

Git教程

代码管理系统对比

1. 集中式与分布式版本控制系统

Git 是分布式版本控制系统，SVN 是集中式版本控制系统

2. 数据存储方式

Git将内容按元数据 方式存储，而SVN按文件

3. 全局版本号

Git 不具备全局版本号，这是跟SVN相比Git缺少的最大的一个特征

4. 代码内容的完整性

5. 分支管理不同

优势

1. 频繁提交，互不干扰

2. 开发过程被打断

3. 分支切换

4. 仓库操作

重要概念

1. 本地仓库

2. 远程仓库

服务器端，多人共享的代码仓库。

3. 工作区

4. 版本库

工作区目录下的 .git 目录就是版本库，里面存储了中心版本库的所有东西，分支，版本记

录还有我们将要介绍的暂存区

5. 暂存区

.git 目录下的 index 文件，它的存在极大的方便了我们修改的撤回或者仅提交我们想提交的代码。

Git 提交代码分为两步，第一步是使用 git add 命令提交到暂存区，第二

步 git commit 命令才会将其提交到仓库

配置

- /etc/gitconfig 文件：系统中对所有用户都普遍适用的配置。若使用 git config 时用 --system 选项，读写的就是这个文件
- ~/.gitconfig 文件：用户目录下的配置文件只适用于该用户。若使用 git config 时用 --global 选项，读写的就是这个文件。
- .git/config：当前项目的 Git 目录中的配置文件（也就是工作目录中的 .git/config 文件）：这里的配置仅仅针对当前项目有效
- 全局配置

```
git config --global user.name "yourname"
git config --global user.email "youremail"
```

1. 局部配置

```
git config user.name "yourname"
git config user.email "youremail"
```

1. 重置

```
git config --global --unset user.name "yourname"
git config --global --unset user.email "youremail"
```

本地仓库

- 创建版本库

```
git init
```

- 提交代码

```
# 将修改增加到暂存区
git add
# 提交代码
git commit
```

- 查看状态及修改内容

```
# 查看状态
git status
# 查看修改的内容
git diff <file>
```

- 查看日志

```
# 查看提交历史
git log
# 查看提交命令历史
git reflog
```

- 撤销修改及版本回退

- 仅修改了工作区，直接丢弃工作区的修改

```
bash git checkout -- <file>
```

- 修改已经提交到暂存区

```
bash git reset HEAD <file> git checkout -- <file>
```

- 修改已经commit

```
bash git reset --hard <版本>
```

- 删除文件

```
git rm <file>
```

- 分支操作

- 创建分支

```
bash git branch <branch name> git checkout <branch name>
```

- 查看分支列表

```
bash git branch
```

- 删除分支

```
``bash
```

删除已合并分支

git branch -d

强行删除

git branch -D

- 查看分支合并日志 bash

git log --graph --pretty=oneline --abbrev-commit

- 分支对比 bash

git diff

- 分支冲突 冲突的原因：不同的分支修改同一位置的代码 删除多余内容，保留所需内容，提交即可 - 分支合并 bash

按照 commit 时间戳合并

git merge

将当前分支移动到branch后面

git rebase

```

## 远程操作

- 关联仓库

```
git remote add origin <仓库地址>
```

- 查看仓库信息

```
查看信息
git remote
```

```
查看详细信息
git remote -v
```

- 推送分支

```
首次推送
git push -u <repository> <branch name>
再次推送
git push <repository> <branch name>
```

- 比较差异

```
查看所有修改
git diff <repository>/<branch name>...HEAD
查看某一个文件的修改
git diff <repository>/<branch name>...HEAD <file>
```

- 拉取代码

- 克隆仓库

```
bash git clone <仓库地址>
```

- 拉取代码

```
bash git pull <repository> <branch name>
```

- 关联分支

```
git branch -set-upstream-to=<repository>/<branch name>
```

## 标签管理

- 默认标签,打在HEAD

```
git tag <tag name>
```

- 指定commit

```
git tag <tag name> <commit id>
```

- 创建说明标签

```
git tag -a <tag name> "description" <commit id>
```

- 查看列表

```
git tag
```

- 查看标签信息

```
git show <tag name>
```

- Push 标签

```
git push -u origin <tag name>
```

- 本地删除标签

```
git tag -d <tag name>
```

- 远程删除标签

```
git tag -d <tag name>
git push origin :refs/tags/<tag name>
```