INF 502 – SOFTWARE DEVELOPMENT METHODOLOGIES

Week 2

NAU NORTHERN ARIZONA UNIVERSITY

First things first…

- Connect to iClicker for in-class participation and attendance

https://join.iclicker.com/7YDD3

Last class…

And this is the basic flow to put your contributions back to the repo

Let’s Branch it out!

- Listing your branches
  - git branch
- Create a branch
  - git branch <NEW_BRANCH>
- Use another branch
  - git checkout <BRANCH_NAME>
- Latest two in one command
  - git checkout –b <NEW_BRANCH>

Updating the Master Branch

- Usually we branch out for versions, features, bug fixes…
  - Later on merging back to master
- How to merge our recently changed file, then?
  - In the master branch:
  - git merge <other_branch>
- If everything goes smooth… sweet

Dealing With Small Conflicts

- Imagine if you change a file in your branch, and someone else changed the same file
  - CONFLICT!!!!
- Can we still merge it?!?!?

Auto-merging <file>
CONFLICT (content): Merge conflict in <file>
Automatic merge failed; fix conflicts and then commit the result.

Usually… For the easy ones

```
Here comes common text before
the area where the conflict happens
and bla bla bla
<<<<<<< HEAD
This is what was in the
master branch
=======
And this…
was in the other branch
>>>>>>> other branch
More text that was common,
and no conflict happened here
```

Your turn!

- Make a new branch called bugfix
- Checkout the bugfix branch with git checkout bugfix
- Create/change a file and commit
- Go back to master with git checkout master
- Change/create a file (different from the previous one) and commit again
- Merge the branch bugfix into master with git merge

Rebasing it all!!!

- Another way of combining branches
- We can take a set of commits, and copy them in another branch

- Master and our branch are not sync'ed
  - check with git log --graph --all

- From the branch, we can
  - git rebase master
  - Point the branch to where the master is…
- Check the log again!

Moving From Here to There

- HEAD is the pointer name for the last checked out commit
- We can “detach” the head by “checking out” a specific commit
  - git checkout <commit SHA>
  - To get the HEAD back to where it was
    - git checkout <branch>
- We can move a branch to where the HEAD is now
  - git branch -f <BRANCH>

iClicker time

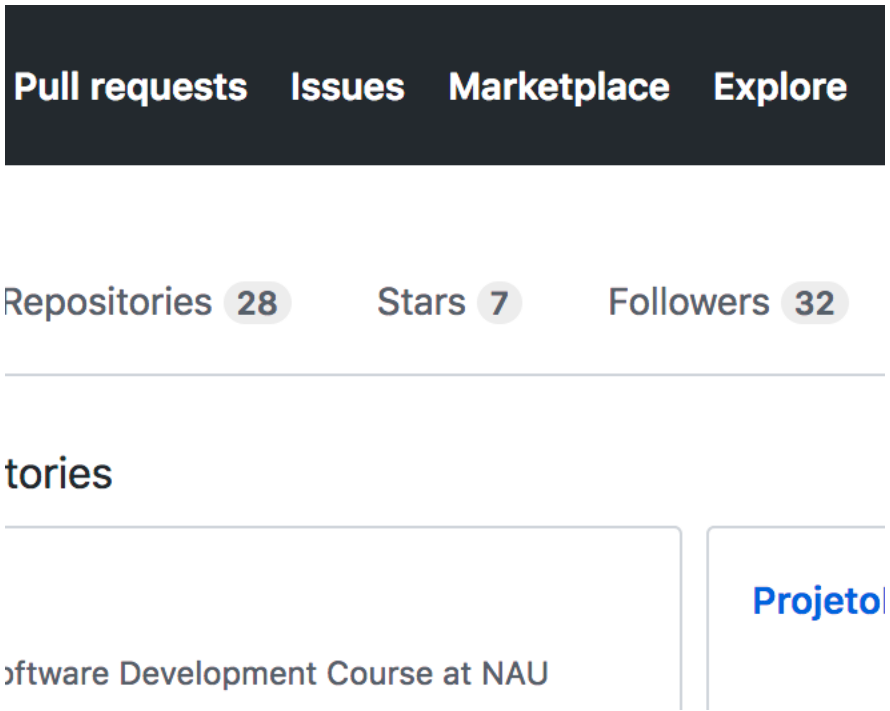
What does a git rebase master command does?

- Move the branch I’m in to point where the master is
- Move the master to where the branch I’m in is
- Detach the master from the HEAD
- Point the HEAD to the master
- None of the above

Dealing with the Remote Repo

github
SOCIAL CODING

Create a Repo in GitHub



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner igorsteinmacher / **Repository name** MyRepo

Great repository names are short and memorable. Need inspiration? How about m

Description (optional)

This is a demo project!

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're repository.

Add .gitignore: None | Add a license: None

Create repository

Cloning To a Local Repo

- Cloning means bringing all the history to a local repo
  - git clone https://github.com/<owner>/<repo>

- Testing it out
  - git clone https://github.com/NAU-OSS/githandson.git

  Cloning into 'githandson'...
  warning: You appear to have cloned an empty repository.

