

# Git Basics

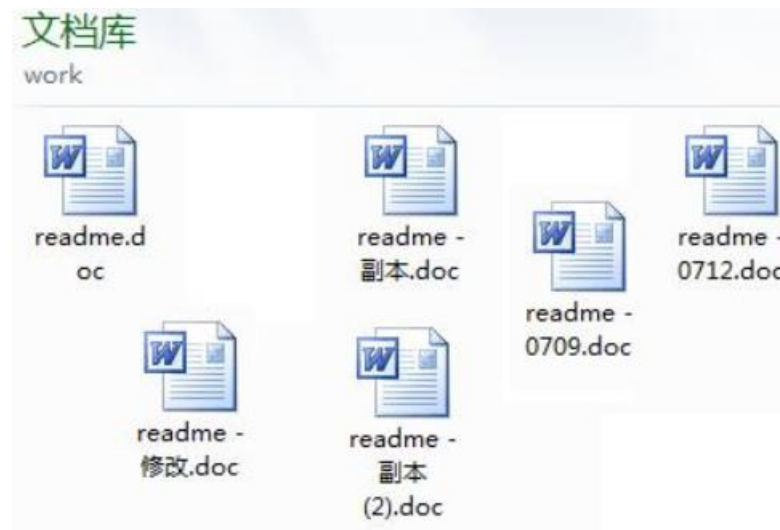
Chenyu Bian  
2020/4/16

- Git 简介
- Git 操作
  - 基础操作
  - 版本回退
  - 解决冲突
  - 分支管理

# Git简介

- Git是一个开源的分布式版本控制系统。

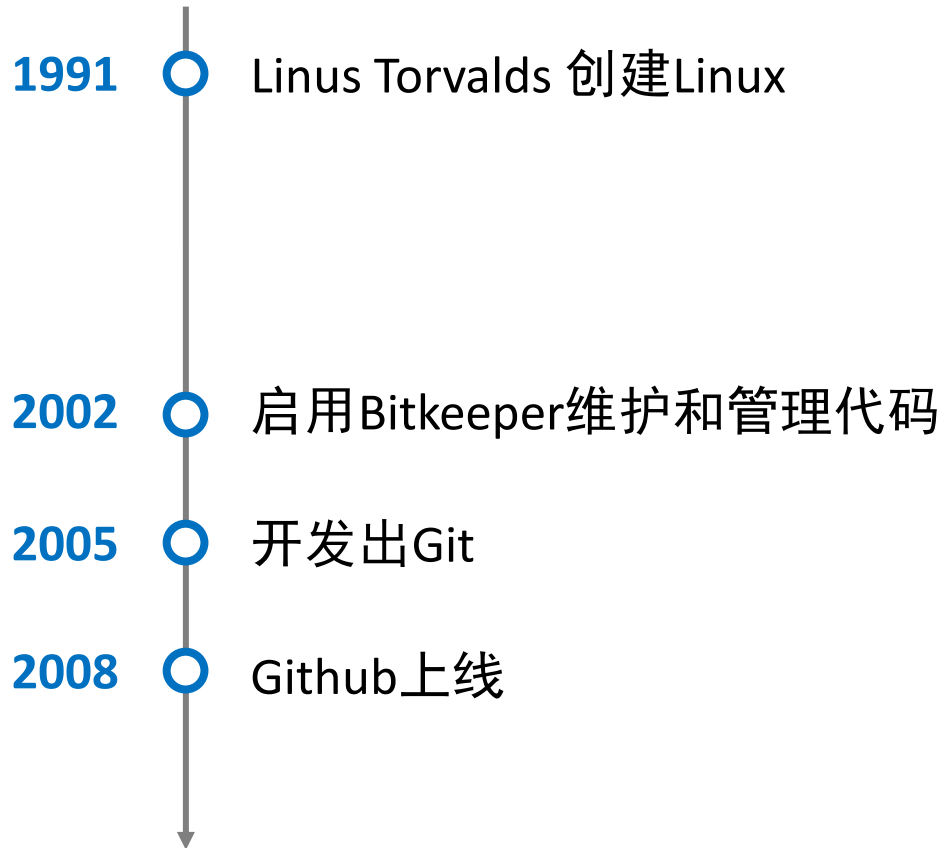
- 版本控制(revision control)是一种记录一个或若干文件内容变化,以便将来查阅特定版本修订情况的系统。



