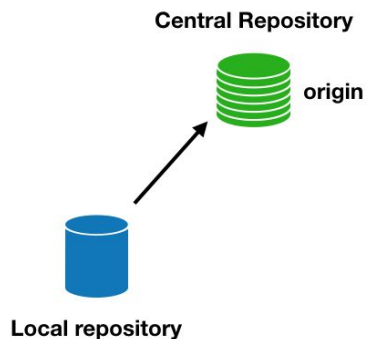


第六章

创建+推到远程仓库

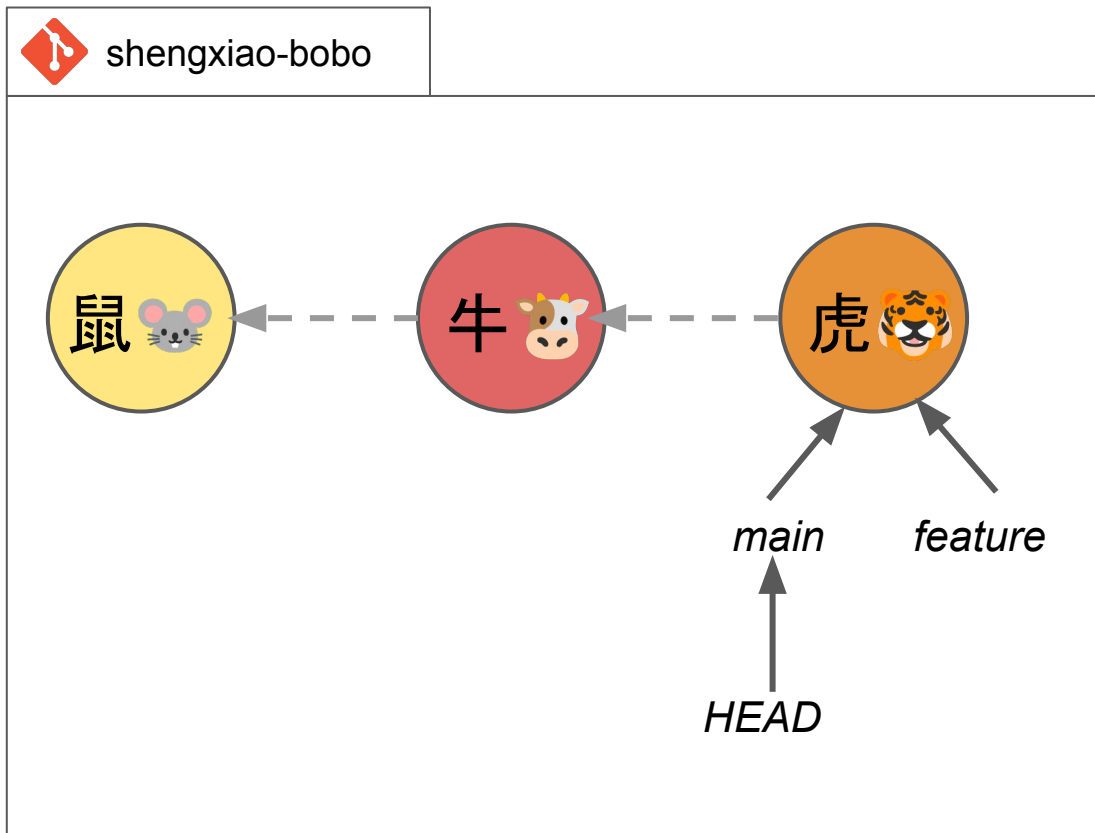


内容大纲

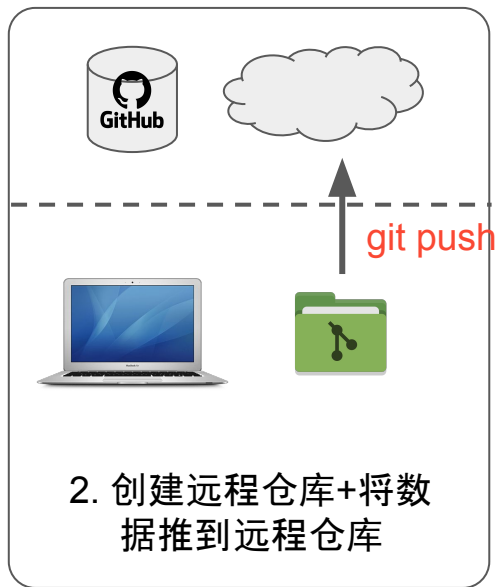
1. 使用Git开始项目的两种方式
 - a. 从本地仓库开始
 - b. 从远程仓库开始
2. 远程仓库的作用
3. 在GitHub上创建个人访问令牌(Personal Access Token)
4. 为shengxiao项目创建一个GitHub远程仓库
5. 将本地内容推(push)到远程仓库



