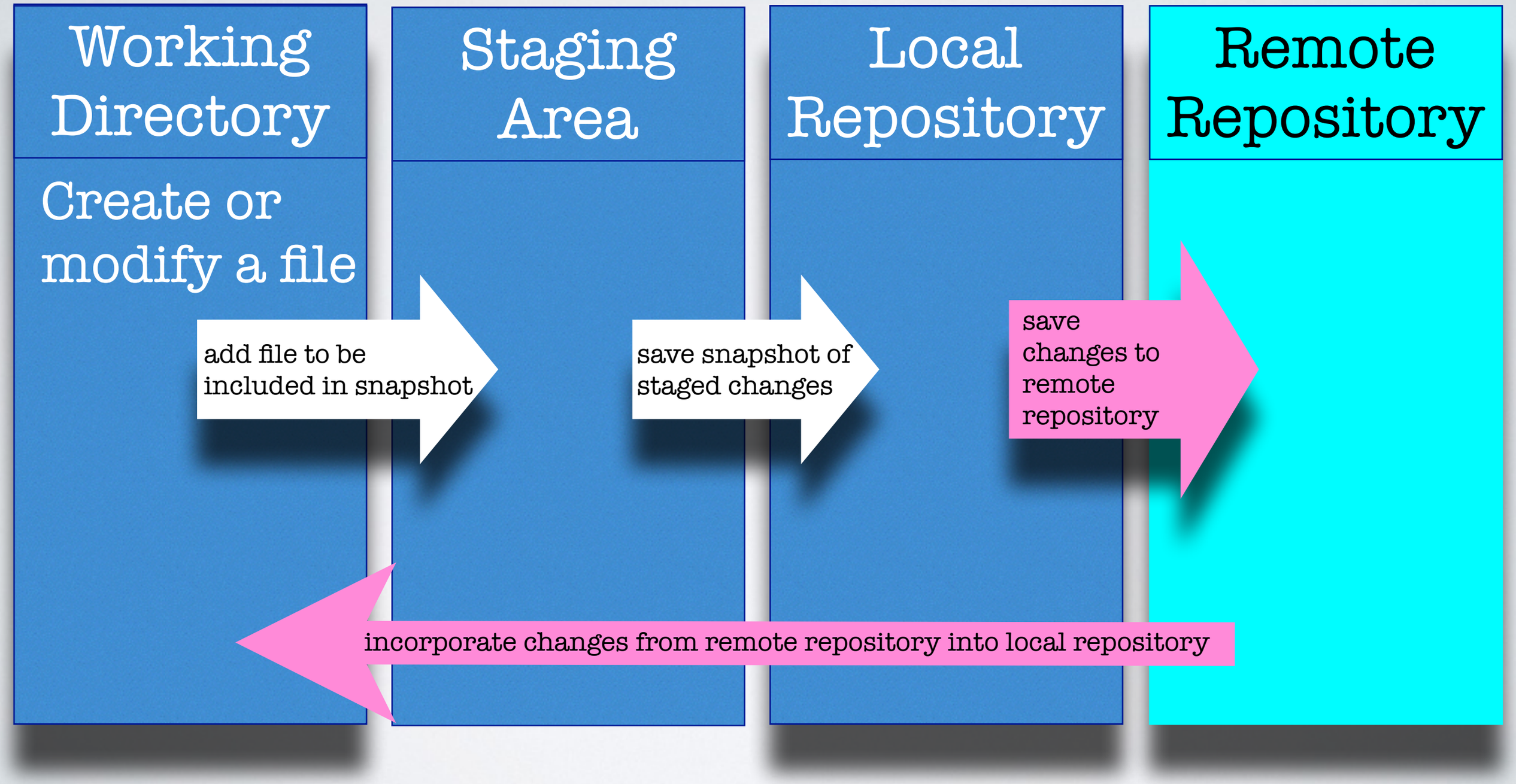
USING GIT WITH A REMOTE SERVER

sharing and collaborating

REVIEW


Working Directory
Staging Area
Local Repository
Remote Repository
Create or modify a file
add file to be included in snapshot
save snapshot of staged changes
save changes to remote repository
incorporate changes from remote repository into local repository

