**[教程]Hexo + Github 搭建自己的专属博客**

[[教程]Hexo + Github 搭建自己的专属博客](#bookmark1)

[1. 安装Git和NodeJS](#bookmark2)

[2. 安装Hexo](#bookmark3)

[3. 加载主题](#bookmark4)

[4. 修改主题配置](#bookmark5)

[5. 将博客部署在GitHub上](#bookmark6)

[6. 写文章并上传](#bookmark7)

[7. 配置一些特效](#bookmark8)

**1. 安装Git和NodeJS**

 在Windows上使用Git，可以从Git官网直接<https://git-scm.com/downloads>，然后按默认选项安 装即可。安装完成后，在开始菜单里找到“Git”->“Git Bash”，蹦出一个类似命令行窗口的东西，就 说明Git安装成功！

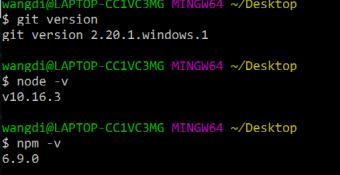
 在Git中绑定Github账号，打开“Git Bash”，在命令框中依次输入两行命令：

|  |
| --- |
| git config --global user.name “Your Name”  git config --global user.email email@example.com  # 其中Your Name和email@example.com替换成上面注册时的账户名和邮箱 |

[ 由于 Hexo 是基于 Node.js 驱动的一款博客框架，所以安装NodeJS](https://nodejs.org/en/download/) <https://nodejs.org/en/downl> [oad/ 并配置环境变量。](https://nodejs.org/en/download/)

 安装之后可以输入以下命令查看是否安装成功：

|  |
| --- |
| git version  node -v  npm -v |



**2. 安装Hexo**

 以上环境准备好了就可使用 npm 开始安装 Hexo 了，在命令行输入执行如下命令：

|  |
| --- |
| npm install -g hexo-cli |

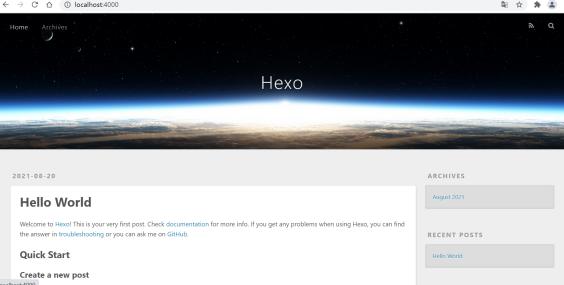
 安装 Hexo 完成后，在指定文件夹下打开“Git Bash”，再执行下列命令， Hexo 将会在指定文件夹中 新建所须要的文件：

hexo init myBlog

cd myBlog

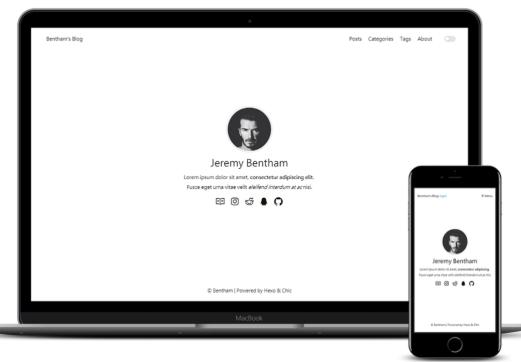
npm install

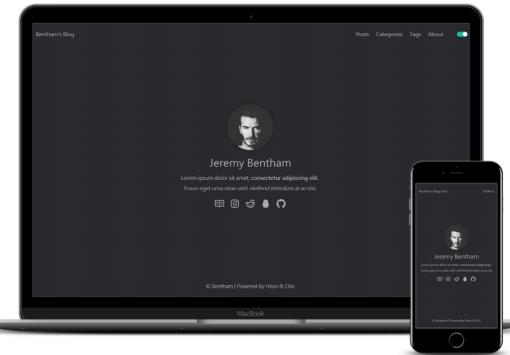
 若是上面的命令都没报错的话，就恭喜了，运行 hexo s 命令，其中 s 是 server 的缩写，在浏览 器中输入 [http://localhost:4000](http://localhost:4000/) 回车就能够预览效果了。