当前仓库状态

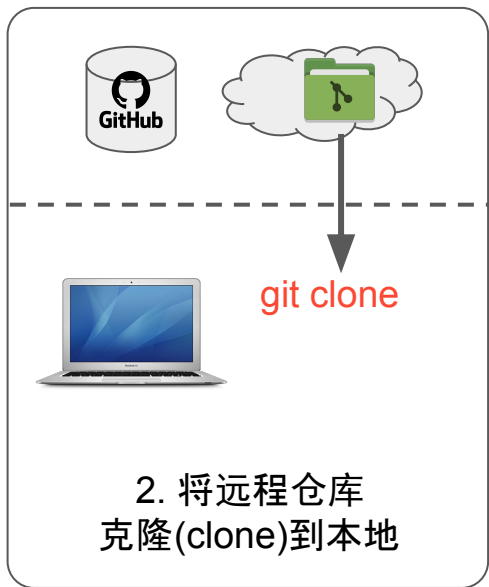


使用Git开始项目方式一～从本地仓库开始



例子: shengxiao

使用Git开始项目方式一～从远程仓库开始



注意

- 本地和远程仓库之间**相互独立**
- 两者之间的交互不会自动发生
- 两者之间的数据同步需通过**命令**执行

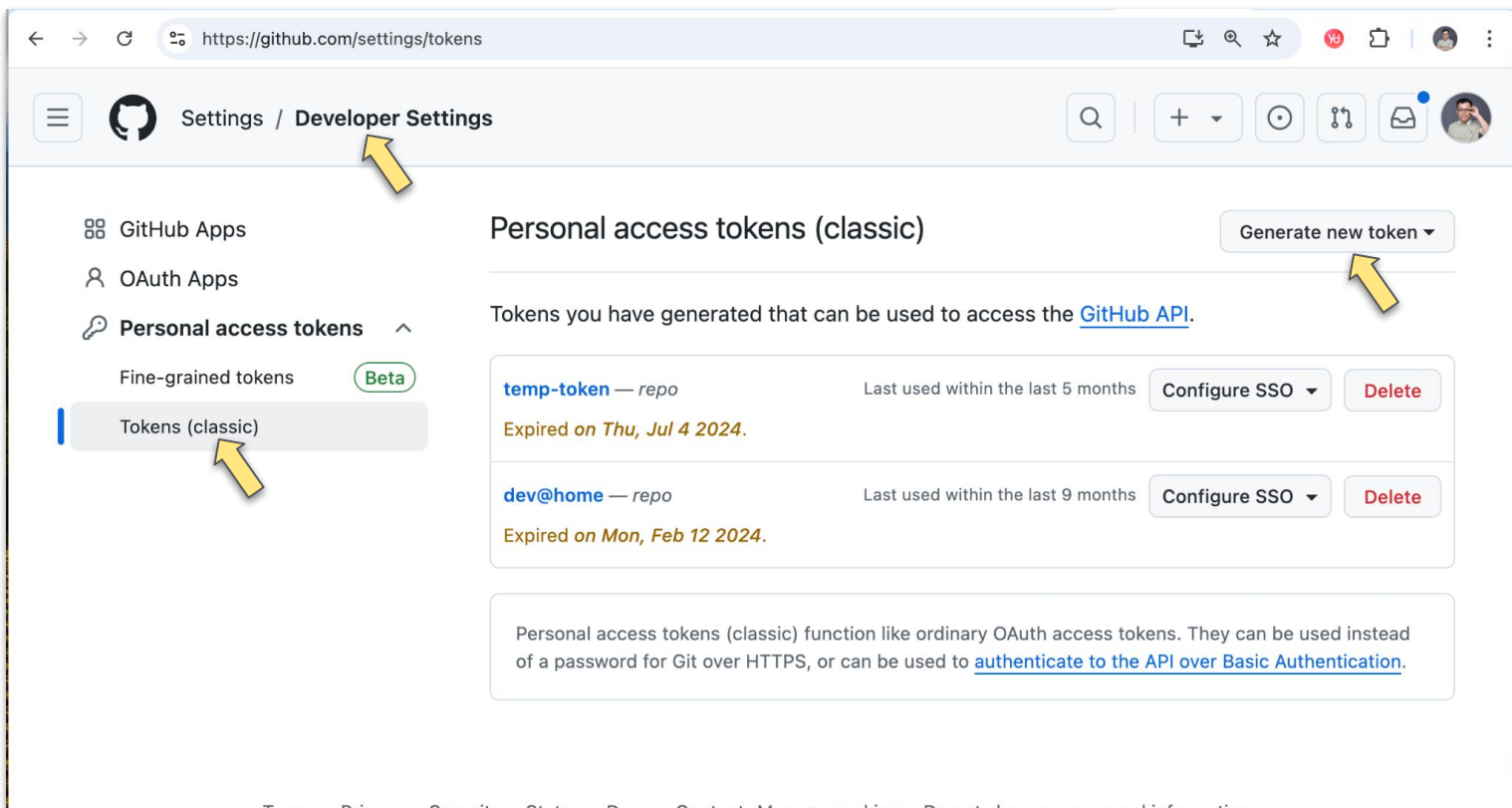


为什么要使用远程仓库？

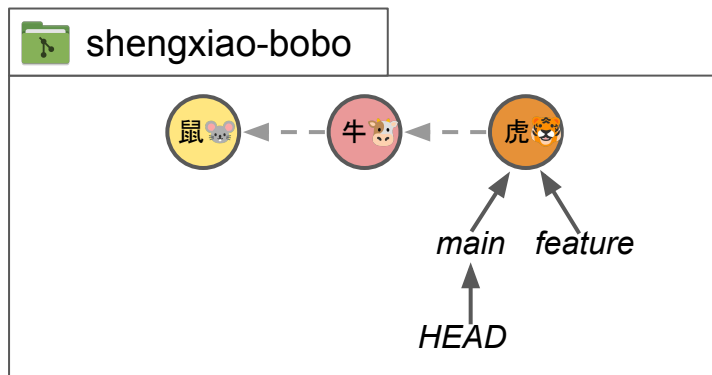
- 将本地数据**备份**到远程服务器
- 从**多台**计算机上访问Git项目
- 和他人**协作**开发Git项目



创建账号和个人访问令牌



目前本地和远程仓库情况



操作步骤～创建+推数据到远程仓库

1. 在GitHub上创建远程仓库
2. 在本地和远程仓库之间建立连接
3. 将数据从本地推(push)到远程仓库



在GitHub上创建一个私有仓库


Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

Repository name *

 boboweike

 /

shengxiao

shengxiao is available.

Great repository names are short and memorable. Need inspiration? How about [ideal-fishstick](#)?

Description (optional)

A sample project to learn Git

☐ Public

Anyone on the internet can see this repository. You choose who can commit.

☒ Private

You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None


A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

☐ You are creating a private repository in the boboweike organization.