版本	文件名	用户	说明	日期
1	service.doc	张三	删除了软件服务条款5	7/12 10:38
2	service.doc	张三	增加了License人数限制	7/12 18:09
3	service.doc	李四	财务部门调整了合同金额	7/13 9:51
4	service.doc	张三	延长了免费升级周期	7/14 15:17

# Git 简史

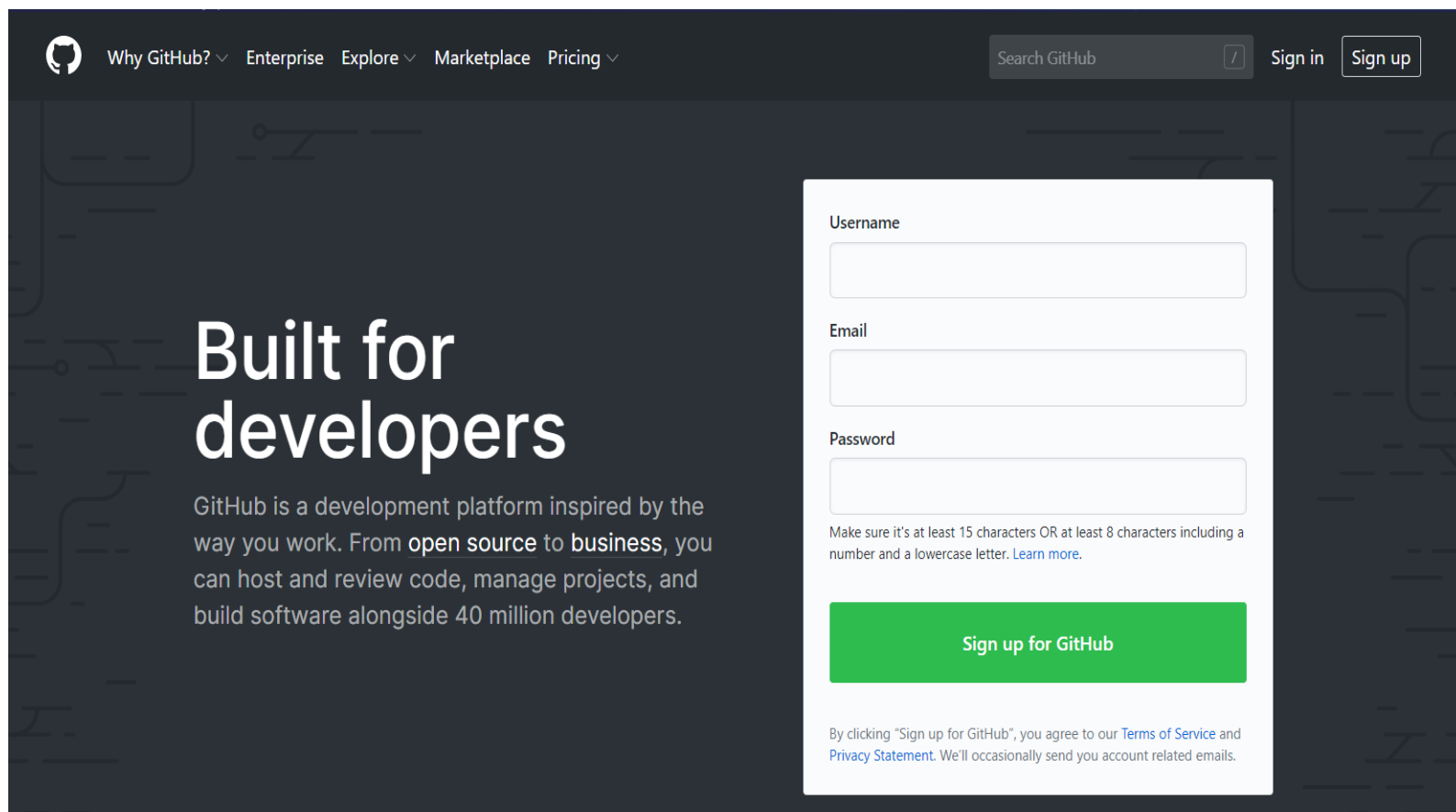
---



Linus Torvalds

# Github 注册

1. 进入Github 官网（<https://github.com/>），输入用户名、邮箱和密码注册



The screenshot shows the GitHub homepage with a dark background. On the left, the text "Built for developers" is prominently displayed in white. Below it, a paragraph describes GitHub as a development platform. On the right, there is a white sign-up form with three input fields: "Username", "Email", and "Password". Below the "Password" field, there is a note about password requirements and a link to "Learn more". A green button labeled "Sign up for GitHub" is positioned below the form. At the bottom of the form, there is a line of text stating that by clicking "Sign up for GitHub", the user agrees to the "Terms of Service" and "Privacy Statement".

