域名污染-dns查询-路由跟踪查询 \(boce.com\)](#) 
- [网站测速|网站速度测试|网速测试|电信|联通|网通|全国|监控|CDN|PING|DNS 17CE.COM](#) 

### • 网站评分

- [PageSpeed Insights \(web.dev\)](#) 

谷歌推出的网站测速网。会给你的网站打分，并给出优化建议。需要「科学上网」。

- [PageSpeed Insights \(huran.icu\)](#) 

上面网站的镜像站。

- [测速网,网站测速,在线网页测速工具,网络宽带测速-谷歌测速网 \(aicesu.cn\)](#) 

## 11.2 压缩静态资源

## 11.2.1 Gulp 压缩文件

[Gulp](#) 能够帮助用户自动压缩静态资源，配合各类下属插件，能够压缩包括 CSS、JS、HTML 和各类格式的图片。但图片文件的压缩效果远不如 `image`、`tinypng` 等压缩方式，不建议使用 Gulp 压缩图片。

以下操作都在 Git Bash here 界面进行。

### 1、安装 Gulp 插件

```
Bash
# 全局安装 gulp 指令集
npm install --global gulp-cli
# 安装 gulp 插件
npm install gulp --save
```

### 2、安装压缩 HTML 的插件

```
Bash
npm install gulp-htmlclean --save-dev
# 用 gulp-html-minifier-teraser 可以压缩 HTML 中的 ES6 语法
npm install gulp-html-minifier-teraser --save-dev
```

### 3、安装压缩 CSS 的插件

```
Bash
npm install gulp-clean-css --save-dev
```

### 4、安装压缩 JS 的插件

```
Bash
npm install gulp-teraser --save-dev
```

### 5、安装压缩字体插件

```
npm install gulp-fontmin --save-dev
```

## 6、创建 `gulpfile.js` 任务脚本

在博客根目录下新建 `gulpfile.js` ，打开 `gulpfile.js` ，添加以下内容。

```
//用到的各个插件
var gulp = require("gulp");
var cleanCSS = require("gulp-clean-css");
var htmlmin = require("gulp-html-minifier-terser");
var htmlclean = require("gulp-htmlclean");
var fontmin = require("gulp-fontmin");
// gulp-tester
var terser = require("gulp-terser");

// 压缩js
gulp.task("compress", () =>
  gulp
    .src(["./public/**/*.js", "!./public/**/*.min.js"])
    .pipe(terser())
    .pipe(gulp.dest("./public"))
);

//压缩css
gulp.task("minify-css", () => {
  return gulp
    .src(["./public/**/*.css"])
    .pipe(
      cleanCSS({
        compatibility: "ie11",
      })
    )
    .pipe(gulp.dest("./public"));
});
```

```

});

//压缩html
gulp.task("minify-html", () => {
  return gulp
    .src("./public/**/*.html")
    .pipe(htmlclean())
    .pipe(
      htmlmin({
        removeComments: true, //清除html注释
        collapseWhitespace: true, //压缩html
        collapseBooleanAttributes: true,
        //省略布尔属性的值, 例如: <input checked="true"/> ==>
        <input />
        removeEmptyAttributes: true,
        //删除所有空格作属性值, 例如: <input id="" /> ==>
        <input />
        removeScriptTypeAttributes: true,
        //删除<script>的type="text/javascript"
        removeStyleLinkTypeAttributes: true,
        //删除<style>和<link>的 type="text/css"
        minifyJS: true, //压缩页面 JS
        minifyCSS: true, //压缩页面 CSS
        minifyURLs: true, //压缩页面URL
      })
    )
    .pipe(gulp.dest("./public"));
});

//压缩字体
function minifyFont(text, cb) {
  gulp
    .src("./public/fonts/*.ttf") //原字体所在目录
    .pipe(
      fontmin({
        text: text,

```

```

        })
    )
    .pipe(gulp.dest("./public/fonts")) //压缩后的输出目录
    .on("end", cb);
}

gulp.task("mini-font", (cb) => {
    var buffers = [];
    gulp
        .src(["./public/**/*.*.html"]) //HTML文件所在目录请根据自身情况修改
        .on("data", function (file) {
            buffers.push(file.contents);
        })
        .on("end", function () {
            var text = Buffer.concat(buffers).toString("utf-8");
            minifyFont(text, cb);
        });
});

// 运行gulp命令时依次执行以下任务
gulp.task(
    "default",
    gulp.parallel("compress", "minify-css", "minify-html", "mini-
font")
);

```

**注意：**压缩字体的格式只能为 `ttf` 格式，且字体应该存放在 `source\fonts` 目录下。你也可以按照你的文件存放位置修改 `gulpfile.js`。

## 7、压缩静态文件

打开 Git Bash here 界面，输入以下指令，开始压缩静态文件。

前提：已 `hexo g` 生成静态文件。

```
Bash
gulp

MINGW64:/e/A_Test_Blog
khi12@jinghu-moon-computer MINGW64 /e/A_Test_Blog (main)
$ gulp
[20:44:57] Using gulpfile E:\A_Test_Blog\gulpfile.js
[20:44:57] Starting 'default'...
[20:44:57] Starting 'compress'...
[20:44:57] Starting 'minify-css'...
[20:44:57] Starting 'minify-html'...
[20:44:57] Starting 'mini-font'...
[20:44:58] Finished 'minify-css' after 1.01 s
[20:44:59] Finished 'compress' after 1.16 s
[20:45:00] gulp-fontmin: Minified 1 font
[20:45:00] Finished 'minify-html' after 2.3 s
[20:45:00] Finished 'mini-font' after 2.31 s
[20:45:00] Finished 'default' after 2.31 s

khi12@jinghu-moon-computer MINGW64 /e/A_Test_Blog (main)
$
```