Create repository

shengxiao Private


Edit Pins Unwatch 1 Fork 0 Star 0



Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Get started with GitHub Copilot



Give access to the people you work with

Ensure the right people and teams have access to this repository.

Manage access

Quick setup — if you've done this kind of thing before


Set up in Desktop or HTTPS SSH

https://github.com/boboweike/shengxiao.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

shengxiao Private


Edit Pins Unwatch 1 Fork 0 Star 0



Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Get started with GitHub Copilot



Give access to the people you work with

Ensure the right people and teams have access to this repository.

Manage access

Quick setup — if you've done this kind of thing before

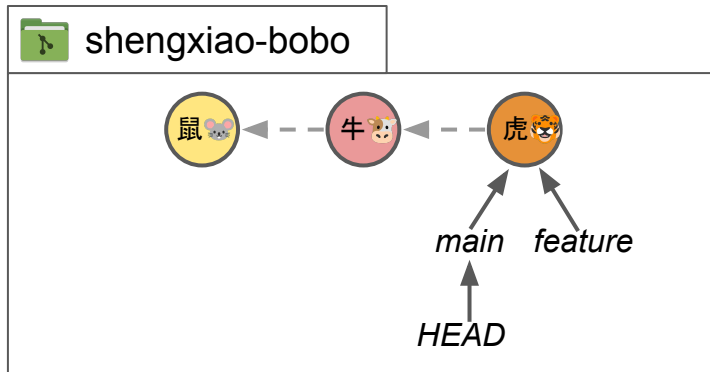
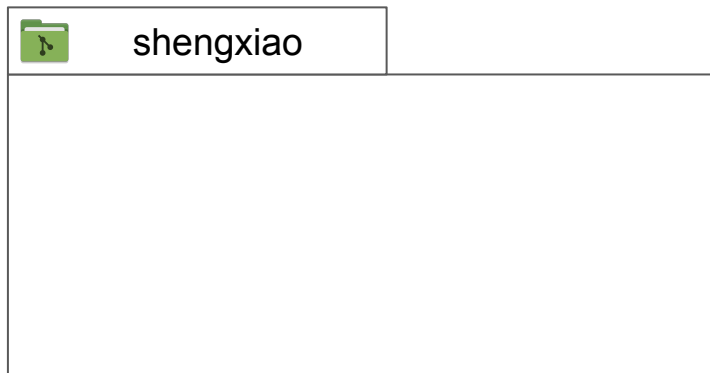
Set up in Desktop or HTTPS SSH

git@github.com:boboweike/shengxiao.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

