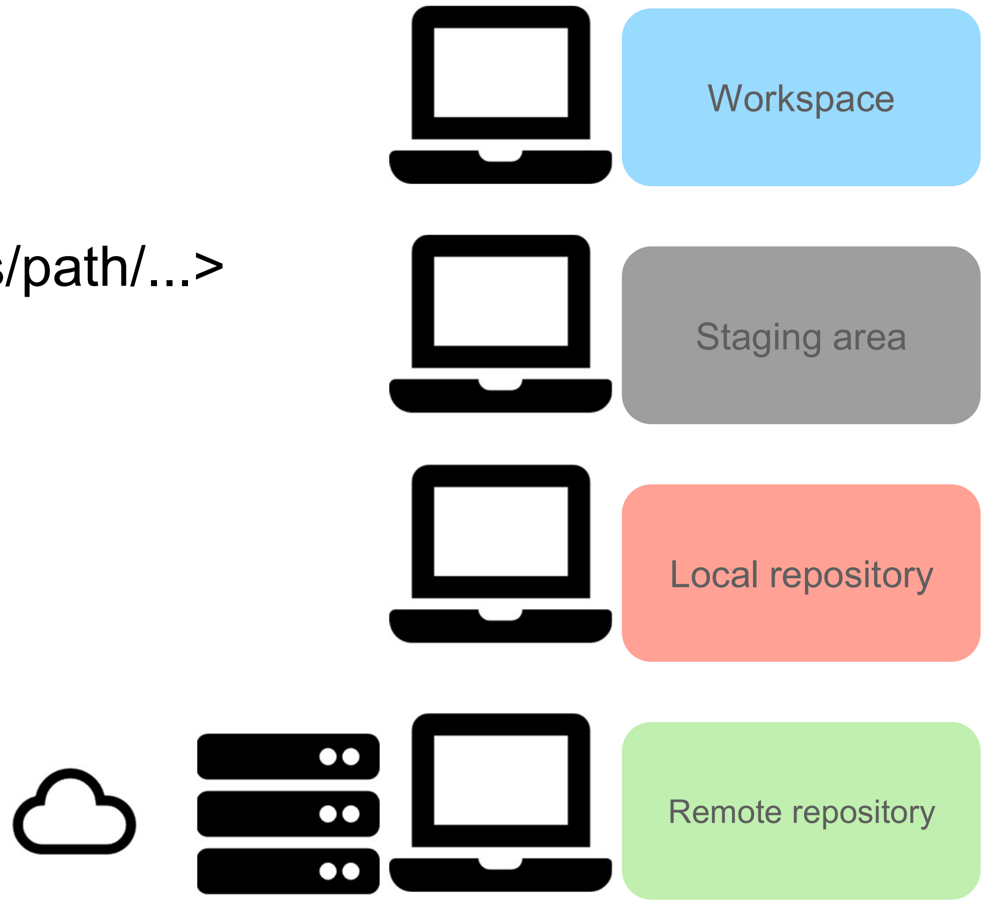
Module 3

Remote version control

IEECR
INSTITUTE OF EXPERIMENTAL EPILEPTOLOGY AND COGNITION RESEARCH

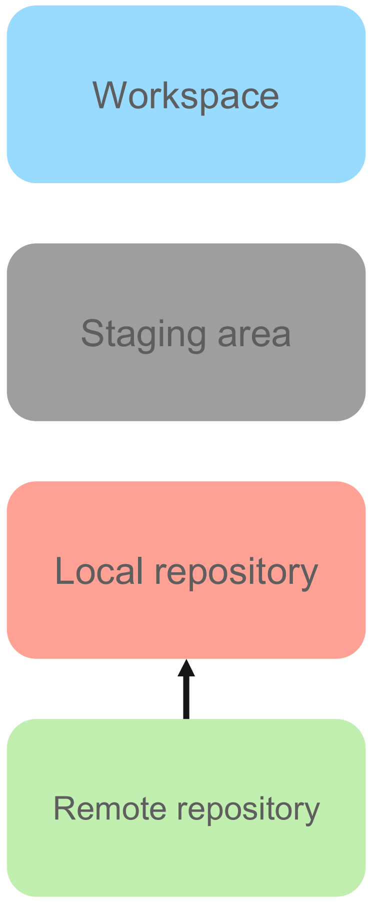
Where is my code?

- Remote ≠ somewhere distant
- git remote add <alias> <remote address/path/...>
- Github:
  - Owned by Microsoft (2018)
  - Code in the cloud
  - Security, collaboration features
  - Integration with Microsoft tools (VSC etc.)