如果存在字体文件，压缩效率可达 70%。

使用 Gulp 后，命令如下：

```
Bash
hexo cl && hexo g && gulp && hexo d
hexo cl && hexo g && gulp && hexo s
```

**参考文章：** [使用 gulp 压缩博客静态资源 - Akilar.top](#)

如果有什么不懂的或者压缩过程报错，建议看参考文章，很详细。

## 11.2.2 常用字字体子集化

在博客上，有一些字，我们想用专门的字体显示，比如博客的标题。中文字体一般都差超过 10MB，只用来显示博客标题，未免有些浪费空间，还减缓网站加载速度。所以我们可以使用 fontmin-app 提取我们想要的字符，并转换成 svg、ttf、woff、eot 等格式。

GitHub 链接: [ecomfe/fontmin-app: fontmin as an OS X, Linux and Windows app](https://github.com/ecomfe/fontmin-app)

fontmin-app 下载链接: <https://xianchengyue.lanzout.com/iTijP00cfaoh>

fontmin-app 使用方法如下:



经测试，使用 fontmin-app 转换的字体，只有使用 `字体名-embed.css` 内的 base64 编码内嵌，才能正确显示。

## 示例

```
/* 以下内容添加至你的 css 文件中 */
@font-face {
  font-family: "LXGW_title";
  src: url(data:application/x-font-ttf;charset=utf-8;base64,xUKEgKTQF6jzao0LxQFDRATDiMdGWC1VpBrLTX9yEzDbU5WKShWik6CTAhCDQgOAAQIVEQcOGQAAAgAu/+QCCwHwAB0ALAAABSInDgEjIiY1NDc2MzIXNTQyFh0BFDEGFRQXFhQGJRQzMjc2NTQ1BiInAEoATABRAFgBAGEDAQQHdW5pNTM1QQd1bmk3Njg0B3Vu aTVCQTI=) format("truetype");
  font-style: normal;
  font-weight: normal;
}
```

## 11.2.3 使用 woff 字体

如果使用 Gulp 压缩字体，网站加载速度没有提高太多，可以使用 woff 字体。

### 什么是 woff 字体？

WOFF 全称为 Web Open Font Format，译作 Web 开放字体格式。是一种网页所采用的字体格式标准。此字体格式发展于 2009 年，现在正由万维网联盟的 Web 字体工作小组标准化。此字体格式能够有效利用压缩来减少字体大小。WOFF 字体文件一般比 TTF 字体文件小 40%。

2018 年，W3C 正式发布 WOFF2。顾名思义，即是 WOFF 2.0。该标准显著提高了压缩效率（比 WOFF 还要小 30%），降低了对网络带宽的使用，同时实现了在移动设备上对字体数据的快速解压缩。

经测试，ttf → woff，缩小 40% 左右；ttf → woff2，缩小 60% 左右。

字体转换网站如下：

- [字体转换器 \(在线自由\) – Convertio](#) 
  - 支持 ttf、otf 格式。
  - 只能转换成 woff 格式。

- [TTF to WOFF2 | CloudConvert](#)

- 支持 ttf、otf 格式。
- 可以转换成 woff 和 woff2 格式。

### 参考文章

- [W3C发布WOFF 2.0正式推荐标准 - W3C中国 \(chinaw3c.org\)](#)
- [TTF、TOF、WOFF 和 WOFF2 的相关概念 - 掘金 \(juejin.cn\)](#)
- [网页开放字体格式\(WOFF\) - Web 开发者指南 | MDN \(mozilla.org\)](#)
- [了解woff2字体及转换 « 张鑫旭-鑫空间-鑫生活 \(zhangxinxu.com\)](#)

## 11.2.4 图片处理

在处理我们博客内的图片前，我们需要了解常见的图片格式，以便我们更好地处理图片。

格式	特点
PNG	<b>Portable Network Graphics</b> 。支持无损压缩，压缩比高，生成文件体积小。支持透明效果。在下载完之前提供浏览者一个基本的图像内容，再逐渐清晰，利于浏览。多用于保存文本，线条或类似的边缘清晰，有大块相同颜色区域的图像。
JPG/JPEG	<b>Joint Photographic Experts Group</b> 。是 JPEG 标准的产物，后缀名为 .jpg或 .jpeg。属于有损压缩，压缩比很高。对色彩的信息保留较好，多用于需要连续色调、低对比，图像颜色过渡平滑且结构不规则的图像。
GIF	<b>Graphics Interchange Format</b> ，译为图形交换格式。压缩比较高。用于制作简短的动画。

格式	特点
WEBP	<p>具有更优的图像数据压缩算法，更小的图片体积，拥有肉眼识别无差异的图像质量；同时具备了无损和有损的压缩模式、Alpha 透明以及动画的特性，在 JPEG 和 PNG 上的转化效果都相当优秀、稳定和统一。截至 2022 年 1 月，已有 94% 的浏览器支持此格式。<b>注意：</b>大多数图像编辑软件都无法编辑 webp 格式的图片。</p> <p><a href="https://isparta.github.io">WebP 示例 (PNG 转 WebP) (isparta.github.io)</a> </p>

