# ECE 361: Computer System Organization

## In-class work session
## Git and GitHub

Roy Kravitz
Electrical and Computer Engineering
Maseeh College of Engineering and Computer Science

Presentation material is drawn from the GitHub Campus-Advisors training material

Portland State UNIVERSITY

## What is Git?

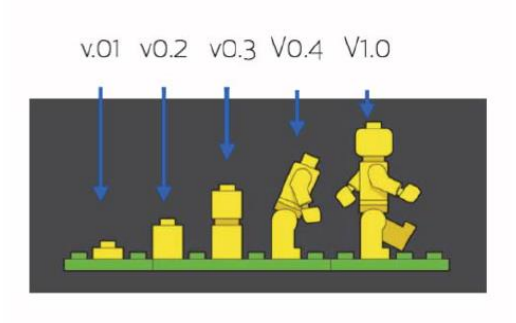
Video: ..\video\1.1 Meet Git.mp4



Portland State UNIVERSITY

## Git is a *version control system*

A tool that lets you track your progress over time.



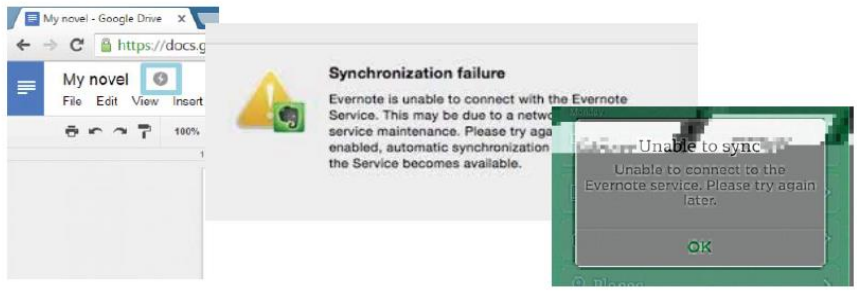



## Git takes snapshots

Save snapshots to your history to retrace your steps.

Also keeps others up-to-date with your latest work.

# Centralized systems require coordination…

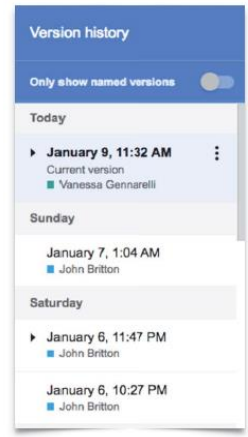





# Order with coordination:

In a centralized system, you can objectively call versions a numerical progression: version 1, version 2, version 3…

Since John made a new version before Vanessa, his is n+1, and Vanessa is n+2.



## Working in parallel: order without coordination

Git goes after this idea of distributed version control, so you can keep track of your versions without coordination.




## Exercise 1 – Configuring Git

- Check is Git is installed on your PC
  - `git --version`
- Install Git if necessary
  - https://git-scm.com/downloads
- Configure Git
  - `git config --global user.name "your name"`
  - `git config --global user.email "your email address"`
  - `gig config --list`
- Create a GitHub account if necessary
  - https://github.com/
- Get a student developer pack (unlimited free repositories, developer tools, and swag)
  - https://education.github.com/pack

## Repositories and the 3-fold model

Video: ..\video\1.2 The three-fold model.mp4



## A repository holds the entire history of your project

A repository is the unit of separation between projects in Git.

Each project, library or discrete piece of software should have it's own repository.

Basics

## Git is like a desk

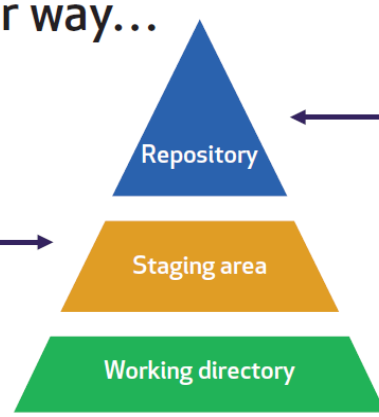
Working directory
where you write

Staging area
rough draft, in a
manila folder

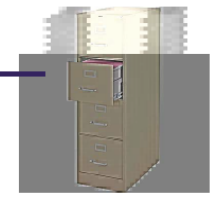
Repository
final draft
in the filing cabinet

