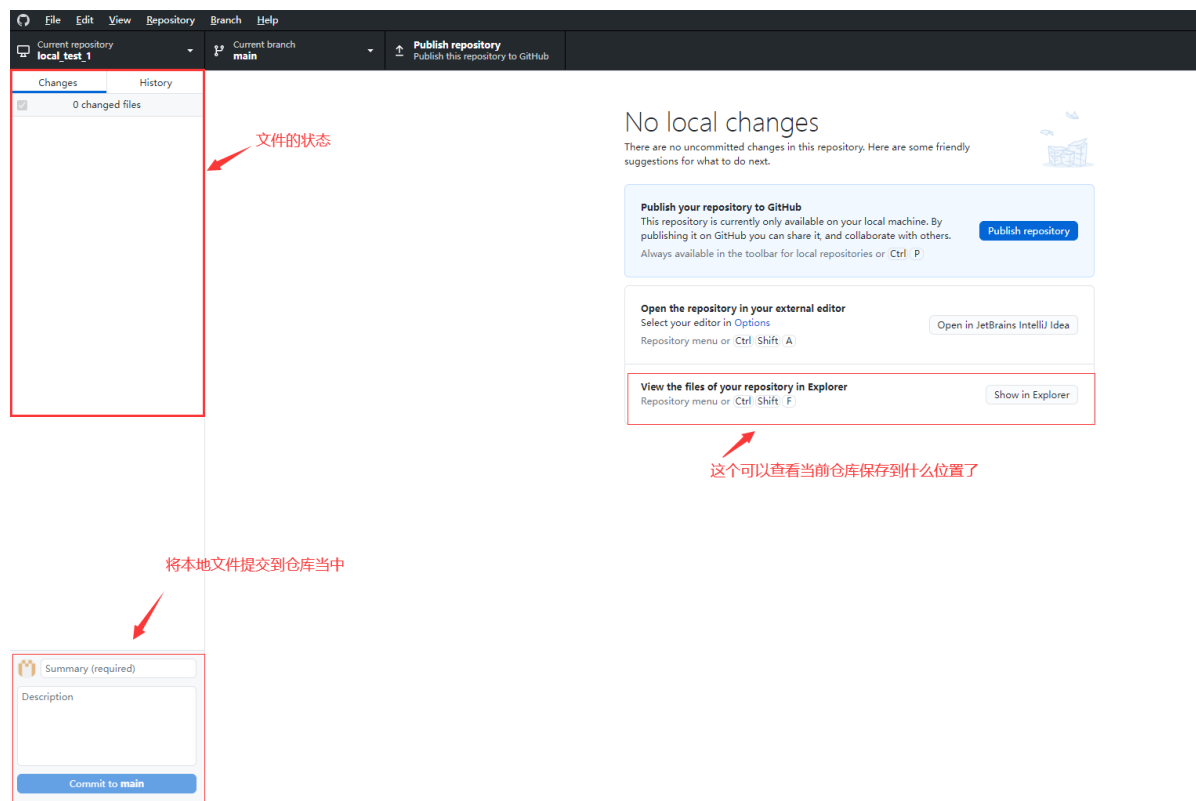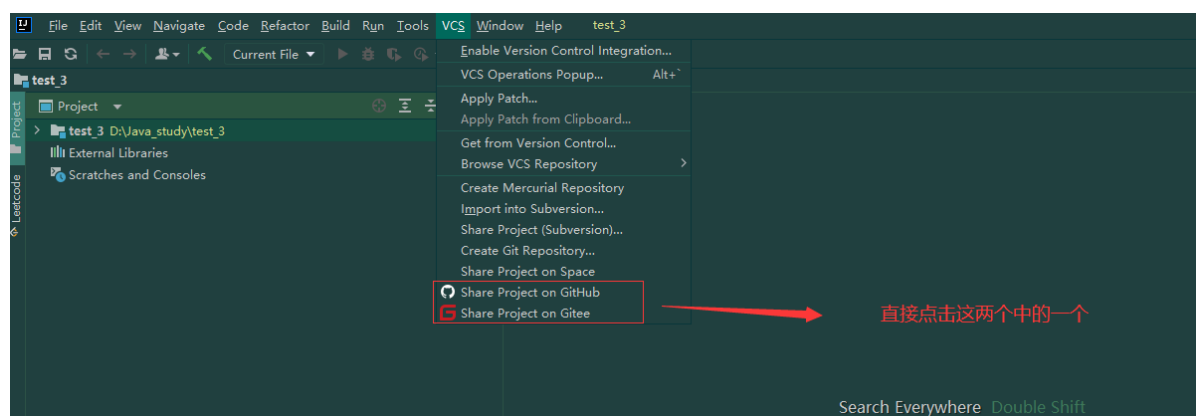# GIt 学习和使用教程

