## Git如何使用

### 注册github用户名

1. 首先在<https://github.com/>注册用户名
2. 然后新建repository库

### 安装git

参考<http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/0013743256916071d599b3aed534aaab22a0db6c4e07fd0000>

### 用git命令初始化仓库

1. 打开git bash
2. 用dos命令定位到要存放新建仓库的目录（例如：cd d:）
3. touch README.md
4. git init
5. git add README.md
6. git commit -m "first commit"
7. git remote add origin https://github.com/luoxufeng/H5.git
8. git push -u origin master

## 其他Git 命令

1. git status
2. git log
3. git reflog
4. git reset –hard HEAD^(回退到前一个版本，前提是没有提交到远程库
5. git reset –hard 版本号（回退到某一个版本，通过git log或者git reflog查找版本号）
6. git checkout –文件名（丢弃工作区的修改，还没有提交到暂存区）
7. git reset HEAD 文件名（把暂存区的修改丢掉，重新回到工作区）

### 小结

又到了小结时间。

场景1：当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命令git checkout -- file。

场景2：当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修改，分两步，第一步用命令git reset HEAD file，就回到了场景1，第二步按场景1操作。

场景3：已经提交了不合适的修改到版本库时，想要撤销本次提交，参考[版本回退](http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/0013744142037508cf42e51debf49668810645e02887691000)一节，不过前提是没有推送到远程库。

## 远程仓库

1. 注册github账号
2. 创建SSH Key （先用dos命令切换到用户文件夹，例如：D:\Users\xfluo）

然后执行命令：$ ssh-keygen -t rsa -C [youremail@example.com](mailto:youremail@example.com) 执行此命令后在用户主目录中找到.ssh目录，里面有id\_rsa和id\_rsa.pub两个文件

1. 登陆GitHub ，打开”Account settings”,”ssh keys”页面，然后点击Add SSH Key,填上任意的title,在key文本框中粘贴id\_rsa.pub文件的内容

### 添加到远程库

1)git remote add origin “git 地址”

2)git push origin master

### 从远程库克隆

git clone “git地址”

## 分支管理

### 创建与合并分支

1,创建dev分支，并且切换到dev分支

Git checkout –b dev

2,查看当前分支

Git branch

3, 修改后的文件提交

Git add 文件名

Git commit –m “branch test”

4,切换到master分支

git checkout master

5,把dev分支的工作合并到master分支上(git merge命令用于合并指定分支到当前分支)

Git merge dev

6,删除分支

git branch –d dev

#### 小结

Git鼓励大量使用分支：

查看分支：git branch

创建分支：git branch <name>

切换分支：git checkout <name>

创建+切换分支：git checkout -b <name>

合并某分支到当前分支：git merge <name>

删除分支：git branch -d <name>

### 解决冲突

当Git无法自动合并分支时，就必须首先解决冲突。解决冲突后，再提交，合并完成。

用git log --graph命令可以看到分支合并图。(参考：<http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/001375840202368c74be33fbd884e71b570f2cc3c0d1dcf000>)

### 分支管理策略

通常，合并分支时，如果可能，Git会用Fast forward模式，但这种模式下，删除分支后，会丢掉分支信息。

如果要强制禁用Fast forward模式，Git就会在merge时生成一个新的commit，这样，从分支历史上就可以看出分支信息。

1. 创建dev分支
2. Add 和 commit文件
3. 切回master
4. 准备合并dev分支，请注意--no-ff参数，表示禁用Fast forward：

$ git merge --no-ff -m "merge with no-ff" dev

Git分支十分强大，在团队开发中应该充分应用。

合并分支时，加上--no-ff参数就可以用普通模式合并，合并后的历史有分支，能看出来曾经做过合并，而fast forward合并就看不出来曾经做过合并。

## Bug分支