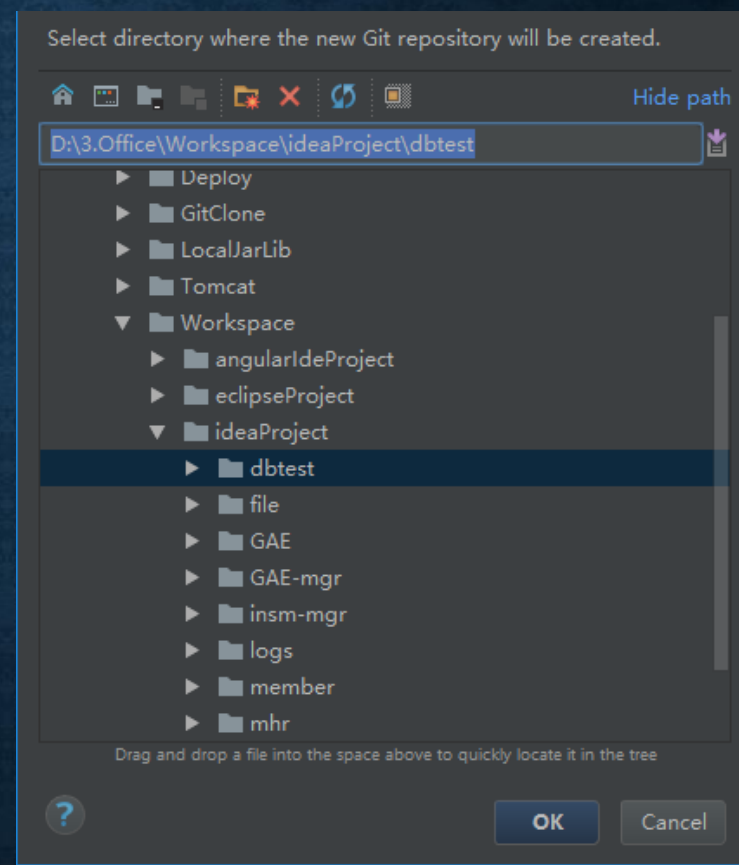
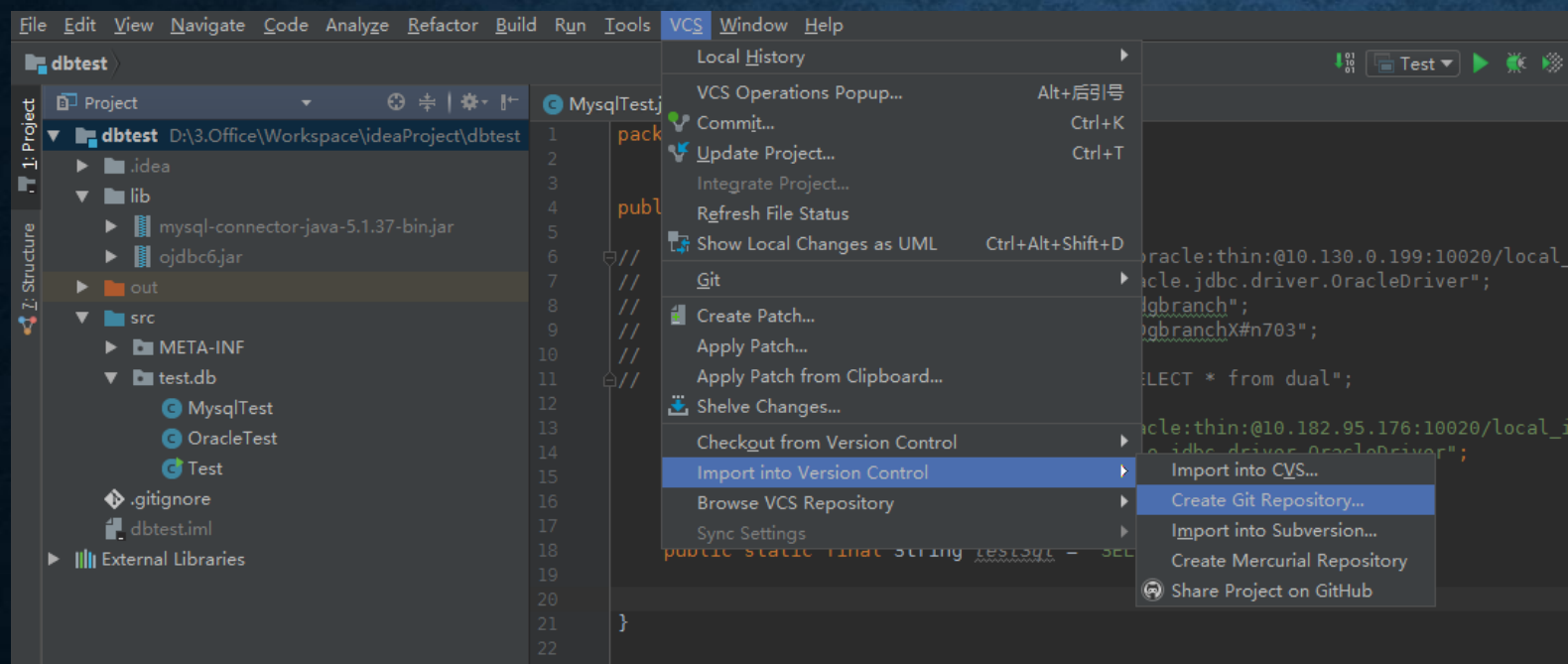


