GitHub

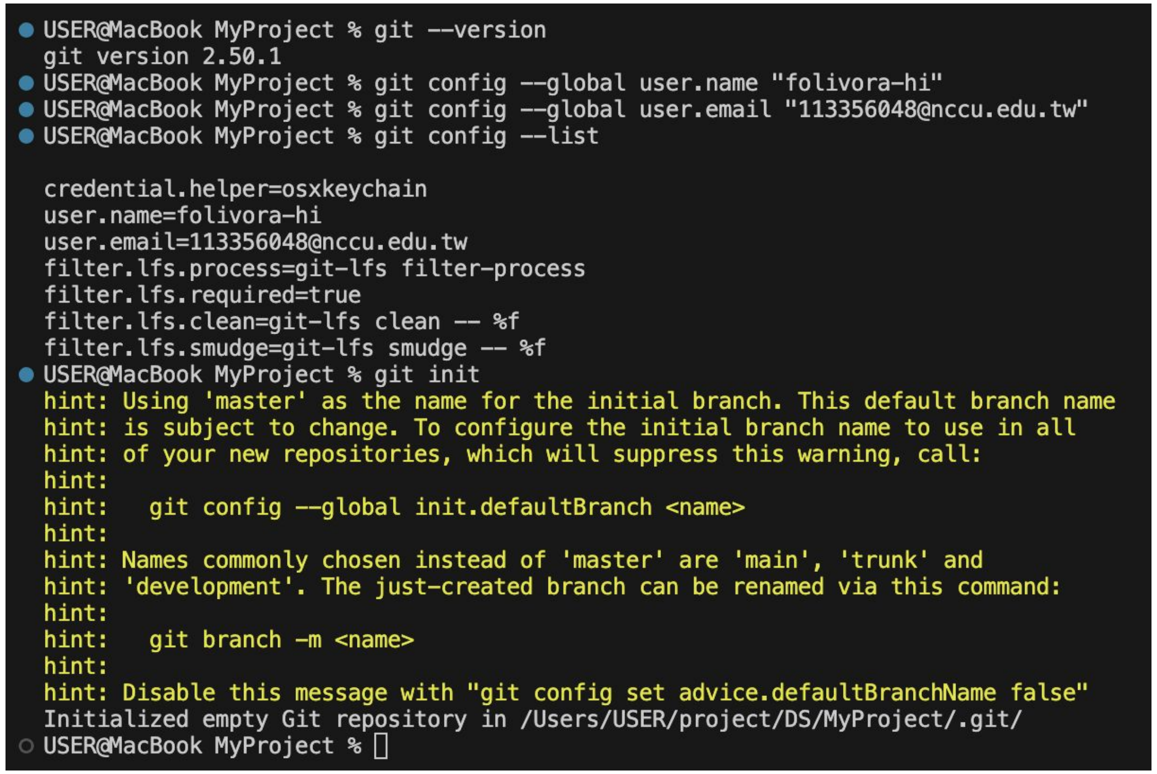
TA
劉亭妤 Joanne

Notices!!

Before the course begins, please register a GitHub account.

0. Preparation

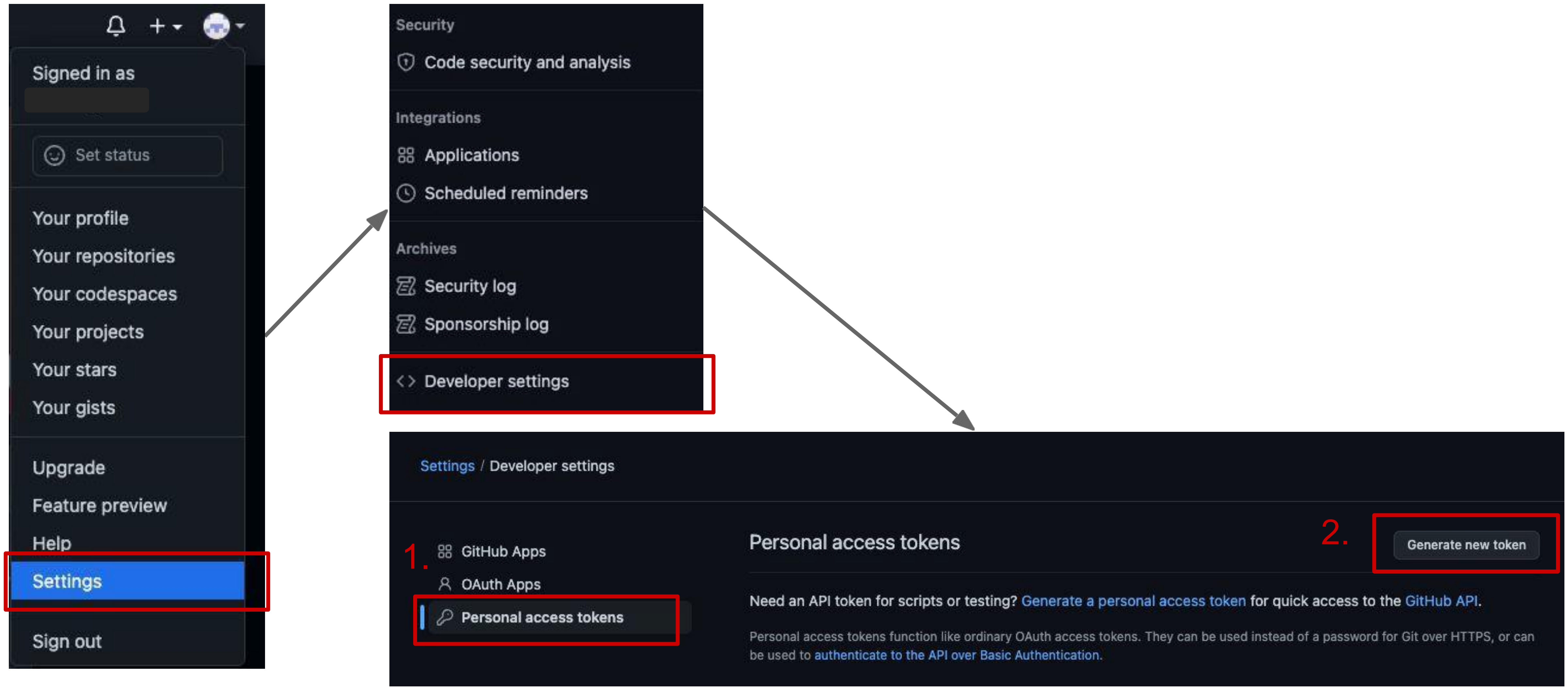
- Install Git and VS Code
- Register a GitHub account
- Configure user information via VScode terminal
  - git config --global user.name "Your Name"
  - git config --global user.email "you@example.com"
  - (check configuration) git config --global --list
  - git init



```
● USER@MacBook MyProject % git --version
  git version 2.50.1
● USER@MacBook MyProject % git config --global user.name "folivora-hi"
● USER@MacBook MyProject % git config --global user.email "113356048@nccu.edu.tw"
● USER@MacBook MyProject % git config --list

  credential.helper=osxkeychain
  user.name=folivora-hi
  user.email=113356048@nccu.edu.tw
  filter.lfs.process=git-lfs filter-process
  filter.lfs.required=true
  filter.lfs.clean=git-lfs clean -- %f
  filter.lfs.smudge=git-lfs smudge -- %f
● USER@MacBook MyProject % git init
  hint: Using 'master' as the name for the initial branch. This default branch name
  hint: is subject to change. To configure the initial branch name to use in all
  hint: of your new repositories, which will suppress this warning, call:
  hint:
  hint:   git config --global init.defaultBranch <name>
  hint:
  hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
  hint: 'development'. The just-created branch can be renamed via this command:
  hint:
  hint:   git branch -m <name>
  hint:
  hint: Disable this message with "git config set advice.defaultBranchName false"
  Initialized empty Git repository in /Users/USER/project/DS/MyProject/.git/
○ USER@MacBook MyProject %
```