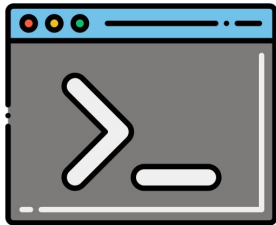
HTTPS <https://github.com/boboweike/shengxiao.git>
SSH [git@github.com:boboweike/shengxiao.git](ssh://git@github.com:boboweike/shengxiao.git)

目前本地和远程仓库情况



命令～在本地和远程仓库之间建立连接

- ***git remote add <shortname> <URL>***
 - 将本地仓库连接到远程仓库, 连接名 **<shortname>**, 远程仓库地址 **<URL>**
 - 后续通过引用shortname访问远程仓库, 惯例命名为 **origin**
 - 远程地址可以是HTTPS URL(本课程使用)或者SSH URL
 - 一个本地仓库可有多到远程仓库的连接(不常见)
- ***git remote***
 - 列出本地到远程仓库的所有连接(connections)的连接名(shortname)
- ***git remote -v***
 - 列出本地到远程仓库的所有连接(connections)的连接名(shortname) + URLs

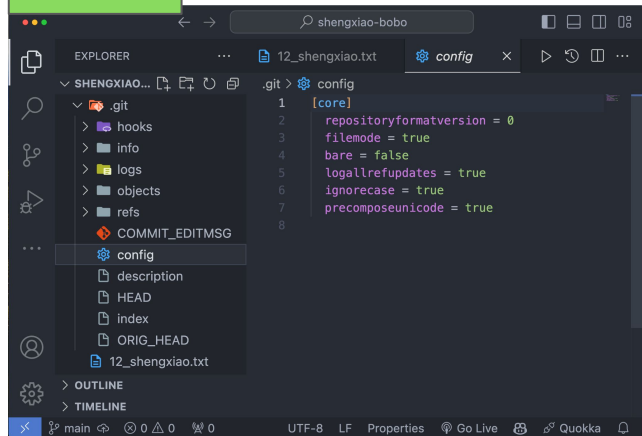


实操～在本地和远程仓库之间建立连接

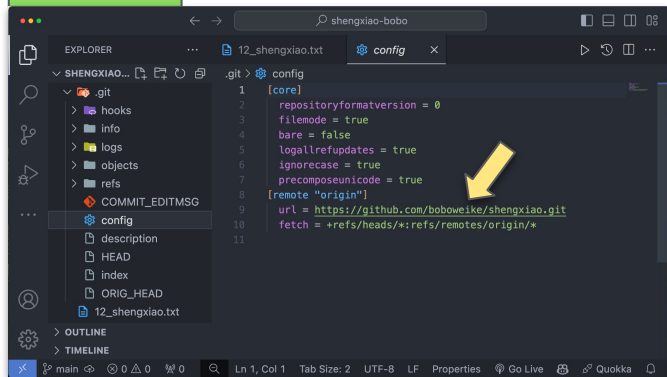
git remote add

```
shengxiao-bobo git:(main) git remote
shengxiao-bobo git:(main) git remote add origin https://github.com/boboweike/shengxiao.git
shengxiao-bobo git:(main) git remote
origin
shengxiao-bobo git:(main) git remote -v
origin https://github.com/boboweike/shengxiao.git (fetch)
origin https://github.com/boboweike/shengxiao.git (push)
shengxiao-bobo git:(main)
```

