GitHub是为开发者提供 Git仓库的托管服务。这是一个让开发者与 朋友、同事、同学及陌生人共享代码的完美场所。

一些功能：

1. 在开发者之间引发化学反应的 Pull Request（Pull Request是指开发者在本地对源代码进行更改后，向 GitHub中 托管的Git仓库请求合并的功能）
2. 对特定用户进行评论（任务管理和 BUG报告可以 通过Issue进行交互）
3. GitHub Flavored Markdown（用户所有用文字输入的功能都可以用GitHub Flavored Markdown（GFM）语法进行描述）

主要功能：

1. Git 仓库
2. Organization（账户）
3. Issue（Issue功能，是将一个任务或问题分配给一个 Issue进行追踪和管理 的功能）
4. Wiki（通过Wiki功能，任何人都能随时对一篇文章进行更改并保存，因 此可以多人共同完成一篇文章）
5. Pull Request（开发者向GitHub的仓库推送更改或功能添加后，可以通过Pull Request 功能向别人的仓库提出申请，请求对方合并）

Git 属于分散型版本管理系统，是为版本管理而设计的软件。

版本管理系统分为集中型和分散型

集中型：只存在一个仓库. 将所有数据集中存放在服务器当中，有便于管理的优点。

分散型：拥有多个仓库，相对而言稍显复杂。由于本地的开发环境中就有仓库，所以开发者不必连接远程仓库就可以进行开发。

双方都各有优缺点，但现在使用分散型的占绝大多数。

在Windows中，下载msysGit之后，git命令就只能在 Git Bash中使用，在 Windows附属的命 令提示符中则无法运行。

基本操作：

1. git init——初始化仓库
2. git status——查看仓库的状态
3. git add——向暂存区中添加文件
4. git commit——保存仓库的历史记录
5. git log——查看提交日志 git reflog——查看当前仓库的操作日志
6. git diff——查看更改前后的差别

分支的操作：

1. git branch——显示分支一览表
2. git checkout -b——创建、切换分支
3. git merge——合并分支
4. git log --graph——以图表形式查看分支

更改提交的操作：

1. git reset——回溯历史版本
2. git commit --amend——修改提交信息
3. git rebase -i——压缩历史