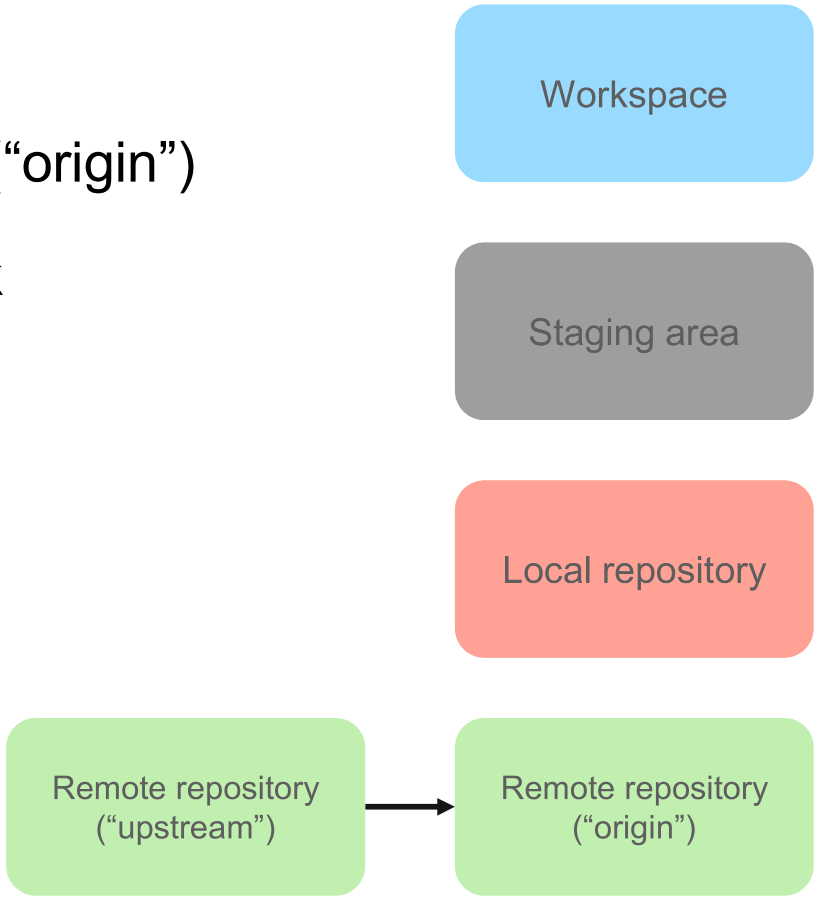
Cloning

- Copy a remote into a new local directory
- Use once (set up local repository)
- Automatic connection (alias) to original remote: “origin”
- Changes in remote not automatically synchronized by git!
    - git pull/fetch, VSC auto-fetch…



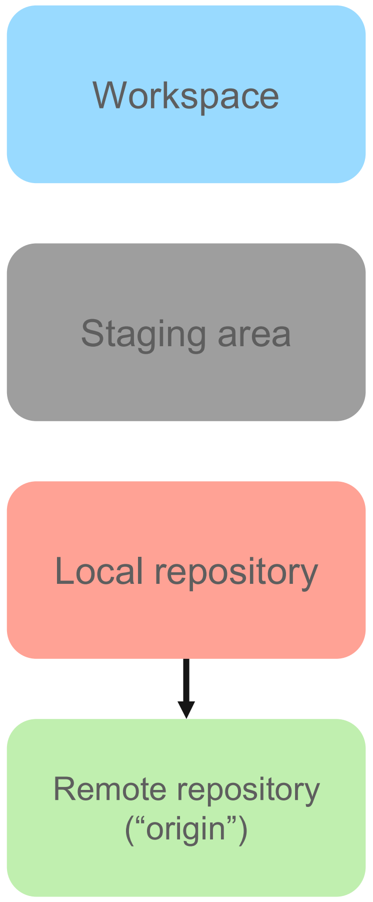
GitHub forking

- Copy a remote (“upstream”) as a new remote (“origin”)
- If forking public repo: you are owner of the fork
  - <u>Private</u> upstream owner is (also) owner!
- Use once
- Ideal for personal experimentation
- To apply your changes: GitHub pull requests
  - (upstream owner has to review, accept)