Basics

## Put another way…




Repository
Final draft,
permanent

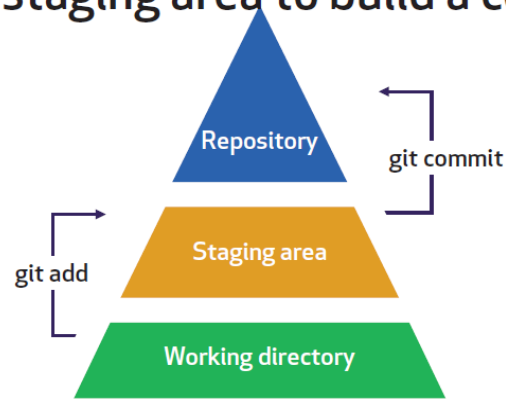


Staging area
Your rough draft

Working directory
Your workspace,
local on your machine

Basics

## Use the staging area to build a commit






## Making commits

`'git commit'`

tells Git to save that portion of the project from the staging area into the repository history.

## Exercise 2 – Making your first commit

- Initialize a new repository
  - Open git bash shell
  - `cd location-for-your-ECE361-projects`
  - `git init ic1_ex1`
  - `cd exercise-2`
  - `ls -al`
- Add an empty readme file to the staging area
  - `touch readme.md`
  - `git status`
  - `git add readme.md`
  - `git status`
- Commit *readme.md* to the repository
  - git commit –m 'initial commit'
- Add/commit a 2nd file to the repository
  - Create a text file that answers the following question:
    - Why did you choose the CompE track instead of the EE track?
- Update the text of *readme.md* in some way and commit it





## Being selective w/ Git

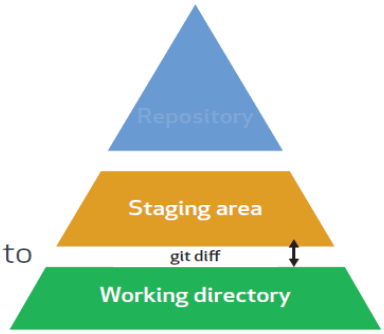
Video: ..\video\1.2 The three-fold model.mp4

## Understanding the state of your repository

```
git status
git diff
git diff --staged
```

## What are comparing w/ `git diff`?

### git diff



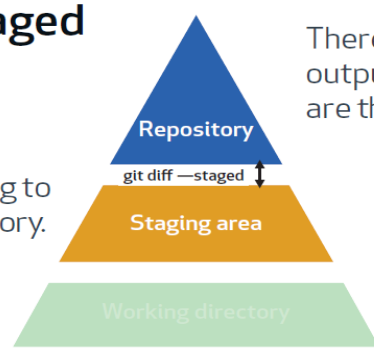

Compares staging to working directory.

There's no output if they are the same.

## What are comparing w/ `git diff`? (cont'd)

### git diff --staged

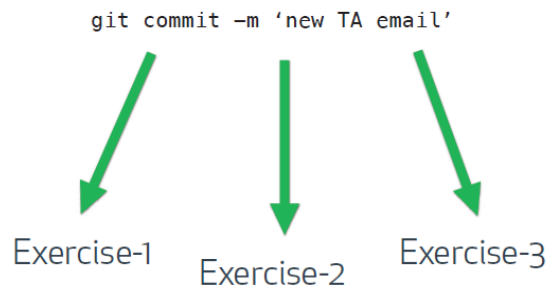
Compares staging to repository directory.




There's no output if they are the same.

## Git allows you to be selective

You can fix a bug across several different files in the same commit.



# But commits should be logically grouped

Don't mix typo corrections and new features.

