GIT安装的初级教程，没有写分支、合并和标签~

## 1.简单介绍

首先，GIT是什么就不介绍了，大本营<https://www.github.com>上资料是非常完整的。简单说一下Msysgit、git for windows 、TortoiseGit和Cygwins。大本营上能下的是git for windows，下载下来的是一个在线安装包，但是本人在校园网环境下，安装不成功，在线资源下载不下来。

安装成功的是开源的Msysgit，该项目托管在googlecode上，包括GIT GUI、GIT Bash、ssh和git for windows没有的toolchain。Msysgit安装完成基本就能满足代码版本控制的需要了。

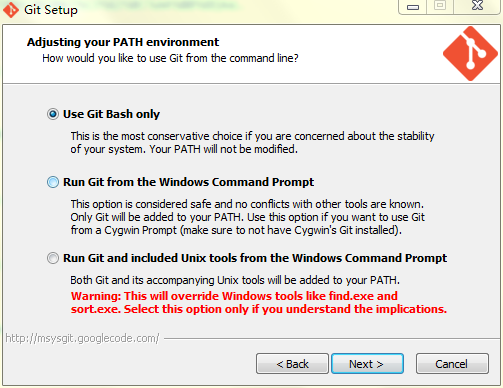
而TortoiseGit其实只是把GIT Bash里操作命令封装，并且图形界面化了，想要跑起来必须安装前两者之一。比如先完成Msysgit，然后安装TortoiseGIT，安装时会自动找到Msysgit下的git.exe，安完右击菜单就有了一系列的嵌好了的命令。

最后的那个Cygwin是Unix的模拟环境，也可以下载GIT相关插件，必须在没安装相关插件的时候先装好Msysgit，不怎么了解。

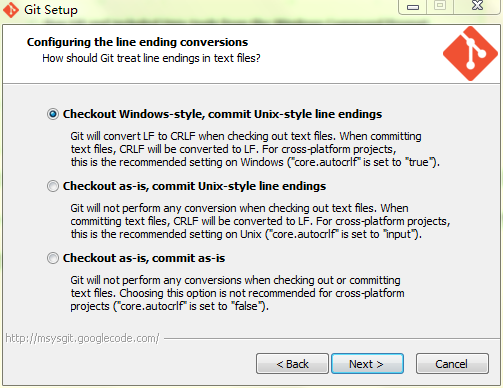
顺带提一点的是，本人用的Sublime，也有插件支持GIT~不愧是最性感的编辑器呐~

## 2．安装Msysgit

下载开源的[Msysgit](https://code.google.com/p/msysgit/downloads/list)，其中几个可选项简单介绍一下：



我选择的是第二项，如果你不想改变自己的PATH（PATH是否存在决定能否在命令行任意位置直接调用），只用Bash，那么就选第一项；如果想要把附带的Unix工具也安上，那就选第三项，不保证安全性。

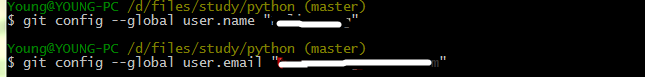


这里本人选的默认，即第一个选项，该选项适合windows平台上开发可迁移的Unix项目；如果平台是Linux等，选第二项，如果确定项目不需要迁移，则选择第三项。该选项页对应于配置的core.autocrlf项。

## 3.配置本地环境

### 3.1本地信息

首先说明，大家应该都了解一些公钥和私钥，由于公钥和私钥是任一都不能泄露的，所以本文图片中的出现的公私钥会被部分涂掉，望谅解；由于本人目前其实项目经验有限，所以直接套用网上最常见设置：



