

Typora+Git搭建云笔记本

前言

熟悉Git的操作

拥有码云账号

注意：由于Typora的限制，此方式只适用于做文本+图片比较，无法存放其他类型的文件

Typora安装与设置

安装软件

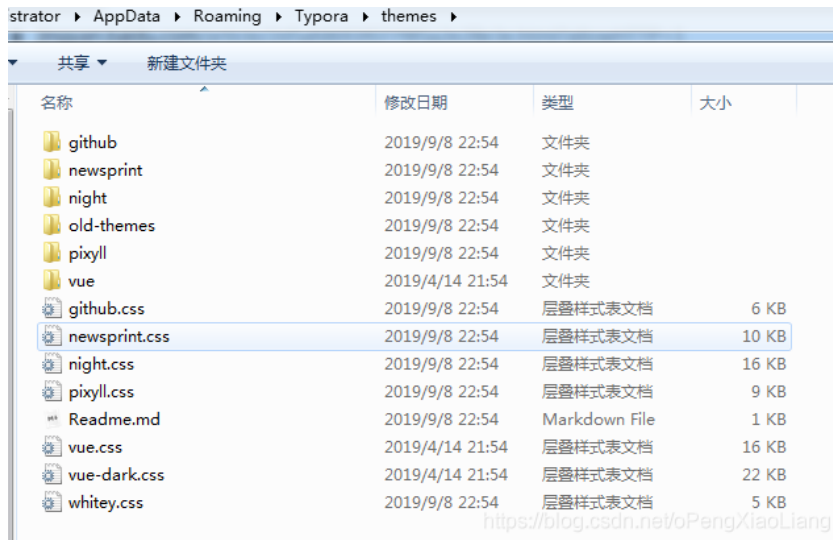
1. Typora
<https://www.typora.io/windows/typora-setup-x64.exe>
2. Git
<https://git-scm.com/download/win>

Git具体操作可以参考教程 <https://www.liaoxuefeng.com/wiki/896043488029600>

安装Typora主题插件

自带的插件不好看，推荐安装主题《Vue》

1. 下载主题压缩包
<http://theme.typora.io/theme/Vue/>
2. 文件 – 偏好设置 – 外观 – 打开主题文件夹



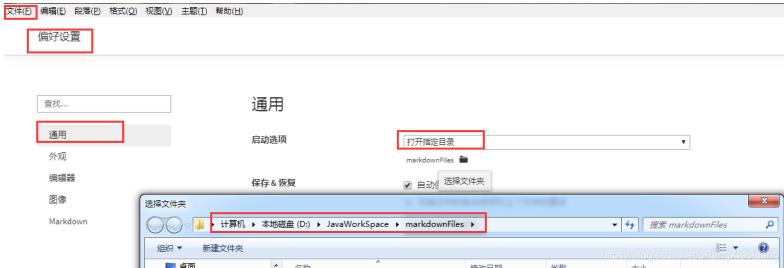
3. 将解压后的内容整个放进去，重启Typora



4. 点击主题，即可看到新加入的主题Vue，选中即可

配置Typora

1. 本地新建存放笔记的文件夹，设置打开Typora时自动打开此文件夹



2. 设置图片的存放模式，这样图片就会被自动分类，从图床中解脱出来



Git仓库与本地笔记本连接

仓库新建markdownFiles项目

过程略

markdownFiles文件夹与项目关联

git全局设置:

```
1 | git config --global user.name "Potato"
2 | git config --global user.email "823353549@qq.com"
```

创建本地 git 仓库，并与Git仓库关联:

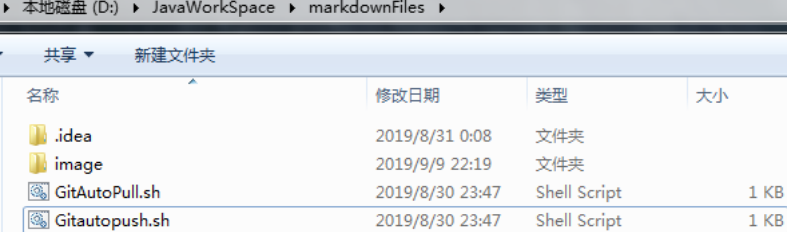
```
1 | #本地之前还没有文件夹，新建一个并与GIT项目关联
2 | mkdir markdownFile
3 | cd markdownFile
4 | #我们之前已经建立markdownFiles文件夹，所以直接从这里开始执行
5 | git init
6 | touch README.md
7 | git add README.md
8 | git commit -m "first commit"
9 | git remote add origin https://gitee.com/xxxx/markdownFile.git
10 | git push -u origin master
```

如果本身是Git本地仓库，通过下面的来建立关联，一般很少用

```
1 | #本地已有文件夹，通过命令与Git项目建立关联
2 | cd existing_git_repo
3 | git remote add origin https://gitee.com/xxxxx/markdownFile.git
4 | git push -u origin master
```

通过脚本推送拉取文件

在markdownFiles文件夹中新建两个sh文件



| 名称 | 修改日期 | 类型 | 大小 |
|----------------|-----------------|--------------|------|
| .idea | 2019/8/31 0:08 | 文件夹 | |
| image | 2019/9/9 22:19 | 文件夹 | |
| GitAutoPull.sh | 2019/8/30 23:47 | Shell Script | 1 KB |
| GitAutoPush.sh | 2019/8/30 23:47 | Shell Script | 1 KB |

Pull脚本

```
1 #!/bin/bash
2 # 上面中的 #! 是一种约定标记, 它可以告诉系统这个脚本需要什么样的解释器来执行;
3
4 git pull origin master
```

Push脚本

```
1 #!/bin/bash
2 # 上面中的 #! 是一种约定标记, 它可以告诉系统这个脚本需要什么样的解释器来执行;
3
4 echo "放弃有道云的利器 GitAutoPush Starting..."
5 time=$(date +%Y-%m-%d %H:%M:%S)
6 git add .
7
8 read -t 30 -p "请输入提交注释:" msg
9
10 if [ ! "$msg" ];then
11     echo "[commit message] 默认提交, 提交人: $(whoami), 提交时间: ${time}"
12     git commit -m "默认提交, 提交人: $(whoami), 提交时间: ${time}"
13 else
14     echo "[commit message] $msg, 提交人: $(whoami), 提交时间: ${time}"
15     git commit -m "$msg, 提交人: $(whoami), 提交时间: ${time}"
16 fi
17
18
19 git push origin master
20 echo "放弃有道云的利器 GitAutoPush Ending..."
```