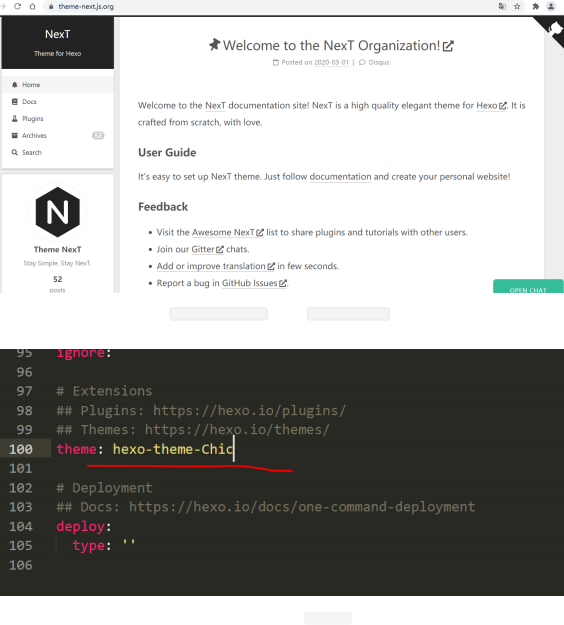


**3. 加载主题**

[ 大家可以去官网上找自己喜欢的主题下载](https://github.com/Siricee/hexo-theme-Chic)<https://hexo.io/themes/> [，自己采用的主题是https://git hub.com/Siricee/hexo-theme-Chic ，这款有明暗两种颜色。](https://github.com/Siricee/hexo-theme-Chic)



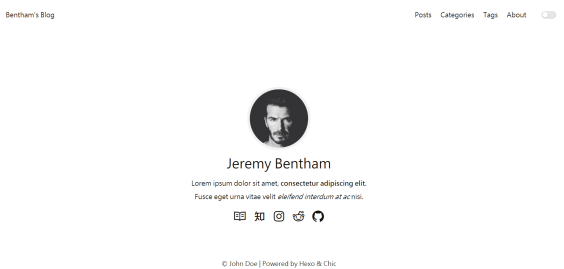




 基本大家用的最多的是NEXT主题： <https://github.com/next-theme/hexo-theme-next>

 随后将下载的主题文件夹放在 myblog/themes 中，在 \_config.yml 文件中修改theme 为hexo- theme-Chic(注意和主题文件名一致)

[ 修改好之后在“Git Bash”中执行 hexo g 命令，然后再 hexo s ，在浏览器中输入 http://localhost:4 000 回车就能够预览修改主题后的效果了。](http://localhost:4000/)



**4. 修改主题配置**

 修改在主题文件夹下的 \_config.yml文件，完成自己个人的配置。

# Header 主页面标题

navname: Bentham's Blog

# navigatior items 四个文件归类

nav:

Posts: /archives

Categories: /category

Tags: /tag

About: /about

# favicon 图标

favicon: /favicon.ico

# Profile 中间显示名字

nickname: Jeremy Bentham

### this variable is MarkDown form.

# 个人描述，可以修改成自己要显示的句子

description: Lorem ipsum dolor sit amet, \*\*consectetur adipiscing elit.\*\* <br>Fusce eget urna vitae velit \*eleifend interdum at ac\* nisi.

# 个人头像图片

avatar: /image/avatar.jpeg

# main menu navigation

## links key words should not be changed.

## Complete url after key words.

## Unused key can be commented out.

# 下方超链接

links:

Blog: /archives

# Category:

# Tags:

# Link:

# Resume:

# Publish:

# Trophy:

# Gallery:

# RSS:

# AliPay:

ZhiHu: https://www.zhihu.com/people/sirice

# LinkedIn:

# FaceBook:

# Twitter:

# Skype:

# CodeSandBox:

# CodePen:

# Sketch:

# Gitlab:

# Dribbble:

Instagram:

Reddit:

# YouTube:

# QQ:

# Weibo:

# WeChat:

Github: https://github.com/Siricee

# how links show: you have 2 choice--text or icon. 图标 or 文字

links\_text\_enable: false

links\_icon\_enable: true

# Post page

## Post\_meta

post\_meta\_enable: true

post\_author\_enable: true

post\_date\_enable: true

post\_category\_enable: true

## Post copyright

post\_copyright\_enable: true

post\_copyright\_author\_enable: true

post\_copyright\_permalink\_enable: true

post\_copyright\_license\_enable: true

post\_copyright\_license\_text: Copyright (c) 2019 <a

href="<http://creativecommons.org/licenses/by-nc/4.0/>">CC-BY-NC-4.0</a> LICENSE

post\_copyright\_slogan\_enable: true

post\_copyright\_slogan\_text: Do you believe in <strong>DESTINY</strong>?

