# https://blog.csdn.net/walkerhau/article/details/77394659?utm\_source=debugrun&utm\_medium=referral

# 选择一款Jekyll的主题

github.io默认采用Jekyll作为建站工具。[Jekyll](https://jekyllrb.com/" \t "https://blog.csdn.net/walkerhau/article/details/_blank)是一款当前火热的开源的静态网站建站工具，拥有非常庞大的使用群里和社区，其[Github](https://github.com/jekyll/jekyll" \t "https://blog.csdn.net/walkerhau/article/details/_blank)截止本文，已经有超过3W+的star，拥有丰富的插件，丰富的主题，并且有无数的人已经帮你早出了无数的轮子可供参考。Jekyll自身的强大功能已经足够你打造自己心仪的静态网站（这里注意的是静态网站，Jekyll没有任何的后台数据库），然而前提是你自己还是得有一定的前端功底，而为了不至于长的太难看，你还得有一定的设计能力。这一下子把大部分人给难住了，咱们只是为了单纯的写写博客啊，至于有这么多要求吗？看到这里，很多人可能觉得此法不怎么方便啊，然则，正如刚刚反复强调的，Jekyll已经有一个非常庞大的社区，这就意味着，你完全可以借鉴别人已经造好的轮子，放在Jekyll这里，咱们应该成为主体（Theme）比较合适。本文推荐国内用户可以考虑一款[国人开发的主题](https://github.com/Huxpro/huxpro.github.io" \t "https://blog.csdn.net/walkerhau/article/details/_blank)。[博主的个人博客](https://keysaim.github.io/" \t "https://blog.csdn.net/walkerhau/article/details/_blank)即是采用了这个主题。

Fork出自己的repo

为了便于管理，建议先把[Huxpro](https://github.com/Huxpro/huxpro.github.io" \t "https://blog.csdn.net/walkerhau/article/details/_blank) fork到自己的账户下

clone主题

$ git clone git@github.com:keysaim/huxpro.github.io.git

* + 1

当然，你也可以直接clone它的样板repo

$ git clone git@github.com:Huxpro/huxblog-boilerplate.git

* + 1

添加自己的github.io git repo   
clone了huxpro的repo之后，需要将自己创建的github.io repo设置到clone的repo：

$ cd huxpro.github.io

$ git remote add mine git@github.com:keysaim/keysaim.github.io.git

* 1
* 2

其中，请自行将其中的repo url换成你自己的。

修改必要的配置

clone之后的repo其实是Huxpro自己的博客网站，里面有非常多作者自己的博文，可根据自己的需要进行必要的删减。基于Jekyll的博客网站，对于配置，非常重要的一个文件是\_config.yml文件，代开这个文件进行必要的修改：

# Site settings

title: 窗外蟋蟀

SEOTitle: 窗外蟋蟀的博客 | Keysaim Blog

header-img: img/home-bg-hill.jpg

email: keysaim@gmail.com

description: "描述"

keyword: "窗外蟋蟀, keysaim"

url: "https://keysaim.github.io"

baseurl: ""...

* + 1
  + 2
  + 3
  + 4
  + 5
  + 6
  + 7
  + 8
  + 9
  + 10
  + 11
  + 12

# 编写发布博客

Jekyll对于博文，都是要求放在\_posts目录下面，同时对博文的文件名有[严格的规定](https://jekyllrb.com/docs/posts/" \l "creating-post-files" \t "https://blog.csdn.net/walkerhau/article/details/_blank)，必须保持格式YEAR-MONTH-DAY-title.MARKUP，通常情况下，咱们采用推荐的Markdown撰写博文，基于该格式，本博文的文件名为2017-08-15-how-to-setup-your-github-io-blog.md。

写好博文之后，就可以通过git提交博文了：

$ git add \_posts/2017-08-15-how-to-setup-your-github-io-blog.md

$ git commit -m "Add how to setup your github.io blog"

$ git push mine master

* 1
* 2
* 3

其中，mine就是前面设置的你自己的repo。等一会（通常几秒到几十秒不等），就可以打开自己的博客网站查看博文了，这里是我的博客网站[https://keysaim.github.io](https://keysaim.github.io/" \t "https://blog.csdn.net/walkerhau/article/details/_blank)。push成功之后，最好是再重新将自己的repo clone到本地。

## 本地查看自己的博客

有时候，在提交到github之前，咱们总想先看看博文的效果如何，既然github采用的也是Jekyll，那么咱们完全可以采用Jekyll在本地构建网站，查看博文效果。

安装Ruby（请务必确保版本在1.9.3以上）

可以参照[官网教程](https://www.ruby-lang.org/en/downloads/" \t "https://blog.csdn.net/walkerhau/article/details/_blank)进行安装，这里，如果你使用的是Mac，可以安装如下：

$ brew install ruby

* + 1

安装Github pages

Github pages其实就是github基于Jekyll用来构建github.io的工具，安装好Ruby之后可以执行：

$ gem install github-pages

* + 1

开启Jekyll本地服务

$ cd keysaim.github.io$ jekyll serve --watch

* + 1
  + 2

默认情况下，该服务会侦听在本地4000的端口上，可以打开浏览器访问http://127.0.0.1:4000，这样就可以在本地查看自己的博文效果了。

# 结语

github.io通过基于Jekyll工具的[Github pages](https://pages.github.com/" \t "https://blog.csdn.net/walkerhau/article/details/_blank)来自动构建网站，同时本身又是github的repo，为使用者提供完全的内容控制，十分便利灵活。当然，要用好这个工具，还是对你有一定的要求：

必须得有基本的git使用基础

必须对前端有一定的概念

必须较为熟悉Markdown撰写

不过，幸运的是，这三点不论哪一点，其实都是相对较为简单的，git的基本使用，Google一下，我相信1小时学会不是太大问题；对前端的要求，由于前人已经造了足够多的轮子，咱们完全可以先借鉴别人的，等以后慢慢熟悉起来之后再考虑自己DIY；对于Markdown，个人觉得其语法真的是非常的简单，只要静下心来，一个下午足够让你编写基本的文章了。

最后，这里再次感谢[Huxpro](https://github.com/Huxpro/huxpro.github.io" \t "https://blog.csdn.net/walkerhau/article/details/_blank)提供本站的Jekyll主题。