REMOTE REPOSITORY OPTIONS

- We will use github.com
  - free version: by default repositories are public
    - with a .edu email you can get a few private repositories
- bitbucket.com
  - free version: all repositories are private and you can only share with 5 people
- gitlab
  - configure your own private server

EXPLORE GITHUB

USING A REMOTE REPOSITORY

- 2 ways to use a remote repository
  - start it locally, create an empty repository, link the two (we will do this first today)
  - start a remote repository (or use an existing one) and clone it (you did this yesterday, we will do it again today)

CREATE YOUR REMOTE REPOSITORY

- go to github.com
- Click New Repository
- Name Repository swc_test_repo
- Create Repository

LINK YOUR REMOTE AND LOCAL REPOSITORIES

- Go back to the command line
- cd into the repo you created yesterday during the git lesson (swc_test_repo)
- make sure you are on the master branch
- Follow directions to "Push an existing repository from the command line"
    - from your local swc_test_repo directory type commands
- View what repositories your local repo is linked to:
    - git remote -v

Cheat sheet:

- git add filename
- git commit -m "message"
- git status
- git help
- git branch branch_name
- git checkout testing
- git merge branch_to_merge
- git branch -d branch_to_delete

EXERCISE: REVIEW

- edit your names.txt file
- add and commit names.txt

Cheat sheet:

- git add filename
- git commit -m “message”
- git status
- git help
- git branch branch_name
- git checkout testing
- git merge branch_to_merge
- git branch -d branch_to_delete

PUTTING CHANGES ON THE REMOTE SERVER

- view names.txt on github.com - no changes
- type in your local repo: git push origin master
- origin: alias to your remote git repository url (git remote -v)
- master: the local branch you are on (default is master)

ADDING REMOTE CHANGES TO YOUR LOCAL REPOSITORY

- git pull origin master

WORKFLOW

- git pull origin master
- make changes
- stage changes (git add)
- commit changes locally (git commit)
- push changes to the remote repository *
    - if a change has been made to the remote repository since you last pulled, you will have to pull before you push
        - why? git wants you to resolve any conflicts locally before you put anything on the remote server
        - You will have to consider whether the change (even if it does not conflict with your change) will affect your tested results

EXERCISE

- Simulate the workflow:
  - pull down any remote changes
  - modify a file
  - add and commit the file
  - push the changes to the remote server (github)

COLLABORATION

ADDING A COLLABORATOR

- If your repo is public, anyone can clone it (and pull) at any time

- To push to someone else's repository you need permission (aka add a collaborator)

HOW TO ADD A COLLABORATOR

- Find a partner
- from your repository on github go to "**settings**" (on the right)
- On the left, click on "**collaborators**"
  - enter your github password if prompted
- Type your partner's github username and click "**add collaborator**"

CLONE PARTNER 1'S REPO

- Partner 1: copy and paste the url to your repo in the ether pad
- Partner 2:
  - go to your partner's repo url (copy from the etherpad)
  - On the righthand side under "**HTTPS** clone URL" is a url copy it
  - cd to your home directory
  - make a directory called collaboration and cd into it
  - git clone <paste url>
  - cd into partner 1's repository (collaborationswc_test_repo)

Exercise 2a: No changes on Remote Server

1. Partner 1: modify a file and add, commit, push
2. Partner 2: pull down the changes to the remote repository, modify the same file then add, commit, and push

If time switch roles and try repeat

Exercise 2b: Non conflicting changes on Remote Server

1. Now, without pulling, both modify different lines in the same file
2. Partner 1: add, commit, and push.
    A. You will receive an error message gently reminding you to pull before you push
    B. pull, save the default commit message and exit editor
    C. push
3. Partner 2: Repeat step 2 on your machine

Exercise 2c: Conflicting changes on Remote Server

1. Partner 1: pull down remote changes
2. Now both modify the same line in a file
3. Partner 1: add, commit, and push
4. Partner 2: add, commit, and push
    A. You will get an error message that you need to pull. When you pull you will be told that you need to resolve conflicts.
    B. resolve conflicts (just like with branches), add, commit, and push

Resources

- This presentation
- http://git-scm.com/book/en/
- http://pcottle.github.io/learnGitBranching/
- github.com