# Git使用及规范

## Git安装配置及基本使用

1. 下载安装包，手动完成安装。
2. 打开Git Bash命令行工具，执行命令ssh-keygen -t rsa -C Email-Addresss生成一个密钥对。
3. 登录到GitLab，点击右上角你的用户头像，点击Edit Profile settings，点击SSH Keys，点击Add SSH Key，填写Title栏，复制用户目录下.ssh/id\_rsa.pub文件的内容到Key，点击Add Key。
4. 点击右上角的New project，填写完成后点击Create project新建一个仓库，点击Activity，点击SSH后复制SSH边上栏里的地址。
5. 打开Git Bash命令行工具，切换到一个合适的目录，使用命令git clone 刚才复制的URL克隆创建的仓库。
6. 进入目录cd 仓库名，执行命令

git config --global user.email your-email，   
git config --global user.name your-name，设置你的个人信息。

1. 执行命令：   
   echo "#Description" > README.md，添加一个文件   
   git status，查看当前状态，发现有未跟踪文件   
   git add .，当前目录所有文件添加到暂存区   
   git diff，比较当前工作区和暂存区有何不同   
   git status，查看当前状态，发现有文件未提交   
   git commit -m "注释"，把暂存区内容提交到本地仓库   
   git push -u origin master，把本地仓库的提交推送到远程仓库   
   git log，查看提交日志

## Git本地分支管理

1. 分支的创建、合并、删除   
   git branch，显示所有分支   
   git branch b1，从当前分支创建一个叫b1的分支   
   git checkout b1，切换到b1分支   
   git checkout -b b1，相当于以上两条命令的组合   
   git checkout master，切换到master主分支   
   git merge b1，把b1分支的代码合并到master上   
   git branch -d b1，删除b1分支，不能在被删除分支上执行

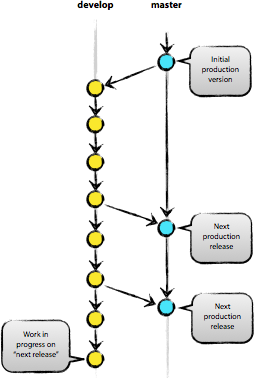
## Git Tag标签管理

1. 标签的创建、删除   
   git tag t1，从当前分支创建一个名为t1的标签   
   git tag -d t1，删除名为t1的标签

## 分支模型

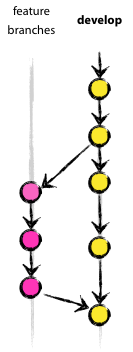
整体流程图:   


### 主要分支（保护分支）

* master 主分支，稳定代码，为生产环境做准备的
* develop 开发分支，为开发服务   
  分支关系类似下图：   
  

### 辅助分支

#### 特性分支

从develop分支创建，用于特性开发，完成后要合并回develop分支。   
操作过程：   
git checkout -b newfeature develop，从develop分支创建newfeature特性分支   
git checkout develop，开发完成后，需要合并回develop分支，先切换到develop分支   
git merge --no-ff newfeature，合并回develop分支，必须加--no-ff参数   
git branch -d newfeature，删除特性分支   
git push origin develop，把合并后的develop分支推送到远程仓库   
分支关系类似下图:   


#### 发布分支

从develop分支创建，用于预发布版本，允许小bug修复，完成后要合并回develop和master。   
操作过程：   
git checkou -b release-1.2 develop，创建一个发布分支   
git checkout master，切换到master分支，准备合并   
git merge --no-ff release-1.2，把release-1.2分支合并到master分支   
git tag 1.2，从master分支打一个标签   
git checkou develop，切换到develop分支，准备合并   
git merge --no-ff release-1.2，把release-1.2分支合并到develop分支   
git branch -d release-1.2，删除这个发布分支

#### 修复分支

从master分支创建，用于生产环境上的Bug修复，完成后要合并回develop和master。   
操作过程：   
git checkout -b hotfix-1.2.1 master，从master分支创建一个Bug修复分支   
git checkout master，切换到master分支，准备合并   
git merge --no-ff hotfix-1.2.1，合并到master分支   
git tag 1.2.1，为master分支创建一个标签   
git checkout develop，切换到develop分支，准备合并   
git merge --no-ff hotfix-1.2.1，合并到develop分支   
git branch -d hotfix-1.2.1，删除hotfix-1.2.1分支   
分支关系类似下图:   