Working In This Repo

git branch --r

git pull

- pulls everything from the remote repo and updates the local repo

git fetch

- pulls changes from remote repos, but it doesn't integrate any of this new data into your working files

git push <remoteName> <branchName>

- push your local changes to an online repository (git push origin master)

Usual Workflow – git clone

branch out to add your changes locally

your adds/commits

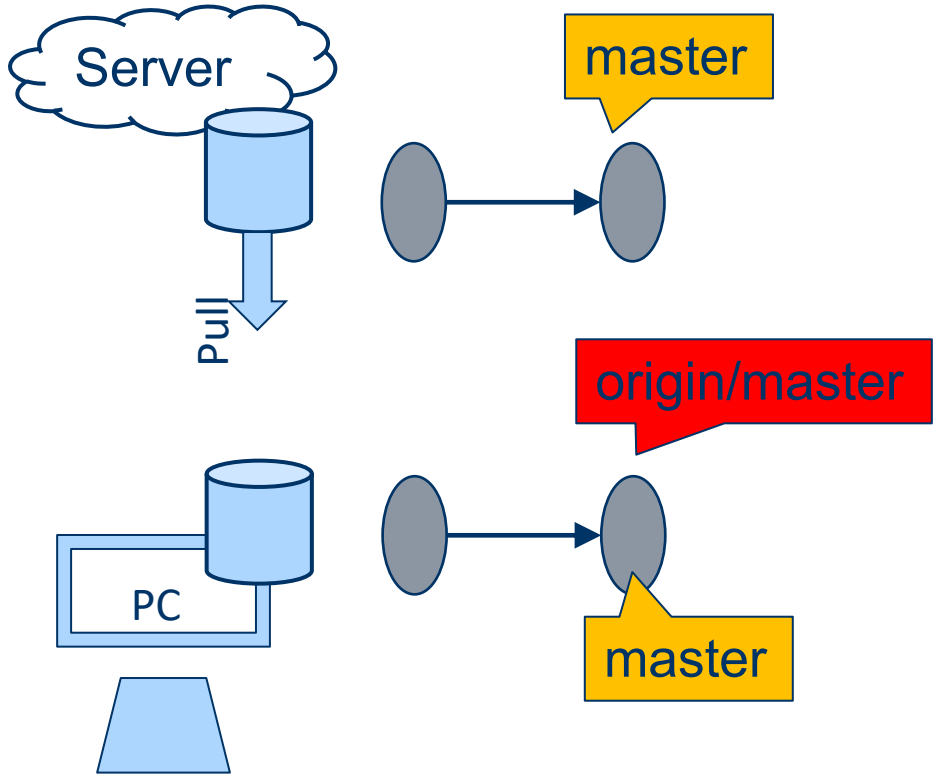
pull remote changes to your local repo

merge your branch back (LOCALLY)