综上，我建议图片格式最好为 png 或者 webp。

## 1、图像压缩

- [在线图片压缩 - docsmall 在线图片压缩工具,在线图片压缩软件](#)   
支持 PNG/JPG 格式。最多上传 30 个文件，单个图片最大为 25MB。 **主用。**
- [在线GIF动图压缩 - docsmall 在线GIF动图压缩工具,在线GIF动图压缩软件](#)   
压缩 GIF 动图，最多上传 30 个文件，单个图片最大为 25 M。
- [Picdiet - 极速在线压缩80%图片体积](#)   
支持 JPG/JPEG 格式。无图像尺寸、图像数量、文件大小限制。
- [Squoosh](#)   
英文网站，支持常见的图像格式。压缩比高。
- [TinyPNG - 智能压缩您的WebP、JPEG和PNG图片 \(tinify.cn\)](#)   
无损压缩。最多 20 张图片，每张图片最大 5MB。

## 2、图片转 webp 格式

- [AnyWebP - 将所有图片批量转换为WebP格式](#)   
转换后图片较小，转换速度快，可调节转换尺寸、质量。单个图片最大为 5MB。 **推荐。**

- [AnyWebP软件--Win/Mac上的WebP转换器](#) 

可以自由使用，完全没有限制。图片大小、数量没有限制。转换速度极快。**推荐。**

- [多功能图片转换器无需上传文件 \(gitee.io\)](#) 

支持常见图像格式。可调节图像质量。最多上传 1000 个文件，单个图片最大为 20MB。**推荐。**

- [Squoosh](#) 

英文网站，支持常见的图像格式。可调节图像质量。压缩比高。

- [WebP - 图片格式的发展趋势 - 又拍云 \(upyun.com\)](#) 

智能压缩，转换速度较快，支持 GIF 转 WebP。只能一张一张转换。

- [TinyPNG - 智能压缩您的WebP、JPEG和PNG图片 \(tinify.cn\)](#) 

如果转换的 webp 图片还不够小，可以使用 TinyPNG 再次压缩。

另外再推荐几个图像放大网站。

- [AI Image Upscaler - Upscale Photo, Cartoons in Batch Free \(imgupscaler.com\)](#) 

- [waifu2x \(udp.jp\)](#) 

- [果核图片AI放大工具 v1.0 - 果核剥壳 \(ghxi.com\)](#) 

## 11.3 建立镜像站

GitHub 访问速度时好时坏，所以我们可以多平台部署我们的博客。

### 11.3.1 Vercel

[Develop. Preview. Ship. For the best frontend teams - Vercel](#) 

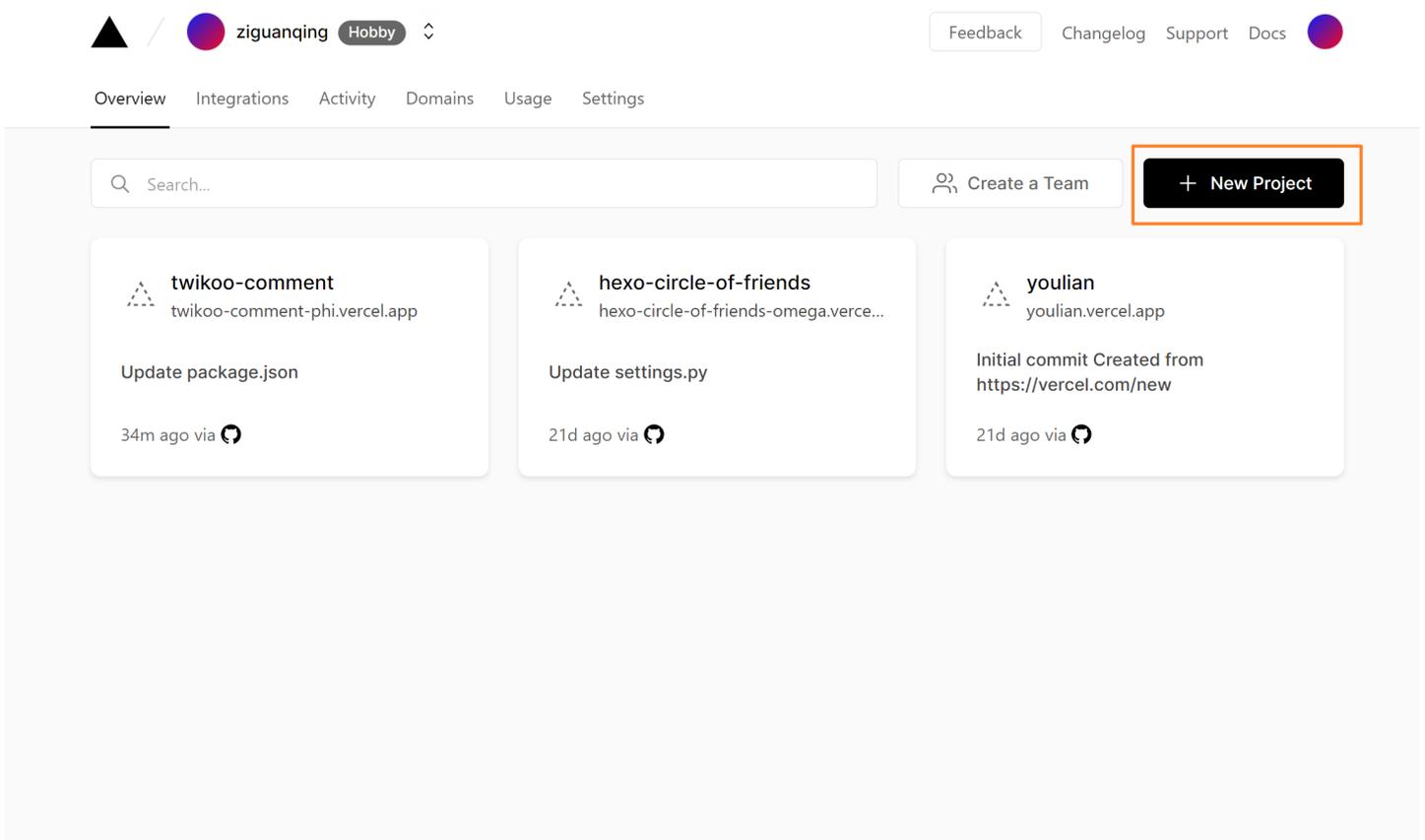
Vercel 是一个前端框架和静态站点的平台。我们可以用它来部署博客。Vercel 的电信线路，访问速度不错。下面是部署流程。