GIT

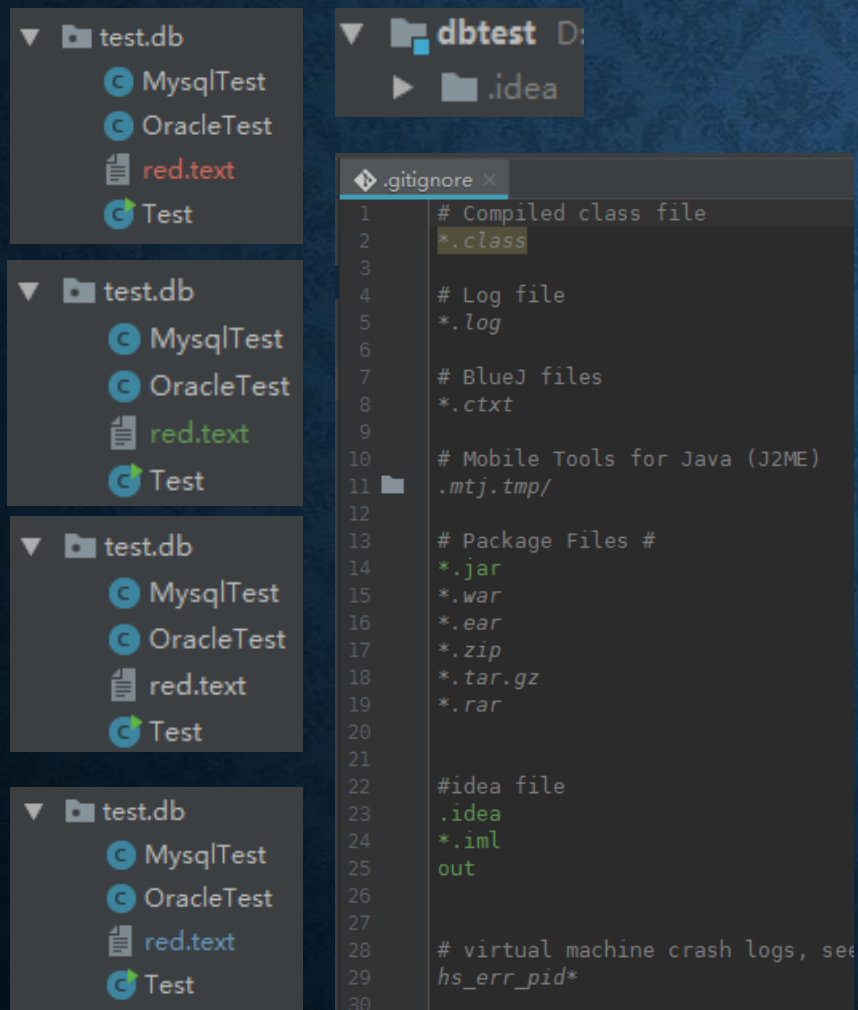
[参考教程](#)

