

# Open Source SW & Lab - Summer 2023

## 4. Git-Remote

**Walid Abdullah Al**

Computer and Electronic Systems Engineering  
Hankuk University of Foreign Studies



**Computer Vision Lab**  
Hankuk University of Foreign Studies

**Based on:**

Pro Git (2022) by Scott Chacon, Ben Straub

# Recap: basics

---

- Git initialization
- Tracking a new file
- Staging changes in a file
- Committing staged changes

# Recap: basics

---

- Undoing a commit
- Going back to a commit

# Recap: branching

---

- Creating a branch
- Switching to a branch
- Merging a branch into current/HEAD branch
- Merge conflicts

# Remote repositories (RR)

---

- versions of your project that are hosted on the Internet
- Collaboration involves
  - Manage remote repositories
  - Pushing and pulling data to and from them
- **Remote management**
  - Adding RR
  - Removing RR
  - Managing RR branches
  - Set them as tracked or non-tracked

# Adding RR

---

- **Run** `git remote add <shortname> <url>`
- **Or, git clone (which implicitly adds the origin remote)**
- **Show remote list using** `remote` **or** `remote -v`

```
$ git clone https://github.com/schacon/ticgit
```

```
$ git remote add pb https://github.com/paulboone/ticgit
$ git remote -v
origin  https://github.com/schacon/ticgit (fetch)
origin  https://github.com/schacon/ticgit (push)
pb      https://github.com/paulboone/ticgit (fetch)
pb      https://github.com/paulboone/ticgit (push)
```

# Fetching / pulling

---

- **Fetch**

- Pulls data from remote that you don't have yet
- Requires explicit merging

```
$ git fetch <remote>
```

- **Pull**

- Automatically fetch and merge into your current branch
- Use `git pull`

# Inspecting a Remote

```
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/schacon/ticgit
  Push  URL: https://github.com/schacon/ticgit
  HEAD branch: master
  Remote branches:
    master                tracked
    dev-branch            tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local ref configured for 'git push':
    master pushes to master (up to date)
```

```
dev-branch                tracked
markdown-strip            tracked
issue-43                  new (next fetch will store in remotes/origin)
issue-45                  new (next fetch will store in remotes/origin)
refs/remotes/origin/issue-11 stale (use 'git remote prune' to remove)
Local branches configured for 'git pull':
  dev-branch merges with remote dev-branch
  master      merges with remote master
Local refs configured for 'git push':
  dev-branch      pushes to dev-branch                (up to
  markdown-strip  pushes to markdown-strip            (up to
  master          pushes to master                    (up to
```



# Remote branches

---

- **<remote>/<branch>**
  - e.g.,
  - origin/master (remote) vs master (local)
  - origin/iss53 (remote) vs iss53 (local)

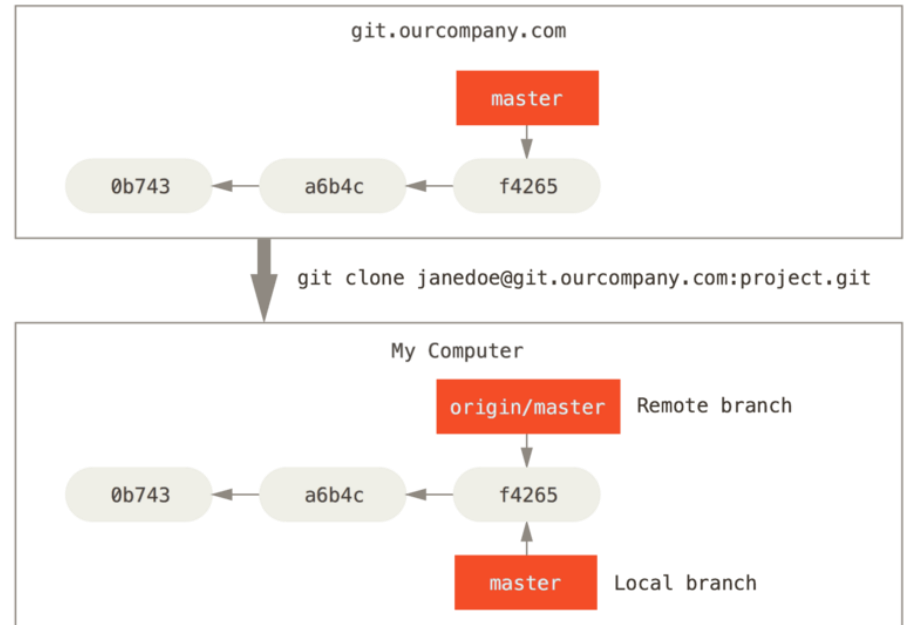
# Remote branches (2)