## 使用说明

- 一天可以部署 100 次。
- 每次部署时间  $\leq$  45 分钟，超过，则部署失败。
- 每个月的免费流量有 100GB，部署时间有 100 小时。
- [Limits – Vercel Docs](#) ，查看使用限制。
- [Usage – Dashboard – Vercel](#) ，查看已使用量。

## 1、注册

进入 [Vercel](#) ，点击右上角的 Sign Up，跳转至注册界面，选择 Continue with GitHub。进入以下界面。



## 2、新建项目

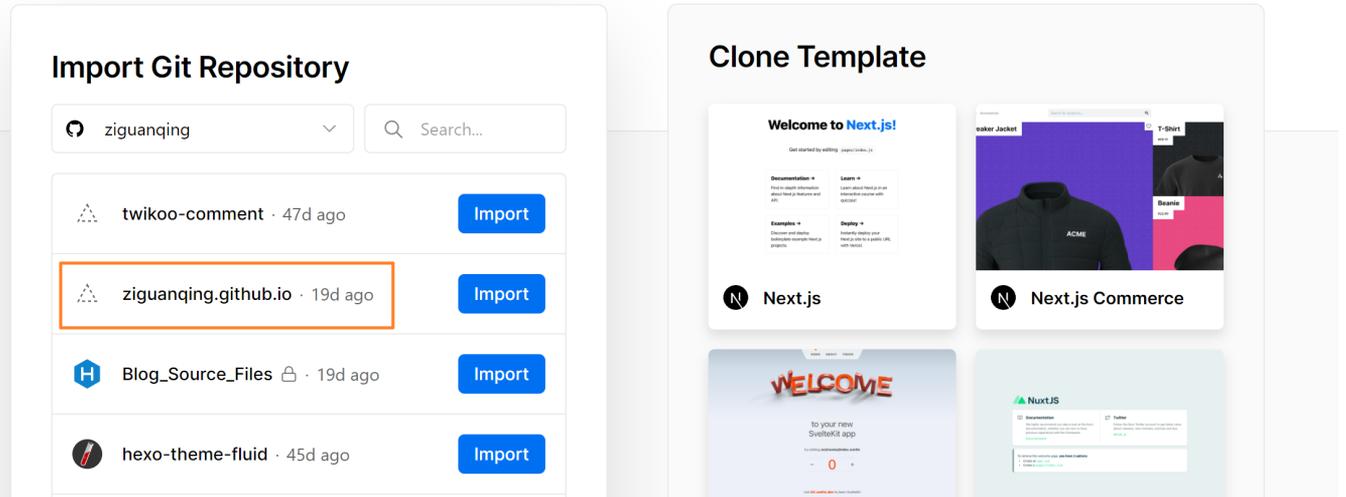
点击 + New Project，进入以下界面。选择 `用户名.github.io` 这个仓库，点击 Import，导入该仓库。



# Let's build something new.

 Collaborate with a Team

To deploy a new Project, import an existing Git Repository or get started with one of our Templates.



## 3、部署设置

基本不用修改，PROJECT NAME 可以修改为你想要的名称。然后点击 **Deploy**，开始部署。

← Back

# You're almost done.

Please follow the steps to configure your Project and deploy it.

ziguanqing.github.io

- Configure Project
- Deploy

## GIT REPOSITORY

ziguanqing/ziguanqing.github.io  
main  
/

Import a different Git Repository →

Browse Templates →

## Configure Project

### PROJECT NAME

ziguanqing-github-io

### FRAMEWORK PRESET

Other

### ROOT DIRECTORY

./

Edit

▶ Build and Output Settings

▶ Environment Variables

Deploy

## Deploy

Preparing deployment...