GIT-IDEA创建版本库



选择项目目录创建git版本库，即完成创建

GIT-IDEA .gitignore 配置文件



忽略的文件，即不提交到版本库的文件
在.gitignore文件中配置，格式如图
忽略的文件通常为IDE文件、日志、class等

idea中

未添加到暂存区文件显示红色

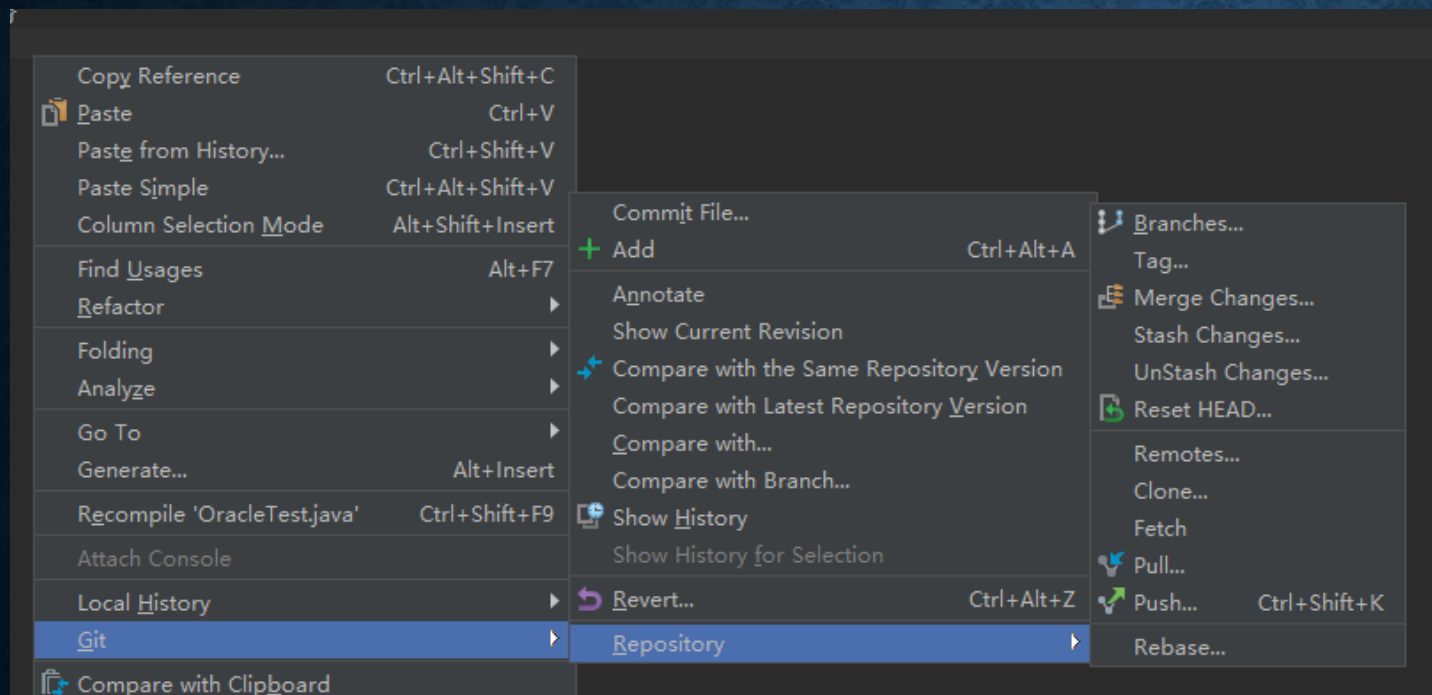
已添加到暂存区文件显示绿色

已提交到版本库文件显示白色

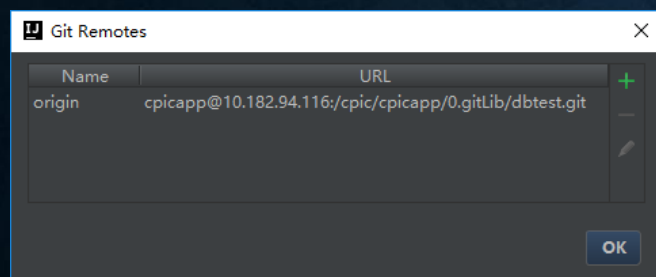
有修改但未提交到版本库的文件显示蓝色

忽略的文件为灰色

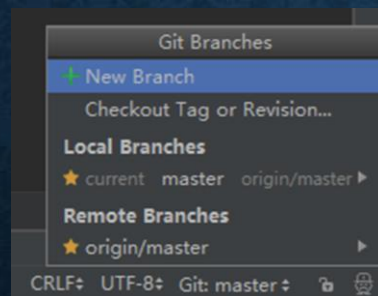
GIT-IDEA GIT功能



IDEA中git功能顾名思义，如
Commit File 提交文件
+Add 添加文件
Revert恢复版本
Reset HEAD 回退版本
Merge Changes 合并分支
Push 推送
Pull 获取
Compare with ...版本对比
....等等



添加远程版本库



分支管理

如果这些看不懂则说明您需要继续往下看

GIT-IDEA版本演绎图

The screenshot displays the Git version control interface within an IDE. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The breadcrumb path is GAE > resources > properties > db.properties. The Version Control tab is active, showing a commit log with the following entries:

Commit Message	Author	Date
修复被保险人清单不能导出bug 对导出Excel清单国际化	BrandonLeaf	2017/7/18 16:38
对修改部分进行国际化	BrandonLeaf	2017/7/18 11:56
修改被保人清单职业类别显示	BrandonLeaf	2017/7/14 17:40