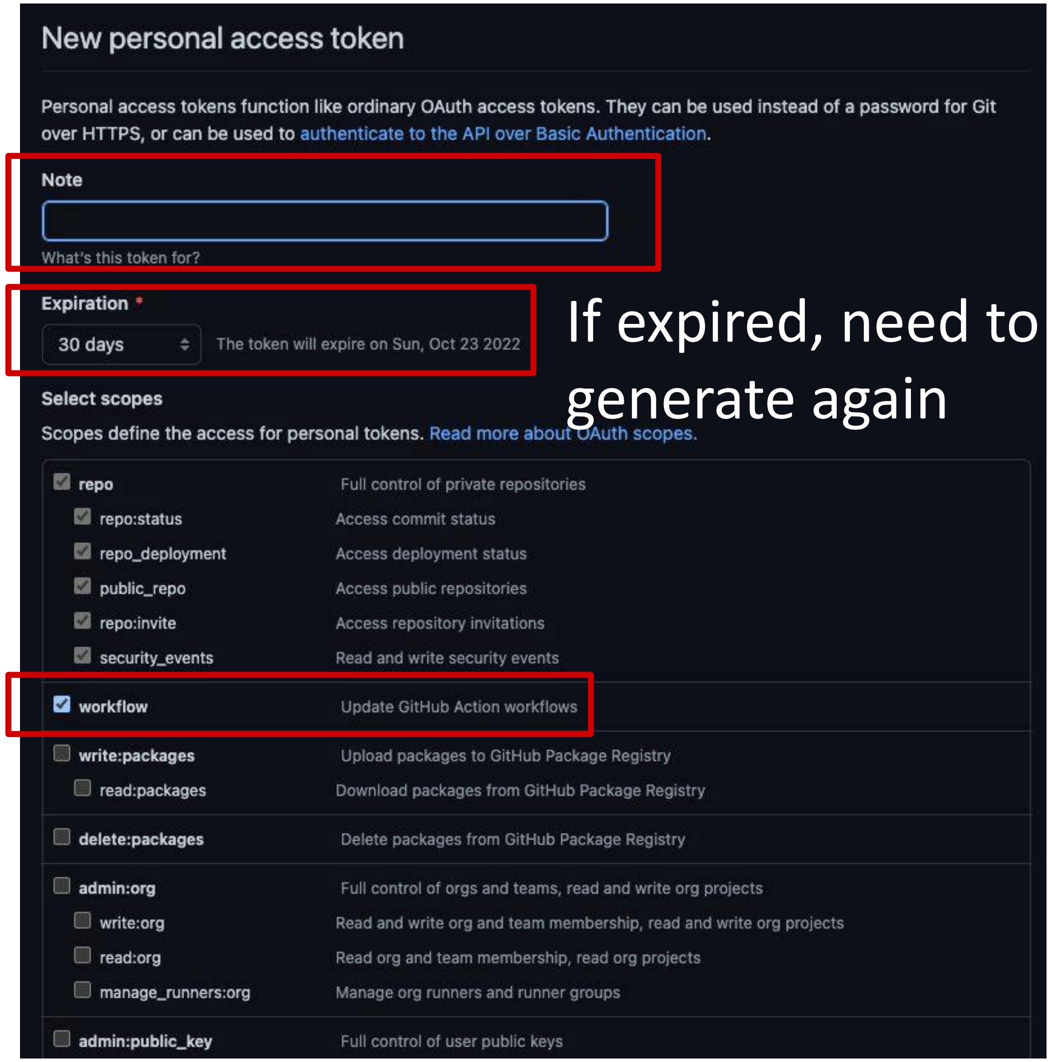
2. Get the access token : PAT key



Need an API token for scripts or testing? Generate a personal access token for quick access to the GitHub API.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

1. Get the access token : PAT key



New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

What's this token for?

