



# 利用 GitHub 和 Hexo 搭建个人博客【保姆教程】

利用 GitHub 和 Hexo 搭建个人博客

## 一、前言

在网络飞速发展的今天，个人博客已成为人们在网络世界中展示独特个性、分享知识见解以及记录生活点滴的理想平台。GitHub 作为全球知名的开源代码托管平台，为我们提供了稳定且免费的存储服务，而 Hexo 作为一款基于 Node.js 的快速、简洁、高效的静态博客框架，两者结合，为我们打造个人博客带来了极大的便利，能让我们轻松搭建出既美观又功能丰富的博客。

看到别人有自己的个人技术博客，自己也想要一个，于是走上了搭建(zhuang)个人(bi)博客之路。

一些大佬的blog：

<https://simonaking.com>

<https://akilar.top>

<https://blog.zhheo.com>

<https://xbxyftx.top>

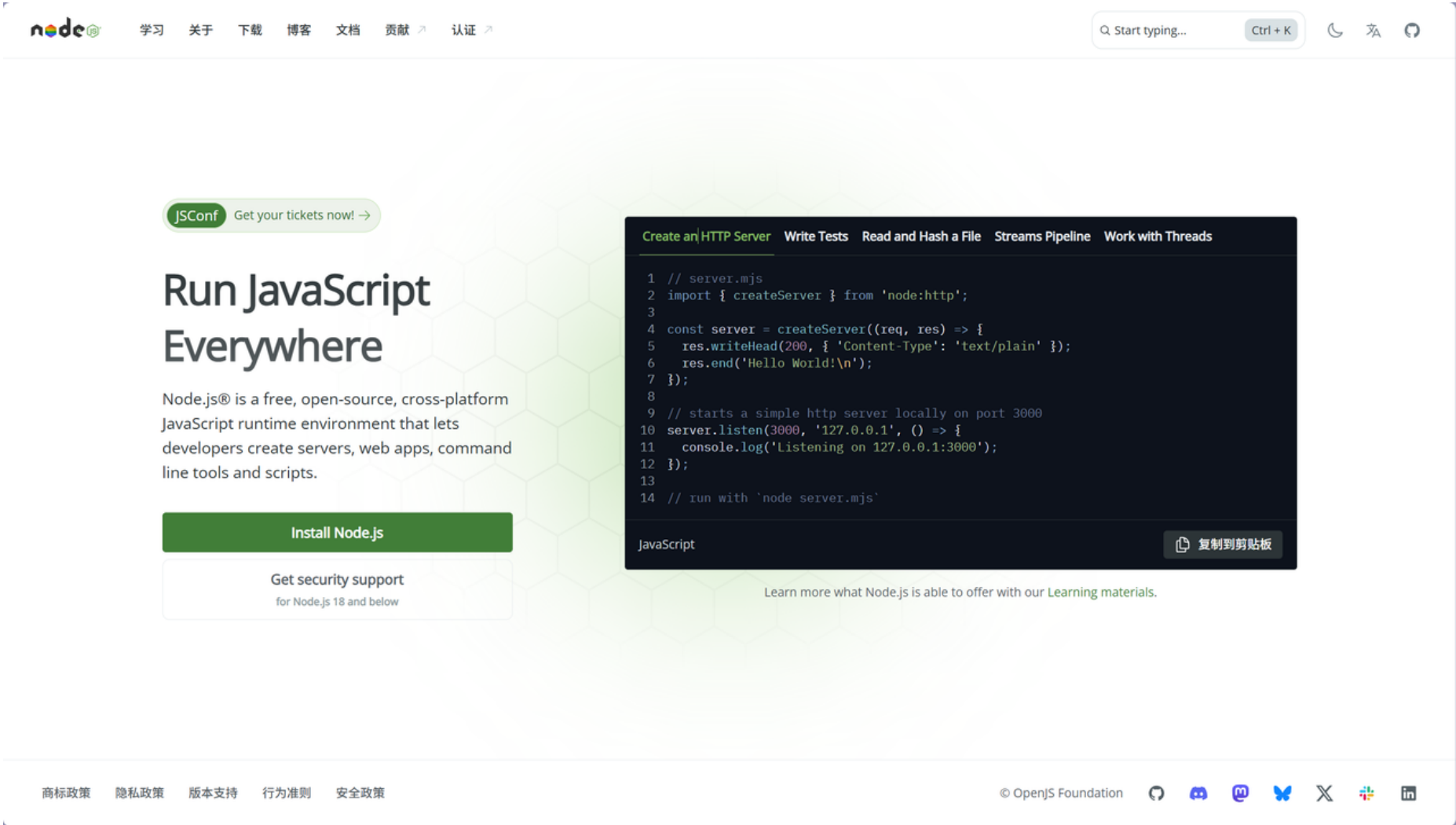
<https://zxjc-niusile.github.io>

## 二、准备工作

### （一）安装 Node.js 和 Git

#### Node.js 安装

Node.js 是 Hexo 运行所依赖的关键环境。访问 Node.js 官方网站（<https://nodejs.org/>），在首页根据自己的操作系统类型（Windows、Mac OS 或 Linux）选择相应的安装包进行下载。