Merge branch 'dev'	BrandonLeaf	2017/7/14 16:34
修复人员清单grid表头文字不对应bug	BrandonLeaf	2017/7/14 16:34
Merge branch 'dev'	BrandonLeaf	2017/7/13 17:33
字体修改为微软雅黑	BrandonLeaf	2017/7/13 17:32
Merge branch 'dev'	BrandonLeaf	2017/7/13 17:22
按需求修改主页面布局, 修改申请记录,批单查询,字段显示	BrandonLeaf	2017/7/13 17:22
修改分页栏样式	BrandonLeaf	2017/7/13 12:10
按需球, 整改主界面, foot部分修改完成 导航态修改完成 加入分页模块	BrandonLeaf	2017/7/13 11:56
Merge branch 'dev'	BrandonLeaf	2017/7/12 18:12
按需球, 整改主界面, 登录界面	BrandonLeaf	2017/7/12 18:01
被保险人国际化完成 登录页面按需修改logo大小	BrandonLeaf	2017/7/12 17:06
批改记录国际化完成	BrandonLeaf	2017/7/12 16:11
申请记录国际化完成	BrandonLeaf	2017/7/12 15:34

The right sidebar shows the project structure:

- GAE 11 files
 - D:\3.Office\Workspace\ideaProject\G
 - jsp 5 files
 - function 4 files
 - detail.jsp
 - endAppPage.jsp
 - endorsement.jsp
 - policyPerson.jsp
 - index 1 file
 - home.jsp
 - static\component 6 files
 - bootstrap\css 1 file

The bottom status bar indicates the current branch is master, and the file encoding is UTF-8.

