

# 一、 版本控制

---

## 集中式(Svn)

### 优点:

- 代码存放在单一的服务器上面，便于管理。

### 缺点:

- 服务器宕机：本地所写的代码得不到保障( 不敢轻易的修改代码(如果喜欢Ctrl-C、和Ctrl-V 那没有影响) )。
- 服务器炸了：历史记录存在丢失的危险。
- 服务器存放的是版本与版本直接的差异不是完整的版本(例如第一次存放是A1，第二次是A2-A1……第100次存放的是A100-A99),所以当需要回滚到特定版本项目的时候需要一步一步的进行打补丁回退,这样会浪费很多时间。

## 分布式(Git)

### 优点:

- 服务器存放的不是版本之间的差异而是完整的版本(例如第一次存放的是A1，第二次存放的是A2， A1， ……第100次存放的是A100， A99……A1)此时想要回到特定的版本会很快回滚回去，不需要打补丁。
- 如果没有网络也可以进行开发因为版本控制在本地。

### 缺点:

- 需要的磁盘空间要不Svn要多一些，但是不会多很多。

# 二、 Git安装

---

打开Git下载网址：根据自己点电脑选择适合的版本下载Git下载地址

。



安装成功之后在桌面右击鼠标如果能出现**Git GUI Here** 和 **Git Bash Here** 则证明安装成功，恭喜你完成了第一步距离成为大神还有99步。



Git GUI Here 是图形界面， Git Bash Here是命令行界面既然要想向着大神的目标出发那肯定要选择命令界面咯！So~ 我们首先应该打开命令行界面并且看一下自己的Git的版本是多少。点击**Git Bash Here** 会弹出命令行界面，我们在命令行界面输入**git version**。

```
MINGW64:/c/Users/dell/Desktop
dell@DESKTOP-V1IAAFN MINGW64 ~/Desktop
$ git version
git version 2.30.0.windows.2
dell@DESKTOP-V1IAAFN MINGW64 ~/Desktop
$
```

至此终于走完的第一步，万事开头难既然第一步走完了那接下来肯定很顺利的。

## 三、 Git初始化

Git可以让一个项目的程序有很多人来同时工作，如果有人修改并且提交了代码，讲道理Git需要给出做出修改的时间以及修改人的姓名等一系列的信息，如果不给出修改人的相关信息，那公司的代码出了问题找不到是谁修改那可不就完蛋了~。所以我们下面要对Git进行一些初始化，补充完整的信息。

对于一台电脑而言会有一个操作系统，一个操作系统可以让不同的用户进行操作，一个用户有会有多个项目。所以对于Git的信息配置也可以配置三个不同的优先级。

- 系统级别(优先级最高):  
`git config --system user.name "Jc"`  
`git config --system user.email`  
`"Jc@example.com"`
- 用户级别(优先级其次):  
`git config --global user.name "Jc"`  
`git config --global user.email`  
`"Jc@example.com"`
- 项目级别(优先级最低):  
`git config user.name "Jc"`  
`git config user.email "Jc@example.com"`
- 通过`git config --list`可以看见自己是否配置成功。

```
init.defaultbranch=master
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=Jc
user.email=Jc@example.com
alias.dnf=log --oneline --decorate --graph --all

dell@DESKTOP-V1IAAFN MINGW64 ~/Desktop
```

## 四、 结尾

---

至此呢Git的基本部署就完成了，如果感觉有用可以点个赞的哦！我会持续更新，如果有错误还请指出来,感谢观众老爷的赏脸。

若想获得上述内容的PDF版本移步到GitHub下载。

**地址:** [Git 学习笔记专区](#)。

----缱绻