- Resolve any conflict

push changes back to the remote repo

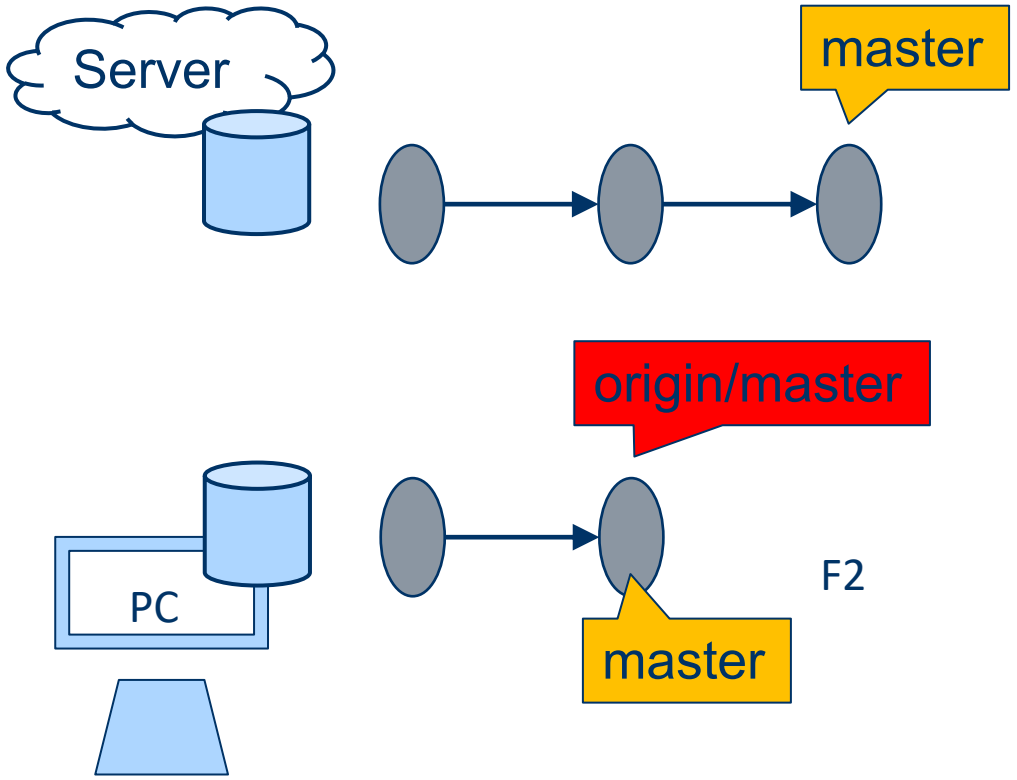
Example



git pull
#updates local repo

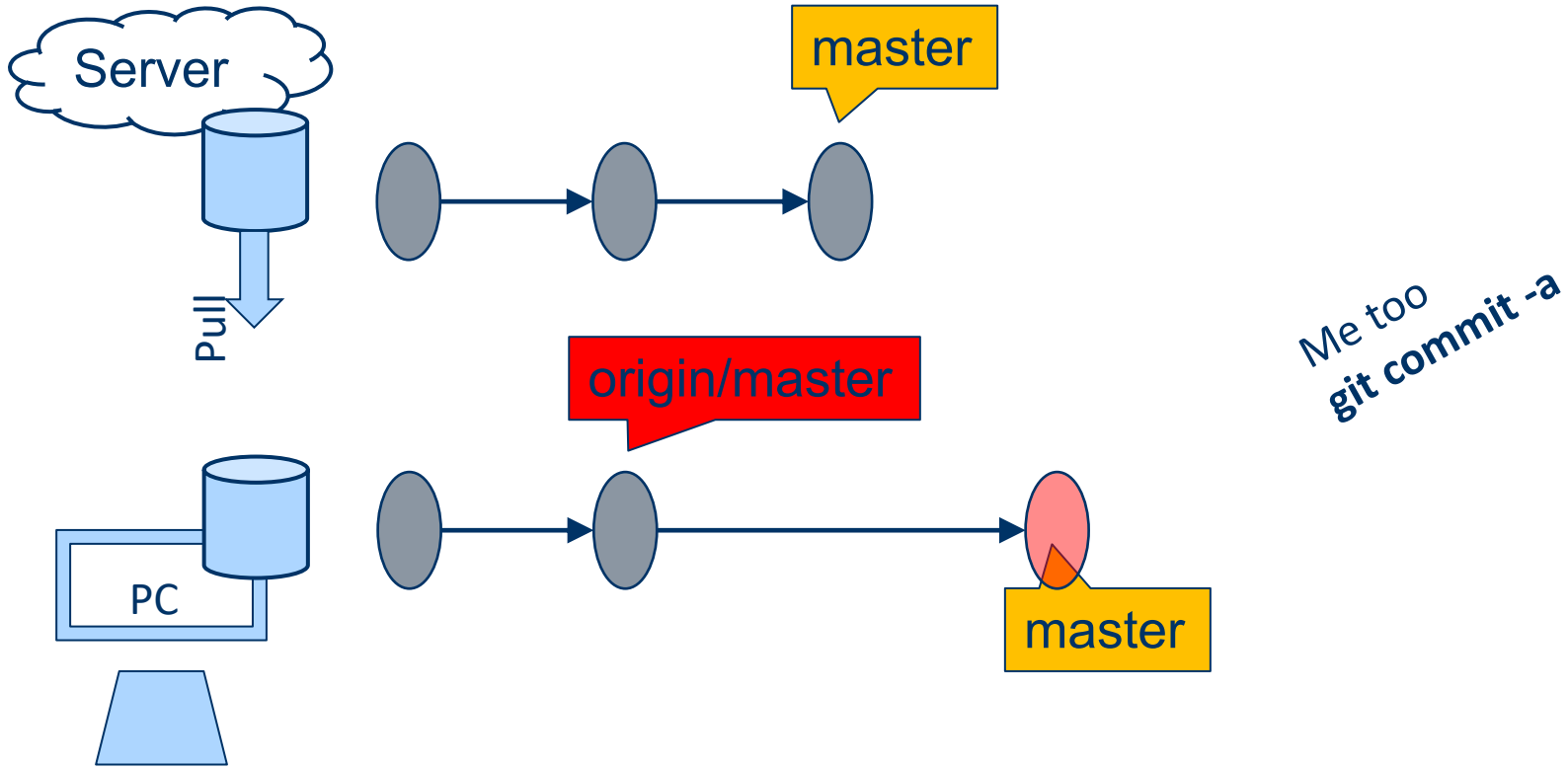
master is a local branch
origin/master is a remote branch (a *local copy* of the branch named "master" on the remote named "origin")
origin is a remote