- **Let's say:**

- git server at git.ourcompany.com
- After cloning, points to remote master
- Names it origin/master

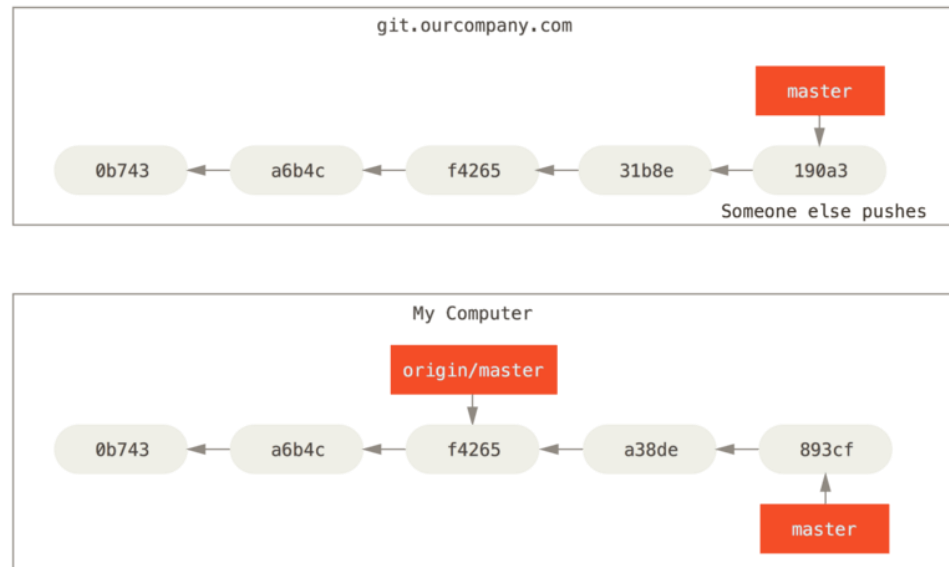
- **Locally**

- master branch pointer
- Starting at the same place as origin/master



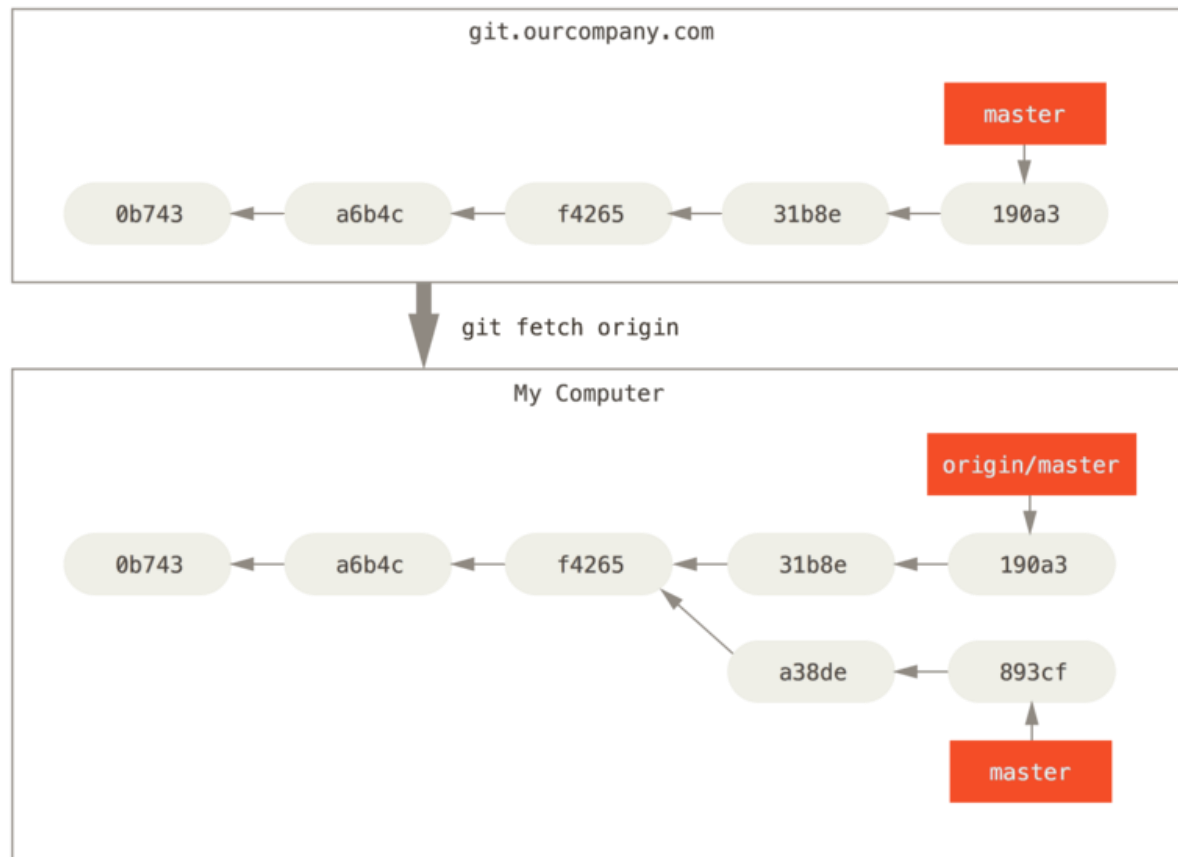
# Remote branches (3)

- **You do some work locally**
  - Which branch is affected?
- **Someone else also pushes to the server**

