

博客就是一个人放自己文章的地方。即，写作的地方。

我换过很多写作的地方。

- 2012开始，我用的是印象笔记；后来换了为知笔记。

换为知笔记是为了无限层级和手机上也能离线阅读笔记。但这两个笔记软件都是自己看的。

- 2014年我尝试了公开写作，当时是用wordpress搭建了一个博客网站。后来域名空间到期了就没再续。

这个阶段，体会到了公开写作带来的好处：可以对写出来的东西设置一个『及格线』，写东西会有底线，如果实在不好，也没脸放出来。另外就是了解了一些博客的设置，但这段写作时间持续了没多久就因为别的事情中断了。而wordpress的好处呢，就是可以高度自定义。

- 2015年，也就是去年的时候我开始在微信公众号上写作。中间断断续续发生了很多事情，但基本上也是一直在思考、一直在写，直到现在。

这个写作平台的好处是比较方便，微信平台已经给你设计好了很多接口，只要写就好了。但缺点也很明显，我觉得很不方便的缺点是：

- i. 没有全文搜索！这很要命，本来写作的一大作用就是促进思考、记录思考过程，而如果没有全文搜索，你想回顾之前的某段、句话的时候，真的很难找到。
- ii. 没有目录，虽然我后来也做了一个手动目录，但也没有归档可以做。尤其像我这种带点强迫症的人，没有系统的归档，真的让我很不爽。
- iii. 发布后不能修改。

- 恩，现在呢，又发生了一些新的事情。我要当个工程师了，搞编程。所以首要的就是开始像个工程师一样学习、干活（装逼）。搭一个方便、又能进行相当程度的自定义的博客是很重要的。最重要的是要有归档，因为没有归档，就没有系统的成长。另外，博客是可以随意修改已经发布的文章的。

## 插一个话题：写博客其实有个坑

比如你每天要发一篇东西在微信公众号上，但如果你时间紧或者时间不紧但你没好好思考，那就写不出有价值的东西来。这时，有两个选择，一个是不写，一个是随便写写先把今天的发了。

这里很关键：如果你选择了后者，随便写写，虽然你完成了今天的任务，但其实对自己来说收

获很小，对读者来说，也是不负责任。谁也不是冲你随便应付的文章来看你的。

其实另外一个选择其实是更好的选择：不管发生什么，一旦你要写作，一定是系统的思考先行。没有思考，就不要写作。要写出对读者有价值的东西，然后才发。不然，干脆就不写或者不发。不要给自己设定今天必须发的限制，这其实不好。但注意，你今天可以不发，但，每天都必须思考、进步。

这两天的时候抽空搜了几个教程，用hexo+github搭了这么一个[博客](#)。

这个过程中踩了很多的坑，而且网上搜到的教程，我翻了好多才找到解决问题的办法，并没有任何一篇教程能一站式地让人顺利建好一个好用的博客。走了过来，就想要写一个尽量说人话的教程，为后人，也为自己备忘用。

## 0. 综述

这里搭建的博客，是用hexo和github pages配合搭建的。

核心的流程是下面这样：

1. hexo是个博客框架，编辑好博客以后，用在git shell上，用『hexo g』命令就能生成一个静态页面，这个网站可以在本地看，也可以在网上看。如果想要所有人都能在网上看到这个页面，那就得上传到网上，具体而言，就是用github上传。
2. 嘿，说来也巧，正好github有个叫github pages就正好有这个功能。新建一个yourname.github.io仓库，这个特殊仓库会自动把github pages功能打开，就是说，会自动把这个仓库中master里的html静态文件显示到http://yourname.github.io这个网站上。
3. 这么一看，hexo和github pages配合一下，奇妙的事情就发生了：你的博客其实就是html，用hexo在本地编辑完以后，把生成的html放到github中上传，然后全世界所有人在浏览器中输入：http://yourname.github.io,就能看到你写的博客了。

## 1. 准备一些东西

下面是基于windows说明的。如果你的电脑也是windows系统，那么，现在可以跟我一起，开始从头搭建一个属于自己的博客。

在后文，我把要准备的东西分成操作（怎么做）和作用（为什么）两部分。

### 1. Git

- 操作：

- 登录[github.com](https://github.com)注册一个github账号。注意，注册账号的时候你起的名字之后会经常用到，考虑好，别想着经常改了。这个名字就是『yourname』
- 下载桌面端github，安装。
- 作用：
  - 要用github账号里的github pages功能 → 其中的master分支里的html文件可以通过自己的yourname.github.io来访问
  - 要用github桌面端来处理进行很多操作，这个基础。

## 2. Node.JS

- 操作：去[官网](#)下载LTS稳定版。
- 作用：Node.JS是Javascript运行环境(runtime)，目前就需要知道，建博客，得先有个它。

## 3. npm

- 操作：不用操作，但要了解作用。很多代码会提到npm这三个字母。
- 作用：NodeJS包管理和分发工具。

## 4. hexo

- 操作：打开git shell，输入 `npm install -g hexo-cli`，回车。
- 作用：输入命令以后，程序会自动下载安装hexo。

到此，需要准备的东西完。

# 2. HEXO搭建本地博客

1. 在电脑里新建一个名为『hexo』的文件夹。
2. 把git-shell打开，开始使用命令行操作（安好github以后有两个.exe文件，一个是图形界面（GUI）形式的，一个是git-shell，命令行形式。）
  - i. 进入到前面的hexo文件夹路径下。可是对新手来说，怎么进入都是个问题：
    - 改变工作盘符：从d到f盘，直接在任意工作目录下输入『f:』，然后回车就好。
    - 用cd命令：cd（change directory），用来改变工作目录；不能改盘符，只能改同一盘符下的路径。
      - cd+空格+ [完整路径]：直接跳进某个路径
      - cd+空格+ 【子目录名】：进入当前工作目录下的某个子目录（即前进）
      - cd+空格+ 【..】：后退一层
      - cd+空格+ 【\】：直接后退到根目录（根目录就是这个盘符的最底层，比如f:）
  - ii. 初始化hexo：输入『hexo init blog』：用来在当前目录下新建一个名为『blog』（可以自己定）的子文件夹，然后在这个文件夹里初始化hexo。这个初始化的过程会新建一些文件。
    - 代码跑完是这个样子的：