Example

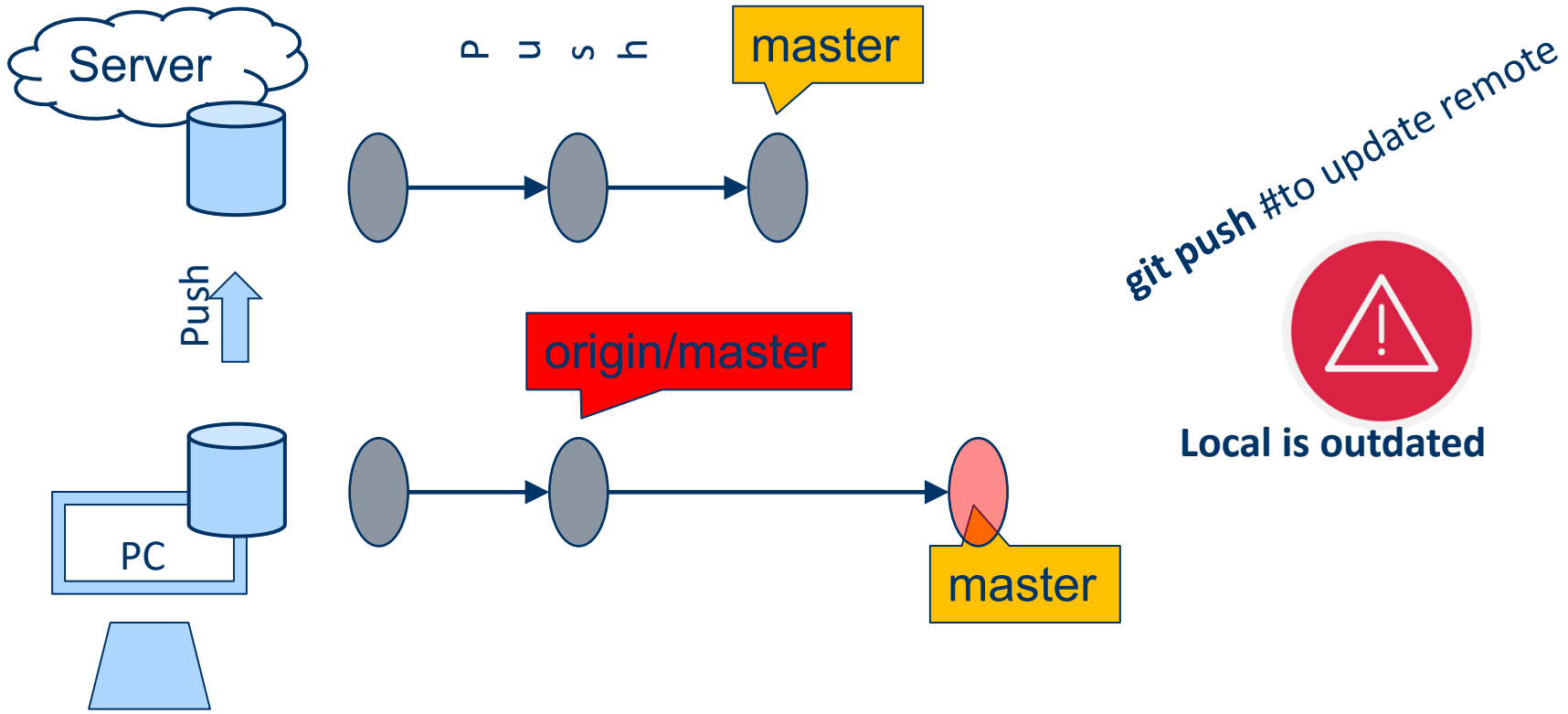


Someone changed the remote

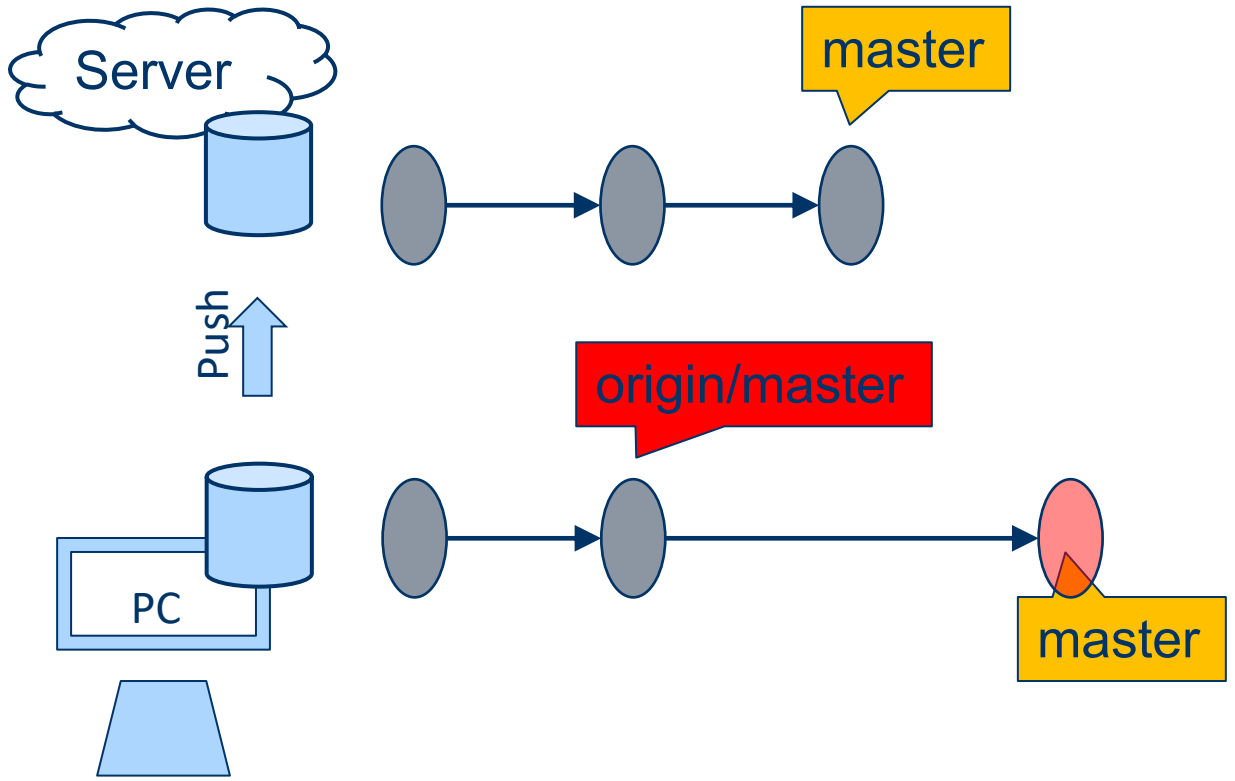
Example


Server
master
Pull
origin/master
PC
master