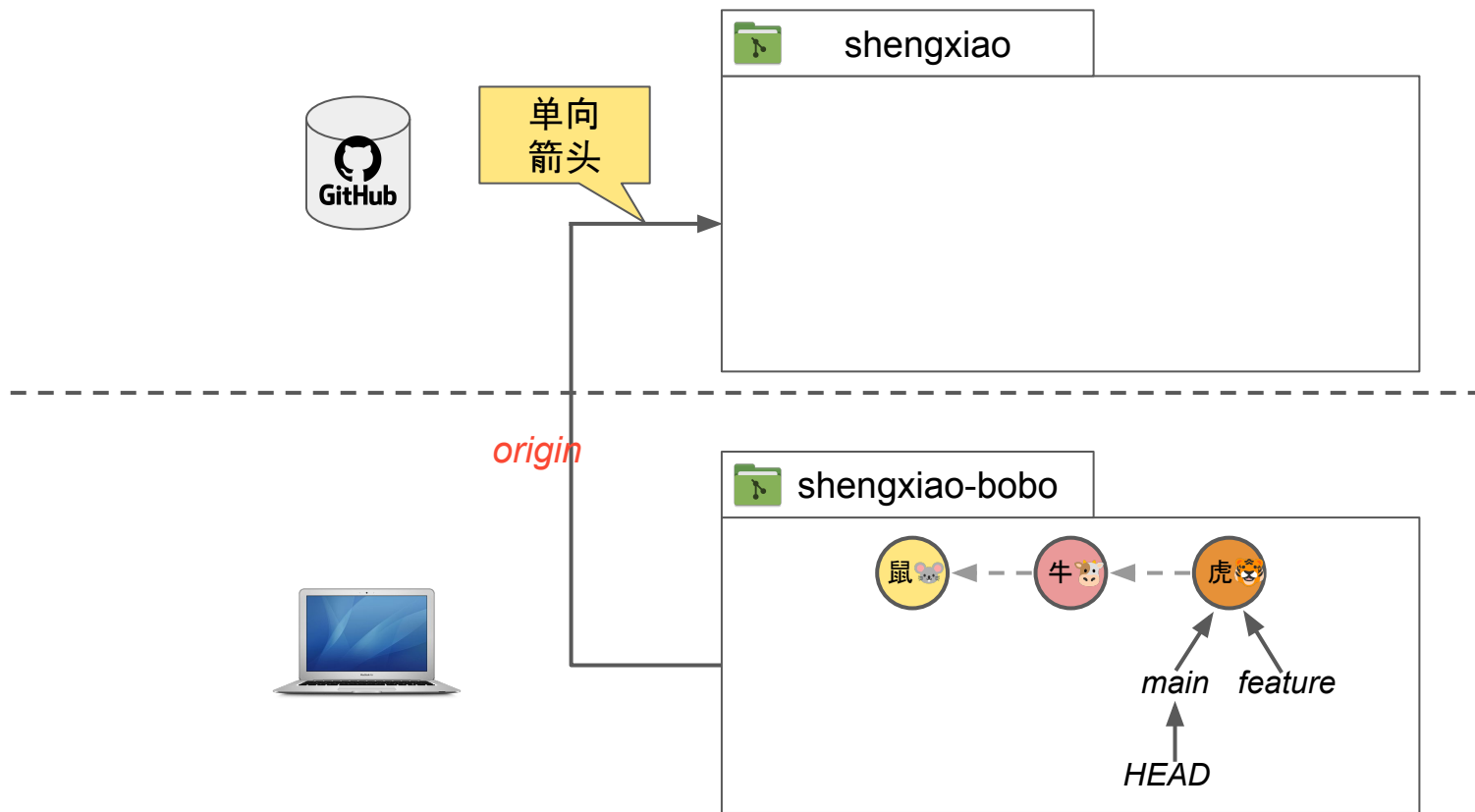
操作前



操作后



目前本地和远程仓库情况



概念

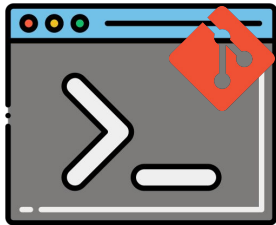
CONCEPT



- **分支(branch)**是指向提交(commit)的可移动指针(复习)
- **本地分支(local branch)** ~ 在本地仓库中的分支
- **远程分支(remote branch)** ~ 在远程仓库中的分支
 - 本地分支变化时, 远程分支并不会自动变化, 需要显式push commits
- **远程跟踪分支(remote-tracking branch)** ~ 在本地仓库中的对远程分支所指向的提交(commit)的一个引用
- **上游分支(upstream branch)** ~ 在本地和远程分支之间建立的跟踪关系, 上游分支是本地分支指向的远程分支。
 - **git push/pull**时, 如果上游分支已经设定, 则不需要跟额外参数

命令～推到远程仓库

- ***git push <shortname> <branch_name>***
 - 将本地分支 **<branch_name>** 中的内容推到和 **<shortname>** 对应的远程分支中
 - 步骤：
 - 在远程仓库中创建一个远程分支
 - 在本地仓库创建一个远程跟踪分支
 - 需求：联网
 - `git push -u <shortname> <branch_name>` 设定上游分支
- ***git branch --all***
 - 列出所有本地分支 + 远程跟踪分支