Expiration *

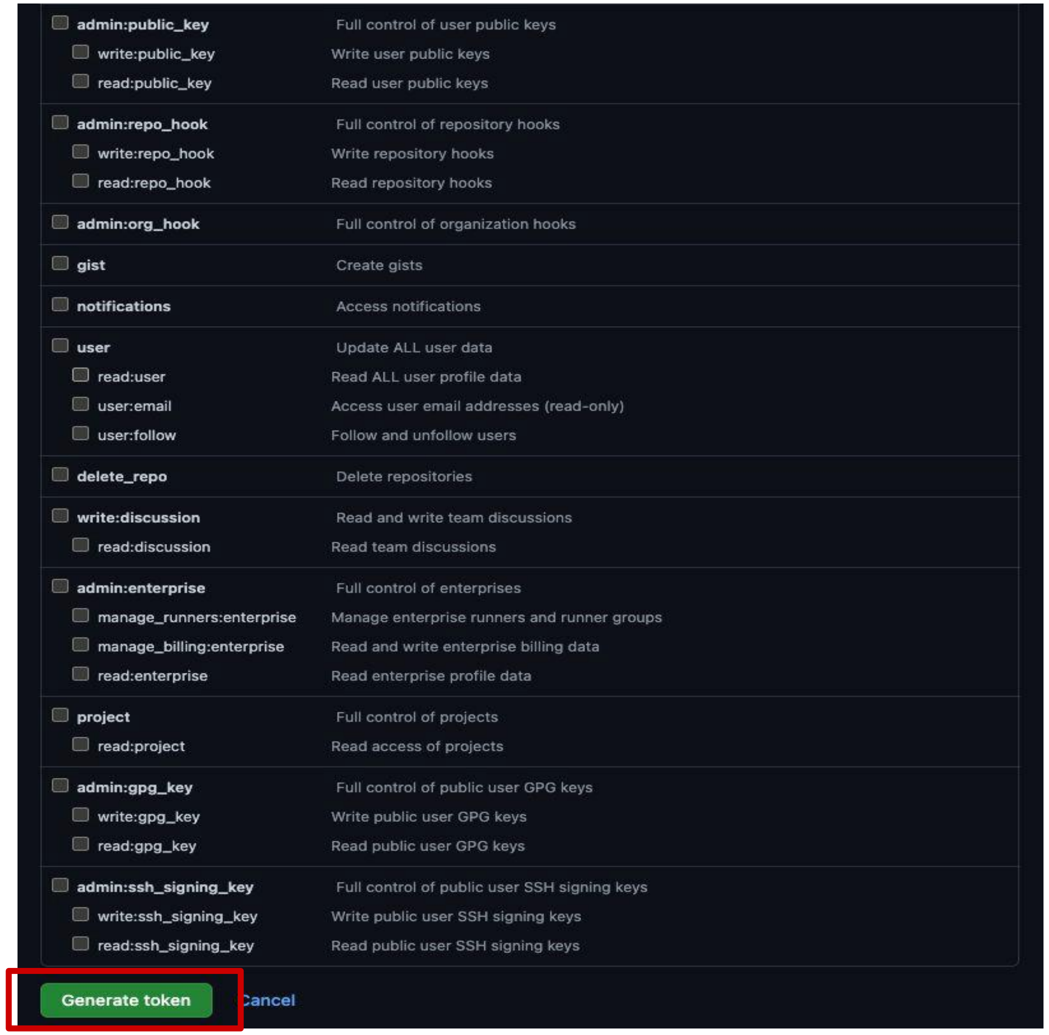
30 days — The token will expire on Sun, Oct 23 2022

If expired, need to generate again

Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

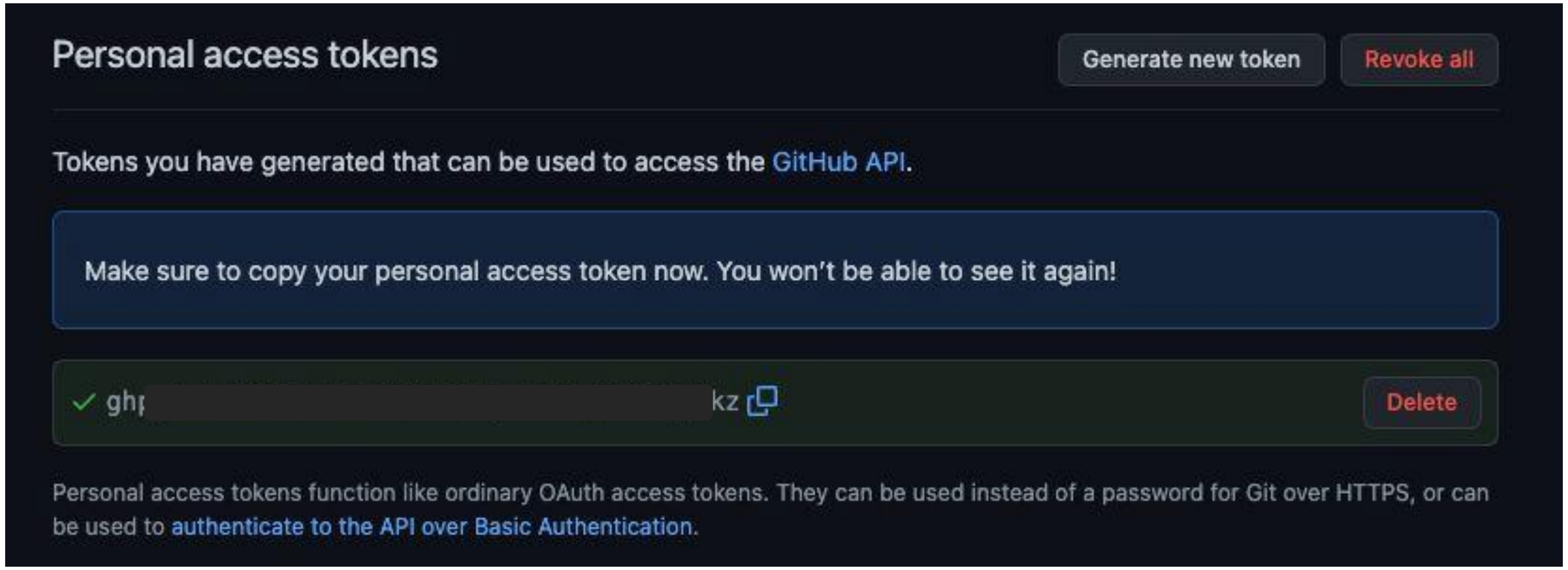
☑ repo	Full control of private repositories
☑ repo:status	Access commit status
☑ repo_deployment	Access deployment status
☑ public_repo	Access public repositories
☑ repo:invite	Access repository invitations
☑ security_events	Read and write security events
☑ workflow	Update GitHub Action workflows
☐ write:packages	Upload packages to GitHub Package Registry
☐ read:packages	Download packages from GitHub Package Registry
☐ delete:packages	Delete packages from GitHub Package Registry
☐ admin:org	Full control of orgs and teams, read and write org projects
☐ write:org	Read and write org and team membership, read and write org projects
☐ read:org	Read org and team membership, read org projects
☐ manage_runners:org	Manage org runners and runner groups
☐ admin:public_key	Full control of user public keys



☐ admin:public_key	Full control of user public keys
☐ write:public_key	Write user public keys
☐ read:public_key	Read user public keys
☐ admin:repo_hook	Full control of repository hooks
☐ write:repo_hook	Write repository hooks
☐ read:repo_hook	Read repository hooks
☐ admin:org_hook	Full control of organization hooks
☐ gist	Create gists
☐ notifications	Access notifications
☐ user	Update ALL user data
☐ read:user	Read ALL user profile data
☐ user:email	Access user email addresses (read-only)
☐ user:follow	Follow and unfollow users
☐ delete_repo	Delete repositories
☐ write:discussion	Read and write team discussions
☐ read:discussion	Read team discussions
☐ admin:enterprise	Full control of enterprises
☐ manage_runners:enterprise	Manage enterprise runners and runner groups
☐ manage_billing:enterprise	Read and write enterprise billing data
☐ read:enterprise	Read enterprise profile data
☐ project	Full control of projects
☐ read:project	Read access of projects
☐ admin:gpg_key	Full control of public user GPG keys
☐ write:gpg_key	Write public user GPG keys
☐ read:gpg_key	Read public user GPG keys
☐ admin:ssh_signing_key	Full control of public user SSH signing keys
☐ write:ssh_signing_key	Write public user SSH signing keys
☐ read:ssh_signing_key	Read public user SSH signing keys

1. Get the access token : PAT key

- Make sure to copy your access token to your note!
- This will use when you want to push code to GitHub.
- (let environment remember your key) git config --global credential.helper manager



Personal access tokens

Tokens you have generated that can be used to access the GitHub API.