▶ Building



▶ Running Checks



▶ Assigning Domains



Cancel Deployment

## 4、部署成功

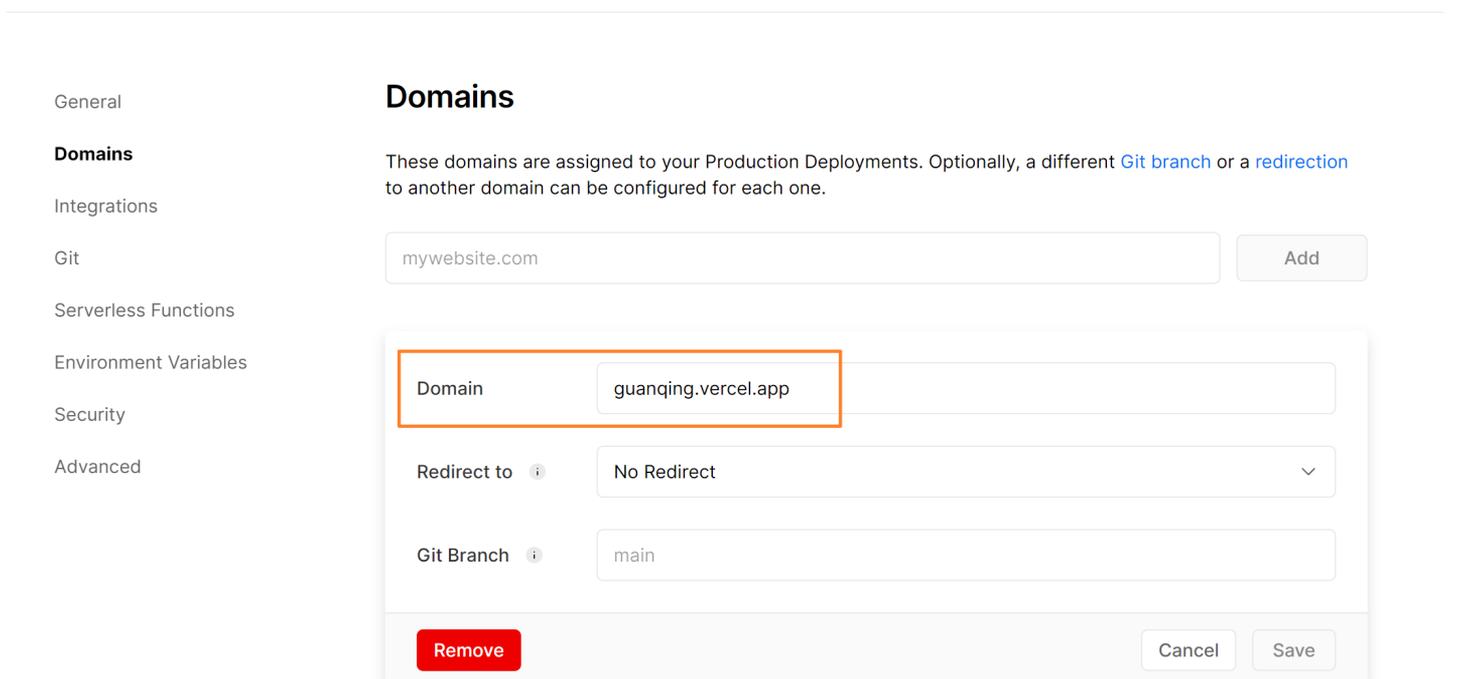


## 5、修改默认域名

博客部署成功后，Vercel 会分配给你一个域名。可以修改。

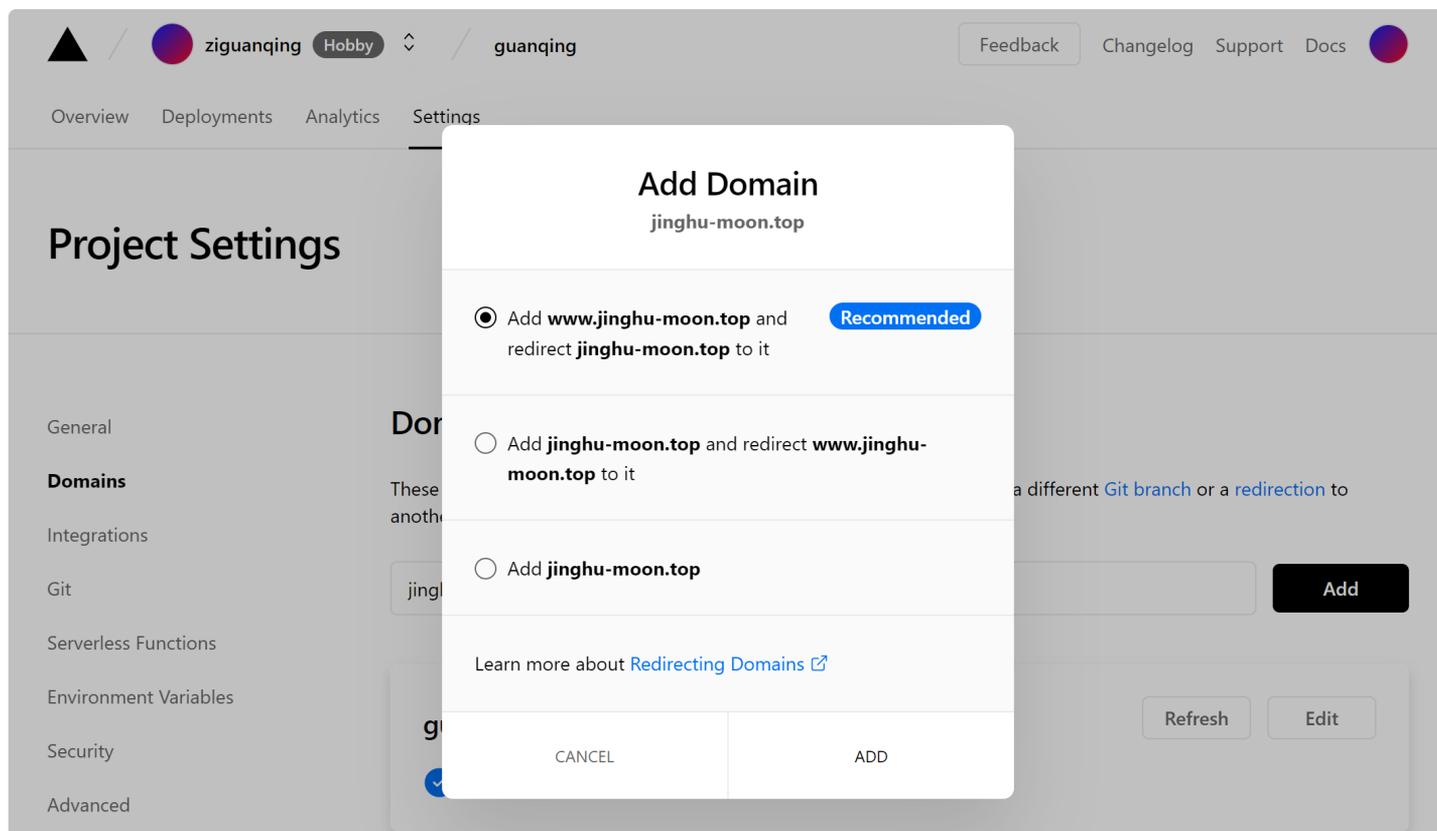
点击 Go to Dashboard，进入项目主界面。点击上方的 Settings，选择 Domains。修改域名。 `guanqing.vercel.app` 这个域名，你只能修改 `guanqing` 这一部分。

