# GitHub和Git入门

## 一些必要到配置

开始

git config --global user.name "xxx" // 用户名

git config --global user.email "xxx@163.com" // 邮箱

查看git配置信息

git config -l

配置完之后，这台电脑上所作的所有修改都会自动将用户名，email发送到主程序中

GitHub的简单使用

第一步 创建一个本地hello-world仓库

1. 在命令行输入 mkdir hello-word，创建一个新文件夹。你可以使用ls命令来查看当前目录下有哪些文件和文件夹。

2. 输入cd hello-world进入新文件夹，注意在输入命令时，你可以用Tab键来自动补全。

3. 输入git init初始化Git仓库。此时用ls -a查看当前目录，可以看到多了一个.git/的文件夹，此文件夹保存了版本控制的所有相关信息。

4. 输入echo "This is a simple practise" > readme.txt，创建一个readme.txt文件。

5. 输入git status查看当前版本库状态，在Untracked files(未跟踪文件)下，会出现红色的readme.txt，代表此文件还未被Git所管理。

1. 使用git add readme.txt，将该文件加入缓冲区，如果你确定所有的修改都需要提交，可以使用git add .来加入所有修改。现在用git status查看，将看到文件名变为绿色。

7. 使用git commit -m "This is my first commit via Git!"来提交修改，-m后面所带的参数是本次提交信息，一般用来记录本次提交的主要意图。

8. 提交成功后，可以用git log查看历史提交记录。每个记录都会有提交id，作者和提交日期。

9. 你可以用git branch查看当前有哪些分支，当然，因为我们没有创建任何分支，目前只会有一个master分支。

10. 使用git checkout -b feature创建一个名为feature的分支，再用git branch查看一下。

11.git remote add origin https://github.com/YourAccountName/name.git -> 这样就可以用origin这个名字来引用添加的远程仓库

12.git fetch origin -> 并不能看到工作目录下有任何变化，只是把远程的数据抓取到本地，而不会把改动合并到当前的分支上

13.git pull https://github.com/YourAccountName/name.git -> 把远程仓库抓取到本地，并合并本地master分支

14.git push -u origin master -> 将本地的数据更新到远程仓库中

## 项目的下载，查看和修改

第一步. 从GitHub上下载我们的项目代码。

1. 以Hello-World项目为例，点击绿色按钮Clone or download，然后在弹出窗口中点击剪切板图标，复制仓库的URL。

2. 在git bash中输入git clone https://github.com/feicui-android/Hello-World.git，下载项目源码。

第二步. 查看版本历史

1. cd到项目文件夹下，使用git log能看到我们的历史提交记录。

2. 要回到某一历史版本，可以使用git checkout commitId，看完后要回到最新代码，使用git checkout master。

第三步. 本地修改代码

你可以在我们的代码基线上任意修改，但为了下载新代码时不出现冲突，请遵循以下步骤：

1. 下载新代码：git pull。

2. 从master出捡出一个新的分支：git checkout -b feature。feature是分支名称，你可以随意取名，但请用英文。

3. 在feature分支上随意修改，改完后你可以提交你的修改：git commit -m "some message"。

4. 此时要同步代码，请先切回主分支：git checkout master，然后更新git pull。

5. 如果想删除自己建立的分支，使用git branch -D feature，注意执行此命令后分支被强制删除，无法恢复。