下载完成后，按照安装向导的提示逐步完成安装。安装结束后，打开命令行工具（在 Windows 上是“命令提示符”或“PowerShell”，在 Mac OS 和 Linux 上是“终端”），  
安装好之后我们可以输入以下指令如果都能正常输出版本号则安装成功。

```
cmd

1  node -v
2  npm -v
```

```
C:\Users\26325>node -v
v22.15.1

C:\Users\26325>npm -v
10.9.2
```

## Git 安装

Git 是用于管理博客代码版本以及推送博客内容到 GitHub 的重要工具，当然它的用处不止于此，总之程序员必备。Git 官方网站（<https://git-scm.com/>），同样根据操作系统类型下载合适的安装包进行安装。安装完成后，在命令行中输入 `git --version` 若显示 Git 的版本信息，则表明安装成功。

```
C:\Users\26325>git --version
git version 2.49.0.windows.1

C:\Users\26325>|
```

## （二）注册 GitHub 账号

打开浏览器，访问 GitHub 官网 (<https://github.com/>)。在首页上找到“Sign up”按钮，点击后按照提示填写相关信息，包括用户名、邮箱地址和密码等，完成注册过程。这个账号将作为存储博客文件的仓库，承载着博客的所有代码和内容。

## 三、安装 Hexo

### （一）创建博客目录

在本地计算机的硬盘上，选择一个你方便管理的位置（例如 D 盘根目录或者用户目录下的某个文件夹），创建一个新的文件夹，将其命名为blog（你也可以根据自己的喜好命名，但要注意避免使用特殊字符和空格）。这个文件夹将作为博客项目的根目录，后续所有与博客相关的文件和操作都将在这个目录下进行。

### （二）安装 Hexo

打开命令行工具，通过cd命令进入到刚刚创建的博客目录（例如，如果你在 Windows 上，且博客目录在 D 盘根目录下，你可以输入 d: 然后 cd blog；如果在其他操作系统下，使用相应的路径导航命令）。进入目录后，运行以下命令安装 Hexo：`npm install -g hexo-cli`。

注意：

如果还不会 目，可以先进行 `npm install -g cnpm --registry=https://registry.npmmirror.com`  
然后进行 `cnpm -v` 验证是否成功  
完成后，在之后所有以 `npm` 开头的步骤，都可以把所有 `npm` 命令替换为 `cnpm`

此命令中的-g参数表示全局安装，hexo-cli是 Hexo 的命令行工具，全局安装后可以在任何目录下使用 Hexo 相关命令。安装过程可能需要一些时间，因为它需要从网络上下载 Hexo 的相关文件和依赖。

### （三）初始化博客

在博客目录下继续运行 `hexo init` 命令。这个命令会自动下载 Hexo 初始化博客项目所需的文件和配置信息，并按照 Hexo 的默认结构进行组织。初始化完成后，博客目录下会出现以下主要的目录和文件：

`_config.yml`：这是 Hexo 的核心配置文件，整个博客的各种设置都在这里完成。它是一个 YAML 格式的文件，通过简单的键值对来配置博客的参数，包括博客的标题、副标题、作者、语言、主题、插件等众多设置。例如，title字段用于设置博客的标题，subtitle字段用于设置副标题，author字段填写作者姓名，language字段可以指定博客使用的语言，如en（英语）、zh-CN（中文简体）等。

source：这个目录是用户创建博客文章和页面的源文件存放处。默认情况下，我们使用 Markdown 格式来撰写文章。在这个目录下创建的 Markdown 文件将被 Hexo 解析并转换为网页内容。例如，您可以在source/\_posts子目录下创建新的博客文章文件。

themes：存放博客主题文件的目录。Hexo 有大量丰富多样的开源主题可供选择，每个主题都有其独特的设计风格和功能。不同的主题可以让您的博客呈现出完全不同的外观和交互体验。

public：这个目录用于存放 Hexo 根据source目录中的内容和\_config.yml的配置生成的静态文件。这些静态文件包括 HTML、CSS、JavaScript 等，它们将被部署到服务器（如 GitHub）上，供用户通过浏览器访问。

