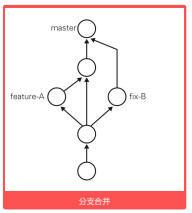
Git与Github使用

Git指令（默认下载git）

- **创造者：Linus Torvalds**
- **SVN与Git对比**
  - SVN — 集中型：仓库都统一存放在服务器中，便于管理；但若开发者无法连接服务器或服务器宕机，则无法获取仓库最新代码
  - Git — 可以通过Fork仓库，将代码获取到本地开发环境中，无需连接远程仓库就可以进行开发
- **部署**
  - 属名：$ git config --global user.name "your name"
  - 邮箱：$ git config --global user.email "电子邮箱"
  - 美化：$ git config --global color.ui auto — 提高命令输出可读性
  - 可以在~/.gitconfig文件下查看配置文件内容
- **代码仓库创建、修改与提交**
  - 创建代码仓库——初始化：$ git init ｜ 克隆代码仓库：$ git clone ssh-code/or/https
    - 获得的.git目录存储着管理当前目录内容所需的仓库数据
  - 添加本地代码仓库的改动：$ git add .
    - 或者添加 -p 参数，指定添加某个文件
    - 该步骤指定untracked files将其加入暂存区中，使其受git仓库管理
  - 查看代码仓库修改状态：$ git status
  - *使用命令：$ git commit -am —步执行*
  - 提交代码改动并添加改动的描述：$ git commit -m "your description about this commit"
    - 1. 若需要详细描述，可以不添加-m参数，直接执行git commit，之后会启动编辑器，第一行填写简单描述，第二行空着，第三行开始输入详细描述
    - 2. 若在编辑器启动后想终止提交，可以直接将描述信息留空，直接关闭编辑器
    - 3. 修改上一条提交信息：$ git commit --amend ,将打开编辑器，修改第一行内容
  - 查看文件提交日志：$ git log
    - 添加参数 --pretty=short : 只显示第一行的日志信息
    - 添加参数 -p [指定文件名] : 查看文件前后差别，可以选择指定需要查看前后改动差别的文件
    - 添加参数 --graph : 美化日志输出
  - 查看工作树、暂存区、最新修改间的差别：$ git diff
    - 添加参数 HEAD：查看本次修改和上一次修改之间的区别，以确认修改内容是否达到目的
    - 当未使用 git add 时，只显示工作树和最新提交修改间的差别
  - 添加远程仓库：$ git remote add origin repository-ssh
  - 将本地仓库上传：$ git push
    - 设置远程仓库后，使用 $ git push -u origin maser ，将本地仓库内容推送至远程仓库的master分支上。（当然也可以推送至其他分支中）
    - origin : 将远程仓库名称设置为origin（标识符）
    - u : 将origin仓库的master分支设为本地仓库当前分支的上游(upstream),将来使用git pull获取远程仓库内容时，本地仓库分支可以直接从origin的master分支中获取内容
  - 获取最新的远程仓库分支：$ git pull
    - 形式：$ git pull origin feature-A ,将本地的feature-A分支更新到远程仓库中的最新状态
- **代码仓库（特性）分支**
  - 理解 — 不同分支代表实现的不同功能，最后我们将不同分支合并，就相当于将不同功能合体。创建分支，每个分支间操作不会互相影响，以达到并行作业效果。
  - 操作命令
    - 显示当前所有分支：$ git branch
    - 创建、切换分支：$ git checkout -b new-branch
      - 该命令等价于：$ git branch new-branch + $ git checkout new-branch
      - 切换回上一个分支：$ git checkout -
    - 获取远程分支：$ git checkout -b branch-name origin/branch-name
      - 为方便理解，本地创建的分支名和从远程仓库获取的origin仓库中的分支名保持一致。再修改后，可以使用 $ git commit -am "description"提交，然后再$ git push 推送本地分支
    - 合并分支：$ git merge --no-ff branch-name
      - 若合并发生冲突，查看发生冲突的文件，并将其修改。解决冲突后，再执行git add 和 git commit提交
      - --no-ff参数：在历史记录中明确记录下本次分支合并
    - 压缩修改历史：$ git rebase -i
      - （例如在修改拼写错误等，将修改包含到前一个提交中）在修改错误后，再次进行$ git commit -am提交，提交描述信息可以写 " Fix typo" ,即拼写错误。再使用$ git rebase -i HEAD~2, 将修正后的与之前一次提交合并，最后在编辑器中将Fix typo 左侧的pick 改为 fixup，这样Fix typo的提交就会在历史中抹去。
    - 回溯分支历史版本：$ git reset
      - 格式：$ git reset --hard 时间哈希值。 使用$ git reflog查看仓库操作日志，找到回溯历史前的哈希值。