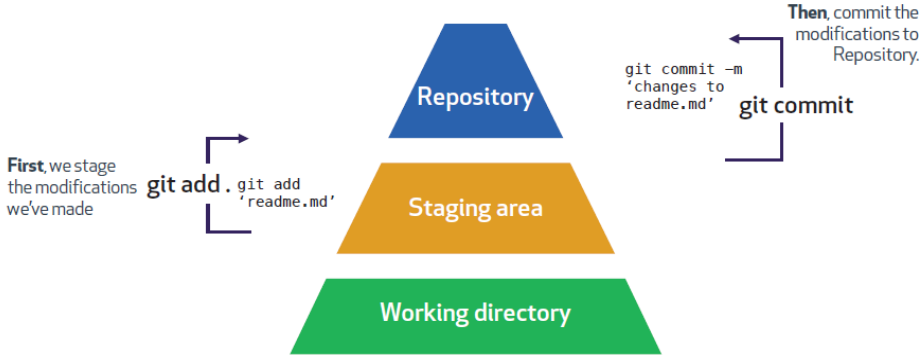
If the feature gets rolled back, you re-introduce the typo.

```
git commit –m 'typo in readme.md'
```

```
git commit –m 'new signup flow.'
```

```
git commit –m 'fix typo, add field to signup flow, create parallax effect'
```

# Order of operations:



## Using GitHub

Video: ..\video\2.1 network activity.mp4



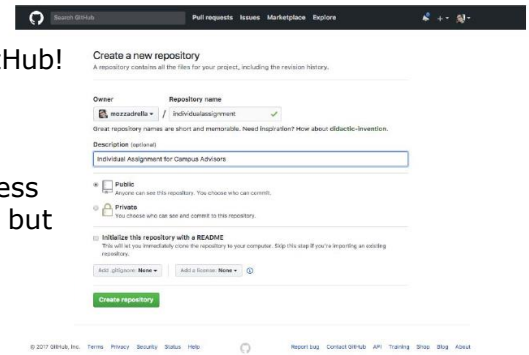
## GitHub

- Hosts your repositories
- Track student progress
- Social features to enable collaboration

Individual assignments

# Let's set up a place to host your code

- A remote repository on GitHub!
- https://github.com/new
- As a student you have access to free private repositories but let's choose public for now






# The bookmarked location is referred to as a "remote."


```
git remote add origin (REPO LOCATION)
```



Adding a remote allows the transfer of your commits to another machine…and provides a cloud-based backup of your work

## Add origin

Send my commits to a location.

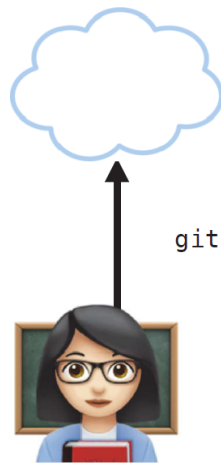
And origin is at this address.

```
git remote add origin (REPO LOCATION)
```

This statement names the remote "origin."

Individual assignments

## How do you get your commits up to the remote?



```
git push origin master
```

Individual assignments

Link remote with local.

-u is short for --set-upstream

```
git push –u origin master
```

Useful because you can just write "git push" when you want to push future commits.

Individual assignments

## Using GitHub

Video: ..\video\2.3 Fetch.mp4

## Exercise 3 – Using GitHub

- Create a remote repository for your *ic1-ex2* project.
  - Open GitHub in your browser (https://github.com/)and sign in
  - Click on `+` and select `New Repository`
  - Name the repository *ic1-ex2* and provide a description. You may make the repository public and you do not need to create a README.md file because there is already a readme file in your local project
  - Click on the green `Clone or Download` button and copy the URL link to the clipboard. You will need it later
- Add a remote bookmark to the repository.
  - `git add remote URL-for-the-project`
- Push your local repository to GitHub.
  - `git push –u origin master`





## Exercise 3 – Using GitHub (cont'd)

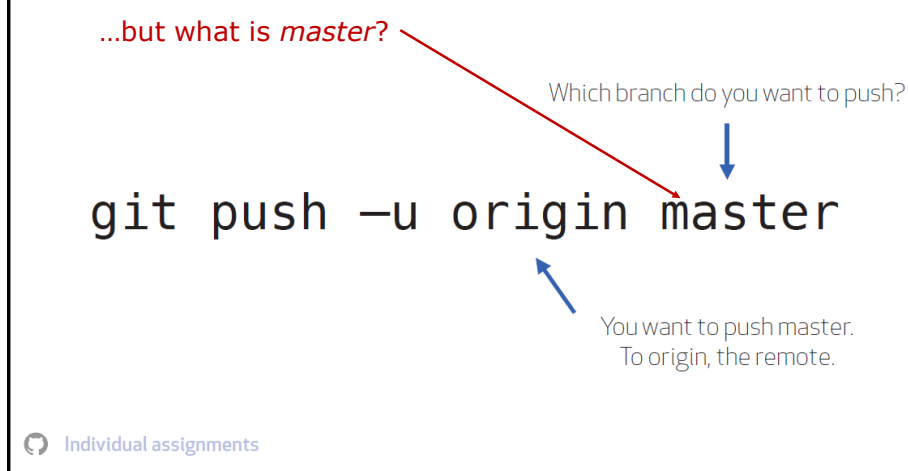
- Make changes to the repository on GitHub
  - Edit *readme.md* in some way on GitHub and commit the changes
  - Add a new file on GitHub that answers the following question:
    - Do you like the Star Wars universe or the Star Trek universe more? Explain why.
  - Commit the new file to GitHub
- Fetch the changes from GitHub to your local repository
  - `git fetch`
- Use `git log`, `git status`, `git diff`, etc. to see the effect of the `Fetch`. Take notes you will need them for the next exercise