Git工具的使用

GitHubDesktop

在Idea上使用Git

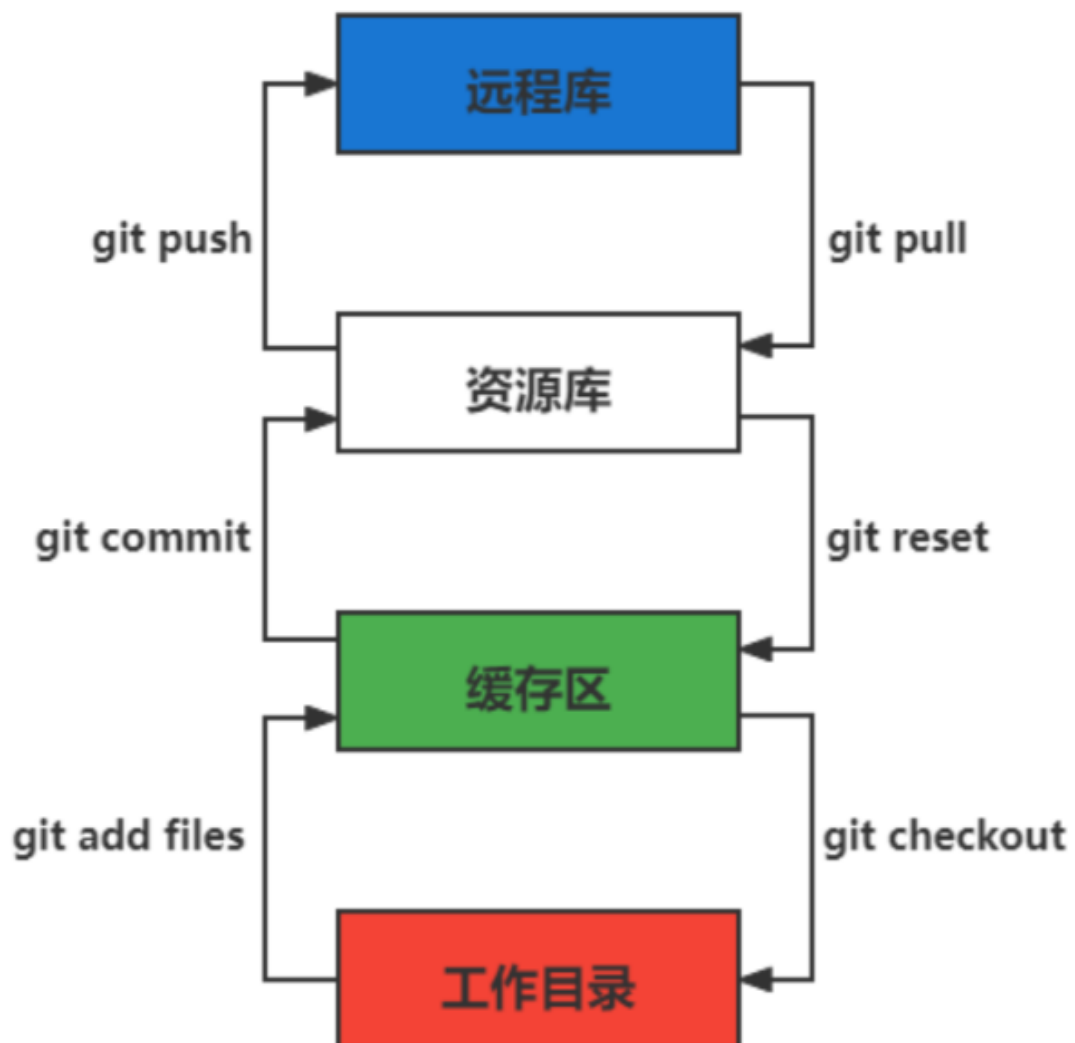