## Android Studio、ADB、Git熟悉。

### Android Studio

a、ctrl + 鼠标左键点击 与vs相同，用于跳转代码

b、ctrl + alt + ←/→ 与vs中ctrl + -/+相同，用于定位至上一次/下一次操作的位置。

### ADB（先不管、等收到pad）

### git 操作

#### git 拉代码（pull/fetch）

##### git fetch

git fetch 是一个从远程仓库获取更新但不自动合并到本地分支的Git命令。它允许你查看远程分支上的更改，并决定何时将这些更改合并到本地分支。这与git pull不同，后者会在获取更新后立即合并。

使用git fetch的场景：

1. 在合并前检查更改：在将远程分支的更改合并到本地分支之前，使用git fetch来浏览更新内容，确保代码合并不会有问题。
2. 保持本地分支与远程分支同步：定期使用git fetch来检查远程分支的状态，而不影响当前分支的代码。

基本用法：git fetch <远程仓库> <分支>

例如从默认的远程仓库origin 拉取所有分支的更新：git fetch origin

结果说明：

运行git fetch后，你的本地仓库会更新远程跟踪分支（如origin/main）以反映远程仓库的最新状态。

本地分支不会被修改，只有在你明确地合并或者变基时，才会将这些更改应用到你的分支中。

##### git log <远程仓库> <分支>

在使用git fetch之后，你可以使用git log origin/main来查看远程分支状态

##### git merge

功能：合并两个分支的历史，创建一个新的合并提交来将他们结合在一起。

特点：合并后的提交历史是非线性的，分支合并点会清晰显示。适用于团队，它保留了每个分支的历史，方便追溯。

使用场景：当你想要保留完整的分支历史或需要明确显示两个分支的合并点时。

示例用法：

##### git pull <远程仓库> <分支>

#### git stash

`git stash` 是 Git 中用于临时存储当前工作目录和暂存区的改动的命令，它让你可以在不提交当前更改的情况下切换分支或进行其他操作，然后稍后再恢复这些更改。

git stash 默认不会包含未跟踪的文件（即新创建的文件）。如果你希望stash这些文件，可以使用git stash -u

git stash 主要用于短期保存改动。如果你需要长时间保存和在多个分支之间共享改动，建议使用分支。

##### git stash

保存当前工作区的改动。这会把当前的修改和未跟踪的文件保存到一个栈中，并恢复工作目录到上一次提交的状态。

##### git stash save “描述信息”

保存并添加描述。这样可以帮助你记住这个存储的目的

##### git stash list

查看已保存的stash列表。这会列出所有已保存的stash记录，按时间顺序显示。

##### git stash apply

恢复最近一次保存的stash。应用最近一次的stash改动到当前分支，但不会从stash列表中删除它。

##### git stash apply stash@{n}

恢复特定的stash。其中n是stash的索引（例如stash@{0}是最新的一个）

##### git stash pop

恢复并删除stash。应用最近的一次stash改动到当前分支，并将该stash从列表中删除

##### git stash drop stash@{n}

删除特定的stash

##### git stash clear

清空所有stash

# Git 问题汇总

## 远端创建一个新的项目（https://gitlab.com/987052850/Learning.git），如何将本地文件夹(C:\Users\e-Yanchang.Zhao\Desktop\新建文件夹)推送到这个项目？

### 进入本地文件夹目录

cd C:\Users\e-Yanchang.Zhao\Desktop\新建文件夹

或者在该文件夹中右键，点击git bash here

### 初始化本地仓库

git init

### 添加远程仓库

git remote add origin <https://gitlab.com/987052850/Learning.git>

### 添加文件并提交

git add .

git commit -m “将本地文件夹推送至远端”

### 推送至远程仓库

git push -u origin main

这个指令会将上一步提交的内容提交

## 基于远端分支user001创建新的远端分支user002

1. 获取远程仓库的最新信息

git fetch origin

1. 创建本地分支并切换到user002

git checkout -b user002 origin/user002

1. 将本地的user002推送到远端

git push -u origin/user002

## 基于远端分支user002创建新的本地分支test001

git checkout -b test001 origin/user002

## 基于本地分支test001创建新的本地分支test002

git checkout -b test002

git branch --unset-upstream

## 新建文件夹并拉取远端指定分支

git init

git remote add origin <https://gitlab.com/987052850/Learning.git>

git fetch origin

git checkout -b localBranch origin/remoteBranch

## 强制本地分支与远端分支状态一致

即覆盖本地分支改动而直接同步远端分支内容

git fetch origin

git reset –hard origin/master

git status