此外，还有其他一些辅助文件和文件夹，如scaffolds用于生成文章或页面的模板，package.json用于管理项目的依赖等。

|                         |                 |          |        |
|-------------------------|-----------------|----------|--------|
| 📁 .deploy_git           | 2025/6/23 13:59 | 文件夹      |        |
| 📁 .github               | 2025/6/22 14:03 | 文件夹      |        |
| 📁 node_modules          | 2025/6/22 21:08 | 文件夹      |        |
| 📁 public                | 2025/6/23 13:59 | 文件夹      |        |
| 📁 scaffolds             | 2025/6/22 14:03 | 文件夹      |        |
| 📁 source                | 2025/6/22 14:03 | 文件夹      |        |
| 📁 themes                | 2025/6/22 21:10 | 文件夹      |        |
| 📄 .gitignore            | 2025/6/22 14:03 | txtfile  | 1 KB   |
| 📄 _config.landscape.yml | 2025/6/22 14:03 | Yaml 源文件 | 0 KB   |
| 📄 _config.yml           | 2025/6/22 21:00 | Yaml 源文件 | 3 KB   |
| 📄 db.json               | 2025/6/23 13:59 | JSON 源文件 | 91 KB  |
| 📄 package.json          | 2025/6/22 19:13 | JSON 源文件 | 1 KB   |
| 📄 package-lock.json     | 2025/6/22 19:13 | JSON 源文件 | 121 KB |

测试：通过 `hexo g && hexo s` 生成一个初始页面来进行测试。

命令提示符不行可以试试在git bash里面执行，（PowerShell的话会报分隔符错误）

在页面启动后按住ctrl左键点击<http://localhost:4000/> 字段即可自动跳转至浏览器进行本地预览。



当你看到这个，庆祝一下吧，我们成功创建了一个本地静态博客网页。

## 四、配置 Hexo

### （一）修改配置文件

使用文本编辑器（如 Visual Studio Code 甚至是记事本）打开\_config.yml文件。在文件中找到并修改以下关键信息：

基本信息修改

博客标题（title）：输入你为博客想好的名称，这个名称将显示在博客的首页和浏览器标签页上，例如“知识探索者的博客”。

副标题（subtitle）：可以填写一个简短的描述性语句，进一步说明博客的主题或特色，如“分享科技与人文的点滴”。

作者（author）：填写你的姓名或者您希望在博客上显示的作者名称。

其他设置

url字段：如果你已经有了自定义的域名，可以填写完整的域名地址，如<https://www.myblog.com>。如果暂时没有域名，可先留空。





或许你并不想自己去找主题，或者短时间找不到自己喜欢的，可以看下这篇博文

### 主题下载与安装

以next主题为例，在博客目录下的命令行中运行 `git clone https://github.com/iissnan/hexo-theme-next themes/next`。这条命令使用 Git 从指定的 GitHub 仓库下载next主题的文件，并将其存放在themes/next目录下。不同的主题可能有不同的安装方式，但通过 Git 克隆仓库是一种常见且便捷的方法。

### 安装渲染器

安装 `pug` 和 `stylus` 渲染器，否则启动之后访问页面会报错。

代码块

```
1 npm install hexo-renderer-pug hexo-renderer-stylus --save
```

### 切换主题

主题下载完成后，需要在\_config.yml文件中修改theme的值。找到theme字段，并将其值修改为您所下载的主题文件夹名称，例如对于next主题，将theme: landscape（如果之前是其他主题）修改为theme: next。修改完成后保存配置文件。

## 五、创建并撰写博客文章

### （一）创建新文章

在博客目录下打开命令行，运行 `hexo new "文章标题"` 命令。这里的“文章标题”是你要创建的新文章的题目，例如 `hexo new "我的第一篇博客文章"`。运行此命令后，Hexo 会在source/\_posts目录下创建一个新的 Markdown 文件，文件名是根据文章标题生成的，同时会自动添加一些基本的头部信息，如文章的创建日期、标题等。

### （二）撰写文章

使用文本编辑器打开新创建的 Markdown 文件。在 Markdown 文件中，您可以按照 Markdown 语法来撰写文章内容。

## 六、本地预览博客

每一次对博客进行修改后都可以启动本地服务器看看修改是否符合预期，再考虑要不要进行上传

在博客目录下的命令行中运行 `hexo clen && hexo g && hexo s` 命令。

访问地址 `http://localhost:4000`，打开浏览器，在地址栏中输入这个地址，就可以看到你的博客在本地的预览效果。在本地预览过程中，你可以对博客的外观、文章内容、布局等进行检查和调整，确保一切符合你的预期。如果在预览过程中发现问题，可以及时回到相应的文件（如文章内容文件、配置文件、主题文件等）进行修改，然后刷新浏览器页面查看修改效果。