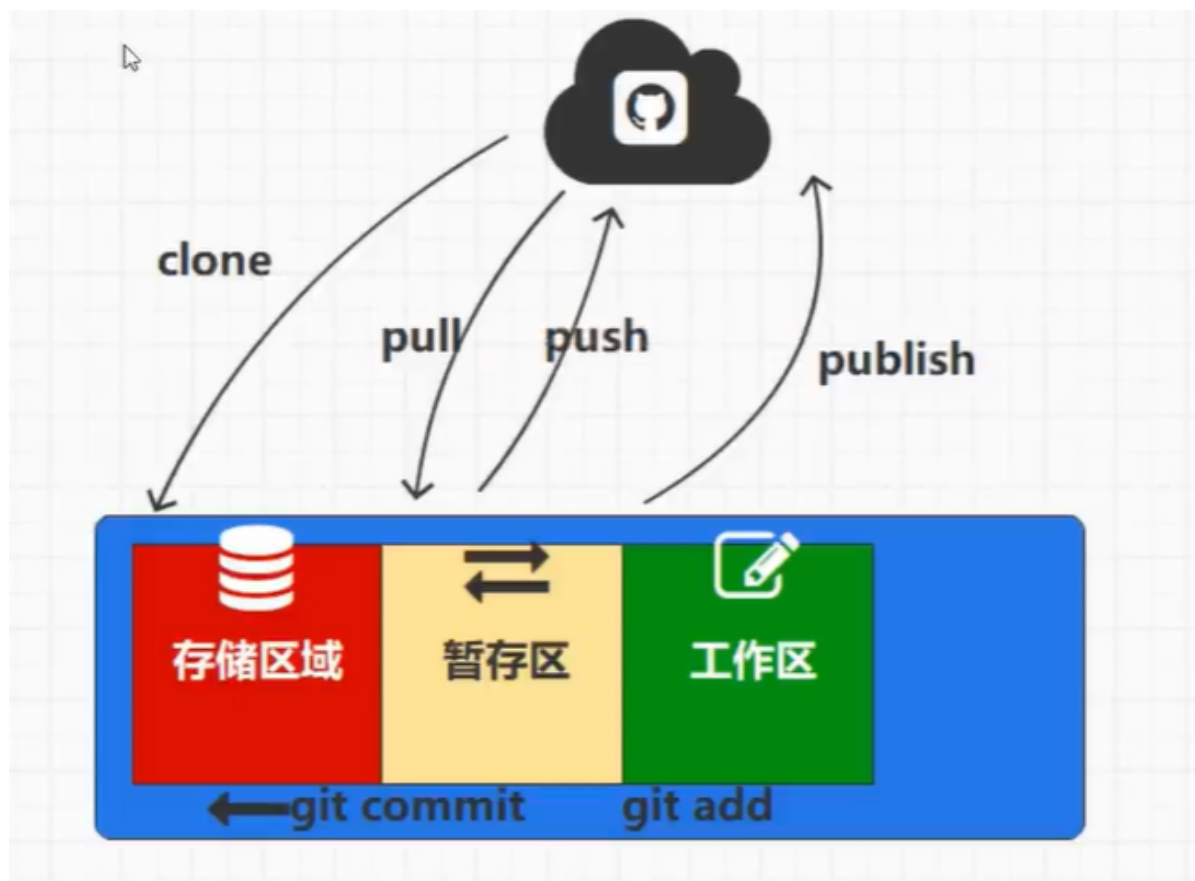# 一、Git工作流程