## toc

post\_toc\_enable: true

# Page

page\_title\_enable: true

# Date / Time format

## Hexo uses Moment.js to parse and display date

## You can customize the date format as defined in

## http://momentjs.com/docs/#/displaying/format/

date\_format: MMMM D, YYYY

time\_format: H:mm:ss

# stylesheets loaded in the <head>

stylesheets:

- /css/style.css

# scripts loaded in the end of the body

scripts:

- /js/script.js

- /js/tocbot.min.js

# tscanlin/tocbot: Build a table of contents from headings in an HTML document.

# https://github.com/tscanlin/tocbot

# plugin functions

## Mathjax: Math Formula Support

## https://www.mathjax.org

# 数学公式

mathjax:

enable: true

import: demand # global or demand

## global: all pages will load mathjax,this will degrade performance and some grammers may be parsed wrong.

## demand: Recommend option,if your post need fomula, you can declare 'mathjax: true' in Front-matter

**5. 将博客部署在GitHub上**

 点击 Start project 或者下面的 new repository 建立一个新的仓库，注意**Github 仅能使用一个同 名仓库的代码托管一个静态站点**，这里注意仓库名一定要是： 用户名 .github.io

 配置 SSH key ，要使用 git 工具首先要配置一下SSH key，为部署本地博客到 Github 作准备。

|  |
| --- |
| git config --global user.name "用户名"  git config --global user.email "邮箱地址"  ssh-keygen -t rsa -C '上面的邮箱 ' |

按照提示完成三次回车，便可生成 ssh key，采用以下指令也可以查看自己的ssh：

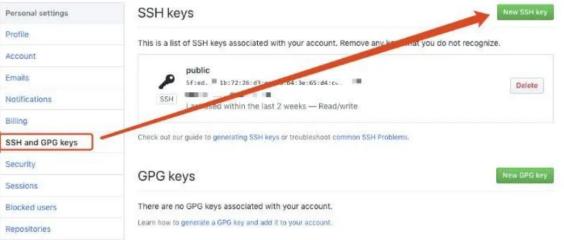
cat ~/.ssh/id\_rsa.pub

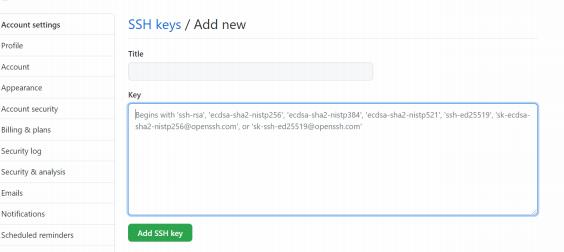
首次使用还须要确认并添加主机到本机SSH可信列表。若返回 Hi xxx! You've successfully authenticated, but GitHub does not provide shell access. 内容，则证实添加成功。

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

 登陆 Github 上添加刚刚生成的SSH key，按如下步骤添加，右上角点击头像-> settings -> SSH

and GPG keys，建立一个新的 SSH key, 标题随便， key 就填刚才生成那个，确认建立，这样在你 的 SSH keys 列表里就会看到你刚刚添加的密钥。





 此时，本地和Github的工做作得差不了，是时候把它们两个链接起来了。你也能够查看官网的部署 教程。先不着急，部署以前还须要修改配置和安装部署插件。第一：打开项目根目录下的

\_config.yml 配置文件配置参数。拉到文件末尾，填上以下配置。



 第二要安装一个部署插件 hexo-deployer-git，打开“Git Bach”，输如以下指令：

|  |
| --- |
| npm install hexo-deployer-git --save |

 最后执行如下两条命令就能够部署上传啦，如下 g 是 generate 缩写， d 是 deploy 缩写

|  |
| --- |
| hexo g # 先生成  hexo d # 部署到Github上 |

 这时用浏览器输入 用户名 .github.io就可以访问刚才的网站啦。

**6. 写文章并上传**

 博客搭好了，就开始写文章了，这里简单介绍一下，详细的文档能够看 hexo 官网。新建文章，输 入如下命令便可

hexo new '文章标题 '