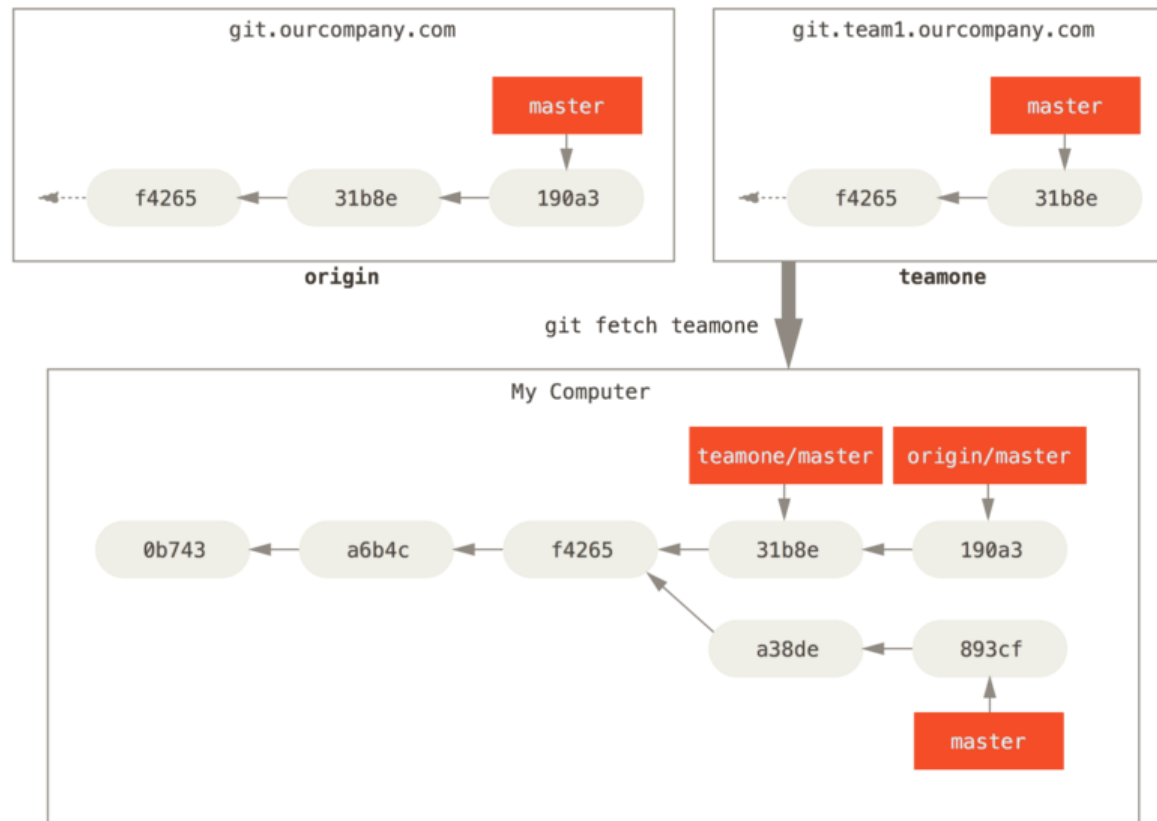


# Remote branches: sync

- What is the local situation called?



# Multiple remotes



# Pushing

---

- **You want to share your `serverfix` branch**

```
$ git push origin serverfix
Counting objects: 24, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (15/15), done.
Writing objects: 100% (24/24), 1.91 KiB | 0 bytes/s, done.
Total 24 (delta 2), reused 0 (delta 0)
To https://github.com/schacon/simplegit
* [new branch]      serverfix -> serverfix
```

- **Push to existing branch**

```
git push origin serverfix:awesomebranch
```

# Working on a new remote branch

---

- **You can't work on a newly fetched branch**
  - You only have a pointer (origin/serverfix)
- **You should merge before you start working**