### Project Settings



## 6、自定义域名（可选）

在 mywebsite.com 一栏输入你的域名。



点击 ADD，在你的 DNS 服务商添加两条记录。现在，你可以通过 [www.xxxxx.com](http://www.xxxxx.com) 或者 [xxxxx.com](http://xxxxx.com) 来访问你的博客。

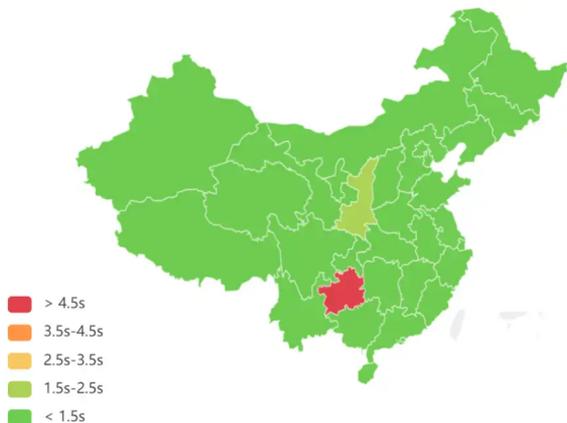


不推荐自定义域名。后面使用 CDN 加速，博客的访问速度只会更快。

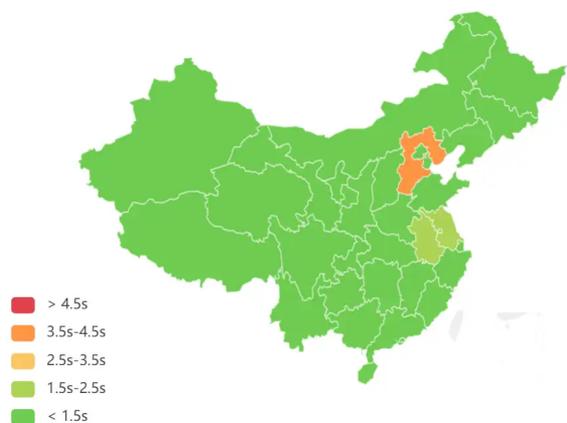
## 7、对比访问速度

“guanqing.vercel.app”与“ziguanqing.github.io”的网站测速对比

guanqing.vercel.app 平均响应: 0.96s VS ziguanqing.github.io 平均响应: 2.426s



网站测速结果 guanqing.vercel.app 2022-03-09 16:02:44



邀请好友注册boce最高可获得2000波点/人奖励, 并可享受好友充值返波点奖励。

分享链接 <https://www.boce.com/http/guanqing.vercel.app/a4ba0d40ef9c> [复制链接](#)

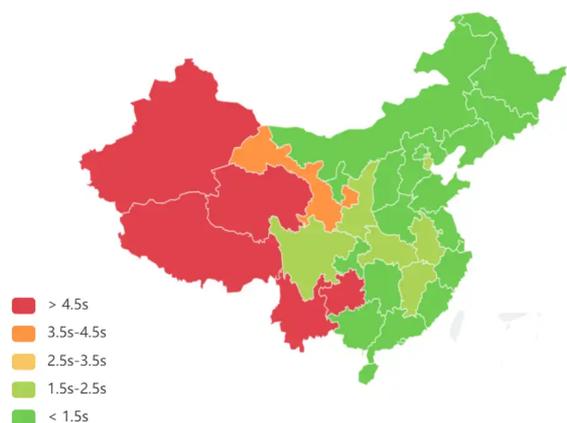
运营商: 全部 电信 移动 联通 教育网

地区: 全部 华北 东北 华东 华中 华南 西南 西北

119	0.068s	8.953s	0.999s
检测节点数	香港hkt.cc(最快)	河北石家庄移动(最慢)	平均响应

线路	最快节点	最慢节点	平均响应
电信	香港hkt.cc 0.068s	江苏无锡电信 4.25s	0.993s
移动	安徽合肥移动 0.45s	河北石家庄移动 8.953s	1.122s
联通	福建莆田联通 0.421s	安徽合肥联通 7.107s	0.881s
教育	山西太原教育网 0.644s	湖北武汉教育网 2.033s	1.039s

网站测速结果 ziguanqing.github.io 2022-03-09 16:06:57



邀请好友注册boce最高可获得2000波点/人奖励, 并可享受好友充值返波点奖励。

分享链接 <https://www.boce.com/http/ziguanqing.github.io/fe061f5a1dc7> [复制链接](#)

