几乎所有的版本控制系统都以某种形式支持分支。 使用分支意味着你可以把你的工 作从开发主线上分离开来，以免影响开发主线。分支指向最新提交。分支的本质就是一个提交对象

1. 创建分支：

命令：git branch 名字

创建 一个新分支，并不会自动切换到新分支中去，在当前所在的提交对象上创建一个指针

1. 切换分支：

命令：git checkout ‘名字’

git branch

不只是可以创建与删除分支。 如果不加任何参数运行它， 会得到当前所有分支的一个列表

git branch -d name

删除分支 ，已合并的

git branch -v

可以查看每一个分支的最后一次提交

git branch name commitHash

新建一个分支并且使分支指向对应的提交对象

git branch –merged

查看哪些分支已经合并到当前分支，在这个列表中分支名字前没有 \* 号的分支通常可以使用 git branch -d 删除掉；

git branch --no-merged

查看所有包含未合并工作的分支

尝试使用 git branch -d 命令删除在这个列表中的分支时会失败。 如果真的想要删除分支并丢掉那些工作，可以使用 -D 选项强制删 除它。

1. 查看分支历史：

命令：git log --oneline --decorate --graph --all

1. 查看当前分支所指对象

命令：git log --oneline --decorate

（提供这一功能的参数是 --decorate）

1. 切换分支：

命令：git checkout 名字

*（分支切换会改变你工作目录中的文件，在切换分支时，一定要注意你工作目录里的文件会被改变。 如果是切换到一个较旧的分支，你的工作目录会恢复到该分支最后一次提交时的样子。如果 Git 不能干净利落地完成这个任务，它将禁止切换分支，每次在切换分支前 提交一下当前分支）*6. 新建分支，并切换

命令：git checkout -b 名字

**最佳实践**：切换分支前，最好是提交状态，否则会污染主分支，在切换分支时，如果当前分支上有未暂存的修改，或者有未提交的暂存，分支切换可以成功，但是会污染其他分支。

1. 合并分支

命令：git merge 分支名