白条处填的分别是用户名（随意）和账户绑定的那个初始的Email地址。一般配置这两项足够，如果有需要，更多的设置命令可以查看：

*git config –h*

如果说同时做多个项目，就去掉--global，每次更新到不同的项目时输入不同的config user.name。此处并不完整需要再自行百度。

### 3.2建立本地文件夹

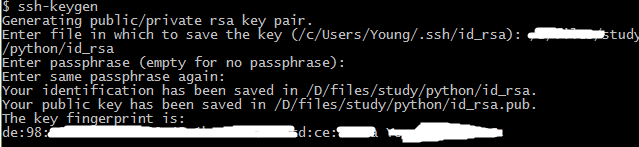
然后建立一个文件夹，可以是随意位置，cd进去之后输入：

*git init*

注意，如果没有输入该指令，无法进行第4部的所有指令。

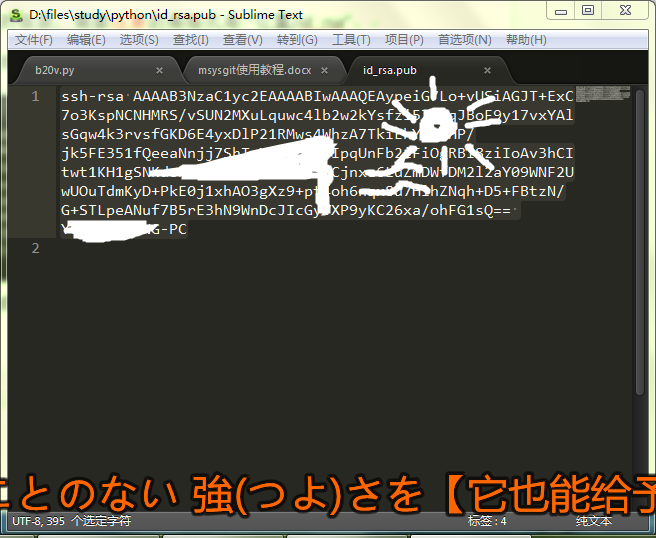
### 3.3密钥对配置

打开Bash输入ssh-keygen生成不对称加密密钥：

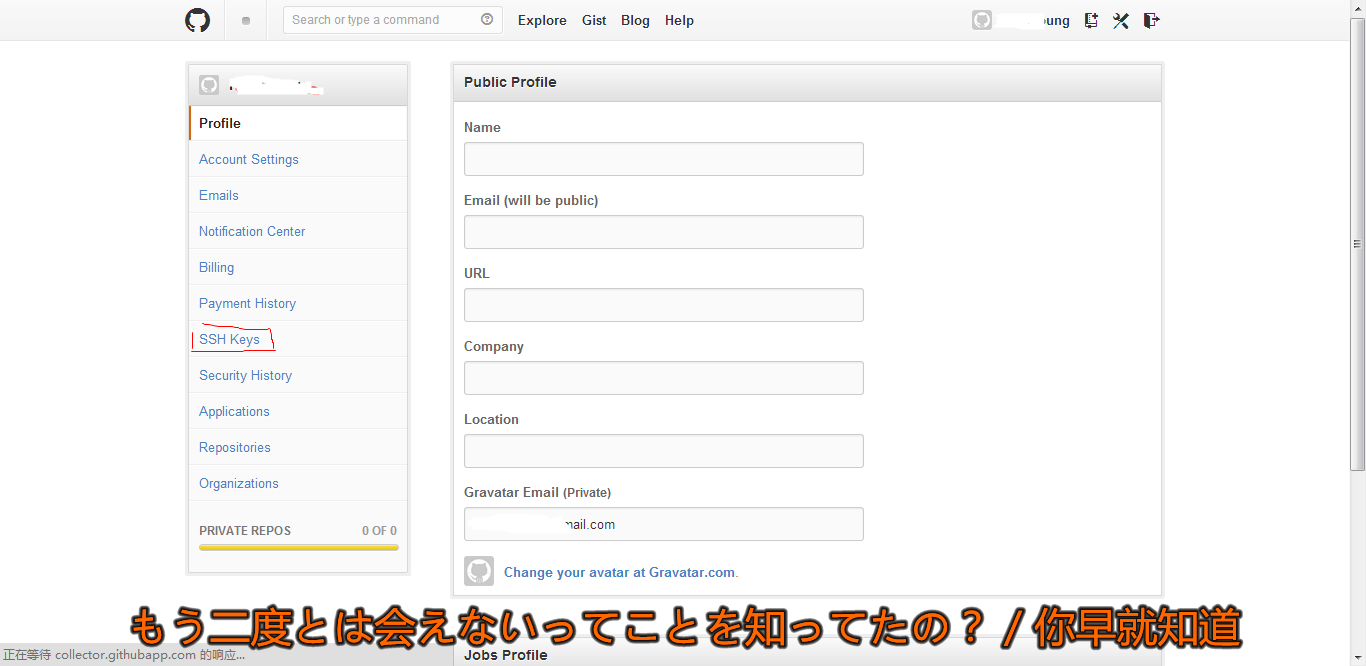


中间会询问保存的目录，需要注意的是格式为:“位置/id\_rsa”。然后询问密码，没有回显，而且需要确认输入。本人没有使用密码，由于每次高权限操作都需要键入密码非常浪费时间，所以直接敲回车两次结束。