Me too
git commit -a

Example

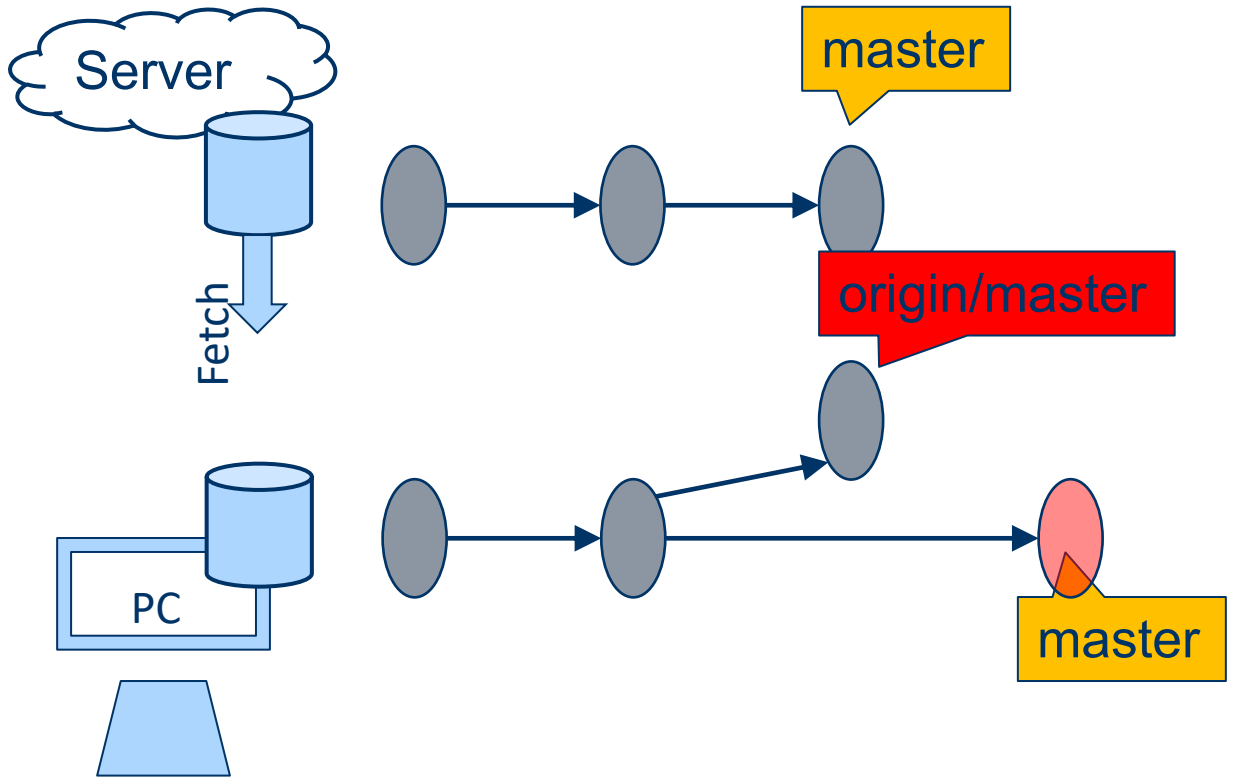

Server
P u s h
master
Push
origin/master
PC
master
git push #to update remote
Local is outdated

