Class #12: git

2/20/19

CS 224 Introduction to Python   Spring 2019
Class #12: git

What is git?

British slang for someone who is annoying or incompetent
But that’s not important right now.

A tool for managing a code base that includes
- version control
- provisions for collaboration
- backup (because it’s in the cloud)

Do we have to use it for our projects?

No, but you should consider it – it’s very useful.

There is a learning curve but it isn’t too bad.

It’s like Eclipse. Mastering Eclipse takes a lot of work because it’s a complex tool. BUT learning to do simple things with Eclipse isn’t difficult.

How does it work?

git uses the concept of a repository.

<table>
<thead>
<tr>
<th>Local repository</th>
<th>Remote repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Code</td>
</tr>
<tr>
<td>Documentation</td>
<td>Documentation</td>
</tr>
<tr>
<td>Test scripts</td>
<td>Test scripts</td>
</tr>
</tbody>
</table>

Each user has a local repo

Usually on a service such as Github or Bitbucket

push

pull
How does it work?

When you edit files, changes are only in your local repo.
- **if changes are good**, you can push them to remote repo where others can access them
- **if not**, you can revert to a previous version of code (multiple ways to do this depending on desired result)

sending local changes to remote

1. determine which files have changes:

   WNG-CS-LM-003:passengers mathias$ git status
   On branch v2
   Your branch is up to date with 'origin/v2'.
   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:   coevolution.py

2. Stage changed files for commit using add:

   WNG-CS-LM-003:passengers mathias$ git add coevolution.py

3. Stage changed files for commit using add:

   WNG-CS-LM-003:passengers mathias$ git commit -m 'meaningful message'

4. Send to remote repo using push:

   WNG-CS-LM-003:passengers mathias$ git push [remote name] [branch]
pull from remote repo

To sync your repo with the most recent code on the remote repo:

```
WNG-CS-LM-003:passengers mathias$ git pull [remote name] [branch]
```

Branches

Branches provide a mechanism for experimenting with new features or significant changes without risking existing code.

Switch between branches

Your project may contain multiple branches:

To see which branch you are currently working in:

```
WNG-CS-LM-003:passengers mathias$ git branch
  * v2
```

To switch to another branch:

```
WNG-CS-LM-003:passengers mathias$ git checkout [branch]
```

You will first need to commit or discard changes in the current branch.

Is pushing always easy?

No!
Can you elaborate?

Yes!

- If remote repo has changed, your changes may be rejected.
- Don’t panic – git will instruct you to pull and merge.
- First, pull the remote branch.
- If there are no collisions, your changes will be seamlessly integrated.
- If there are collisions, git will put “pointers” in the files for you.
- You then edit the files to choose manually which changes to keep.

Any additional advice?

- GIT COMMIT
- GIT PUSH
- LEAVE BUILDING