## Branches and Merges

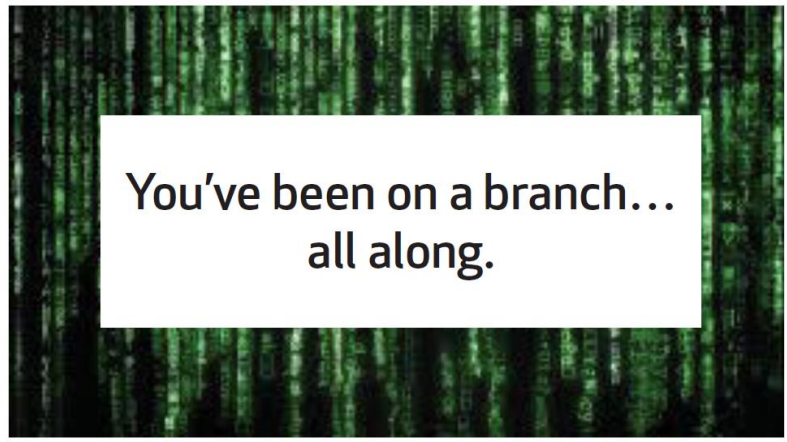
Video: ..\video\2.4 Branches.mp4
Video: ..\video\2.5 Merge.mp4



Portland State UNIVERSITY




Individual assignments



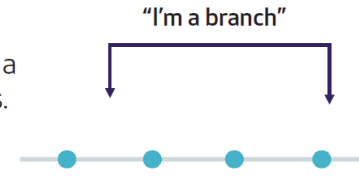


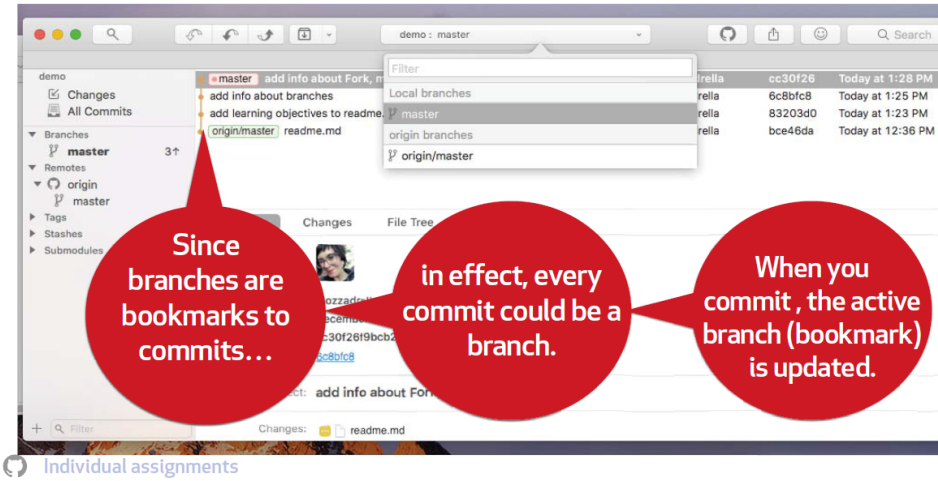

## Branches are bookmarks to commits

"Master" is the default, it's a naming convention.

Can think about branches as either a bookmark or a pointer for commits.

As we add commits, the active branch updates to point to the newest commit (HEAD).