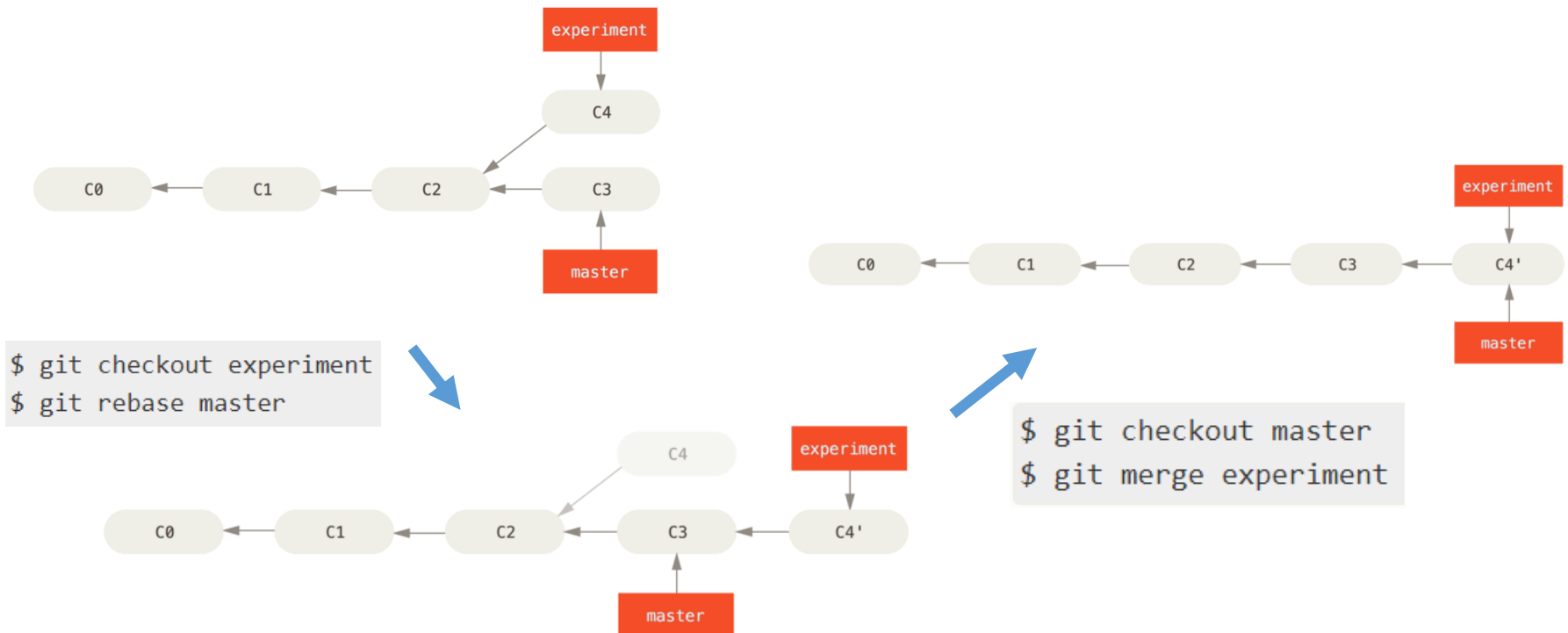
```
git merge origin/serverfix
```

or

```
$ git checkout -b serverfix origin/serverfix  
Branch serverfix set up to track remote branch serverfix from origin.  
Switched to a new branch 'serverfix'
```

# Rebasing

- Alternative to merging
- Replay changes that were committed on a different branch
- E.g., checkout <experiment> and rebase it onto <master>