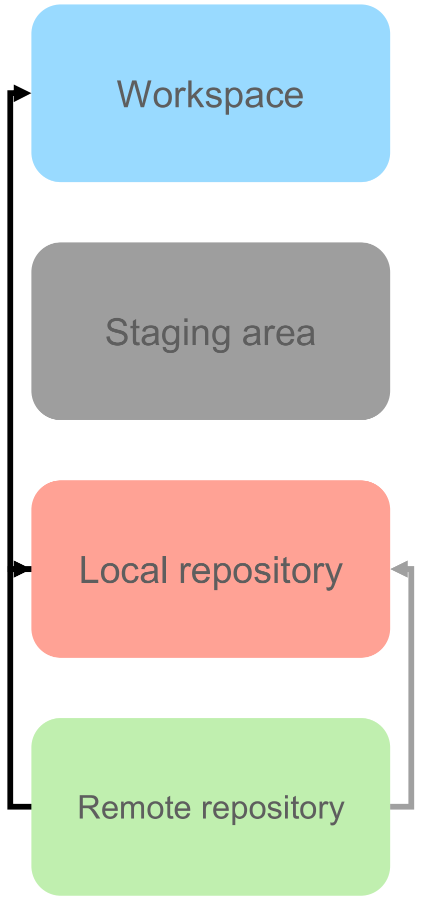
Push

- Move changes in local repository (commits) to remote
- “git push <remote repository> <branch>”
  - e.g. git push origin master:
    push changes in local master branch to its counterpart in “origin” (mostly also master)



Pull & fetch

- pull = fetch + merge
  - Import (“fetch”) possible changes, synchronize remote with local repository
  - Try to apply these changes (“merge”)
- Different from pull requests (GitHub)
- Recommended to separate:
  - “git fetch origin” (VSC can automatize it), then
  - (“git status”)
  - “git merge”