运营商: 全部 电信 移动 联通 教育网

地区: 全部 华北 东北 华东 华中 华南 西南 西北

119	0.051s	15.002s	1.767s
检测节点数	香港hkt.cc(最快)	贵州黔东南布依族苗族自治州电信(最慢)	平均响应

线路	最快节点	最慢节点	平均响应
电信	香港hkt.cc 0.051s	贵州黔东南布依族苗... 15.002s	2.592s
移动	福建厦门移动 0.271s	云南红河哈尼族彝族... 15.002s	1.92s
联通	上海联通 0.379s	天津联通 3.333s	1.005s
教育	北京教育网 0.285s	广东东莞电信 2.113s	0.721s

# 11.3.2 Netlify

Netlify: Develop & deploy the best web experiences in record time [🔗](#)

Netlify 将现代工具和服务的整个生态系统整合到一个单一、简单的工作流中，用于构建高性能的网站和应用程序。我们可以使用 Netlify CMS 搭建博客后台，实现在线编辑、预览、发布文章。

## 使用说明

- 每个月的免费流量有 100GB，部署时间有 300 分钟。
- 网站数量可以无限个。
- [Netlify Pricing and Plans](#) ，查看定价。
- 在「Team overview」处，查看已使用量。

## 1、注册

进入 [Netlify](#) ，点击右上角的 Sign Up，开始注册。以下内容，按照自己的意愿填写。

# Get started

## Tell us about yourself

Your responses will help us improve your experience.

I'm using Netlify for

Work

Client

Personal

School

 Required field

I'm a:

Student

My first project will be:

Blog

## Personalize your team space

Make your team your own.

Name your team

jinghu-moon

 People who use Netlify for school often use a school, cohort or project name.

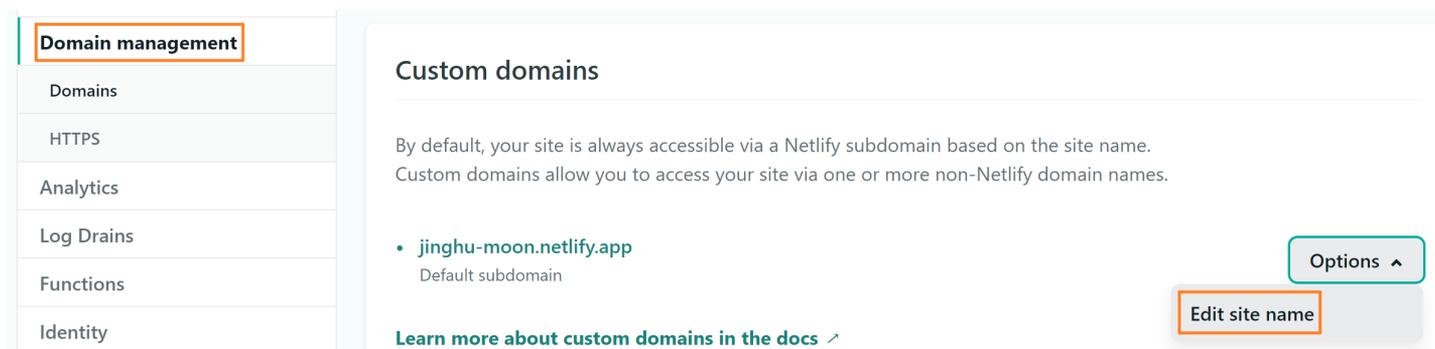
Set up and continue

## 2、导入仓库

点击 Set up and continue，进入下一步。选择 Import from Git → GitHub。授权。选择 `用户名.github.io` 这个仓库。直接点击 Deploy site，开始部署。

### 3、修改默认域名

点击 Domain settings → Edit site name，修改域名。这个域名，你只能修改 `jinghu-moon` 这一部分。



### 4、自定义域名（可选）

点击 Add custom domain → 输入域名 → 点击 Add domain，进入下一界面 → 点击 Check DNS configuration，弹出界面有两条 DNS 解析记录 → 在 DNS 服务商添加两条记录