Why GitHub? ▾ Enterprise Explore ▾ Marketplace Pricing ▾

Search GitHub / Sign in Sign up

## Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 40 million developers.

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more](#).

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.

## 2. 验证账户

Join GitHub

# Create your account

### Verify your account

请解决接下来的问题，  
以便让我们知道您是真人而非机器

验证

### Email preferences

☒ Send me occasional product updates, announcements, and offers.

Next: Select a plan

## 4. 邮箱地址确认



Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.

An email containing verification instructions was sent to [redacted]

Resend verification email

Change your email settings

## 3. 选择用户计划

Choose a plan

### Individual

Pick the plan that's right for you, personally.

### Team

Choose a plan to help your team grow and collaborate.



Free  
\$0 USD

The basics of GitHub for every developer

Choose Free



Pro  
\$7 USD

Per month

Pro tools for developers with advanced requirements

Choose Pro

The screenshot shows the GitHub homepage. At the top, there's a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below the navigation bar, there's a section titled 'Learn Git and GitHub without any code!' which includes a 'Read the guide' button and a 'Start a project' button. To the right of this section, there's a 'Explore repositories' section listing several repositories with their names, descriptions, and star counts. At the bottom, there's a section titled 'Get the Student Developer Pack' which includes a 'Get the Pack' button and an illustration of two people working on a laptop.

# Git 安装

Git 各平台安装包下载地址: <https://git-scm.com/>

## 1. 进入官网



## 2. 选择系统对应安装包



### Linux上安装Git:

`sudo apt-get install git`

### Windows上安装:

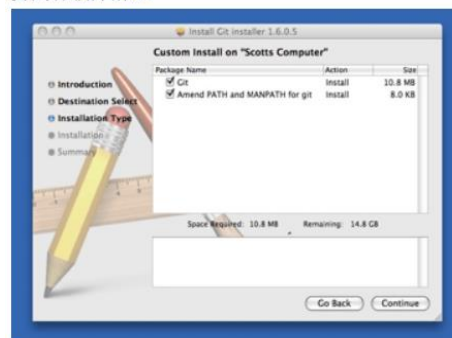
双击.exe 安装文件, 一路next即可

### Mac上安装:

在 Mac 平台上安装 Git 最容易的当属使用图形化的 Git 安装工具, 下载地址为:

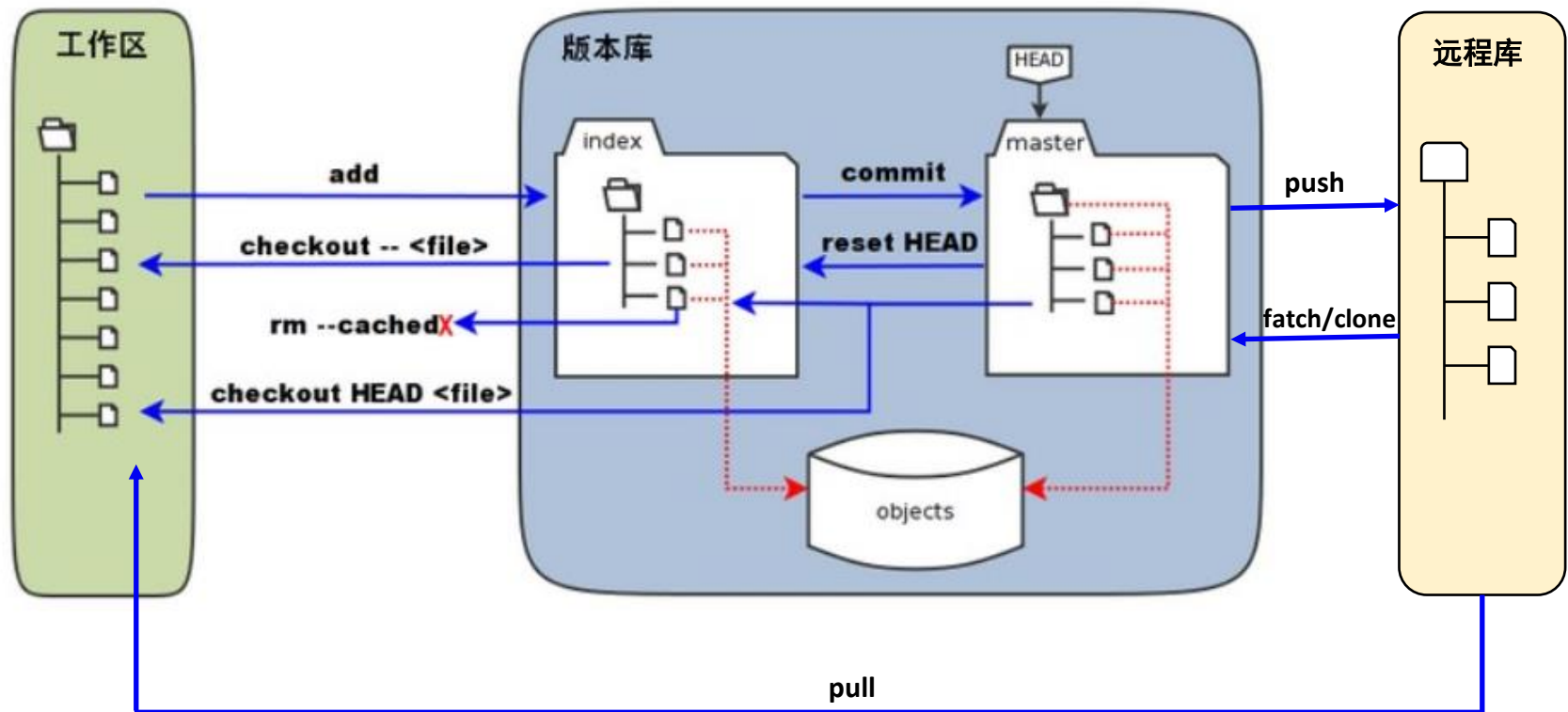
<https://sourceforge.net/projects/git-osx-installer/>

安装界面如下所示:



Git for windows 镜像: <https://npm.taobao.org/mirrors/git-for-windows/>  
<https://github.com/waylau/git-for-win>

# Git 工作流程



Git 工作流程:

1. 在工作区中添加、修改文件;
2. 将需要进行版本管理的文件放入暂存区;
3. 将暂存区文件提交到本地git仓库;
4. 将本地git仓库推送到远端 (可选)。



