Git and Github Example and Tutorial
Sync with Multiple Repositories, Store Password etc.

Go back to fan’s Tex4Econ and Miscellaneous Repository.

1 Download Programs

1. Install git for windows: after install, try “cd ~/PyFan”
2. Install atom
3. Might need to install putty possibly

2 git and Github Security Set-up

1. open up git bash
2. generate rsa
   - *ssh-keygen -t rsa -C “wangfanbsg75@live.com”*
   - when prompted, do not enter “file in which save the key”, when prompted for passphrase, enter “WHATEVERPASSWORDIS”
3. copy key
   - WINDOWS: *clip < ~/.ssh/id_rsa.pub*
   - LINUX: *cat ~/.ssh/id_rsa.pub*
4. log on to github ssh section, generate new ssh rsa key
   - SSH and GPG keys, choose New SSH key, paste in clipped text.

```
# inside git bash
ssh-keygen -t rsa -C "wangfanbsg75@live.com"
# this copies the text in the .pub file generated
clip < ~/.ssh/id_rsa.pub
```

3 Start and Sync Key Repositories

Inside git bash (open as *administrator*), Sync Several Key Repositories that should be synced on all computers:

1. fanwangecon.github.io: Github root repo
2. PyFan: Private Repo with Python Support Files
3. Tex4Econ: Latex, installation, and various other support files, public repo.
4. R4Econ: Public repo, R package, R research programs
5. M4Econ: Public repo, Matlab package, Matlab examples and research files
6. Py4Econ: Public repo, Python package, Python research programs
7. Teaching: Private teaching repository

Other repositories can be synced when needed on an ad-hoc basis. The repositories above are essential repositories.

3.1 Repo Folders and Global config

Remember git bash or bash should be in windows opened as *administrator*.

```
# cd to root folder
cd ~
# generate all needed key repositories
mkdir fanwangecon.github.io PyFan Tex4Econ R4Econ M4Econ Py4Econ Teaching

# Set global config settings, in ~/.gitconfig
git config --global user.name "Fan Wang"
git config --global user.email wangfanbsg75@live.com
```

3.2 Initalize and add remote repo for each:

```
# Initialize Repositories
cd ~/fanwangecon.github.io
git init
git remote add github git@github.com:fanwangecon/fanwangecon.github.io.git
cd ~/PyFan
git init
git remote add github git@github.com:fanwangecon/PyFan
cd ~/Tex4Econ
git init
git remote add github git@github.com:fanwangecon/Tex4Econ
cd ~/R4Econ
git init
git remote add github git@github.com:fanwangecon/R4Econ
cd ~/M4Econ
git init
git remote add github git@github.com:fanwangecon/M4Econ
cd ~/Py4Econ
git init
git remote add github git@github.com:fanwangecon/Py4Econ
cd ~/Teaching
git init
git remote add github git@github.com:fanwangecon/Teaching
```

3.3 Sync Repo in SSH Secure Session

Pull latested from multiple repositories. Just Paste the following lines.

1. start ssh-agent secure session
2. pull from multiple Repositories upon start of working session.
3. commit changes as work on one computer
4. upon leaving a computer with commited changes, push all
5. create bash file to make it easier to repeat steps

Start and end ssh session:

```
# To avoid having to enter password each time, start background authentication agent.
eval "$(ssh-agent)"
ssh-add ~/.ssh/id_rsa

# to stop the ssh session
kill $SSH_AGENT_PID
```

Pull from repos:

```
# Pull from Repositories, do these one by one first
cd ~/fanwangecon.github.io
git pull github master
cd ~/PyFan
git pull github master
cd ~/Tex4Econ
git pull github master
cd ~/R4Econ
git pull github master
cd ~/M4Econ
git pull github master
cd ~/Py4Econ
git pull github master
cd ~/Teaching
git pull github master
```

Push from repos committed changes

```
# Push to Repositories, do these one by one first
cd ~/fanwangecon.github.io
git push -u github master
cd ~/PyFan
git push -u github master
cd ~/Tex4Econ
git push -u github master
cd ~/R4Econ
git push -u github master
cd ~/M4Econ
git push -u github master
cd ~/Py4Econ
git push -u github master
cd ~/Teaching
git push -u github master
```

to create bash file to store these (from git bash in windows), generate a pull_repos and push_repos file.

```
# store bash file in ~\PyFan\bin
mkdir ~/PyFan/bin
cd ~/PyFan/bin

# create bash file
vim pull_repos
vim push_repos
```

What *pull_repos* could look like, *push_repos* will look similar:

```
# paste the text below over, comfirm bash loc with: which bash
#!/usr/bin/bash
echo start fan github pull

# Security, will prompt for password once.
eval "$(ssh-agent)"
ssh-add ~/.ssh/id_rsa

# Pull from Persistent Repos
cd ~/fanwangecon.github.io
```

```
git pull github master
cd ~/PyFan
git pull github master
cd ~/Teaching
git pull github master

# Stop Secured SSH
kill $SSH_AGENT_PID

# Open Atom with the Projects
atom ~/fanwangecon.github.io ~/PyFan ~/Teaching
```

Possibly do this:

```
# change permission to make file an executable
chmod u+x ~/PyFan/bin/pull_repos
chmod u+x ~/PyFan/bin/push_repos
# execute script
bash ~/PyFan/bin/pull_repos
```

3.4 Quick Pull and Push Single Repo and Delete

Suppose temporarily need to work on a repo, but don't want it to take up too much space. Create two bash files with these single repo info

1. initialize
2. secure and pull
3. push
4. delete

Steps 1 and 2, note do not leave space when defining bash variables around the equality symbol:

```
#!/usr/bin/bash
STRREPO='EconStatClass'
echo pull single repository
echo going to pull from $STRREPO

# generate all needed key repositories
cd ~
mkdir $STRREPO

# Set global config settings, in ~/.gitconfig
git config --global user.name "Fan Wang"
git config --global user.email wangfanbsg75@live.com

# Initialize
cd ~
cd $STRREPO
git init
git remote add github git@github.com:fanwangecon/$STRREPO

# Secure
eval "$(ssh-agent)"
ssh-add ~/.ssh/id_rsa

# Pull
```

```
cd ~
cd $STRREPO
git pull github master

# Open Repository in Atom
atom ~/$STRREPO
```

Steps 3 and 4:

```
# push
cd ~
cd $STR_REPO
git push -u github master

# remove fully
cd ~
cd $STR_REPO
git rm -r $STR_REPO
```

bash scripts

```
# change permission to make file an executable
chmod u+x ~/PyFan/bin/pull_one
chmod u+x ~/PyFan/bin/push_one
# execute script
bash ~/PyFan/bin/pull_one
```