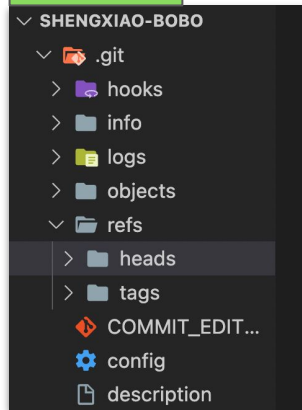


实操 ~ 将main分支推到远程分支

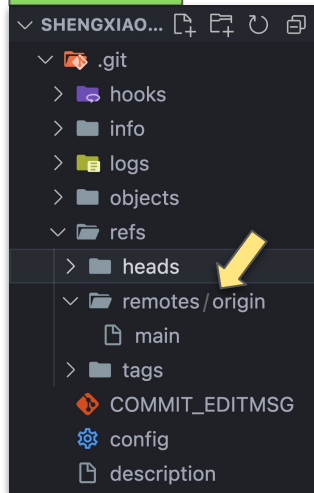
1. git push origin main

```
shengxiao-bobo git:(main) git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 10 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (9/9), 748 bytes | 748.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/boboweike/shengxiao.git
* [new branch]      main -> main
shengxiao-bobo git:(main)
```

push前



push后



2. git branch --all

```
shengxiao-bobo git:(main) git branch --all
* main
remotes/origin/main
(END)
```

3. git log

```
shengxiao-bobo git:(main) git log
commit 9508fc1f62ab16cb32654bf448c9e53f23e5d72 (HEAD -> main, origin/main, feature)
Author: bobo <bobo@boboweike.cn>
Date: Sun Oct 13 15:27:26 2024 +0800

虎

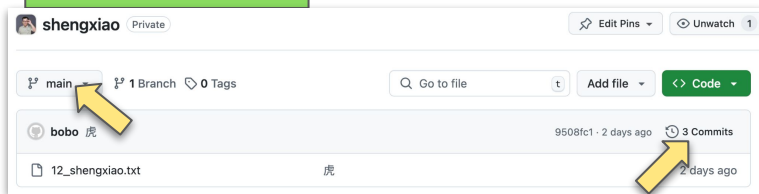
commit 779dcdcfacdf279ce53993d46518a66e29a7007
Author: bobo <bobo@boboweike.cn>
Date: Sun Oct 13 13:23:39 2024 +0800

牛

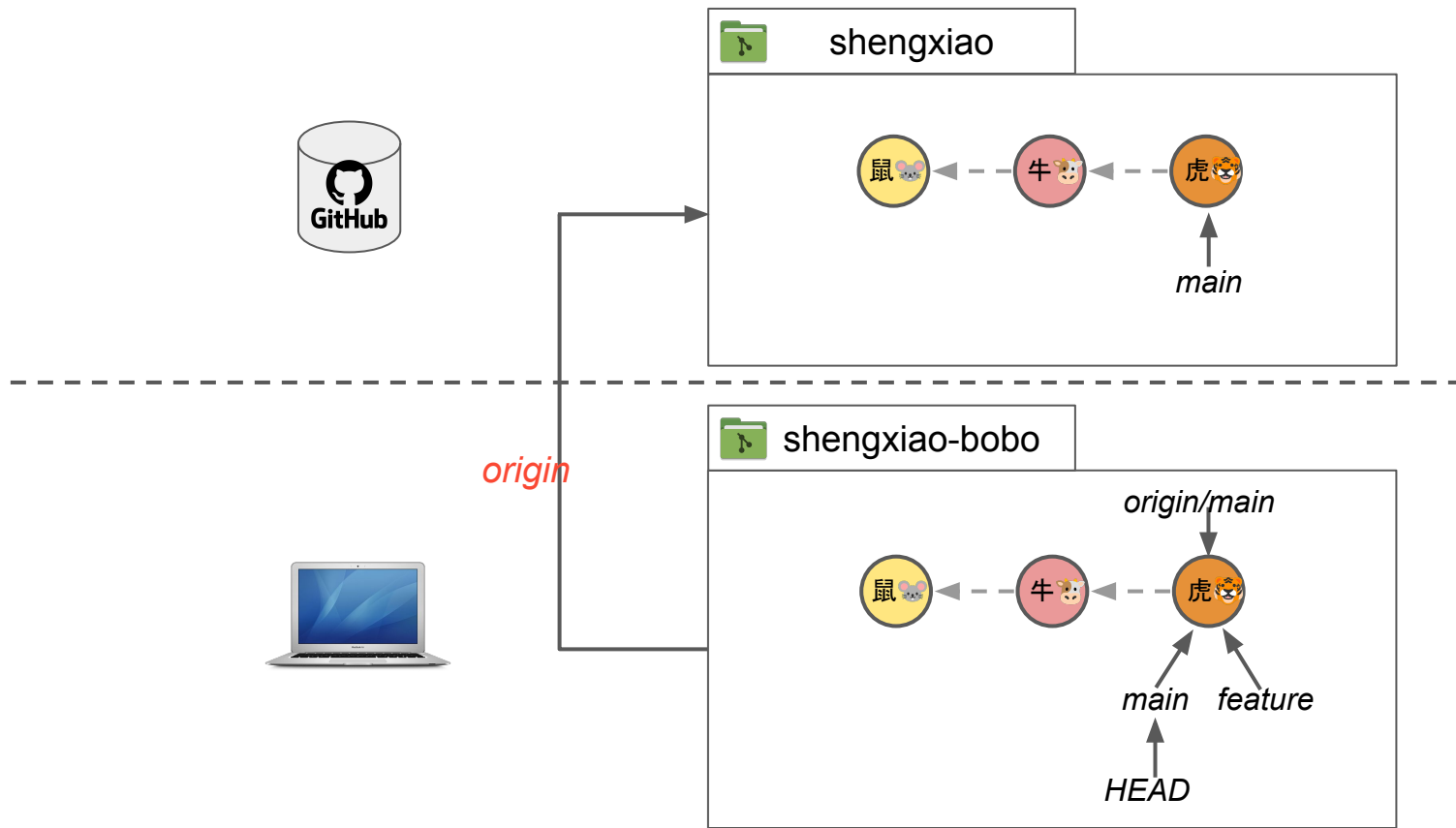
commit 52d496d246c90b0ba13a1b25b5060af2f73dedad
Author: bobo <bobo@boboweike.cn>
Date: Sun Oct 13 09:43:56 2024 +0800

鼠
(END)
```