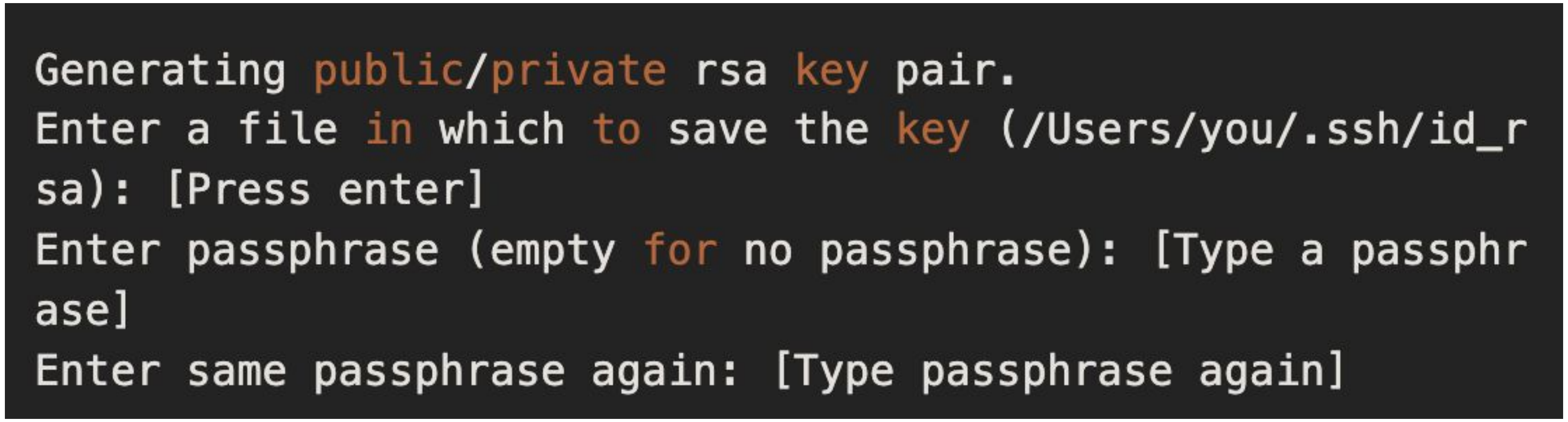
Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp[redacted]kz

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

1. Get the access token : SSH key

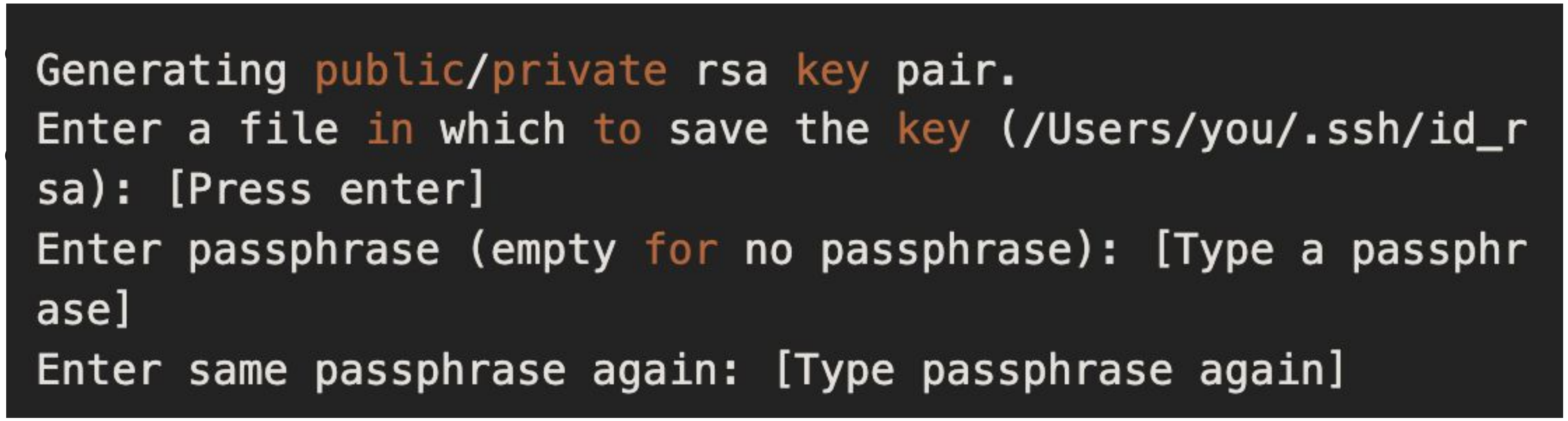
- terminal
  - ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
  - Enter your passphrase twice



```
Generating public/private rsa key pair.
Enter a file in which to save the key (/Users/you/.ssh/id_r
sa): [Press enter]
Enter passphrase (empty for no passphrase): [Type a passphr
ase]
Enter same passphrase again: [Type passphrase again]
```

