github常见操作和常见错误！错误提示：fatal: remote origin already exists.

如果输入$ git remote add origin [git@github.com:djqiang（github帐号名）/gitdemo（项目名）.git](mailto:git@github.com:djqiang/gitdemo.git)

    提示出错信息：fatal: remote origin already exists.

    解决办法如下：

    1、先输入$ git remote rm origin

    2、再输入$ git remote add origin [git@github.com:djqiang/gitdemo.git](mailto:git@github.com:djqiang/gitdemo.git) 就不会报错了！

    3、如果输入$ git remote rm origin 还是报错的话，error: Could not remove config section 'remote.origin'. 我们需要修改gitconfig文件的内容

    4、找到你的github的安装路径，我的是C:\Users\ASUS\AppData\Local\GitHub\PortableGit\_ca477551eeb4aea0e4ae9fcd3358bd96720bb5c8\etc

    5、找到一个名为gitconfig的文件，打开它把里面的[remote "origin"]那一行删掉就好了！

    如果输入$ ssh -T [git@github.com](mailto:git@github.com)  
    出现错误提示：Permission denied (publickey).因为新生成的key不能加入ssh就会导致连接不上github。

    解决办法如下：

    1、先输入$ ssh-agent，再输入$ ssh-add ~/.ssh/id\_key，这样就可以了。

    2、如果还是不行的话，输入ssh-add ~/.ssh/id\_key 命令后出现报错Could not open a connection to your authentication agent.解决方法是key用Git Gui的ssh工具生成，这样生成的时候key就直接保存在ssh中了，不需要再ssh-add命令加入了，其它的user，token等配置都用命令行来做。

    3、最好检查一下在你复制id\_rsa.pub文件的内容时有没有产生多余的空格或空行，有些编辑器会帮你添加这些的。

    如果输入$ git push origin master

    提示出错信息：error:failed to push som refs to .......

    解决办法如下：

    1、先输入$ git pull origin master //先把远程服务器github上面的文件拉下来

    2、再输入$ git push origin master

    3、如果出现报错 fatal: Couldn't find remote ref master或者fatal: 'origin' does not appear to be a git repository以及fatal: Could not read from remote repository.

    4、则需要重新输入$ git remote add origin[git@github.com:djqiang/gitdemo.git](mailto:git@github.com:djqiang/gitdemo.git)

    使用git在本地创建一个项目的过程

    $ makdir ~/hello-world    //创建一个项目hello-world  
    $ cd ~/hello-world       //打开这个项目  
    $ git init             //初始化   
    $ touch README  
    $ git add README        //更新README文件  
    $ git commit -m 'first commit'     //提交更新，并注释信息“first commit”  
    $ git remote add origin [git@github.com:defnngj/hello-world.git](mailto:git@github.com:defnngj/hello-world.git)     //连接远程github项目    
    $ git push -u origin master     //将本地项目更新到github项目上去

    gitconfig配置文件

         Git有一个工具被称为git config，它允许你获得和设置配置变量；这些变量可以控制Git的外观和操作的各个方面。这些变量可以被存储在三个不同的位置：   
         1./etc/gitconfig 文件：包含了适用于系统所有用户和所有库的值。如果你传递参数选项’--system’ 给 git config，它将明确的读和写这个文件。   
         2.~/.gitconfig 文件 ：具体到你的用户。你可以通过传递--global 选项使Git 读或写这个特定的文件。  
         3.位于git目录的config文件 (也就是 .git/config) ：无论你当前在用的库是什么，特定指向该单一的库。每个级别重写前一个级别的值。因此，在.git/config中的值覆盖了在/etc/gitconfig中的同一个值。  
        在Windows系统中，Git在$HOME目录中查找.gitconfig文件（对大多数人来说，位于C:\Documents and Settings\$USER下）。它也会查找/etc/gitconfig，尽管它是相对于Msys 根目录的。这可能是你在Windows中运行安装程序时决定安装Git的任何地方。

        配置相关信息：

        2.1　当你安装Git后首先要做的事情是设置你的用户名称和e-mail地址。这是非常重要的，因为每次Git提交都会使用该信息。它被永远的嵌入到了你的提交中：

　　$ git config --global user.name "John Doe"

　　$ git config --global user.email [johndoe@example.com](mailto:johndoe@example.com)

       2.2    你的编辑器(Your Editor)

　　现在，你的标识已经设置，你可以配置你的缺省文本编辑器，Git在需要你输入一些消息时会使用该文本编辑器。缺省情况下，Git使用你的系统的缺省编辑器，这通常可能是vi 或者 vim。如果你想使用一个不同的文本编辑器，例如Emacs，你可以做如下操作：

　　$ git config --global core.editor emacs

      2.3 检查你的设置(Checking Your Settings)

　　如果你想检查你的设置，你可以使用 git config --list 命令来列出Git可以在该处找到的所有的设置:

　　$ git config --list

      你也可以查看Git认为的一个特定的关键字目前的值，使用如下命令 git config {key}:

　　$ git config user.name

      2.4 获取帮助(Getting help)

　　如果当你在使用Git时需要帮助，有三种方法可以获得任何git命令的手册页(manpage)帮助信息:

　　$ git help <verb>

　　$ git <verb> --help

　　$ man git-<verb>

　　例如，你可以运行如下命令获取对config命令的手册页帮助:

　　$ git help config