push后(github)



目前本地和远程仓库情况



实操 ~ 将feature分支推到远程分支

1. git switch feature + git push origin feature

```
shengxiao-bobo — boboweike@bobos-Mac — ..hengxiao-bobo — -zsh — 70x11
→ shengxiao-bobo git:(feature) git push origin feature
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature' on GitHub by visiting:
remote:   https://github.com/boboweike/shengxiao/pull/new/feature
remote:
To https://github.com/boboweike/shengxiao.git
 * [new branch]      feature -> feature
→ shengxiao-bobo git:(feature)
```

2. git branch --all

```
shengxiao-bobo — git branch --all — git — less — git branch --all — 37...
* feature
main
remotes/origin/feature
remotes/origin/main
~
(END)
```

3. git log

```
shengxiao-bobo — git log — git — less — git log — 75x20
commit 9508fc1f62ab16bcb32654bf448c9e53f23e5d72 (HEAD -> feature, origin/main, origin/feature, main)
Author: bobo <bobo@boboweike.cn>
Date:   Sun Oct 13 15:27:26 2024 +0800

    虎

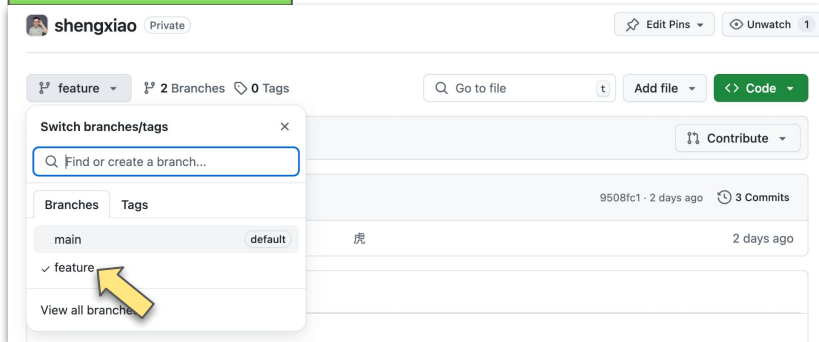
commit 779dcdcfacdf279ce53993d46518a666e29a7007
Author: bobo <bobo@boboweike.cn>
Date:   Sun Oct 13 13:23:39 2024 +0800

    牛

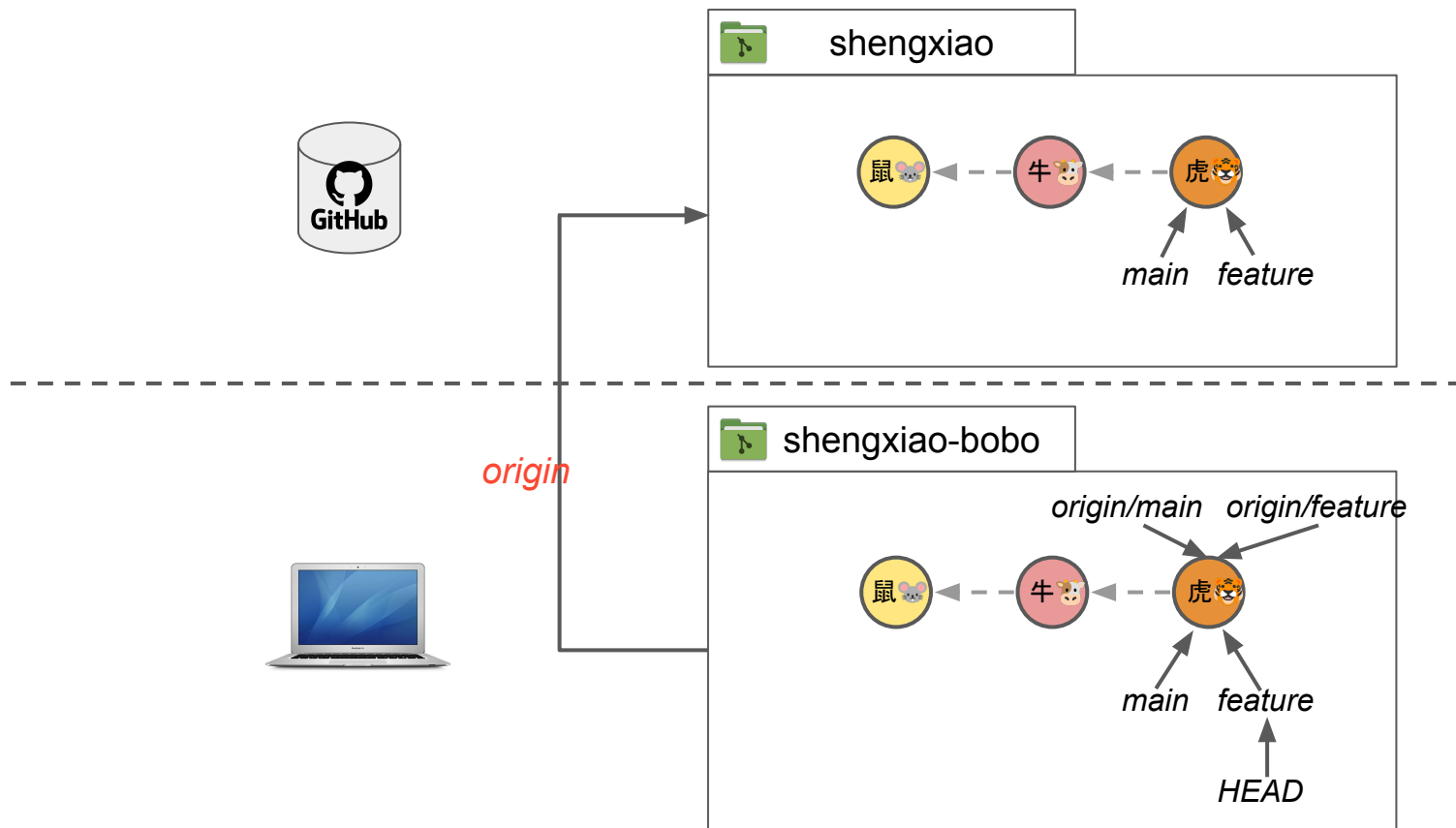
commit 52d496d246c90b0ba13a1b25b5060af2f73dedad
Author: bobo <bobo@boboweike.cn>
Date:   Sun Oct 13 09:43:56 2024 +0800

    鼠
~
(END)
```

push后(github)



目前本地和远程仓库情况

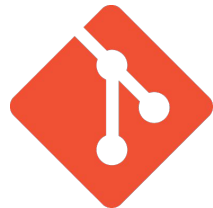


修改远程仓库的两种方式

1. 先修改本地仓库, 再推到远程仓库
 - a. 本课程主要演示命令行方式
2. 直接在GitHub上通过界面修改
 - a. 提交/创建远程分支/通过pull request合并分支
 - b. 具体参考官方文档



小结和预告



- 本课小结

- 什么是远程仓库, 为什么要使用远程仓库
- 使用Git开始项目的两种方式
 - 从本地仓库开始
 - 从远程仓库开始
- 从本地仓库开始的三个步骤
 - 在GitHub上创建一个远程仓库
 - 在本地和远程仓库之间建立连接
 - 将数据从本地仓库推(push)到远程仓库
- 学习了一些新概念
 - 远程分支(remote branch)
 - 远程跟踪分支(remote tracking branch)
 - 上游分支(upstream branch)

- 下一课

- 协作开发, 克隆(cloning)远程仓库到本地