Example



Solution 1:
Fetch +
Rebase +
Push

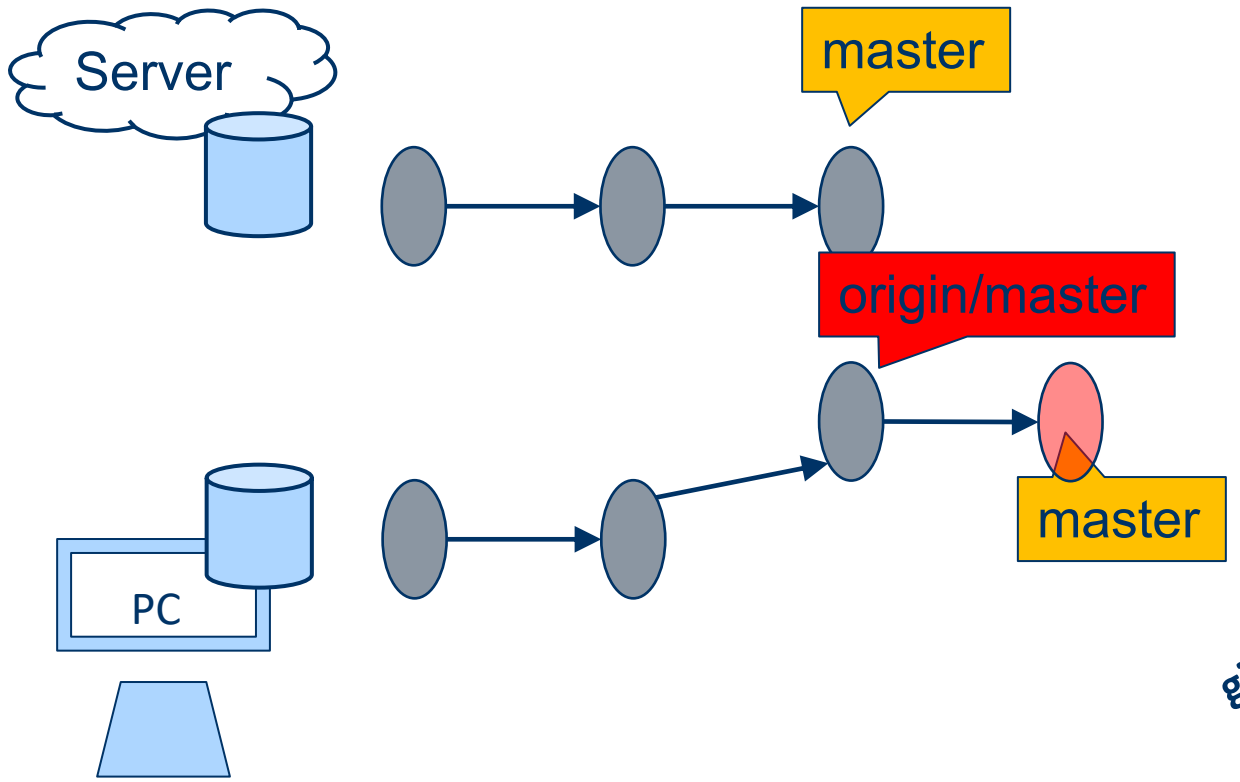
Example



Solution 1:
Fetch +
Rebase +
Push

git fetch
#download remote changes

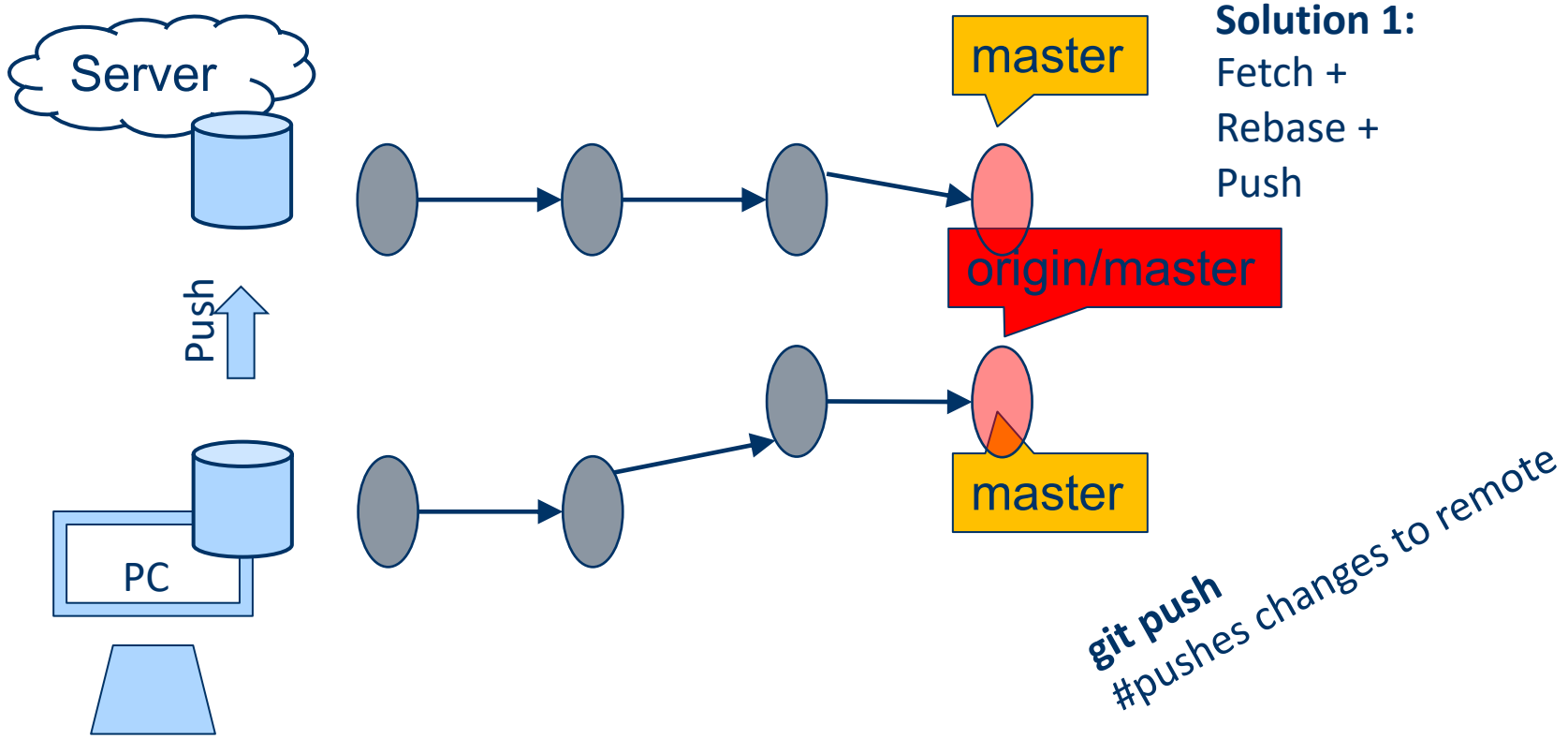
Example



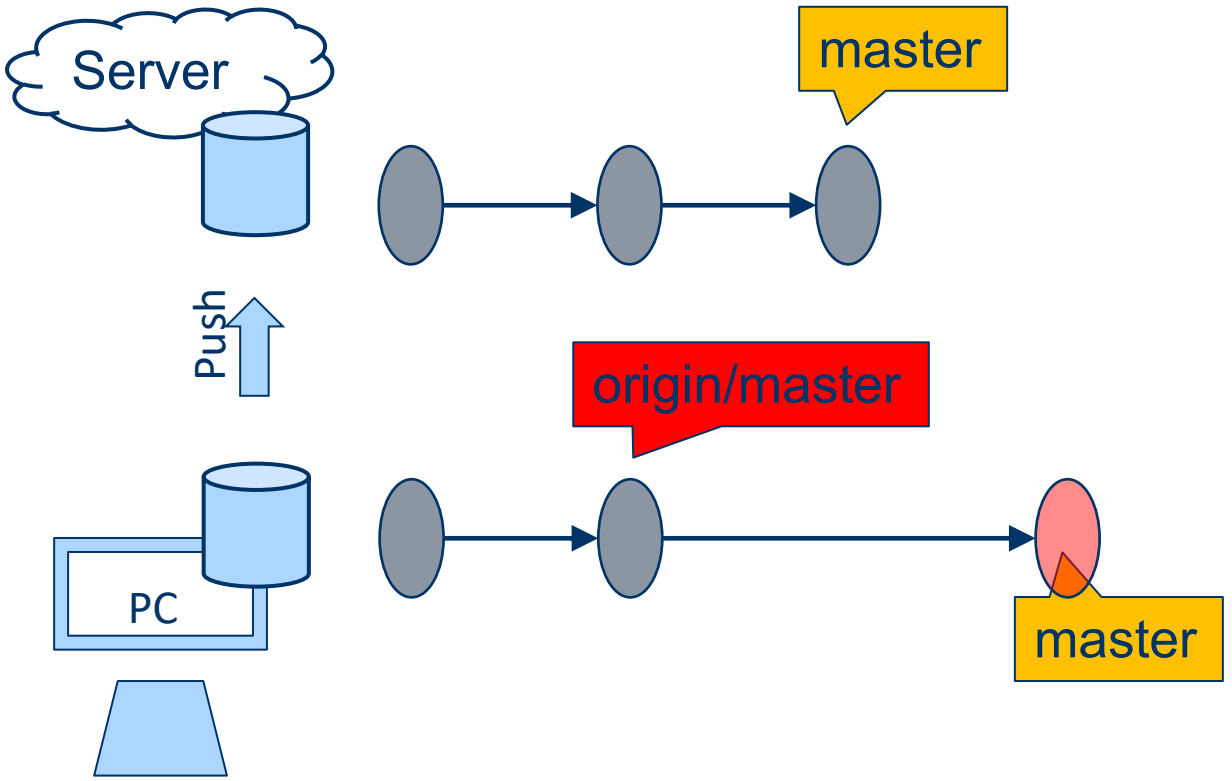