1. Get the access token : SSH key

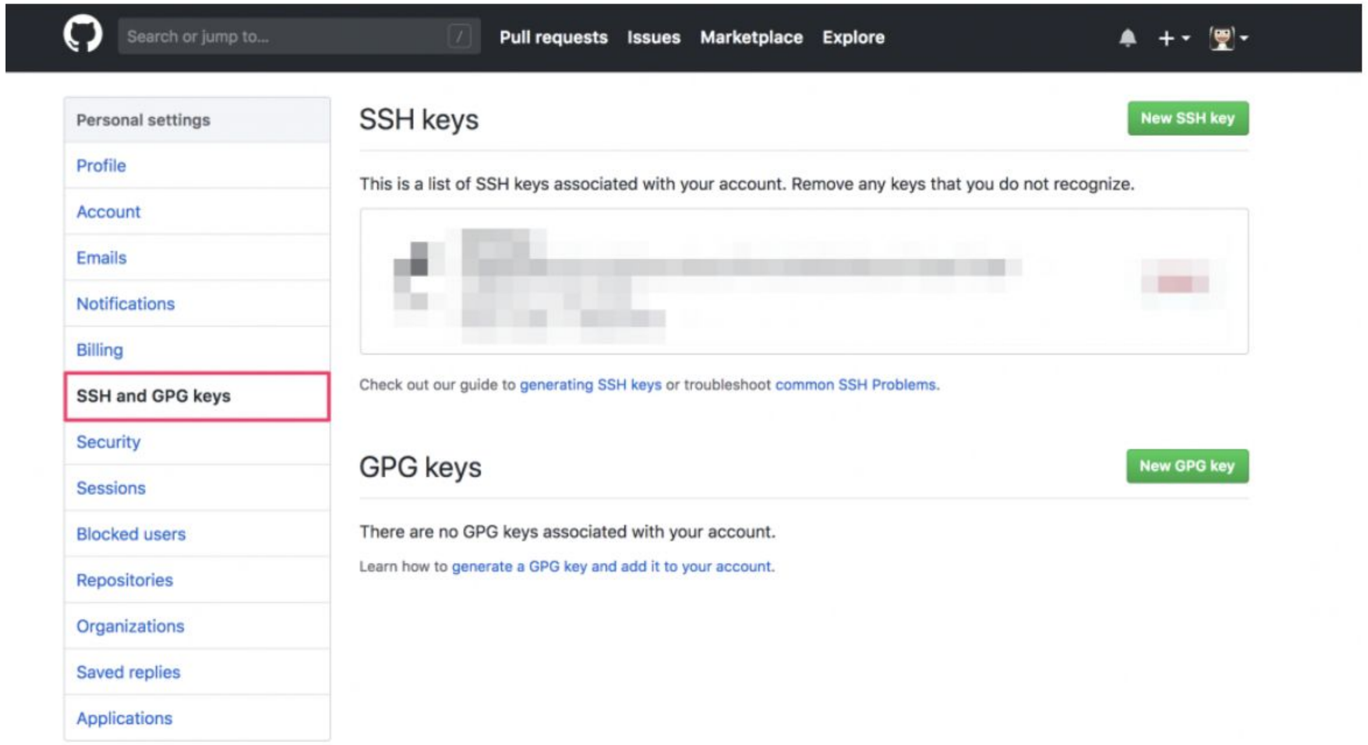
- terminal
  - (generate ssh key) ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
  - Enter your passphrase twice
  - (open ssh agent) eval "$(ssh-agent -s)"
  - (add key to agent) ssh-add -K ~/.ssh/id_rsa



```
Generating public/private rsa key pair.
Enter a file in which to save the key (/Users/you/.ssh/id_r
sa): [Press enter]
Enter passphrase (empty for no passphrase): [Type a passphr
ase]
Enter same passphrase again: [Type passphrase again]
```

1. Get the access token : SSH key

- terminal
  - (show ssh key) cat ~/.ssh/id_rsa.pub
  - copy the key
- GitHub
  - register key



SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Check out our guide to generating SSH keys or troubleshoot common SSH Problems.

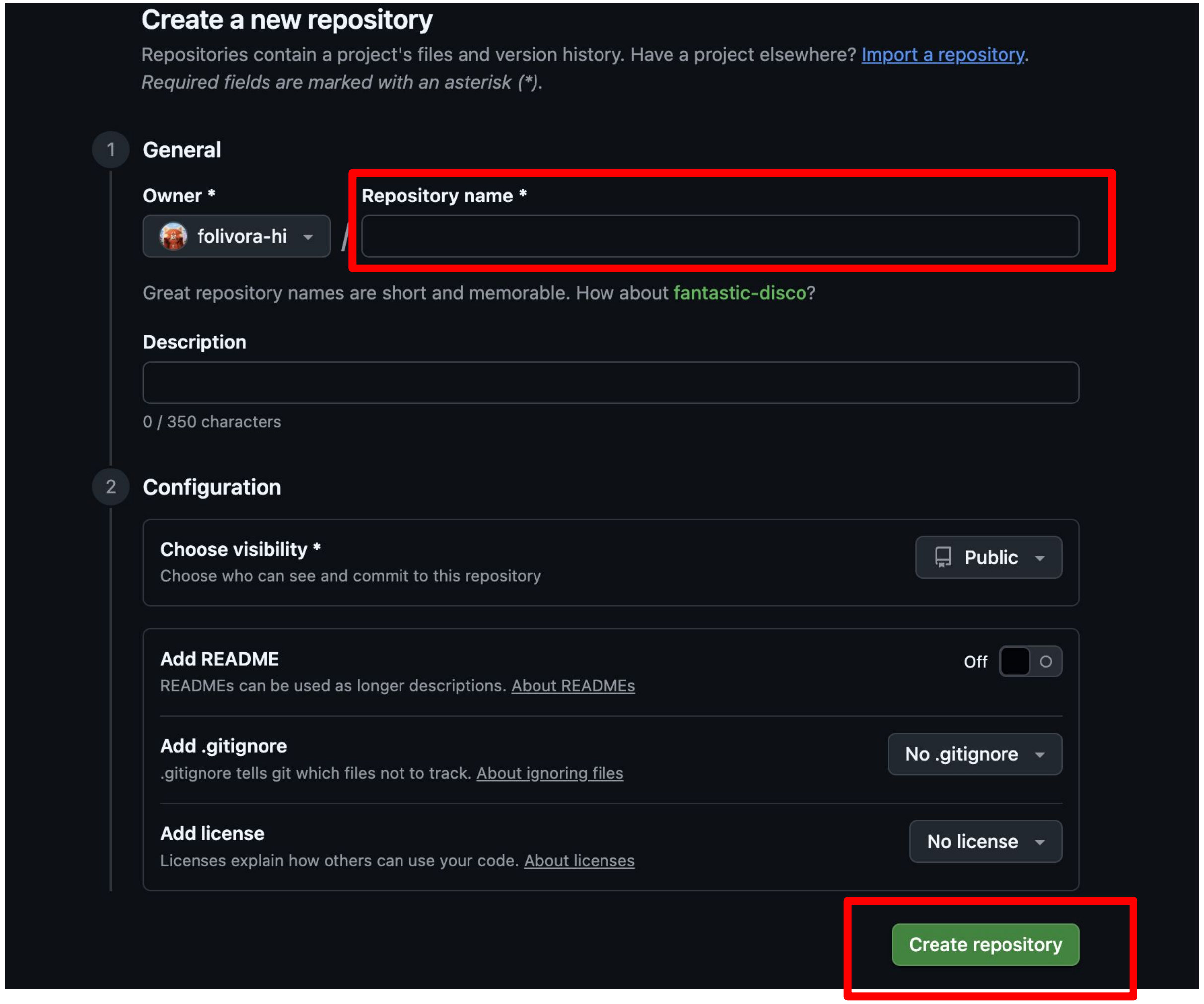
GPG keys

There are no GPG keys associated with your account.