Git有四个区域

- 3个本地区域
  - 工作区（Workspace）：存放项目代码的地方
  - 暂存区（Stage）：存放临时的改动，事实上它只是一个文件，保存即将提交的文件列表信息。

- 资源库（Repository）：安全存放数据的位置，这里面有提交到所有版本的数据。其中 HEAD 指向最新放入仓库的版本。
- 1个远程区域
  - 远程库（Remote）：托管代码的服务器
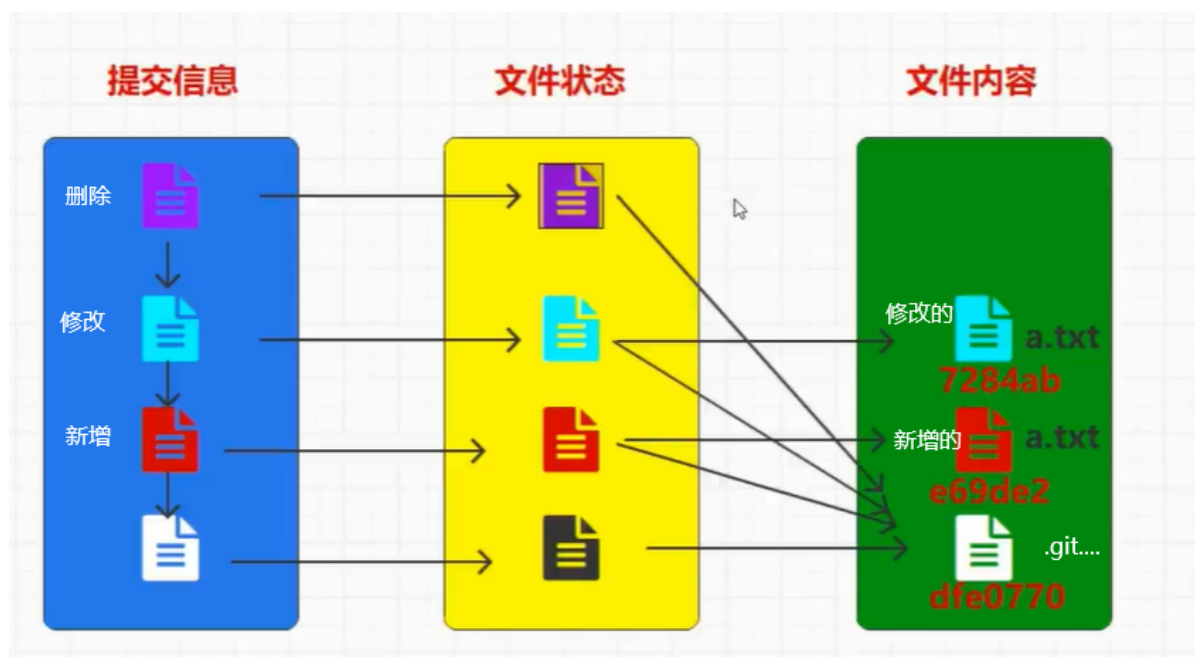
```
tang3@DESKTOP-773TJHF MINGW64 /e/test/local_test_4 (main)
$ git cat-file -p 846745898c02b54ed80cfa4c58fdafc56ca90ecu        当前指针所对应
tree 2c55199b96284f5decfb50117b7c51869ab598ca                      的版本号
parent a1fd338280b50f43df0508f50dec614b4699816a            上一次提交的版本号
author tyd <tang3816697@foxmail.com> 1691841638 +0800
committer tyd <tang3816697@foxmail.com> 1691841638 +0800

Update a.txt

修改了一下

tang3@DESKTOP-773TJHF MINGW64 /e/test/local_test_4 (main)
$
```