- 之后的blog是这个样子的：
- 注意：『blog』就是你的博客根目录，所有的操作都在里面进行。『在blog路径下输入一个命令』 = `$+空格+命令`

iii. 为hexo安装插件hexo-deployer-git

- 操作： `$ npm install hexo-deployer-git --save`
- 作用：其实是用来往git中上传html时用的，这里先准备好。
- hexo的博客插件可以在(blog\node\_modules)路径中看到。

iv. 生成静态页面： `$ hexo g` 或 `$ hexo generete` （前者是简写）

v. 本地预览

- 输入 `$ hexo s / $ hexo server` ,就能在本地进行预览和调试。
- 输入完以后，在浏览器里输入『 <http://localhost:4000> 』 就能预览。

## 3. 部署/发布到github

1. 通过浏览器进入 <https://github.com> ，登录你注册好的账号。
2. 新建仓库『New Repository』，名字为yourname.github.io。yourname就是你之前注册账号时用的名字。
  - 一般新建仓库都很快，但这个是一个特殊的仓库。仓库的settings里有一个可供打开github pages的功能，但如果你起的仓库名是这个特殊的格式，审核通过了就会自动把这个功能打开。
3. hexo 和 github pages建立连接。
  - 前面说过，这种建博客的核心就是：用hexo搭好本地博客，然后用github传到网上。所以现在要做的就是告诉hexo，本地博客要传到github上。
  - 用来建立连接的就是blog文件中的\_config.yml文件，用任何一个文本编辑器打开它。（这个文件用来配置博客的基本功能）
    - 翻到最下面，改成下面这样：

```
deploy:
  type: git
  repo: git@github.com:icearl/icearl.github.io.git
  branch: master
```

（这里有个坑：『: 』后面要跟一个空格!!! 程序设计就是这样，一个符号错了和少了都不行

- 这一步就是告诉hexo，在发布/deploy/部署的时候，1. 部署类型是用git发2.要发到自己指定的github的指定仓库里3.要发到这个指定仓库的master分支里。
  - 其实这个.yml文件里还可以配置博客页面上的其他东西，比如作者的名字。
4. 发布博客：输入 `$hexo d / $ hexo deploy`
    - 这就发布完了，现在在浏览器里输入yourname.github.io，就能看到你在本地预览时看到的helloworld博客了。

## 4. 用markdown写文章

---

### 1. 文章编辑器:

- 选一款文章编辑器，我用的是atom。
- 以后跟文本相关的编辑任务（比如写作和编程），就都用它了。

### 2. markdown相关的配置

- 为atom安装插件：安装方法是ctrl+,打开settings，然后搜索以下插件名install就好。
  - markdown-scroll-sync
  - linter-markdown
  - markdown-writer
  - markdown-toc
  - markdown-pdf
  - markdown-preview: 这个要经常用，快捷键：ctrl + shift +m，就能呼出一个.md文件右侧的渲染预览。
- 为hexo安装插件：
  - markdown是纯文本标记语言，需要解析器来渲染。前面的markdown-preview是atom里的解析器。现在我们在hexo生成html博客的时候还需要自己的解析器。
  - hexo自带的是hexo-renderer-marked解析器，不够吊，要换成hexo-renderer-markdown-it。但在换之前要把旧的卸载了，只要输入代码就好：

```
$ npm un hexo-renderer-marked -save
$ npm i hexo-renderer-markdown-it -save
```

### 3. 编辑文章:

- 打开git shell → `$ atom .`
- atom会被自动打开，进到（source\_posts）：编辑一下hello-world.md文档（编辑的时候可以打开ctrl+shift+m进行实时预览）
- `$ hexo g`：生成静态html文件
- `$ hexo s`：本地查看（在浏览器中输入这个地址：localhost:4000）
- `$ hexo d`：发布到github中

### 4. 发布新文章

- 在source\_posts目录中创建新的.md文件并编辑。
- `$ hexo g` 生成。
- `$ hexo s` 预览 → `$ hexo d` 发布。

---

再补两个坑：

- 一开始新建的yourname.github.io，只要10分钟以后审核通过了，这个网站就能访问了，但是如果你没有新建hexo博客，而是直接先打开这个网址，那会出现404。这时，你需要提供一个html文件。为了预览，可以在这个仓库里随便添加一个后缀为html的文件，随便写点什么，就不会404了。

2. 如果你按我的教程一步步做了，而且以前用过markdown的话，就会发现，这默认的blockquote（> 符号触发的引用块）真是垃圾啊，什么鬼东西，字体比正文大不少，而且还自动居中，看着非常难受，根本无法排版。这个问题当时我都不知道怎么回事，后来还是找到了。。
- 进入 `blog\themes\landscape\source\css\_partial` 这个路径
  - 用atom打开`article.styl`这个文件，`ctrl+f`，输入`blockquote`,在`blockquote`和`footer`之间输入以下三行代码，定制一下比较正常的引用块格式：

```
border-left:5px solid #DDD;
margin:15px 30px 0 10px;
padding-left:20px;
```

---