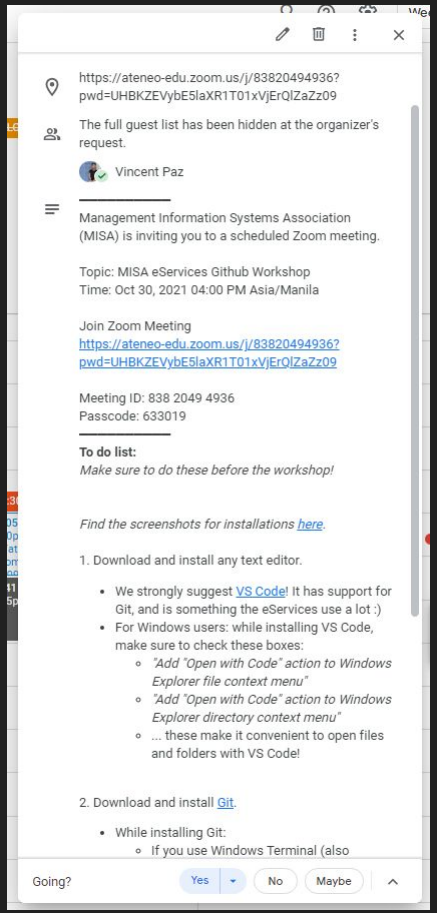
eServices@workshop MISA ~/intro (main)

$ git and github workshop

Before we start, please make sure you've installed all needed programs!

Instructions are listed in the Google Calendar event details, and have also been emailed to you!



MISA eServices

eServices@workshop MISA ~/intro (main)

$ git and github workshop

MISA eServices

eServices@workshop MISA ~/intro (main)

$ before we start ...

- Do not be afraid to ask any questions :)
- We may or may not be able to answer all questions
  - But we'll try our best to answer everything!
  - If you have unanswered questions, email me at sid.paz@obf.ateneo.edu or messenger fb.com/paz10s
- No need to take notes!
  - We'll provide resources, even a **cheat sheet**!
    - https://tinyurl.com/GitHubWorkshopCheatSheet2021
  - All you need to do is listen and follow along :)

eServices@workshop MISA ~/intro (main)

$ why Git and Github?

- Why learn this?
  - Version control - **checkpoints / save points**
  - **Collaboration** with other programmers

eServices@workshop MISA ~/intro/git (main)

$ what’s Git?

- Free and open source software
- **Tracks changes** in any set of files and allows the user to choose between specific versions if need be

eServices@workshop MISA ~/intro/github (main)

$ what's Github?

- A **repository service** for Git
- Used for **collaboration** through actions such as pull requests
- People are also able to share their projects with others through Github

eServices@workshop MISA ~/terminal-commands (main)

$ terminal commands

Command	**Explanation**
ls	*lists items in current directory*
cd <path>	*go to path*
cd ..	*go back one directory*
touch <filename>	*create file, only works in git bash*

eServices@workshop MISA ~/git-tut (main)

$ git tutorial

eServices@workshop MISA ~/git-tut/commands (main)

$ basic commands

Command	Explanation
git init	*create a local git repo*
git add <filename>	*add files to staging area*
git add .	*add all files in current directory*
git add *.html	*add all files with .html extension*
git reset <filename>	*remove file from staging area*
git commit -m “<message>”	*make commit*
git commit -am “<message>”	*add modified files and make commit*
git log	*view history of commits*

eServices@workshop MISA ~/git-tut/commands (main)

$ basic commands

main

eServices@workshop MISA ~/git-tut/commands/committing (main)

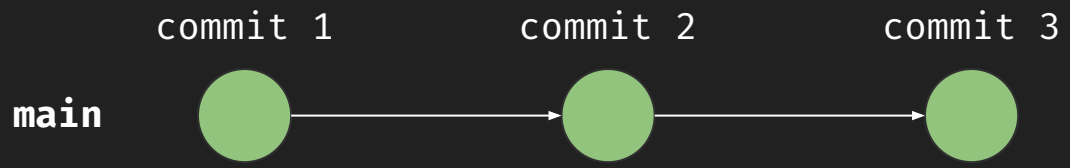
$ basic commands - committing

eServices@workshop MISA ~/git-tut/commands/committing (main)

$ basic commands - committing

eServices@workshop MISA ~/git-tut/commands/committing (main)

$ basic commands - committing



eServices@workshop MISA ~/git-tut/commands/committing (main)

$ basic commands - committing

Command	Explanation
`git reset`	*unstage files*
`git reset HEAD~1`	*go back 1 commit* • *HEAD refers to the current commit*
`git reset <commit hash>`	*go back to commit corresponding to hash*
`git reset --hard <commit hash>`	*unstage changes and remove all changes since that commit*

eServices@workshop MISA ~/git-tut/commands/committing (main)

$ basic commands - committing

eServices@workshop MISA ~/git-tut/commands/committing (main)

$ basic commands - committing

eServices@workshop MISA ~/git-tut/gitignore (main)

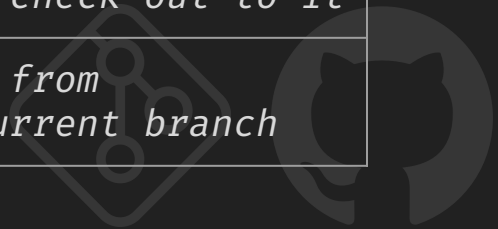
$ .gitignore file

- Create this through ***touch .gitignore***
- Type in filenames you wish for Git to ignore
- For folders: ***/<foldername>***

eServices@workshop MISA ~/git-tut/branches (main)