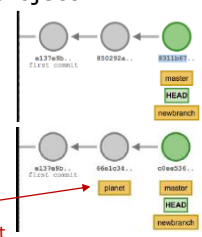

Individual assignments

## Branch-related commands

- `git show` *master* – shows the commit that the *master* branch is pointing to

```
commit cc30f26f9bcb27fc45338961a3f09b269ecd0931
(HEAD -> master)
Author: Mozzadrella <mozzadrella@github.com>
Date:   Sat Dec 16 13:28:05 2017 -0500
```

- `git branch` – shows you the branches in your project
  - * - the currently active branch
- `git branch <name>` – creates a new branch
  - Ex: `git branch` *newbranch*
- `git branch <name> REF`
  - **REF** is HEAD, a branch name, a commit or a commit-ish (^, master~3)
  - Ex: `git branch planet master^`




- `git checkout <branch>` - switch to <branch> by updating the index and the files in the working tree and by pointing HEAD at the branch
  - Local modifications to the files in the working tree are kept so that they can be committed to the <branch>

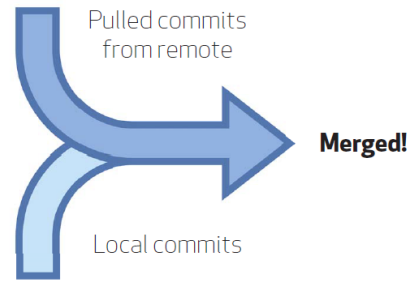
# Pull = fetch + merge

Pull first fetches the commits and stores them locally.

Merge takes the two divergent commits, puts them together in the staging area and makes a new commit with two parents.

Merge updates the active branch to point to the new merge commit

You'll see the new commits reflected in your local project when you run "git log."




Individual assignments

# Watch what happens when we run "pull."

I'd like the latest commits on the branch that my active branch tracks.

```
git pull origin master
```

```
C02T40YZFVH4:individual-work mozzadrella$ git pull origin maste
From https://github.com/mozzadrella/Module-1-Individual-Work
 * branch            master     -> FETCH_HEAD
Updating 1fa65cf..0869a21
Fast-forward
 readme.md | 4 ++++
 1 file changed, 4 insertions(+)
C02T40YZFVH4:individual-work mozzadrella$
```

Individual assignments

## Exercise 4 – Branch and Merge

- Create a new repository on our local PC called `ic1-ex4` with a *`readme.md`* file. Commit.
- Go to https://www.pottermore.com/news/discover-your-hogwarts-house-on-pottermore and select your favorite Hogwarts house. Paste the Sorting Hat verse for that house into your *`readme.md`* file. add and commit.
- Create a new branch called *`2ndfavorite`* and switch to it
  - `git branch 2ndfavorite`
  - `git checkout 2ndfavorite`
  - *Note: you can combine the commands with `git checkout -b 2ndfavorite`*
- Edit *`readme.md`* by pasting the sorting hat verse of your second choice for house (after all, the Sorting Hat does take your opinion into account). Save and commit
- Use `cat` to display the contents of *`readme.md`*. Note which house is listed



Portland State UNIVERSITY

## Exercise 4 – Branch and Merge (cont'd)

- Switch to the `master` branch and display *`readme.md`*. Note which house is listed.
- Create a remote repository on GitHub and push the `master` branch to the remote repository.
- Edit `readme.md` in the remote repository to replace the house with your least favorite house. Save and commit the changes to the remote repository.
- Pull the `master` branch from your remote repository to the `master` branch of your local repository. Display *`readme.md`* and note which house is listed.
- Use `git diff` to see the differences between `readme.md` on the `master` and `2ndfavorite` branches
- Create a new file on your remote repository called lessons-learned and summarize your findings. Please comment on how useful you found this lesson and what could be done to improve it. Save and commit the changes to the remote repository.
- Download the remote repository as a .zip file and upload the .zip file to your `Git/GitHub work session` dropbox.



Portland State UNIVERSITY

## Next Time

- Topics:
  - Data structures and algorithms
  - Big O Notation
- You should:
  - Read Wengrow Ch. 1 - 3
- Homework, Projects, Quizzes:
  - Homework #1 will be assigned Sun, 30-Sep  Due to D2L by 10:00 PM on Sun, 06-Oct