GIT-工作区、暂存区和分支

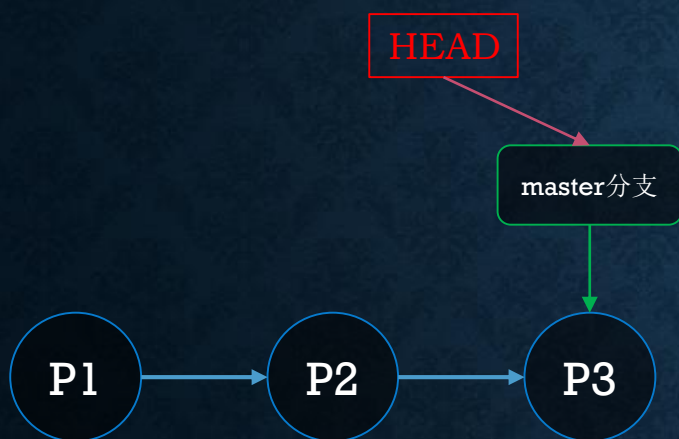


GIT-分支和HEAD指针

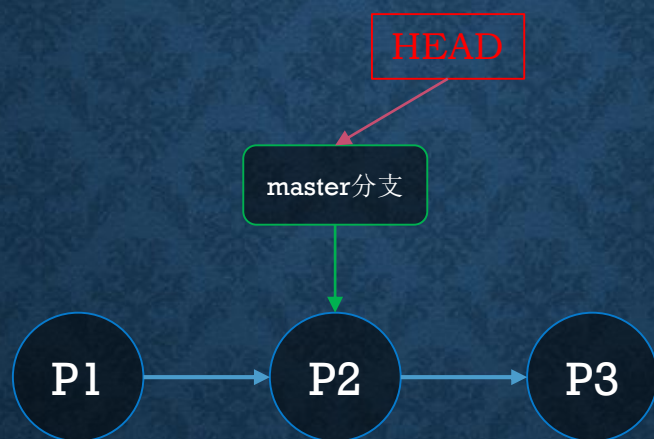


GIT-分支和HEAD指针

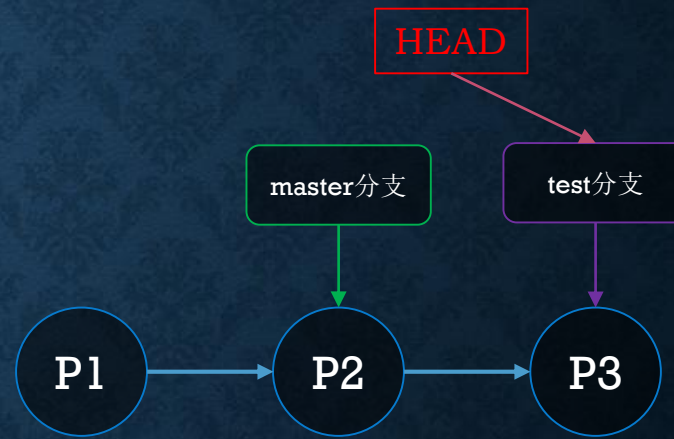
HEAD 指向git版本库的分支，表示此分支是当前使用的分支



HEAD表示当前git所指向的分支
此时HEAD指向master分支
而master分支指向P3版本



#回退到上一个版本
\$ git reset --hard HEAD^
HEAD指向master分支没变
而master分支所指向的版本回退到上一个版本



#创建并切换到test分支
\$ git checkout -d test
此时HEAD指向test分支，即当前使用test分支
此时test指向版本是P2版本
#test分支指向P3版本（用git log 查看commit_id）
\$ git reset --hard <P3的commit_id>
此时test分支指向的版本是P3版本(如上图)
而master分支指向的版本仍然是P2版本

GIT-分支策略

其他分支可灵活处理，比如尝试新功能或优化代码可建**Feature**分支，重构项目可建**rebuild**分支，适用于此类“失败则丢弃成功则应用”的修改

GIT命令-初始化

#安装完后需要配置您的用户名和邮箱，用于记录谁干的操作

```
$ git config --global user.name "Your Name"
```

```
$ git config --global user.email "email@example.com"
```

#创建版本库(cd到项目目录下)

```
$ git init
```

初始化后项目目录下会多一个.git的文件夹，默认是隐藏的，这就是版本库不要动它就好了

GIT命令-文件操作

#查看当前git状态

```
$ git status
```

#查看文件修改状态\$ git diff <file>

```
$ git diff readme.text
```

#撤回工作区文件的修改\$ git checkout -- <file>

```
$ git checkout -- readme.text
```

#添加工作区文件到暂存区\$ git add <file>

#commit后添加此文件到版本库

```
$ git add readme.text
```

#添加工作区文件到暂存区\$ git rm <file>

#commit后从版本库中移除此文件

```
$ git rm readme.text
```

#移除暂存区中的文件\$ git reset HEAD <file>

```
$ git reset HEAD readme.text
```

#提交暂存区中的文件到版本库

```
$ git commit -m 'commit node'
```

#查看工作区和版本库里面最新版本的差别

```
$ git diff HEAD -- readme.txt
```

GIT命令-版本回退

#查看git commit历史记录

```
$ git log
```

#查看git commit历史记录（简短）

```
$ git log --pretty=oneline
```

```
3628164fb26d48395383f8f31179f24e0882e1e0 append GPL  
ea34578d5496d7dd233c827ed32a8cd576c5ee85 add distributed  
cb926e7ea50ad11b8f9e909c05226233bf755030 wrote a readme file  
前面一串随机码是commit_id，即版本号
```

