

尚硅谷-Git

版本控制工具

集中式版本控制工具

CVS、SVN、VSS.....

- 通过单一的集中管理服务器获取最新的版本
- 一定程度上可以看到别人的操作
- 可以轻松掌控每个开发者的权限

服务器一旦宕机，则所有人都无法更新、协同

分布式版本控制工具

Git、Mercurial、Bazaar.....

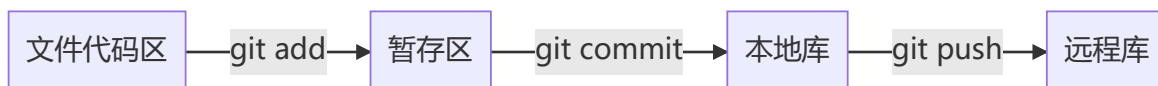
- 把代码克隆到本地，即使发生远程发生故障也可以继续操作
- 在断网情况下也可以进行开发（版本控制是在本地进行的）
- 每个客户端保存的都是完整的项目（包含历史记录，更加安全）

安装Git（2.35.1）

[简书-Git安装教程（Windows安装超详细教程）](#)

初次运行git的配置

git将文件的版本控制分为四个板块



1.配置用户信息（姓名、邮箱）

```
1 | git config --global user.name "YourName"
2 | git config --global user.email "YourEmail@example.com"
```

global是全局，只需配置一次，后续的操作都会读取全局配置

如果在某个目录中使用没有 `--global` 选项的命令来配置，可以达到针对特定项目使用不同的用户信息

2.查看配置文件

```
1 | git config --list
```

可以使用 `git config user.name` 来获取用户名，其他同理

3. 初始化并查看本地库状态

```
1 | git init # 初始化，在指定的目录下初始化即可在所在位置创建本地库
2 | git status # 可以查看当前本地库使用的分支和当前是否有内容需要提交
```

```
Administrator@DESKTOP-CJIHGIK MINGW64 /f/git-demo (master)
$ git status
On branch master
nothing to commit, working tree clean

Administrator@DESKTOP-CJIHGIK MINGW64 /f/git-demo (master)
$ |
```

vim相关操作

vim操作	描述
yy	复制一行到缓冲区
yn	复制n行到缓冲区
p	粘贴缓冲区的内容
np	粘贴缓冲区的内容n次

linux命令

命令	含义
<code>tail -n 3 hello.txt</code>	查看hello.txt的末尾3行

3. 添加到暂存区

- 1 `git add fileName` #添加到暂存区
- 2 `git rm --cached fileName` #从暂存区删除

```
Administrator@DESKTOP-CJIHGIK MINGW64 /f/git-demo (master)
$ git add hello.txt 警告:hello.txt中的LF将被CRLF替换
warning: LF will be replaced by CRLF in hello.txt.
The file will have its original line endings in your working directory
```

4. 提交本地库

- 1 `git commit -m "日志信息" fileName` # 将文件从暂存区提交到本地库
- 2 `git reflog` #显示所有的版本信息 `git log` #显示详细的版本信息

```
Administrator@DESKTOP-CJIHGIK MINGW64 /f/git-demo (master)
$ git reflog
3ecf554 (HEAD -> master) HEAD@{0}: reset: moving to HEAD
3ecf554 (HEAD -> master) HEAD@{1}: commit (initial): 尚硅谷git笔记1, luigit
↑ 头七位
Administrator@DESKTOP-CJIHGIK MINGW64 /f/git-demo (master)
$ git log 详细版本号
commit 3ecf5540ec714482d8ac4fd4e9583ed2353178e (HEAD -> master)
Author: luigit <luigit@aliyun.com> 提交者
Date: Tue Mar 15 08:07:07 2022 +0800 提交日期
    提交的文件, 提交者
    尚硅谷git笔记1, luigit
```

5. 修改文件

使用vim编辑文件即可

```
Administrator@DESKTOP-CJIHGIK MINGW64 /f/git-demo (master)
$ vim hello.txt

Administrator@DESKTOP-CJIHGIK MINGW64 /f/git-demo (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

- 1 | `git add fileName`
- 2 | `git commit -m "Second commit" fileName`

```
Administrator@DESKTOP-CJIHGIK MINGW64 /f/git-demo (master)
$ git commit -m "second hello.txt" hello.txt
warning: LF will be replaced by CRLF in hello.txt.
The file will have its original line endings in your working directory
[master 7c8772f] second hello.txt
1 file changed, 1 insertion(+), 1 deletion(-)
```

基于行进行维护的，
删除原有保存新增。

查看版本信息

```
Administrator@DESKTOP-CJIHGIK MINGW64 /f/git-demo (master)
$ git reflog
7c8772f (HEAD -> master) HEAD@{0}: commit: second hello.txt
d3747a0 HEAD@{1}: commit: hello.txt first
3ecf554 HEAD@{2}: reset: moving to HEAD
3ecf554 HEAD@{3}: commit (initial): 尚硅谷git笔记1, luigit
```

历史版本

查看历史版本

- 1 | `git reflog # 查看版本简略信息`
- 2 | `git log # 查看版本详细信息`

```
Administrator@DESKTOP-CJIHGIK MINGW64 /f/git-demo (master)
$ git log
commit 7c8772f4d8614558cd734b4947359d0121a30cfa (HEAD -> master)
Author: luigit <luigit@aliyun.com>
Date: Tue Mar 15 13:15:56 2022 +0800

    second hello.txt

commit d3747a0cb3cd8a1f3362bf781c333ce7474406c4
Author: luigit <luigit@aliyun.com>
Date: Tue Mar 15 13:13:25 2022 +0800

    hello.txt first

commit 3ecf5540ec714482d8ac4fd4e9583ed2353178e
Author: luigit <luigit@aliyun.com>
Date: Tue Mar 15 08:07:07 2022 +0800

    尚硅谷git笔记1, luigit

Administrator@DESKTOP-CJIHGIK MINGW64 /f/git-demo (master)
$ git reflog
7c8772f (HEAD -> master) HEAD@{0}: commit: second hello.txt
d3747a0 HEAD@{1}: commit: hello.txt first
3ecf554 HEAD@{2}: reset: moving to HEAD
3ecf554 HEAD@{3}: commit (initial): 尚硅谷git笔记1, luigit
```

