



Intro to Git and Github

Part I

UChicago TechTeam

October 12, 2016

Introduction



- Track and manage changes
 - Revert back to previous versions
 - Compare changes over time
- Collaborate with other TechTeam members on the same project

Basic structure

Basic Structure

Local repository: files on your computer

Working area

Where you
make changes
to files



Staging area

Bring changes
to Git's
"attention"



Repository

Where Git
stores versions
of the files

`git init` creates a new repository

`git add filename` adds the file to the staging area

`git commit -m "[commit message]"` stores the changes as a new version in the repository

Practice

Practice with a local repository

1. Make a new directory called "git-practice" and enter it

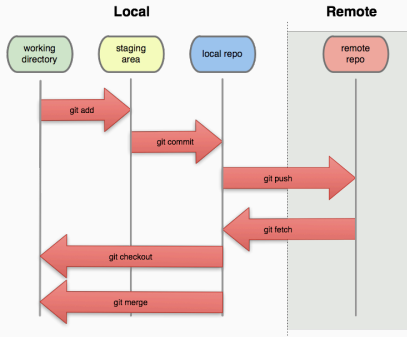
```
mkdir git-practice  
cd git-practice
```
2. Create a new Git repository in the directory with `git init`
3. Create two new files, make changes to the files (add some text), and save the changes
4. Check the status of the files with `git status`
5. Add the files using `git add [filenames]`, check the status again
6. Commit the files with a message using `git commit -m "[commit message]"`

Remote Repositories: push and pull

Remote repositories are hosted on the Internet or other network, allowing you to collaborate with others on a project

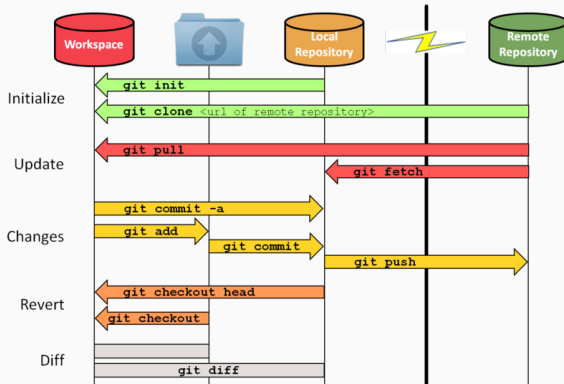
This is where Github comes in!

Remote Repositories: push and pull



- `git clone <url>` to make a local copy of the remote repository hosted at that URL
- `git push` to push changes from local to remote repository
- `git pull` updates all local files to the versions stored in the remote repository

Remote Repositories: push and pull



Practice exercise

Practice with a remote repository on Github, pt. 1

Initializing

1. Log into your Github account and go to your profile (top right)
2. Go to the Repositories tab and press New to create a remote repository
3. Name your repository, check the box "Initialize this repository with a README", and create the repository
4. Hit "Create new file". Give your file a name, some text, and press "Commit new file" at the bottom when you're done.
5. Click the name of your repository near the top of the page to go to the main repository page, then copy the URL of the repository in the address bar
6. Go to Terminal and clone your repository with the command `git clone URL-that-you-copied`
7. Change into your new repository directory and list the files contained in it

Pushing to a remote

1. In your cloned local copy of the repository (in Terminal), create a new file, add some text, and save it
2. Add and commit your new file
3. Push these changes to the remote repository with `git push`
4. Go back to Github and check that the file has been pushed successfully from your local repository

Pulling from a remote

Used when changes are made to the remote repository and you want to update your local repository to match it.

Usually you won't add new files or changes directly through Github like we're doing for this exercise; new files would get to your remote repository because a collaborator pushed them there.

1. Go back to your Github (remote repository). Add another new file and commit it
2. In Terminal, list the files in your local repository
3. Update your local repository with **git pull**
4. List the files again to check that the new file has been pulled

Questions?