## 七、将博客部署到 GitHub

### （一）创建 GitHub 仓库

登录到你的 GitHub 账号，在 GitHub 页面右上角点击“+”号，在弹出的下拉菜单中选择“New repository”。在创建仓库页面，填写以下信息：

仓库名称：建议命名为 `username.github.io`，这里的username是你在 GitHub 注册的账号用户名。这个命名方式是 GitHub Pages 的特殊要求，用于识别和部署个人网站或博客。

仓库描述（可选）：可以简要描述一下这个仓库的用途，比如“我的个人博客仓库”。

仓库类型：选择“Public”（公开）保证博客能够被任何人访问和浏览。

填写完成后，点击“Create repository”按钮完成仓库创建。

### （二）配置 SSH 密钥

!!! 打起精神我觉得这是最麻烦的部分，无论是编写这一教程还是自己搭建博客的时候。

#### 一：打开 Git Bash

1. 点击 Windows 的“开始”菜单。
2. 输入 `Git Bash` 并找到它。
3. 点击打开，你会看到一个类似于 Linux 终端的黑色窗口。接下来的操作都在这里进行。

#### 二：生成新的 SSH 密钥

1. 在 Git Bash 窗口中，复制并粘贴以下命令。

代码块

```
1 git config --global user.name "XXXX" #XXXX换成自己的名字或者在GitHub上显示的昵称
2 git config --global user.email "XXXXXXXXXX@XXX.com" #"XXXXXXXXXX@XXX.com" 换成注册GitHub时使用的电子邮箱
3 ssh-keygen -t rsa -C "XXXXXXXXXX@XXX.com" #邮箱推荐和上面一样
```

2. 按下回车键。接下来终端会询问你几个问题，事实上这里可以一路回车直到生成成功。

#### 三：将 SSH 公钥添加到 GitHub

1. 按照默认路径生成的密钥都会储存在C:\Users\用户名\.ssh\id\_rsa.pub这个地址下，复制这串内容，打开GitHub设置，添加ssh密钥，将该文件的全部内容黏贴到文本框中即可。
  - a. 登录你的 GitHub 账户。
  - b. 点击右上角的个人头像，选择 Settings。
  - c. 在左侧菜单中，点击 SSH and GPG keys。
  - d. 点击右上角的 New SSH key 按钮。
  - e. Key: 将刚才复制的那一长串公钥内容完整地粘贴到这个文本框中。
  - f. 点击 Add SSH key 按钮。GitHub 可能会要求你再次输入登录密码以确认操作。

## 四：测试 SSH 连接

1. 回到 Git Bash 窗口，输入以下命令：

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

如果连接成功，你会看到一条欢迎信息： `Hi [your-username]! You've successfully authenticated, but GitHub does not provide shell access.`

2. 在测试之前可以输入以下命令：

```
eval "$(ssh-agent -s)"
```

```
ssh-add ~/.ssh/id_rsa
```

## 五：配置 Hexo 使用 SSH 方式部署

打开 hexo 的配置文件 `_config.yml` 修改 `deploy` 位置的配置

代码块

```
1  deploy:
2    type: git
3    repo: git@github.com:你的GitHub用户名/你的GitHub用户名.github.io.git
4    branch: main
```

这里需要注意，要将上述配置中的“你的用户名”替换为自己的 GitHub 用户名。

### （三）安装 hexo - deployer - git 插件

在博客目录下的命令行中运行 `npm install hexo-deployer-git --save`。这个插件是 Hexo 用于将生成的静态文件部署到 GitHub 的关键工具。`--save` 参数表示将这个插件添加到项目的依赖列表中，以便在项目的其他环境中也能正确使用。安装完成后，插件会被下载并安装到博客项目的 `node_modules` 目录下。

### （四）部署博客

在博客目录下运行 `hexo cl && hexo g && hexo d`

这个命令会首先根据 `source` 目录中的内容和配置信息生成静态文件，然后使用 `hexo - deployer - git` 插件将这些静态文件推送到 GitHub 仓库中。在推送过程中，命令行可能会提示你输入 GitHub 的用户名和密码（如果你没有配置 SSH 密钥）。输入正确的信息后，推送操作会继续进行。推送完成后，稍等片刻（一般几分钟内），就可以通过你的用户名.github.io 访问你的个人博客了。如果遇到访问问题，可以检查仓库设置、域名解析（如果有自定义域名）以及文件推送是否完整等情况。