Learn how to generate a GPG key and add it to your account.

2. Create a Project

- Login GitHub (https://github.com/)
- New a Repository
    - git remote add origin git@github.com:{user.name}/{repo.name}.git
    - git branch -M main
    - git push -u origin main



Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? Import a repository.
Required fields are marked with an asterisk (*).

1 General

Owner * folivora-hi / Repository name *

Great repository names are short and memorable. How about fantastic-disco?

Description

0 / 350 characters

2 Configuration

Choose visibility * — Public
Choose who can see and commit to this repository

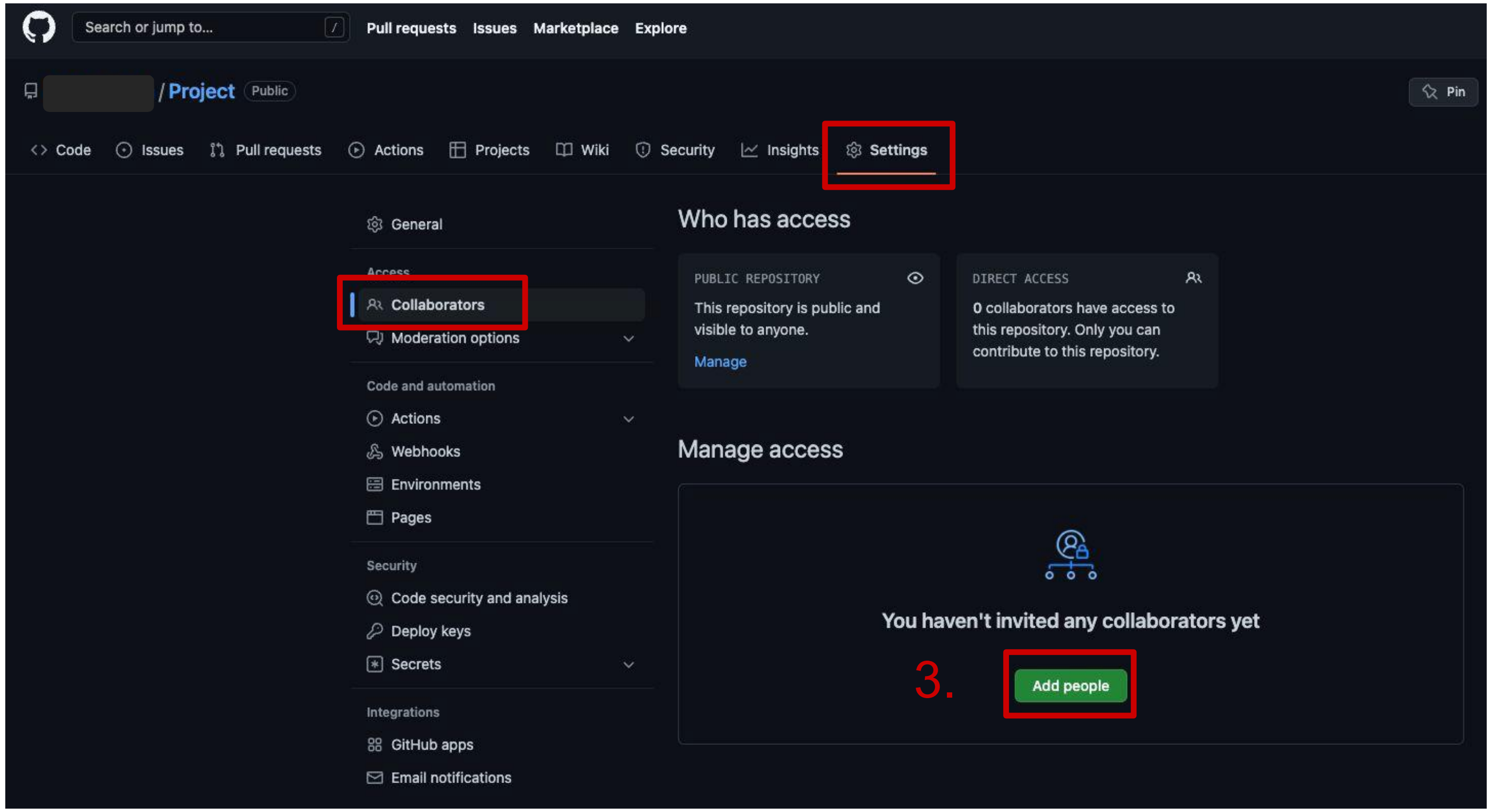
Add README — Off
READMEs can be used as longer descriptions. About READMEs

Add .gitignore — No .gitignore
.gitignore tells git which files not to track. About ignoring files

Add license — No license
Licenses explain how others can use your code. About licenses

2. Create a Project

- Add your members to repository


Search or jump to... /
Pull requests Issues Marketplace Explore

[redacted] / Project Public

Pin

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Actions

Webhooks

Environments

Pages

Security

Code security and analysis

Deploy keys

Secrets

Integrations

GitHub apps

Email notifications

Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

Manage

DIRECT ACCESS

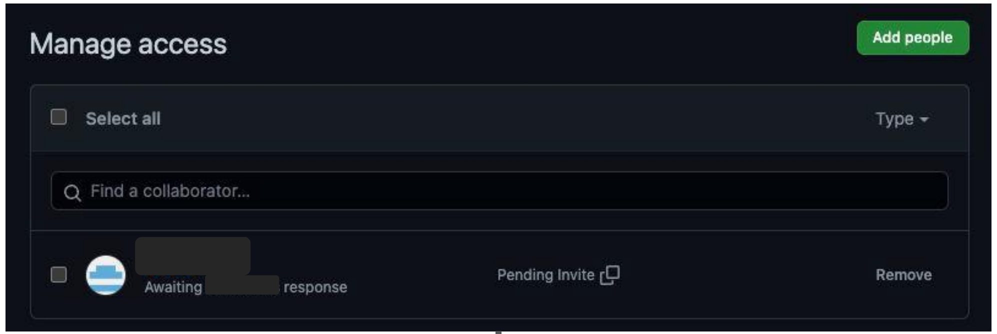
0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

You haven't invited any collaborators yet

3.

Add people


Manage access

Add people

Select all

Type

Find a collaborator...