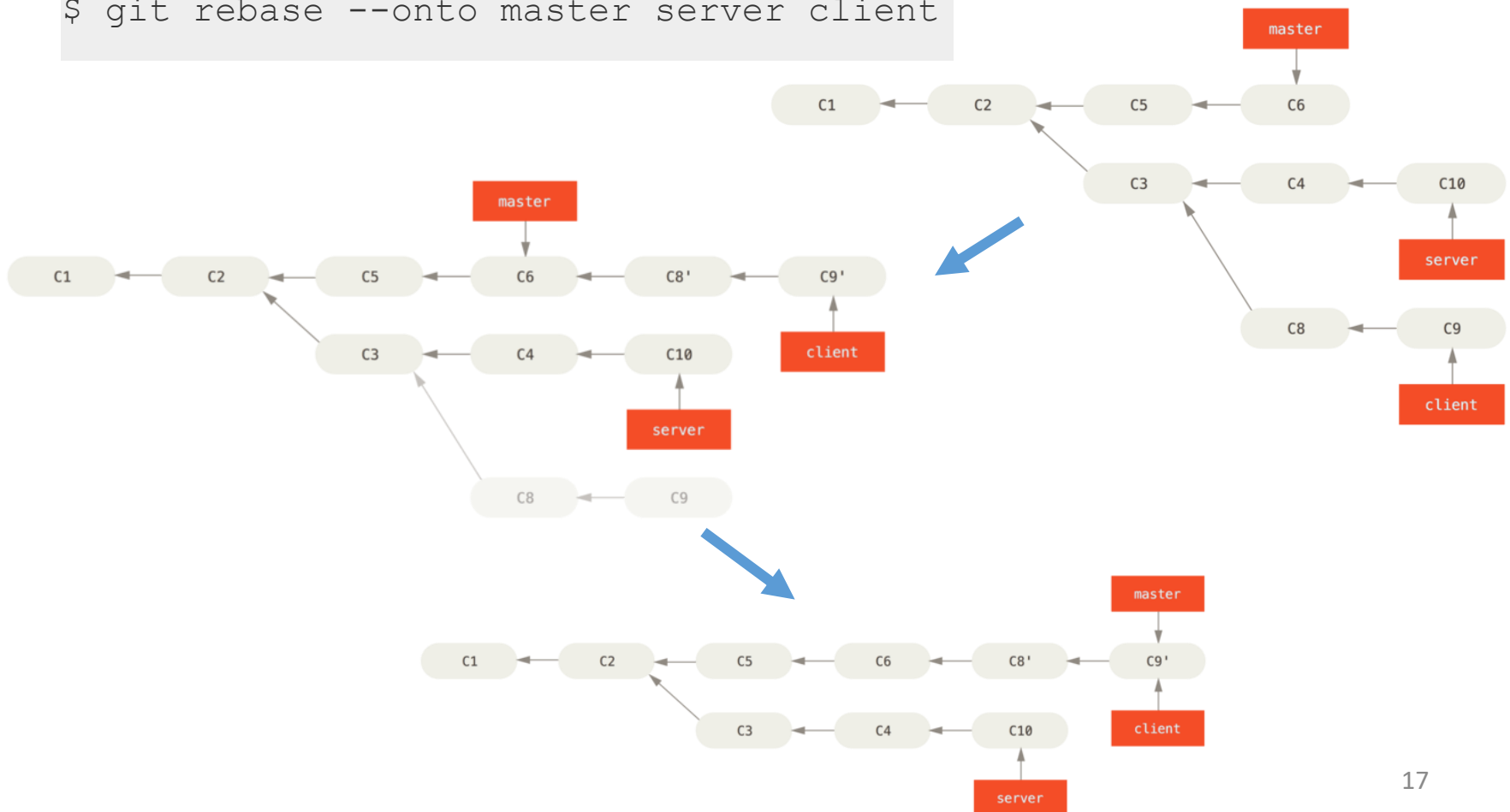




# Rebasing (2)

- We want to merge the client to the master first

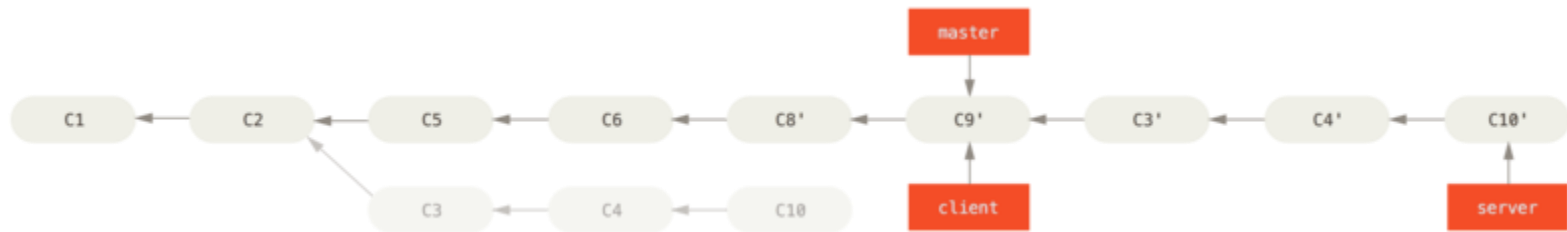
```
$ git rebase --onto master server client
```



# Rebasing (3)

---

- **git rebase master server**



- **Note:**
  - Do not rebase commits that exist outside your repository and that people may have based work on.

# Labwork on remote

---

- **Instructions available at:**
  - [https://github.com/awjibon/remote\\_exercise.git](https://github.com/awjibon/remote_exercise.git)
- **To access this,**
  - Open a Github account
  - Fetch the above remote repository
  - Follow the instructions in readme.txt file

# Done for today!

---

- I will check your attendance based on your push to the remote