#回退到提交的版本（3628164），输入版本号前几位就可以了，Git会自动去找

```
$ git reset --hard 3628164
```

#回退到上个版本

```
$ git reset --hard HEAD^
```

#回退到上上个版本

```
$ git reset --hard HEAD^^
```

#回退到上100个版本

```
$ git reset --hard HEAD~100
```


GIT命令-分支操作

#创建分支

```
$ git branch bug
```

#切换到分支

```
$ git checkout bug
```

#创建并切换到分支

```
$ git checkout -b dev
```

#合并分支到当前分支

```
$ git merge bug
```

#删除分支

```
$ git branch -d bug
```

git merge合并分支默认是**Fast-forward** “快进模式”

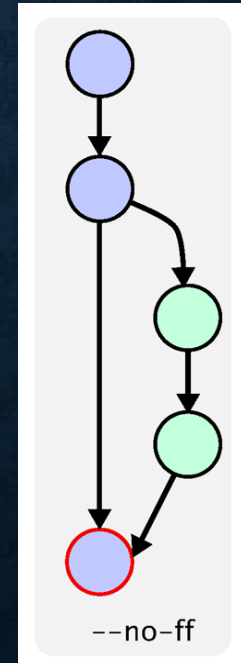
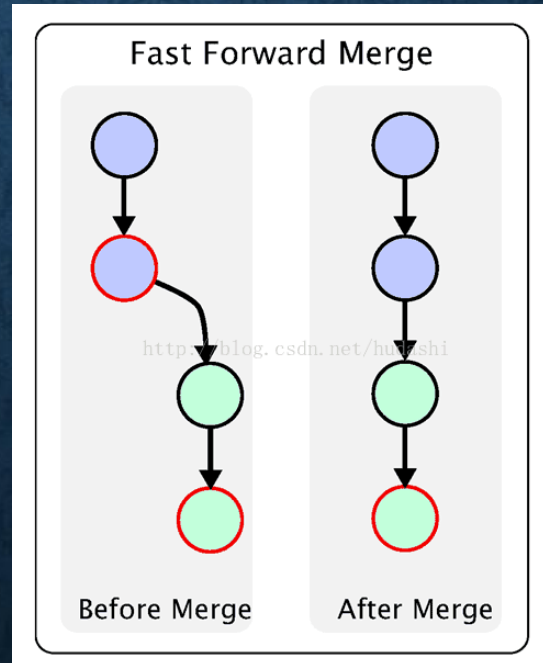
快进模式会把当前分支直接指向合并分支的版本，

git log 查看时看不出曾经创建过分支，因此加上

--no-ff 参数后合并时会在当前分支创建一个节点，

使用**git log**查看时就能看到曾经创建过分支，因为要提交所以要加参数 **-m** 记录修改内容，对比如下

```
$ git merge --no-ff -m 'merge message' bug
```



GIT命令-分支合并冲突解除

#合并分支时可能出现冲突的情况

\$ git merge bug

Automatic merge failed; fix conflicts and then commit the result.

提示合并失败

存在冲突的文件内容会显示如下:

Git is a distributed version control system.

Git is free software distributed under the GPL.

Git has a mutable index called stage.

Git tracks changes of files.

<<<<<< HEAD 当前分支

Creating a new branch is quick & simple.

===== 冲突内容分隔符

Creating a new branch is quick AND simple.