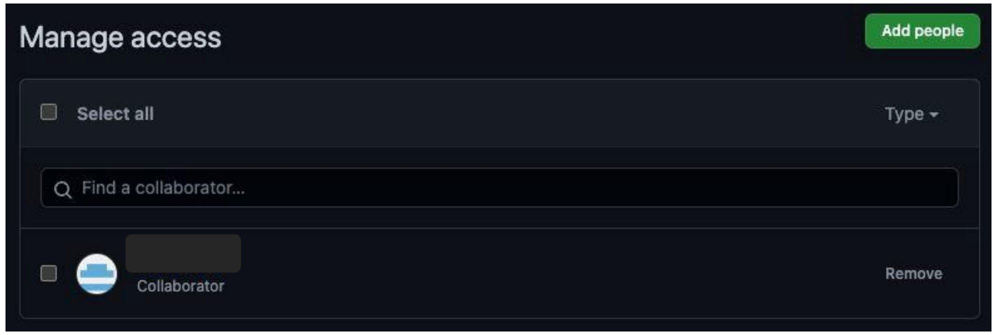
[redacted]

Awaiting [redacted] response

Pending Invite

Remove

After your members join


Manage access

Add people

Select all

Type

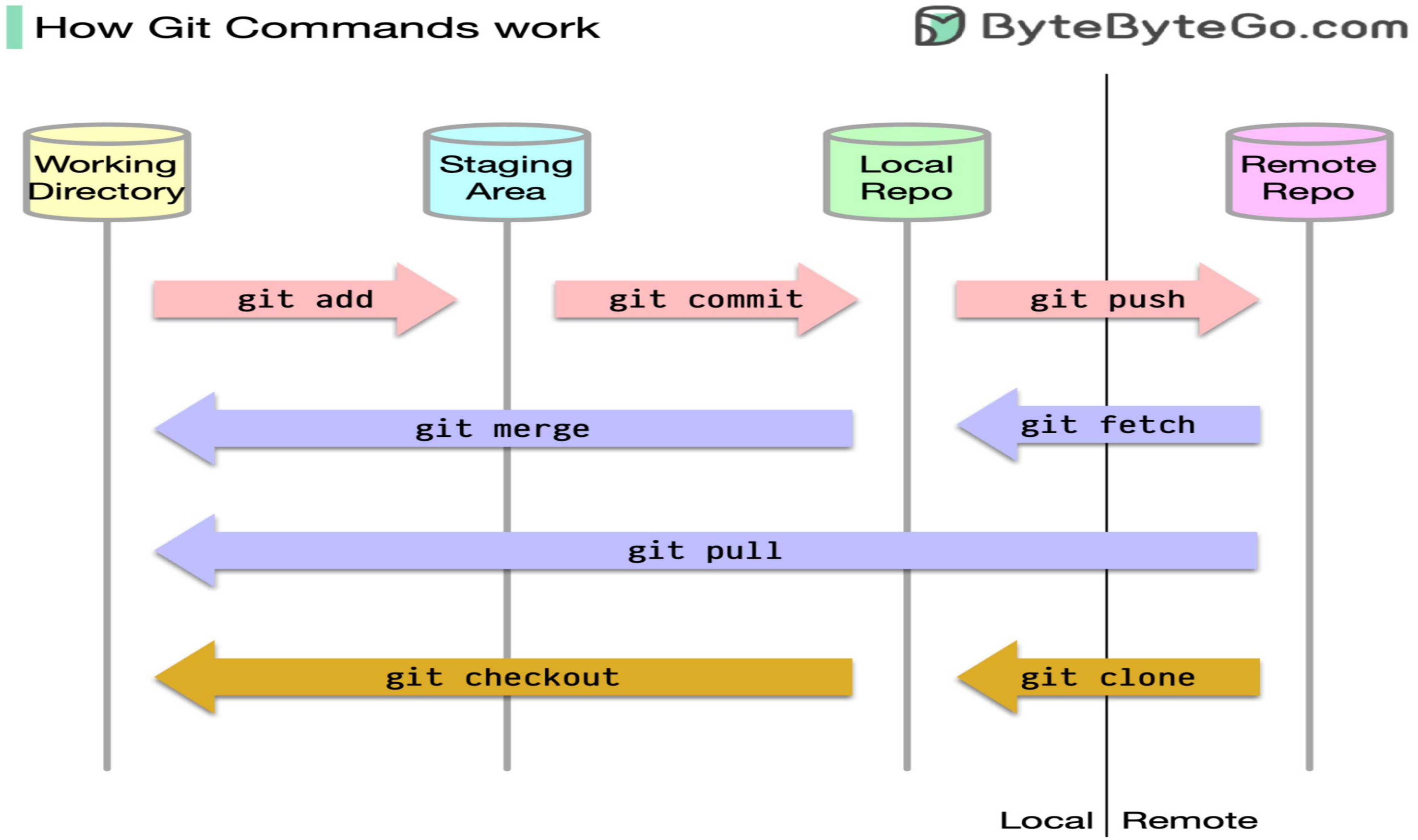
Find a collaborator...

[redacted]