Solution 1:
Fetch +
Rebase +
Push

git rebase origin/master

Example

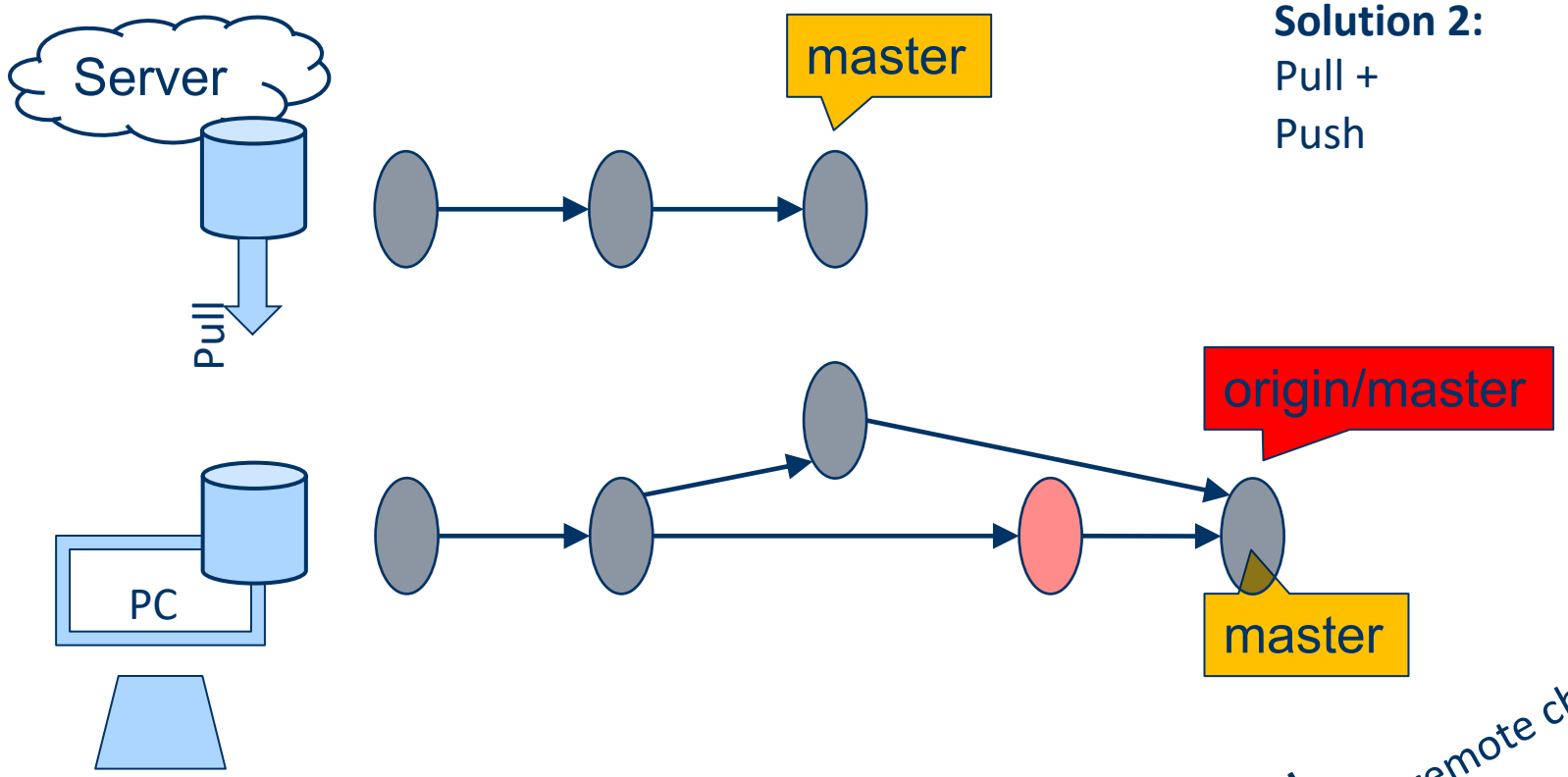

Server
Push
PC
master
origin/master
master
Solution 1:
Fetch +
Rebase +
Push
git push
#pushes changes to remote