分支合并

Github

- **缔造者**
  - Chris Wanstrath
  - PJ Hyett
  - Tom Preston-Werner
- **编写语言** — Ruby on Rails
- **创建一个新仓库**
  - 1. 在Repository界面点击New
  - 2. 填写仓库名称和描述（选），以及是否添加.gitignore 和 license。
    - .gitignore文件可以帮助记录不需要在Git仓库中进行版本控制的文件
    - license是我们仓库内容的许可协议,具体可见https://opensource.org/licenses
  - 3. 选择是否添加一个 README.md文件，若选项打勾，Github自动初始化仓库，并且用户可以直接克隆仓库
    - 如何想将本地git仓库上传，则建议不要勾选
- **Pull Request (PR)**
  - CIA (Contributor License Agreement) 当向一些大组织或机构公司提交PR时，可能需要签署该协议
  - 解释：PR指开发者克隆下代码后，对源代码进行修改后，请求与原项目合并。
    - 通常有修改BUG、添加新功能等
- **Issue**
  - 用法
    - 通过 @ 用户名 指定并通知用户
      - 或者 @ 组织名/团队 通知组织或开发团队
    - 通过 # 编号 连接代码仓库中对应编号的Issue
      - 或者 用户名/仓库 # 编号 来连接指定用户的仓库对应的issue
    - 关闭issue
      - 使用格式：" fix/fixes/fixed/close/closes/closed/resolve/resolves/resolved ＃编号" 描述提交信息
  - 用途
    - 任务管理
    - BUG报告
  - 子主题 3
  - Issue与PR的编号相互通用，给Issue添加源代码，即变为PR
- **连接Git** — 通过SSH认证
  - 1. 在git下生成SSH KEY : $ ssh-keygen -t rsa -C "your_email@example.com"
    - 第一次设置可能需要设置认证密码
    - 此步骤会生成私有密钥id_rsa和公开密钥id_rsa.pub
  - 2. 向Github中添加公开密钥（setting-SSH and GPG keys-New SSH key-随意添加一个标题并输入id_rsa.pub内容-Add SSH key
  - 3. 验证ssh是否添加成功：$ ssh -T git@github.com

```
$ ssh -T git@github.com
The authenticity of host 'github.com (207.97.227.239)' can't be established.
RSA key fingerprint is fingerprint值 .
Are you sure you want to continue connecting (yes/no)? 输入yes
```

验证图例

- **Github技巧**
  - 查看仓库不同分支间的差别 — 形式：https://github.com/electron/electron/compare/1-3-x...1-4-x (两个分支间用..隔开)
  - 查看分支在最近几天的改变
    - 形式：https://github.com/electron/electron/compare/master@{7.day.ago}...master
    - 时间单位：day,week,month,year
  - 处理PR信息
    - 获取diff格式文件：https://github.com/用户名/仓库名/pull/xx.diff
    - 获取patch格式文件：https://github.com/用户名/仓库名/pull/xx.patch
  - 应用评论 — 选中需要引用的内容，按R键