记录值就是 Netlify 分配给你的二级域名。

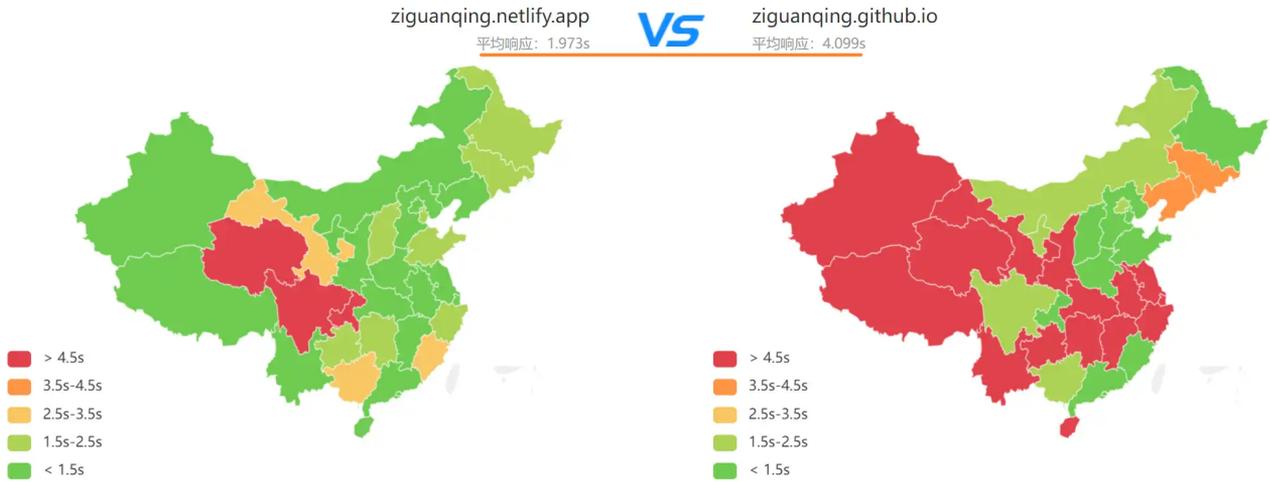
可能你添加了 DNS 解析记录，不会立即生效。一般最迟 24 小时后生效。



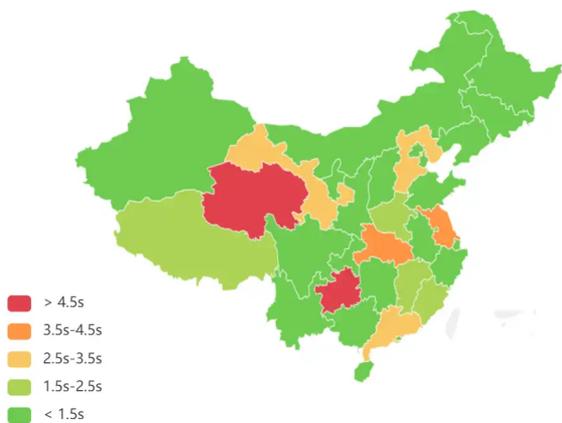
不推荐自定义域名。后面使用 CDN 加速，博客的访问速度只会更快。

### 5、对比访问速度

## “ziguanqing.netlify.app”与“ziguanqing.github.io”的网站测速对比



### 网站测速结果 ziguanqing.netlify.app 2022-03-09 16:35:52



邀请好友注册boce最高可获得2000波点/人奖励, 并可享受好友充值返波点奖励。

分享链接 <https://www.boce.com/http/ziguanqing.netlify.app/a394e46189> [复制链接](#)

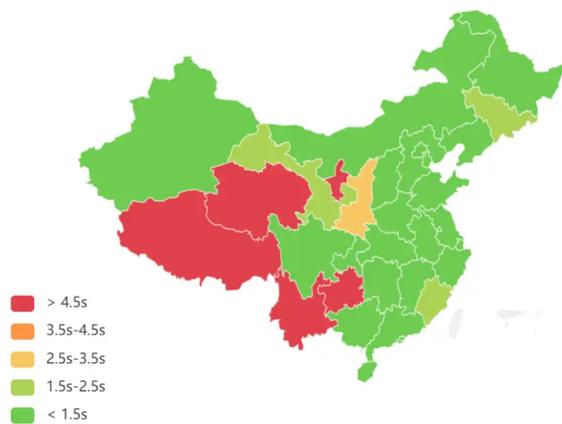
运营商: [全部](#) [电信](#) [移动](#) [联通](#) [教育网](#)

地区: [全部](#) [华北](#) [东北](#) [华东](#) [华中](#) [华南](#) [西南](#) [西北](#)

120 检测节点数    0.232s 香港hkt.cc(最快)    12.811s 青海西宁电信(最慢)    1.588s 平均响应

线路	最快节点	最慢节点	平均响应
电信	香港hkt.cc 0.232s	青海西宁电信 12.811s	2.152s
移动	江西吉安移动 0.281s	辽宁鞍山移动 1.848s	0.975s
联通	江西赣州联通 0.311s	江苏徐州联通 10.493s	1.688s
教育	北京教育网 0.592s	辽宁大连教育网 1.641s	0.99s

### 网站测速结果 ziguanqing.github.io 2022-03-09 16:21:23



邀请好友注册boce最高可获得2000波点/人奖励, 并可享受好友充值返波点奖励。

分享链接 <https://www.boce.com/http/ziguanqing.github.io/e1fe372b024f> [复制链接](#)

运营商: [全部](#) [电信](#) [移动](#) [联通](#) [教育网](#)

地区: [全部](#) [华北](#) [东北](#) [华东](#) [华中](#) [华南](#) [西南](#) [西北](#)

120 检测节点数    0.245s 河南洛阳移动(最快)    15.001s 西藏拉萨电信(最慢)    1.765s 平均响应

线路	最快节点	最慢节点	平均响应
电信	香港hkt.cc 0.576s	西藏拉萨电信 15.001s	2.412s
移动	河南洛阳移动 0.245s	云南红河哈尼族彝族... 15.001s	1.928s
联通	浙江绍兴联通 0.394s	陕西西安联通 3.077s	1.004s
教育	安徽合肥教育网 0.274s	福建福州教育网 6.052s	1.157s

## 11.3.3 总结

### 访问速度

Vercel > Netlify ≈ Github。Netlify 没比 Github 访问快多少, 有时候还不如 Github。

## 打开几率

Vercel > Netlify  $\approx$  Github。这是四川地区的结果。不代表别的地区。总的来说，相较于其他两家，Vercel 访问又快又稳，比较推荐。

# 11.4 CDN 加速

## 11.4.1 CDN 简单介绍

### 1、什么是 CDN?

CDN，全称 Content Delivery Network，译为 [内容分发网络](#)。是一种透过互联网互相连接的计算机网络系统，利用最靠近每位用户的服务器，更快、更可靠地将音乐、图片、视频、应用程序及其他文件发送给用户，来提高用户打开网站的速度。

### 2、CDN 如何实现加速?

以网购举例。为缩短用户收到快递的时间，商家一般会在多地建立货运仓库。当用户下单后，商家会选择距离用户收货地址最近的仓库发货。CDN 加速与之类似，但不完全相同。

一个网站主要由静态资源和动态资源两部分组成。Hexo 搭建的博客为静态网站，所以我们主要讲静态资源。静态资源主要有 HTML、CSS、JS、图片、字体等。