 执行完成后能够在 /source/\_posts 下看到一个“文章标题.md”的文章文件啦。 .md 就是 Markdown 格式的文件，具体用法能够在网上找一下，语法仍是比较简单的。

---

title: blogTest

date: 2021-08-20 18:07:21

tags: Test

categories: blog1

---

### 1 . This is a blog Test

\* First

\* Second

---

### 2 . Show Text

\* \*\*这是加粗\*\*

> \*这是斜体\*

 之后依次输入以下命令：

|  |  |  |
| --- | --- | --- |
| hexo hexo hexo | g  s  d | # 生成文件  # 本地服务器查看网站  # 部署到Github 上 |

**7. 配置一些特效**

 雪花特效： themes\hexo-theme-Chic\layout\index.ejs 中添加如下代码：

|  |
| --- |
| <!-- 雪花特效 -->  <script type="text/javascript"  src="<https://libs.baidu.com/jquery/1.8.3/jquery.js>"></script>  <script type="text/javascript"  src="<https://libs.baidu.com/jquery/1.8.3/jquery.min.js>"></script>  <script type="text/javascript" src="/js/snow.js"></script> |

蜘蛛网特效： themes\hexo-theme-Chic\layout\layout.ejs 中添加如下代码：

<script>

!

function() {

function n(n, e, t) {

return n.getAttribute(e) || t

}

function e(n) {

return document.getElementsByTagName(n)

}

function t() {

var t = e("script"),

o = t.length,

i = t[o - 1];

return {

l: o,

z: n(i, "zIndex", -1), //置于主页面背后

o: n(i, "opacity", .5), //线条透明度

c: n(i, "color", "0,0,0"), //线条颜色

n: n(i, "count", 100) //线条数量

}

}

function o() {

a = m.width = window.innerWidth ||

document.documentElement.clientWidth || document.body.clientWidth, c = m.height = window.innerHeight ||

document.documentElement.clientHeight || document.body.clientHeight }

function i() {

r.clearRect(0, 0, a, c);

var n, e, t, o, m, l;

s.forEach(function(i, x) {

for (i.x += i.xa, i.y += i.ya, i.xa \*= i.x > a || i.x < 0 ? -1 : 1, i.ya \*= i.y > c || i.y < 0 ? -1 : 1, r.fillRect(i.x - .5, i.y - .5, 1,

1), e = x + 1; e < u.length; e++) n = u[e],

null !== n.x && null !== n.y && (o = i.x - n.x, m = i.y - n.y, l = o \* o + m \* m, l < n.max && (n === y && l >= n.max / 2 && (i.x -= .03 \* o, i.y -= .03 \* m), t = (n.max - l) / n.max, r.beginPath(), r.lineWidth = t /

2, r.strokeStyle = "rgba(" + d.c + "," + (t + .2) + ")", r.moveTo(i.x, i.y), r.lineTo(n.x, n.y), r.stroke()))

}),

x(i)

}

var a, c, u, m = document.createElement("canvas"),

d = t(),

l = "c\_n" + d.l,

r = m.getContext("2d"),

x = window.requestAnimationFrame || window.webkitRequestAnimationFrame || window.mozRequestAnimationFrame || window.oRequestAnimationFrame ||

window.msRequestAnimationFrame ||

function(n) {

window.setTimeout(n, 1e3 / 45)

},

w = Math.random,

y = {

x: null,

y: null,

max: 2e4

};

m.id = l,

m.style.cssText = "position:fixed;top:0;left:0;z-index:" + d.z + ";opacity:" + d.o,

e("body")[0].appendChild(m),

o(),

window.onresize = o,

window.onmousemove = function(n) {

n = n || window.event,

y.x = n.clientX,

y.y = n.clientY

},

window.onmouseout = function() {

y.x = null,

y.y = null

};

for (var s = [], f = 0; d.n > f; f++) {

var h = w() \* a,

g = w() \* c,

v = 2 \* w() - 1,

p = 2 \* w() - 1;

s.push({

x: h,

y: g,

xa: v,

ya: p,

max: 6e3

})

}

u = s.concat([y]),

setTimeout(function() {

i()

},

100)

} ();

</script>

· 评论区采用的时Valine <https://valine.js.org/> 和LeanCloud