# Git 操作

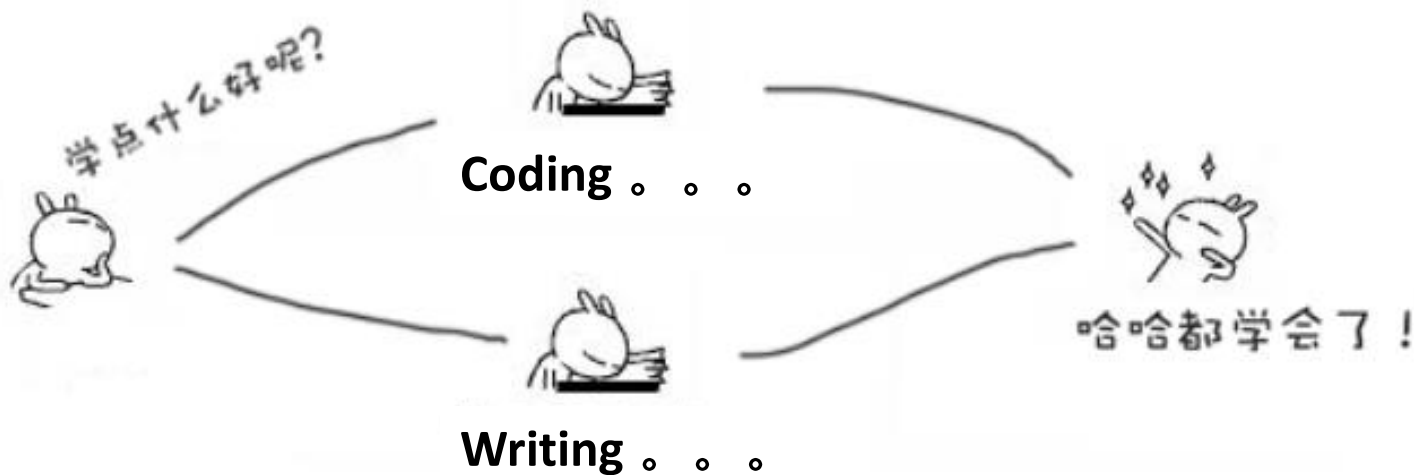
---

1. 基础操作：将本地非空目录转化为Git仓库并推送到远端；
2. 版本回退：将新版本的文件回退到之前的版本；
3. 解决冲突：从其他服务器克隆一个已存在的Git仓库进行修改并推送；
4. 分支管理：分支的基本操作和分支管理策略