2.使用文本编辑器内置命令

如果你正在使用文本编辑器（如vi、nano等）编辑文件，并且想要保存并退出该文件，请按照以下步骤操作：

-在vi中，首先按下Esc键，然后输入":wq"（不包括引号），然后按下回车键即可保存并退出。

-在nano中，首先按下Ctrl键加X键，然后输入Y键以保存更改并退出。

Git一些常用的基础指令

- git -v 查看git的版本
  -
    ```
    tang3@DESKTOP-773TJHF MINGW64 /
    $ git -v
    git version 2.40.0.windows.1
    ```

- git init 在当前目录下创建仓库
  -
    ```
    $ git init
    Initialized empty Git repository in E:/test/local-rep-1/.git/
    ```

- git clone ... 将远程仓库克隆到本地
  -
    ```
    $ git clone https://gitee.com/aitiger/test-gitee-project.git
    Cloning into 'test-gitee-project'...
    remote: Enumerating objects: 21, done.
    remote: Counting objects: 100% (21/21), done.
    remote: Compressing objects: 100% (7/7), done.
    remote: Total 21 (delta 0), reused 0 (delta 0), pack-reused 0
    Receiving objects: 100% (21/21), done.
    ```

- 配置邮箱和地址
  - git config user.name ....
  - git config user.email ....
  -
    ```
    18801@LAPTOP-J9IRK5BM MINGW64 /e/test/remote-git-rep (master)
    $ git config user.name lihaibo

    18801@LAPTOP-J9IRK5BM MINGW64 /e/test/remote-git-rep (master)
    $ git config user.email 15811009164@163.com
    ```

- git status   查看暂存区的状态
  -
    ```
    18801@LAPTOP-J9IRK5BM MINGW64 /e/test/local-rep-1 (master)
    $ git status
    On branch master

    No commits yet

    nothing to commit (create/copy files and use "git add" to track)
    ```

- git add 将工作区的数据，推送到暂存区做比对操作
  -
    ```
    tang3@DESKTOP-773TJHF MINGW64 /e/test/local-rep-1 (master)
    $ git add a.txt
    ```
  -
    ```
    tang3@DESKTOP-773TJHF MINGW64 /e/test/local-rep-1 (master)
    $ git status
    On branch master

    No commits yet

    Changes to be committed:
      (use "git rm --cached <file>..." to unstage)
            new file:   a.txt
    ```

- git rm --cached a.txt 将暂存区的文件，移动到工作区去

```
tang3@DESKTOP-773TJHF MINGW64 /e/test/local-rep-1 (master)
$ git rm --cached a.txt
rm 'a.txt'

tang3@DESKTOP-773TJHF MINGW64 /e/test/local-rep-1 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        a.txt
```

- git commit -m 将暂存区的文件，移动到存储区

```
tang3@DESKTOP-773TJHF MINGW64 /e/test/local-rep-1 (master)
$ git commit -m 新增文件
[master (root-commit) fb5c1de] 新增文件
 1 file changed, 1 insertion(+)
 create mode 100644 a.txt

tang3@DESKTOP-773TJHF MINGW64 /e/test/local-rep-1 (master)
$ git status
On branch master
nothing to commit, working tree clean
```

- git log 查看提交日志

```
tang3@DESKTOP-773TJHF MINGW64 /e/test/local-rep-1 (master)
$ git log
commit fb5c1de8ee460ca847329f5af80c8947ef44d677 (HEAD -> master)
Author: tyd <tang3816697@foxmail.com>
Date:   Sat Aug 12 21:02:30 2023 +0800

    新增文件
```

- git restore 文件误删后，可以通过这个语句来恢复文件

  - 若是没有通过 git commit提交前:

```
tang3@DESKTOP-773TJHF MINGW64 /e/test/local-rep-2 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    a.txt

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

tang3@DESKTOP-773TJHF MINGW64 /e/test/local-rep-2 (master)
$ git restore a.txt
```

- 若是用了git commit 提交后，则需要采用 git reset --hard 版本号 来恢复 不过缺点是，把我们的提交过程丢失了

- 使用 git revert 版本号，相当于创建了一个新的提交

- 创建分支 git branch ....

- 切换分支 git checkout user

- 创建并切换分支 git checkout -b ....

- 删除分支 git branch -d user

- git merge 合并操作

- 增加标签 git tag 标签名 版本号

- 远程仓库 git remote add url

- git push url

- git pull url