Git & SVN

Git Command

command	description
`git clone url [dir]`	copy a git repository so you can add to it
`git add files`	adds file contents to the staging area
`git commit`	records a snapshot of the staging area
`git status`	view the status of your files in the working directory and staging area
`git diff`	shows diff of what is staged and what is modified but unstaged
`git help [command]`	get help info about a particular command
`git pull`	fetch from a remote repo and try to merge into the current branch
`git push`	push your new branches and data to a remote repository
others: `init, reset, branch, checkout, merge, log, tag`	

Get ready to use Git

Set the name and email for Git to use when you commit:

```
1   git config --global user.name "<username>"
2   git config --golbal user.email <email_addres>
```

Verify personal information in Git

```
1   git config --list
```

Create a local copy of a repo

Clone an already existing repo

```
1   git clone <url> [<local dir name>]
```

Create a Git repo in your current directory

```
git init
git add <file_address_related_to_current_address>
git add . // 添加所有文件
git commit -m "<commit_message>"
```

Committing files

Add file to staging area

```
git add <file_address_related_to_current_address>
```

Move staged changes into the repo we commit

```
git commit -m "<commit_message>"
```

To unstage a change on a file before you have committed it

```
git reset HEAD -- <filename>
```

Remove unstaged changes

```
git checkout -- <filename>
```

Fetch net data

```
git fetch origin // check whether there is an update in remote server
git merge origin/master // merge file
```

Status and Diff

To view the status of your files in the working directory and staging area

```
git status
```

To see what is modified but unstaged

```
git diff
git diff --cached // To see staged changes
```

Viewing logs

To see a log of all changes in your local repo

```
git log
git log --oneline // to show a shorter version
git log -5 // to show only the 5 most recent updates, etc.
```

Pulling and Pushing

```
git pull origin <remote_branch>
git push origin <remote_branch>
```

- `origin`: an alias for the URL you cloned from

Branch

Create a new branch

```
git branch <newbranch>
```

Switch to a branch

```
git checkout <newbranch>
```

Create a new branch and switch to it

```
git checkout -b <newbranch>
```

Merge

Switch to main develop branch

```
git checkout <main_develop_branch>
```

Merge

```
git merge <newbranch>
```

Submodule

```
git submodule add <link> <submodule_name>
git submodule update --init --recursive
```

SVN Command

```
// svn checkout https://subversion.ews.illinois.edu/svn/fa15-cs427/<netid> cs427
svn checkout <remote_link> <name_of_local_dir>

// commit some modification
svn commit -m "<message>"

// update your local copy with the changes on the server
svn up

// To tell svn about a new file to track
svn add <name_of_new_file>

// shows the status of files in the current svn directory
svn st

// removes a file from the set of tracked files (will be removed on the remote server as well)
svn rm

// moves a file from one directory to another (or renames if in same directory
svn mv

// diff between two revisions, or diff a file to see uncommitted local changes
svn diff
```