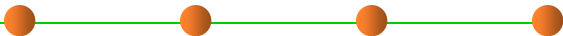




# Git实战



# 在本地机器上安装Git

## ■ Linux:

- \$ yum install git-core #在redhat等系统下用yum
- \$ apt-get install git-core #在debian, ubuntu等系统下用apt-get

## ■ Windows: <http://msysgit.github.io/> 下载安装包

## ■ 远程Git服务器: Github(<http://www.github.com>)

## ■ 说明:

- Linux和Windows下的Git安装操作二选一即可
- 在Github上申请账号
- 使用命令行方式完成实验, 避免图形界面下的操作。

## 实验场景(1)：仓库创建与提交

- R0：针对R1和R7，在进行每次git操作之前，随时查看工作区、暂存区、git仓库的状态，确认项目里的各文件当前处于什么状态；
- R1：本地初始化一个git仓库，随机创建n个文件（ $n > 10$ ）创建项目的全部源文件加入进去，纳入git管理；
- R2：提交；
- 手工修改3个文件；
- R3：查看上次提交之后都有哪些文件修改、具体修改内容是什么（查看修改后的文件和暂存区域中相应文件的差别）；
- R4：重新提交；
- 再次对3个文件进行修改；
- R5：重新提交
- R6：把最后一次提交撤销；
- R7：查询提交记录；

## 实验场景(2)：分支管理

- 在你的Github网站上，通过web界面建立一个project，将不少于10个文件（程序代码、文档等）加入进去，形成初始分支B1；在B1基础上建立两个并行的分支B2、B3，手工对B2和B3上的某些文件进行不同程度的修改并提交；
- R8：从Github上(URL)克隆一个已有的git仓库到本地；
- R9：获得该仓库的全部分支；
- R10：在B2分支基础上创建一个新分支C4；
- R11：在C4上，对4个文件进行修改并提交；
- R12：在B3分支上对同样的4个文件做不同修改并提交；
- R13：将C4和B3分支合并，若有冲突，手工消解；
- R14：查看目前哪些分支已经合并、哪些分支尚未合并；
- R15：将C4和B3合并后的分支删除，将尚未合并的分支合并到一个新分支上，分支名字为你的学号；



## 实验场景(3)：远程分支管理

- R16：将本地以你的学号命名的分支推送到Github上；
- R17：将R1到R7各步骤得到的结果推送到Github上。
- R18：在Github网站以web页面的方式查看你的两个仓库的当前状态。



結束