Module 3

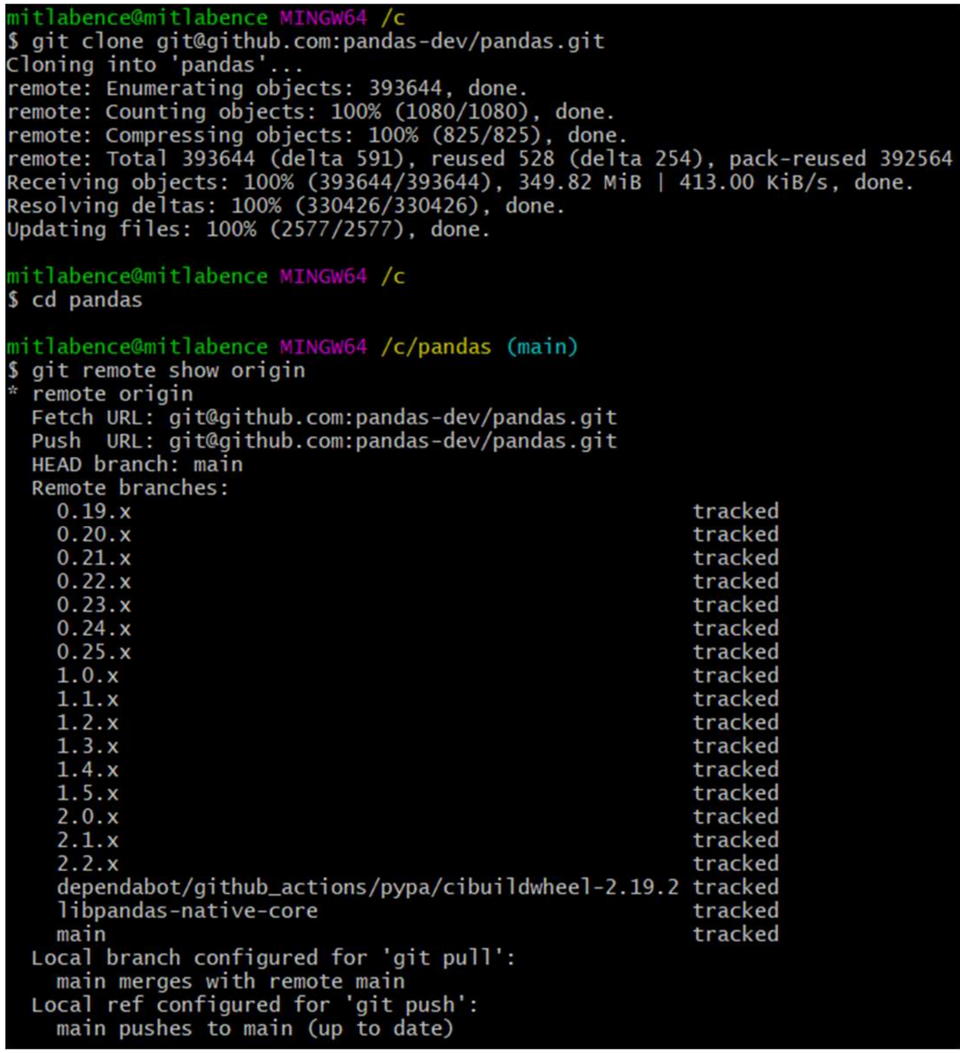
Extra slides

IEECR

INSTITUTE OF EXPERIMENTAL EPILEPTOLOGY AND COGNITION RESEARCH

Branches

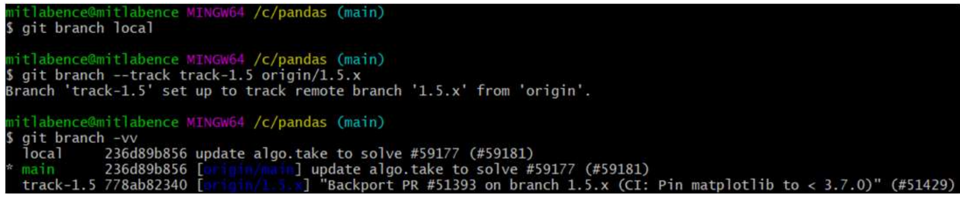
- Pointer to a commit in the chain (tree)
- Cloning: creates local main branch
  - Tracks origin/main remote branch
- Local branches:
  - Local non-tracking (e.g. *local*)
  - Local tracking (e.g. *track-1.5*)
- Remote-tracking branches:
  - Local information about remote branches
  - View: git branch –r
  - Update information: git fetch
- Remote branches:
  - View: git remote show <remote>



```
mitlabence@mitlabence MINGW64 /c
$ git clone git@github.com:pandas-dev/pandas.git
Cloning into 'pandas'...
remote: Enumerating objects: 393644, done.
remote: Counting objects: 100% (1080/1080), done.
remote: Compressing objects: 100% (825/825), done.
remote: Total 393644 (delta 591), reused 528 (delta 254), pack-reused 392564
Receiving objects: 100% (393644/393644), 349.82 MiB | 413.00 KiB/s, done.
Resolving deltas: 100% (330426/330426), done.
Updating files: 100% (2577/2577), done.

mitlabence@mitlabence MINGW64 /c
$ cd pandas

mitlabence@mitlabence MINGW64 /c/pandas (main)
$ git remote show origin
* remote origin
  Fetch URL: git@github.com:pandas-dev/pandas.git
  Push  URL: git@github.com:pandas-dev/pandas.git
  HEAD branch: main
  Remote branches:
    0.19.x                                             tracked
    0.20.x                                             tracked
    0.21.x                                             tracked
    0.22.x                                             tracked
    0.23.x                                             tracked
    0.24.x                                             tracked
    0.25.x                                             tracked
    1.0.x                                              tracked
    1.1.x                                              tracked
    1.2.x                                              tracked
    1.3.x                                              tracked
    1.4.x                                              tracked
    1.5.x                                              tracked
    2.0.x                                              tracked
    2.1.x                                              tracked
    2.2.x                                              tracked
    dependabot/github_actions/pypa/cibuildwheel-2.19.2 tracked
    libpandas-native-core                              tracked
    main                                               tracked
  Local branch configured for 'git pull':
    main merges with remote main
  Local ref configured for 'git push':
    main pushes to main (up to date)
```



```
mitlabence@mitlabence MINGW64 /c/pandas (main)
$ git branch local

mitlabence@mitlabence MINGW64 /c/pandas (main)
$ git branch --track track-1.5 origin/1.5.x
Branch 'track-1.5' set up to track remote branch '1.5.x' from 'origin'.

mitlabence@mitlabence MINGW64 /c/pandas (main)
$ git branch -vv
  local     236d89b856 update algo.take to solve #59177 (#59181)
* main      236d89b856 [origin/main] update algo.take to solve #59177 (#59181)
  track-1.5 778ab82340 [origin/1.5.x] "Backport PR #51393 on branch 1.5.x (CI: Pin matplotlib to < 3.7.0)" (#51429)
```

Branches

Creation, maneuvering

- git checkout: Swiss army knife: create new branch, change branch, move to previous commit...
- git switch <branch>: switch to local <branch> (-c: create new branch)
- git branch –d <branch>: delete branch

Remote repository - locally

Check behaviour without risking code

- Create folder <remote-repository-folder>, e.g. C:/repos/test-remote
- cd inside, git init –bare
- Create folder <local-repository-folder>, e.g. C:/repos/test-local
- cd inside, git clone <remote-repository-folder>, or
  - git init
  - git remote add origin <remote-repository-folder>
- Create simple initial commit, then git push origin master
- Useful commands (Linux, Windows WSL, Git CLI; MacOS?)
  - cd folder_to_enter, cd ..
  - ls (small L): ls –l, ls –a
  - echo “some text” > some_file.txt
  - cat some_file.txt
  - mkdir new_folder
  - rm file_to_remove, rm –r folder_to_remove

git diff

- Show difference between two versions of file
  - git diff <file-name>: see changes between unstaged (workspace) file and staged (committed) file
  - git diff master origin/master: show all changes between local master and remote master
  - git diff branch1 branch2: show all changes between two branches...