$ branches

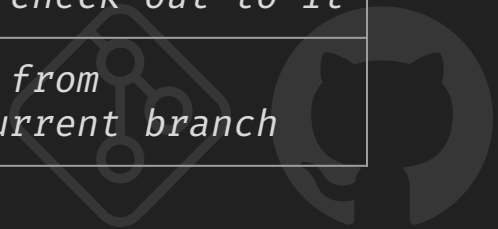
Command	Explanation
git branch	*list branches*
git branch <branchname>	*create branch*
git tag <tagname>	*name current commit*
git checkout <branchname>	*check out to branch*
git checkout -b <branchname>	*create branch and check out to it*
git merge <branchname>	*apply all commits from <branchname> to current branch*



eServices@workshop MISA ~/git-tut/branches (main)

$ branches

Command	Explanation
git branch	*list branches*
git branch <branchname>	*create branch*
git tag <tagname>	*name current commit*
git checkout <branchname>	*check out to branch*
git checkout -b <branchname>	*create branch and check out to it*
git merge <branchname>	*apply all commits from <branchname> to current branch*

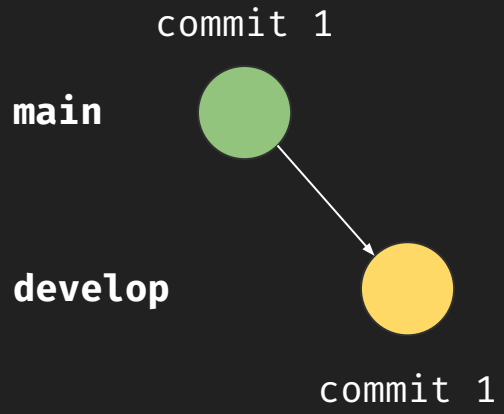


eServices@workshop MISA ~/git-tut/commands/committing (main)

$ basic commands - committing

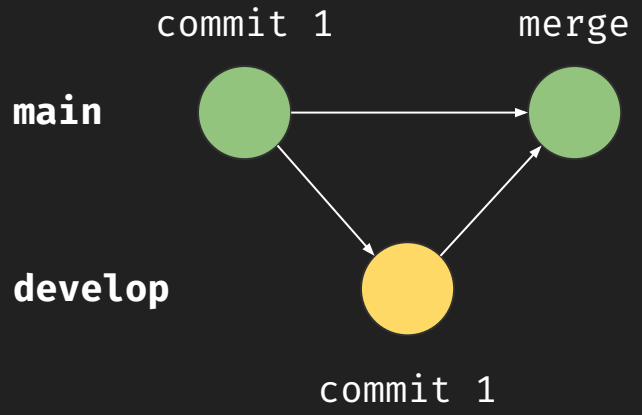
eServices@workshop MISA ~/git-tut/branches (main)

$ branches



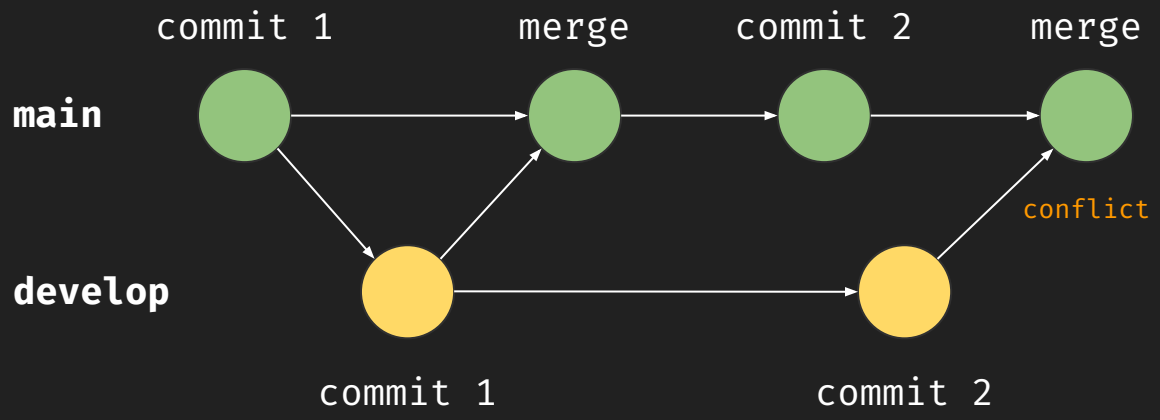
eServices@workshop MISA ~/git-tut/branches (main)

$ branches



eServices@workshop MISA ~/git-tut/branches (main)

$ branches



eServices@workshop MISA ~/github-tut (main)

$ github tutorial

eServices@workshop MISA ~/github-tut/linking-repos (main)

$ linking your local repo to Github

1. Create a remote repo in github.com
2. Use the following commands:

Command	**Explanation**
git remote add origin <link>	*link your local repo to the remote repo*
git push -u origin <branchname>	*set an upstream branch to <branchname>*
git push	*push your local branch to the remote branch*

eServices@workshop MISA ~/github-tut/cloning-repos (main)

$ cloning a remote repository

1. Create a remote repo in github.com
2. Use the following command:

Command	**Explanation**
git clone <link>	*clone remote repo to your system*

eServices@workshop MISA ~/github-exercise (main)

$ github exercise

eServices@workshop MISA ~/github-exercise (main)

$ pushing your commits to Github

1. Fork and clone the repo
2. Make your own branch. You can name it anything you want!
3. Create a <your-name-here>.txt file:
4. You can write anything in it!
5. git add, git commit, git push
6. Create a pull request
7. Wait for me to accept your request!