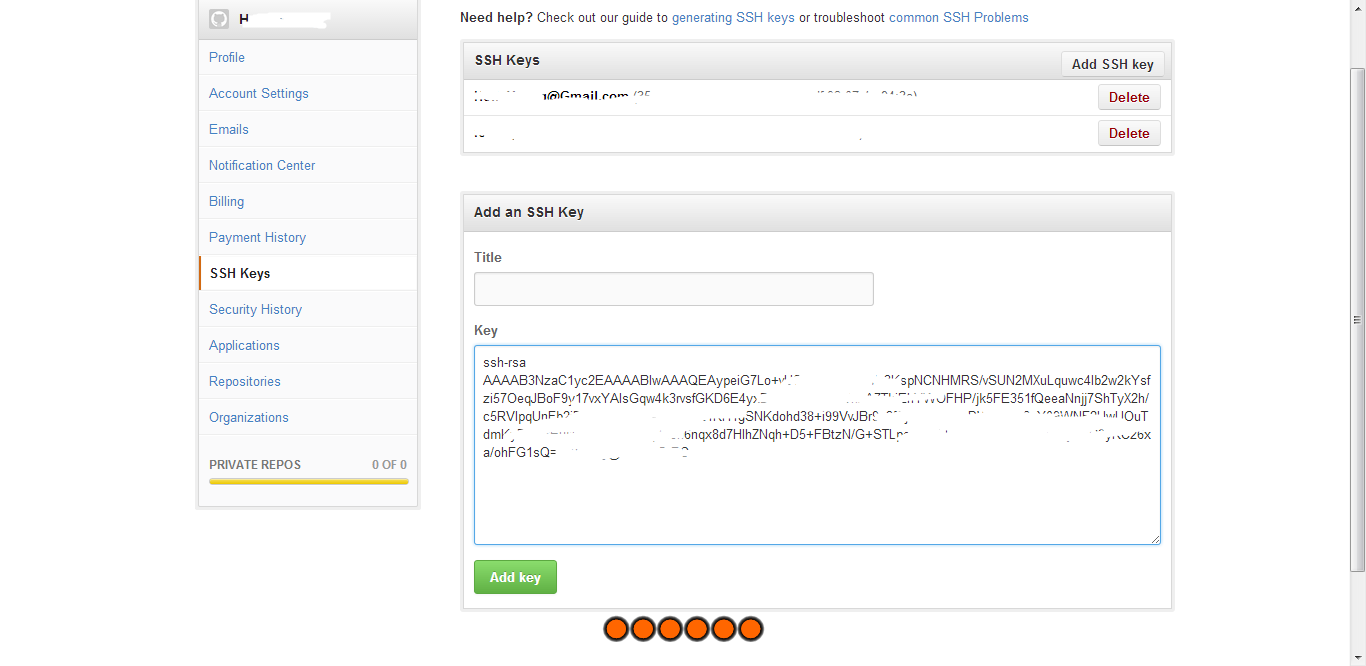
打开目录位置，用记事本打开 id\_rsa.pub，复制所有的字符串：



打开官网，登录或注册登录，点右上Account Setting（请无视桌面歌词）：



点红色框出来的ssh keys，点Add，直接复制到Key文本框：



点绿色的Add key，不需要Title。

至此，密钥配对部分完成。

### 3.4SSH连接方式

配置完公私玥，想查看ssh连接是否可用可以输入命令（无需改动）：

*ssh –T* [*git@github.com*](mailto:git@github.com)*。*

这里可能出现的错误是:

*ssh: connect to host github.com port 22: Bad file number。*

出现这种错误主要是ssh要用到的Port 22可能由于本地网关的安全策略而关闭。但是可以跳过查看连接可用性，直接进行之后的操作。如果之后的ssh操作仍然不可用，就需要改换成https形式进行操作。

然后检查是否存在ssh授权代理：

*ps aux |grep agent*

应该出现一行。出行多行需要全部kill并重启Bash，未出现不需要操作，也不需要上网搜索解决方法，继续做agent会自动启动。

一如后面会出现的那样，ssh连接的远程地址参数格式是（所有git三个字符不要改动）：

*git@github.com:username/repository\_name.git*

另一个可行的地址参数格式是：

*username@github.com:path/to/repository*

### 3.5https连接方式

https连接方式的远程地址参数格式是:

[*https://github.com/username/path/to/repository*](https://github.com/username/path/to/repository)

但是每次都需要键入用户名和密码，想本地保存密码需要下载安装：[GIT密码缓存助手](http://blob.andrewnurse.net/gitcredentialwinstore/git-credential-winstore.exe)。

## 4与远程通信命令

首先需要用指定字符串代换远程地址：

*git remote add addname git@github.com:username/repo\_name.git*

可以用remove取消字符串与地址代换。

完整的-add选项说明如下，入门可以跳过：

|  |  |
| --- | --- |
| -t <branch> | 分支名，见后续 |
| -m <master> | 远程的分支名，master是默认的主分支 |

### 4.1add和commit命令

常用格式为：

*git add [–a|’文件名’]*

*git commit –m ‘描述信息’*

该命令用于提交修改，只影响本地文件，并不影响远程文件，注意每次commit必须要有描述信息。

### 4.2push命令

常用格式为：

*git push [远程地址|字符串代换（addname）] [本地分支]：[远程分支]*

push命令会检查远程的文件和文件目录，如果在本地文件未上传期间，又有其他人修改了文件，那么本次命令会失败，并且需要先将远程文件pull（4.3pull命令）下来。push后远程的repository才发生了修改。

### 4.3pull命令

常用格式为：

git pull addname 本地分支名（一般为master）

## 5常用网址

这是一个非常好看的多国语言版网址：

<http://rogerdudler.github.io/git-guide/index.zh.html>

其它的就算了，不是很好用。实在有问题建议去github官网。