CS/COE 1520
Recitation Week 2

TA: Jeongmin Lee
TA: Jeong-Min Lee

- jlee@cs.pitt.edu
Office Hours and Location

• Monday: 2:00 – 4:00pm and by appointment.
• Location: 5324 Senott Square
Plan for today

- GIT demo
- JavaScript Discussion
Disclaimer

- Note that slides on Git tutorial were taken from a slide by “Radoslav Georgiev
- https://www.slideshare.net/GameCraftBulgaria/github-basics
Some basic terminology

- **git** = the shell command to work with Git
- **repo** = Repository, where the code for a given project is kept
- **commit** = verb, means push the code to the server (in Git, commit = (commit + push)
- **diff** = the difference between two versions of a file
- **SSH** = Secure SHell – Network protocol for communication between machines
- **RSA** = Rivest, Shamir, Adleman – public-key cryptography algorithm
Let’s Do It: Configure

1. Make your own GitHub Repository
   • Make an Github account first
   • Make a your own repository on Github

2. On your computer, install git installer and open a terminal/bash
   • Windows: [https://git-for-windows.github.io/](https://git-for-windows.github.io/)
   • On terminal/bash, type `>> git`
Let’s Do It: Configure

3. Configure your info
   - Name and email address

```bash
$ git config --global user.name "Firstname Lastname"
$ git config --global user.email "email@email.com"
```

- Github API token
  - On the GitHub website, click user image > “Settings” > “Personal access tokens” and click “generate new token” button.

```bash
$ git config --global github.user username
$ git config --global github.token the_token
```
Personal access tokens

Need an API token for scripts or testing? Generate a personal access token for quick access to the GitHub API.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.
Let’s Do It: Create a repo

• Okay, let’s add it!
  
  ```
  $ git add omgrofl.txt
  ```

• And commit it 😊
  
  ```
  $ git commit –m ‘This is a commit message’
  Some gitorish output
  ```

• And for the sake of learning, let’s edit it again
  
  ```
  $ echo “roflcopter” >> omgrofl.txt
  $ cat omgrofl.txt
  rofllo
  rofl
  roflcopter
  ```
Let’s Do It: Create a repo

• And now, let’s see:

```sh
$ git status
```

• Outputs:

```sh
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   omgrofl.txt
```

• Almost there

```sh
$ git add omgrofl.txt
$ git status
```
How it works? Staging area.
What about Github? Remotes?

• Okay, you suck, there’s nothing @ Github
• Damn. Enter magic!

```
$ git remote add origin git@github.com:UserName/ProjectName.git
```

• **Git commits locally, pushes remotely !!!!!!**
• Add the remote when the repo is created (git init, remember? 😊)

```
$ git remote add [name] [url]
```

• Want to see the remotes?

```
$ git remote -v
```
What about Github? Push it up

• Okay, we have committed and added a remote to Github. It’s time to push 😊

  $ git push origin master
  Enter passphrase! 😊

• Open up the repo in Github and enjoy ^_^
• The push command explained:

  $ git push [remote_name] [branch]

• Branches are black magic for later 😊
• There’s a big chance that the branch you are pushing to will be named “master”
RECAP

• git init
• git add somefile.txt
• git commit –m "msg"
• git push origin master
Pull from Remote to Local

- You can get what’s freshly in remote repository to your local with git pull command

```
$ git pull origin master
```

```
$ git pull [remote_name] [branch]
```
Quiz

• Ungraded