Example



Solution 2:
Pull +
Push

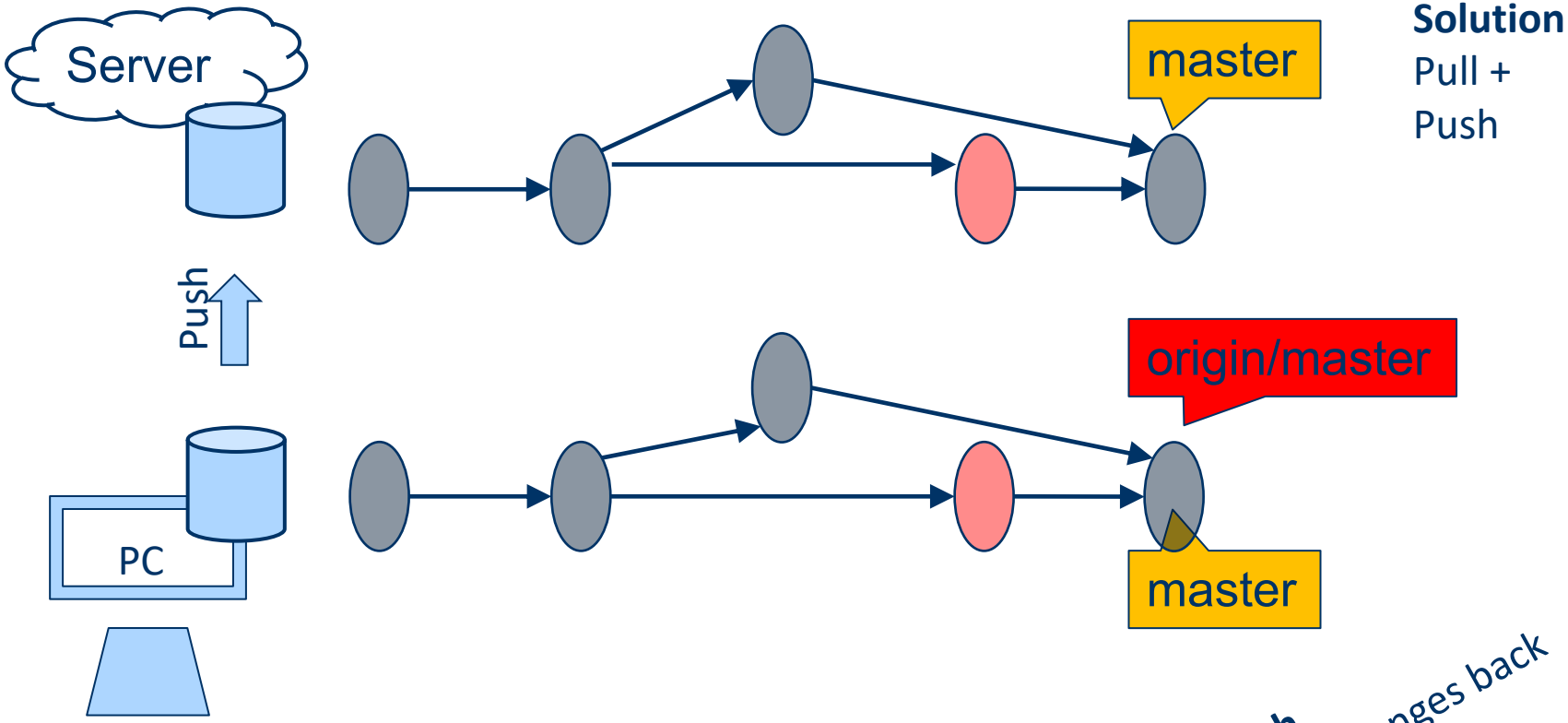
Example



Solution 2:
Pull +
Push

git pull
#download remote changes
and update local

Example

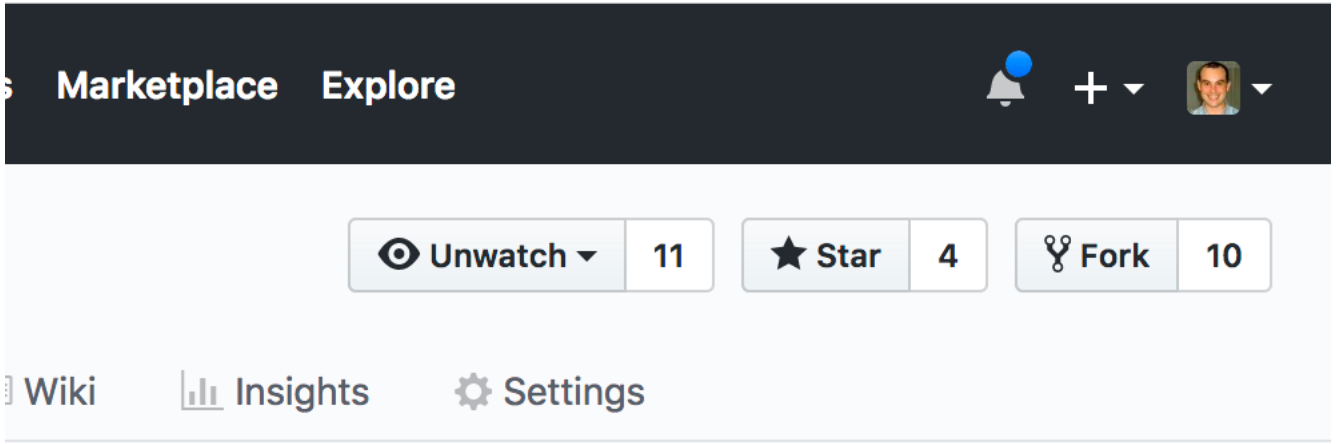


Solution 2:
Pull +
Push

git push
#pushes changes back

GitHub Workflow

- Fork + pull-request
  - You usually creates a fork for your repo
    - “A **fork** is a copy of a repository. **Forking** a repository allows you to freely experiment with changes without affecting the original project”
  - Creating a fork



  - Then, you usually clone your fork… work, and send a pull request against the main repo

A Pull Request Example

(GitHub) Fork

(CLI) Clone (the fork)

(CLI) Commits

(CLI) Push

(GitHub) Send Pull Request

- Use the GitHub interface and follow the instructions
- See more here

Keep your fork up-to-date

- https://help.github.com/articles/syncing-a-fork/

iClicker time

What is the meaning of a pull request?

- A request to merge the changes from the workspace to the local repository
- A request to merge the changes from the local clone to the remote repository
- A request to merge the changes from the local clone to the forked repository
- A request to merge the changes from the fork to the original repository
- None of the above

Be ready for what's next…

Python essentials

NAU NORTHERN ARIZONA UNIVERSITY