# 分支管理

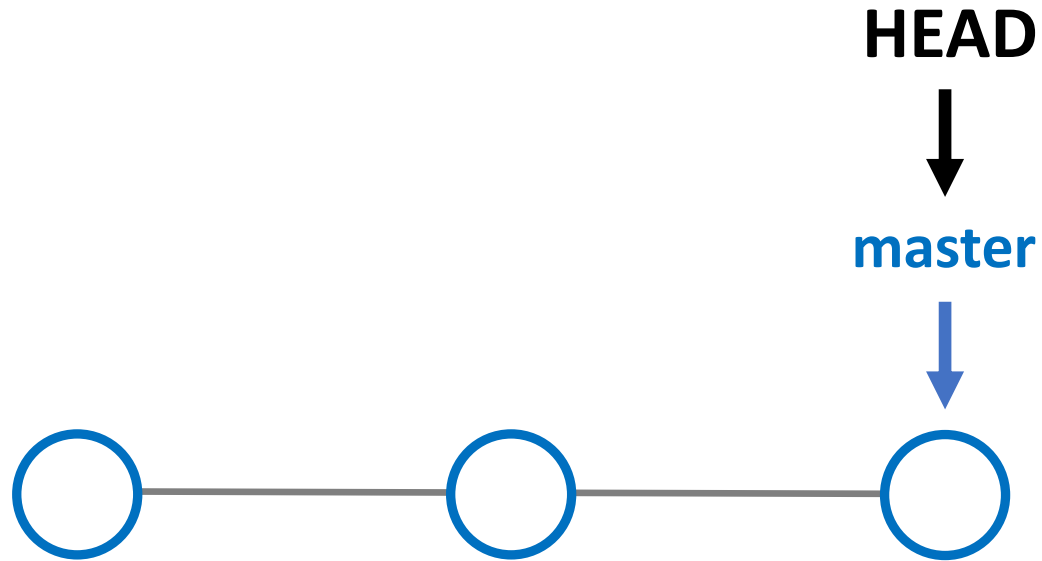
---

什么是分支？



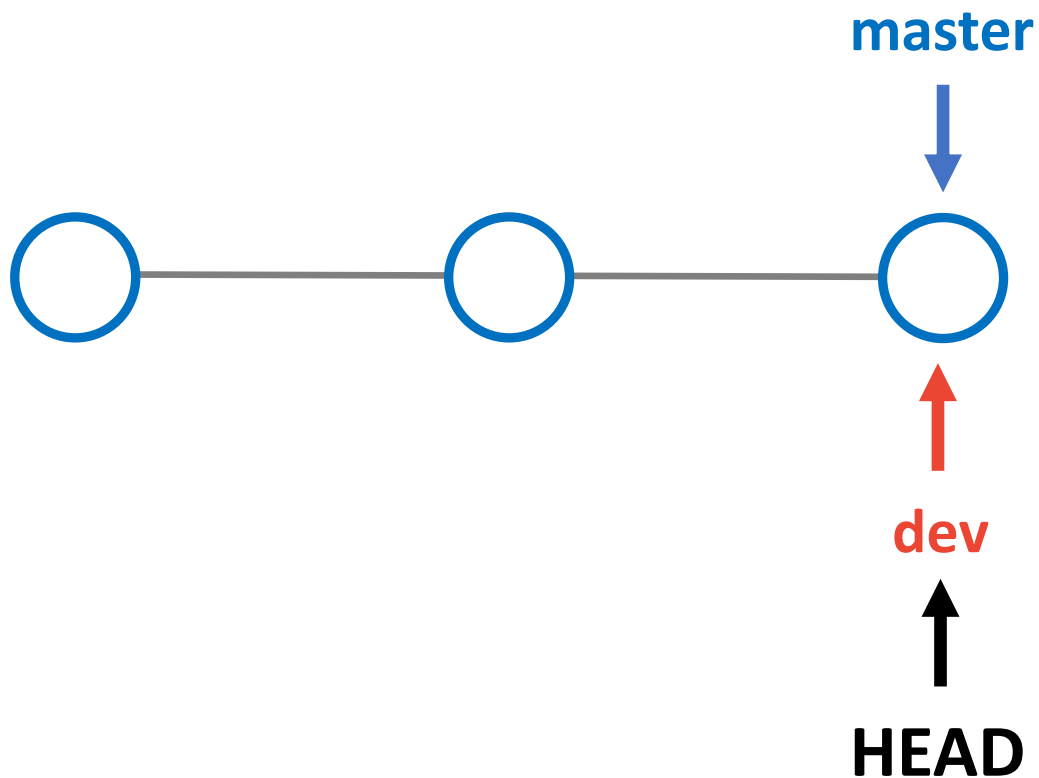
# 分支管理

---



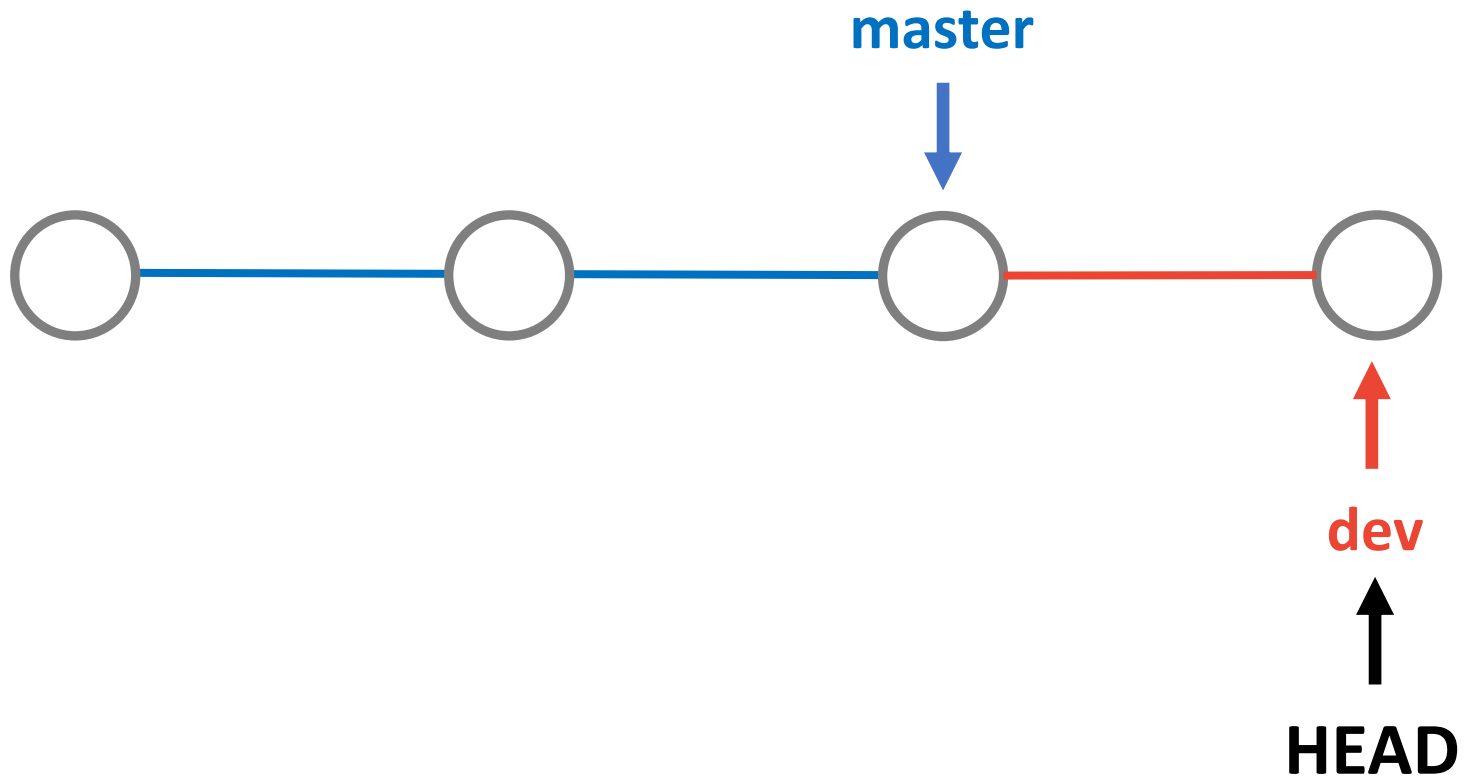
# 分支管理

---



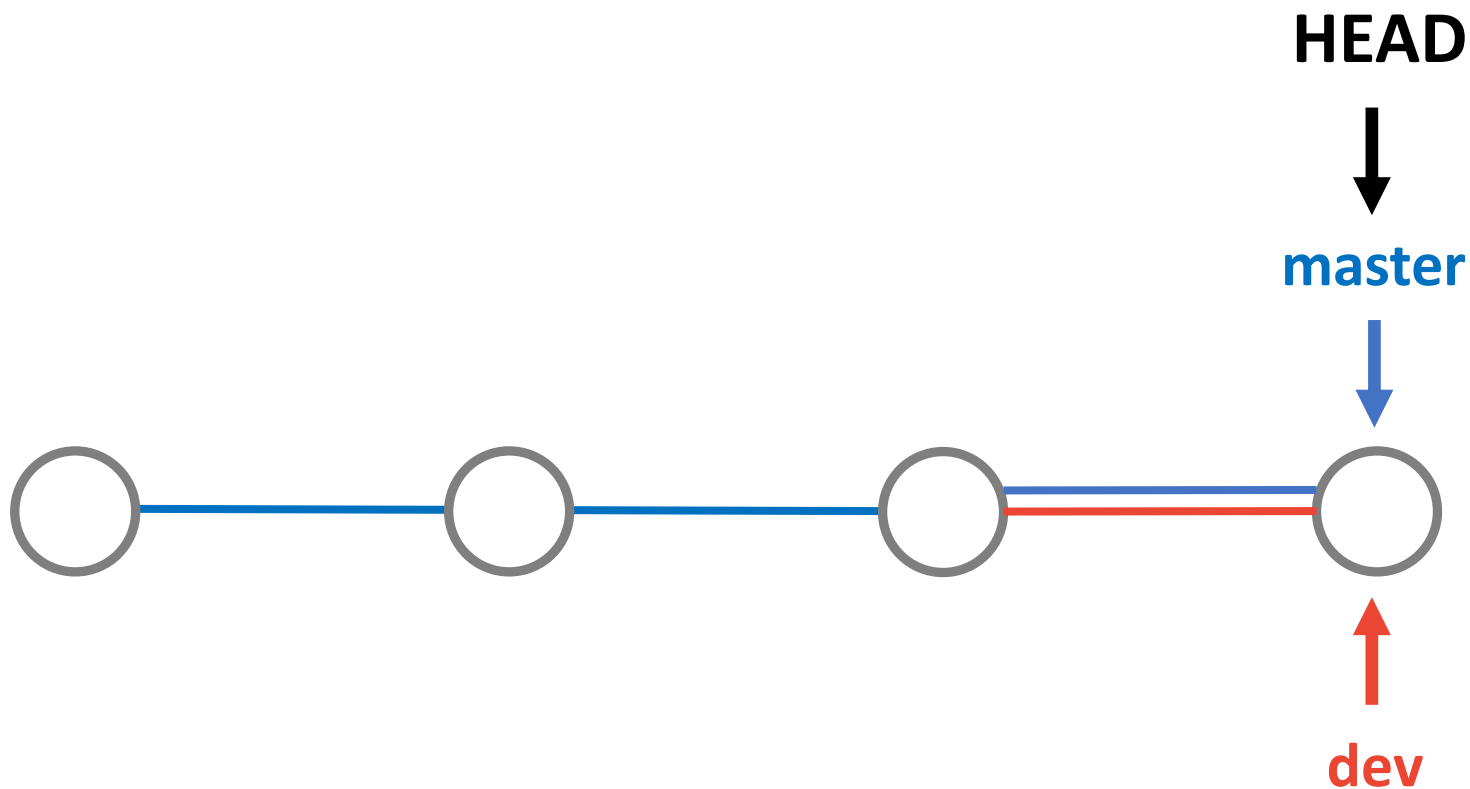
# 分支管理

---



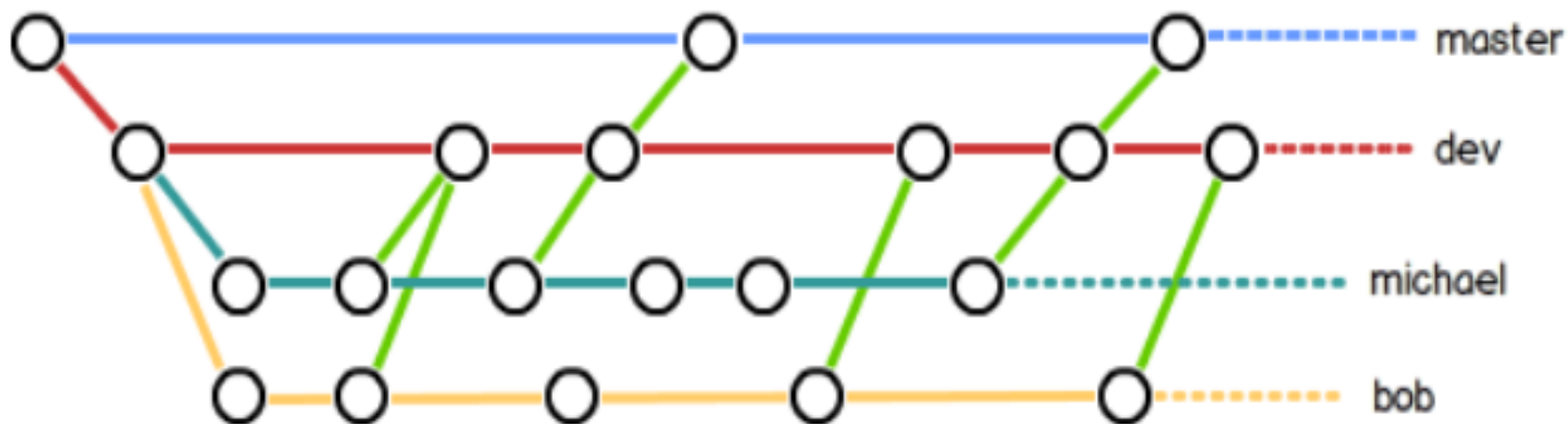
# 分支管理

---



# 分支管理

## 分支管理基本策略



# 总结

---

## # 基本操作

<code>git init</code>	# 创建仓库
<code>git clone &lt;url&gt;</code>	# 从远程克隆仓库
<code>git add &lt;filename&gt;</code>	# 将文件从工作区添加到暂存区
<code>git commit -m 'message'</code>	# 将文件从暂存区添加到版本库
<code>git push &lt;url&gt;</code>	# 将本地仓库推送到远端
<code>git reset --hard &lt;commit_id&gt;</code>	# 回退到之前的版本
<code>git status</code>	# 查看当前仓库状态
<code>git log</code>	# 查看历史记录 (commit)
<code>git reflog</code>	# 查看所有操作记录 (commit和reset)
<code>git pull</code>	# 从远端拉取文件到本地

## # 分支管理

<code>git branch &lt;branch_name&gt;</code>	# 创建分支
<code>git checkout &lt;branch_name&gt;</code>	# 切换分支
<code>git switch &lt;branch_name&gt;</code>	# 切换分支
<code>git checkout -b &lt;branch_name&gt;</code>	# 创建并切换分支
<code>git merge &lt;branch_name&gt;</code>	# 合并指定分支到当前分支
<code>git branch -d &lt;branch_name&gt;</code>	# 删除分支



Thank You!

Any questions ?