Collaborator

Remove

3. Git Command : Basic

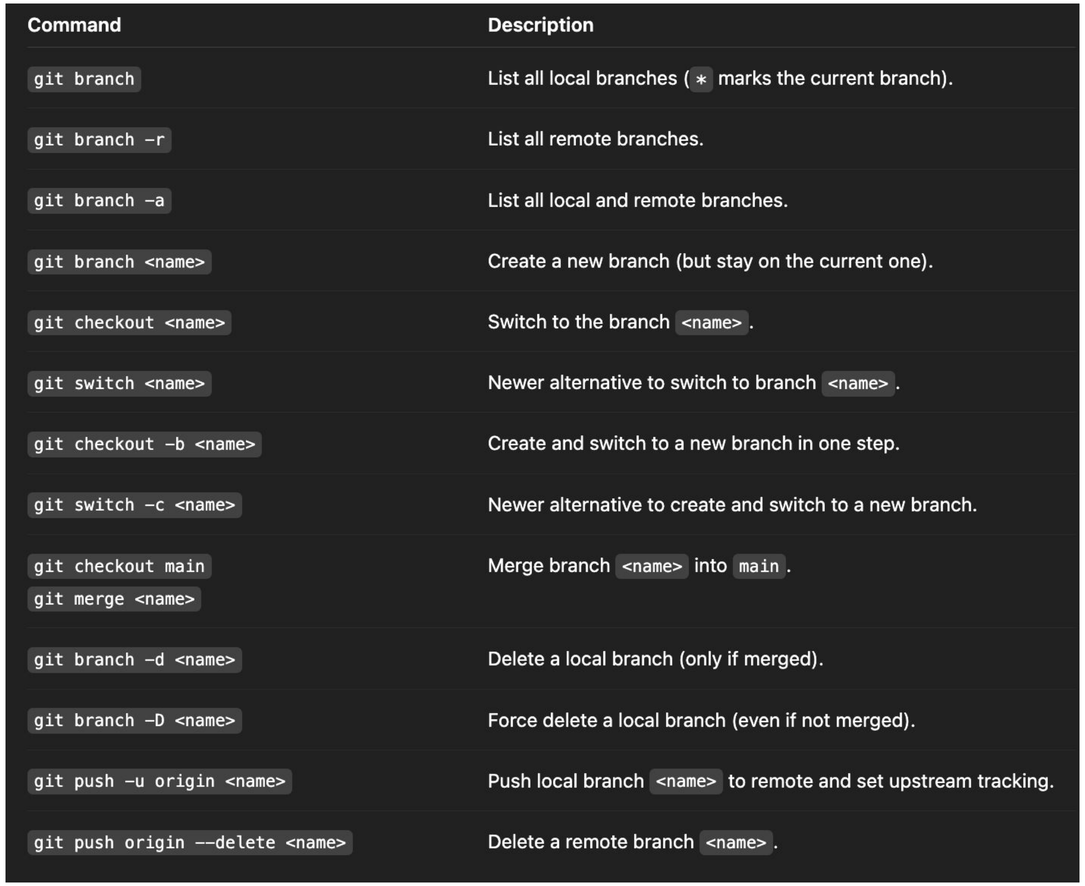


3. Git Collaboration

- Always work on a branch.
- Keep main synced with remote.
  - Always git pull origin main before creating a new branch to ensure it is based on the latest code.
- Clear Commit Messages.
  - Avoid vague messages like "fix bug." Be descriptive, e.g., fix: resolve header alignment issue on mobile.
- Don’t Upload Unnecessary Files.
  - Use .gitignore to exclude compiled files, temp files, and environment configs.
- Merge via Pull Requests (with reviews).

3. Git Collaboration : Branch

Never develop directly on main. Always create a new branch for features or fixes, e.g., feature/login or bugfix/header.



Command	Description
`git branch`	List all local branches (`*` marks the current branch).
`git branch -r`	List all remote branches.
`git branch -a`	List all local and remote branches.
`git branch <name>`	Create a new branch (but stay on the current one).
`git checkout <name>`	Switch to the branch `<name>`.
`git switch <name>`	Newer alternative to switch to branch `<name>`.
`git checkout -b <name>`	Create and switch to a new branch in one step.
`git switch -c <name>`	Newer alternative to create and switch to a new branch.

`git checkout main`
`git merge <name>`
	Merge branch `<name>` into `main`.

`git branch -d <name>`	Delete a local branch (only if merged).
`git branch -D <name>`	Force delete a local branch (even if not merged).
`git push -u origin <name>`	Push local branch `<name>` to remote and set upstream tracking.
`git push origin --delete <name>`	Delete a remote branch `<name>`.

3. Git Collaboration : Merge & Conflict

Don’t merge directly. Use Pull Requests, get code reviewed, and then merge into main.

[Link] https://ithelp.ithome.com.tw/articles/10339487

When merge conflicts occur, don’t force push. Resolve locally, test, then push again.

[Link] https://heidiliu2020.github.io/git-github/

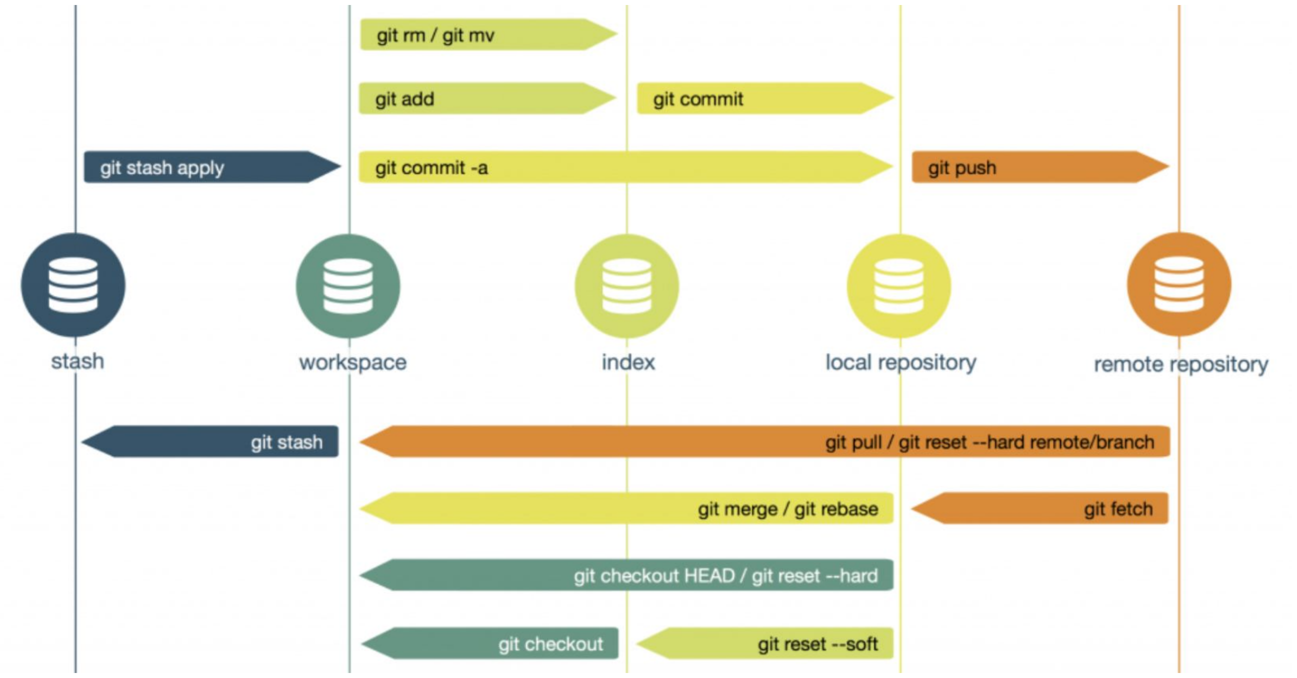
Notice

- Send your GitHub link and contact information via Google Sheets!

  [Link]

Supplement

- Basic Git & GitHub Concept : https://www.youtube.com/watch?v=FKXRiAiQFiY
- Git Stash (暫存) : https://www.maxlist.xyz/2018/11/02/git_tutorial/
- https://www.youtube.com/watch?v=VShhhq_5sMc&list=PLBd8JGCAcUAF2_im__k ZTfEAKnlmfPJy


git rm / git mv
git add
git commit
git stash apply
git commit -a
git push
stash
workspace
index
local repository
remote repository
git stash
git pull / git reset --hard remote/branch
git merge / git rebase
git fetch
git checkout HEAD / git reset --hard
git checkout
git reset --soft