>>>>>> bug bug分支

Git用<<<<<<, =====, >>>>>>标记出不同分支的内容, 根据提示手动解除后再合并, 即冲突解除

GIT命令-远程版本库关联

#创建裸库-命名规则: <项目名>.git

```
$ git init --bare dbtest.git
```

#添加远程服务器

```
$ git remote add origin git@github.com:/BrandonLeaf/learn.git
```

#查看添加的远程库

```
$ git remote
```

#删除本地的远程库

```
$ git remote rm origin
```

#推送分支到远程库 (-u 设定默认关联服务器)

```
$ git push -u origin master
```

如果当前分支与多个主机存在追踪关系, 则可以使用 **-u** 参数指定一个默认主机, 这样后面就可以不加任何参数使用 **git push**
git push默认只推送当前分支

#抓取远程版本分支, 添加默认主机后可简化为以下命令, 默认只抓取当前分支

```
$ git pull
```

抓取远程版本时会遇到与本地版本冲突的问题, 与分支合并时一样的方式解除冲突

#克隆远程版本库

```
$ git clone git@github.com:/BrandonLeaf/learn.git
```

克隆会在当前目录下项目名称创建文件夹, 文件夹里是项目文件

GIT-WIN-GIT终端版本演绎图

```

* 2203555 - 修复登录验证码显示问题 添加 " 关于我们 " 文本内容 (6 weeks ago) <BrandonLeaf>
* 37ddf9b - 修复被保险人清单不能导出bug 对导出Excel清单国际化 (6 weeks ago) <BrandonLeaf>
* 5b33a1b - 对修改部分进行国际化 (6 weeks ago) <BrandonLeaf>
* 36c71c2 - 修改被保险人清单职业类别显示 (6 weeks ago) <BrandonLeaf>
* 036e092 - Merge branch 'dev' (6 weeks ago) <BrandonLeaf>
| \
| * 3dad92d - 修复人员清单 g r i d 表头文字不对应 b u g (6 weeks ago) <BrandonLeaf>
* | 829b585 - Merge branch 'dev' (6 weeks ago) <BrandonLeaf>
| \
| \
| * 90996de - 字体修改为微软雅黑 (6 weeks ago) <BrandonLeaf>
* | ffe84d3 - (tag: V2.1.1) Merge branch 'dev' (6 weeks ago) <BrandonLeaf>
| \
| \
| * 3d0c1aa - 按需求修改主页面布局, 修改申请记录,批单查询,字段显示 (6 weeks ago) <BrandonLeaf>
| * 101665b - 修改分页栏样式 (6 weeks ago) <BrandonLeaf>
| * de49713 - 按需求, 整改主界面, f o o t 部分修改完成 导航态修改完成 加入分页模块 (6 weeks ago) <BrandonLeaf>
* | 983c912 - (tag: V2.1) Merge branch 'dev' (7 weeks ago) <BrandonLeaf>
| \
| \
| * e32910b - 按需求, 整改主界面, 登录界面 (7 weeks ago) <BrandonLeaf>
| * 1ed2d5a - 被保险人国际化完成 登录页面按需求修改 l o g o 大小 (7 weeks ago) <BrandonLeaf>
| * 1437890 - 批改记录国际化完成 (7 weeks ago) <BrandonLeaf>
| * a4fad9c - 申请记录国际化完成 (7 weeks ago) <BrandonLeaf>
| * 23a1ab6 - 完成承保信息国际化 登录界面增加语言切换按钮 (7 weeks ago) <BrandonLeaf>
* | 229250f - (tag: V2.0.3) Merge branch 'dev' (7 weeks ago) <BrandonLeaf>
| \
| \
| * 7fec91f - 完成设置功能, 新增账户信息、修改密码功能 完成语言切换功能, 实现简繁转换 (目前主界面、登录界面、换人申请实现, 其余部陆续实现) (7 weeks ago) <BrandonLeaf>
* | acf5291 - Merge branch 'dev' (7 weeks ago) <BrandonLeaf>
| \
| \
| * 4922104 - 修复登录页面bug (7 weeks ago) <BrandonLeaf>
* | 5c922df - Merge branch 'dev' (7 weeks ago) <BrandonLeaf>
| \
| \
|

```

```
#有大神把git log 使用别名，配置成如上图，显示比较不错，执行下面命令后，使用 git lg即可
$ git config --global alias.lg "log --color --graph --pretty=format:'%Cred%h%Creset -
%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit"
```