Lab1

Python

Github

VirtualEnv

Curl

Ali Sajedi Jiangwei Yu

1- Exercises …

1. Python:

- Design and implement a class named “Student” with three methods;
- __init__(self, courseName)
- addCourseMark(self, course, mark)
- average(self)

Then test the class with an instance. The details like member variables are up to you. For example, you can use lists or dictionaries to implement the class.

You can utilize this code (that uses a dictionary) too:

```
class Student:
    courseMarks={}
    name= ""
    def __init__(self, name, family):
        …
    def addCourseMark(self, course, mark):
        self.courseMarks[course] = mark

    def average(self):
        …
```

1- Exercises (continued)

2. Github:

- Make a new online repository and push your python file (titled “*name-family.py*”) in it.
- You have to have some additional files in it.

3. VirtualEnv:

- Create a virtual environment (like the example) and install a package in it.

4. Curl:

- Make some request(s) for some arbitrary web site.

2- Tutorials

1. Python:
2. Github:
3. VirtualEnv:
4. Curl:

2-1- Python Tutorial

- Run Python programs online with help:
  - http://www.learnpython.org/

- Other resources:
  - http://docs.python.org/2/tutorial/
  - Download a Python Tutorial/Reference (pdf):
  - http://www.tutorialspoint.com/python/python_pdf_version.htm

2-2- Github Tutorial …

- Github: A version control system
- Guide to git:
  - $ apt-get install git ▭ no need
  - GUI: Download and install Github software: http://github.com/ (bottom of page)
  - Bring your Version Controlled projects to the Web: Create an account in github.com

2-2- Github Tutorial …

- http://readwrite.com/2013/09/30/understandin

- http://readwrite.com/2013/10/02/github-for-be

- http://rogerdudler.github.io/git-guide/

- Other resources:
  - http://marklodato.github.io/visual-git-guide/index-en.html
  - https://www.kernel.org/pub/software/scm/git/docs/everyday.html

2-2- Github Tutorial …

Concepts:

- **Command Line:** The computer program we use to input Git commands. On a Mac, it’s called Terminal. On a PC, it’s a non-native program that you download when you download Git for the first time (we’ll do that in the next section). In both cases, you type text-based commands, known as prompts, into the screen, instead of using a mouse.
- **Repository:** A directory or storage space where your projects can live. Sometimes GitHub users shorten this to “repo.” It can be local to a folder on your computer, or it can be a storage space on GitHub or another online host. You can keep code files, text files, image files, you name it, inside a repository.
- **Version Control:** Basically, the purpose Git was designed to serve. When you have a Microsoft Word file, you either overwrite every saved file with a new save, or you save multiple versions. With Git, you don’t have to. It keeps “snapshots” of every point in time in the project’s history, so you can never lose or overwrite it.
- **Commit:** This is the command that gives Git its power. When you commit, you are taking a “snapshot” of your repository at that point in time, giving you a checkpoint to which you can reevaluate or restore your project to any previous state.
- **Branch:** How do multiple people work on a project at the same time without Git getting them confused? Usually, they “branch off” of the main project with their own versions full of changes they themselves have made. After they’re done, it’s time to “merge” that branch back with the “master,” the main directory of the project.

2-2- Github Tutorial …

Commands:

- **git init:** Initializes a new Git repository. Until you run this command inside a repository or directory, it's just a regular folder. Only after you input this does it accept further Git commands.
- **git config:** Short for "configure," this is most useful when you're setting up Git for the first time.
- **git help:** Forgot a command? Type this into the command line to bring up the 21 most common git commands. You can also be more specific and type "git help init" or another term to figure out how to use and configure a specific git command.
- **git status:** Check the status of your repository. See which files are inside it, which changes still need to be committed, and which branch of the repository you're currently working on.
- **git add:** This does *not* add new files to your repository. Instead, it brings new files to Git's attention. After you add files, they're included in Git's "snapshots" of the repository.
- **git commit:** Git's most important command. After you make any sort of change, you input this in order to take a "snapshot" of the repository. Usually it goes git commit -m "Message here." The -m indicates that the following section of the command should be read as a message.
- **git branch:** Working with multiple collaborators and want to make changes on your own? This command will let you build a new branch, or timeline of commits, of changes and file additions that are completely your own. Your title goes after the command. If you wanted a new branch called "cats," you'd type git branch cats.
- **git checkout:** Literally allows you to "check out" a repository that you are not currently inside. This is a navigational command that lets you move to the repository you want to check. You can use this command as git checkout master to look at the master branch, or git checkout cats to look at another branch.
- **git merge:** When you're done working on a branch, you can merge your changes back to the master branch, which is visible to all collaborators. git merge catswould take all the changes you made to the "cats" branch and add them to the master.
- **git push:** If you're working on your local computer, and want your commits to be visible online on GitHub as well, you "push" the changes up to GitHub with this command.
- **git pull:** If you're working on your local computer and want the most up-to-date version of your repository to work with, you "pull" the changes down from GitHub with this command.

2-2- Github Tutorial …

- 1- Introduce yourself to Git:
  - git config --global user.name "Your Name Here"
  - git config --global user.email "your_email@youremail.com"
- 2- Go to https://github.com/ and Create and account
- 3- Login to your account
- 4- Create your own new repository there (repo1) and make it public
- 5- Go to terminal and run:
- a. mkdir repo1
- b. cd repo1
- 6- git init
- 7- ls
  - a. ls –a
  - b. ls .git –a
- 8- touch readme.txt
- 9- git status
- 10- git add readme.txt
- 11- git commit –m “first commit – added readme.txt”
- 12- Connect your local repository to your online one:
  - a. git remote add origin https://github.com/username/repo1.git
  - b. To confirm, type: git remote –v
- 13- To upload the changes to the online repo: git push –u origin master
- 14- Go to the online repository (your browser) and check the repository and readme.txt there.

2-2- Github Tutorial …

- 15- Now change the file readme.txt in the terminal:
  - a. vi readme.txt
  - b. press “I” and edit the file. Then press “Esc” and “:x” and “Enter” to save and exit.
- 16- git add readme.txt
- 17- git commit –m “second commit”
- 18- git push –u origin master
- 19- Go to the online repository (your browser) and check the repository and readme.txt there.

2-2- Github Tutorial …

- 20- Change the readme.txt in the browser.
- 21- Create a new file there in the online repository (readme2.txt) and type something in it.
- 22- Go to the terminal and create a new text file (readme3.txt).
- 23- git add readme3.txt
- 24- git commit –m “second commit”
- 25- git push –u origin master
  - a. ERROR……
- 26- git pull origin master
- 27- ls
  - a. check the files you’ve made online are pulled in your local repository.
- 28- Go to online repository and check the files there (unchanged).
- 29- git push –u origin master
- 30- Go to online repository and check the files there (Now changed).

2-3- VirtualEnv Tutorial …

- Having Different versions of Python with different libraries installed on them without interfering

- http://www.virtualenv.org/en/latest/virtualenv.html#installa
- http://simononsoftware.com/virtualenv-tutorial/

2-3- VirtualEnv Tutorial …

- sudo easy_install virtualenv □ no need
- mkdir virt_env
- Create first virtual environment:
  - ls virt_env/virt1
  - virtualenv virt_env/virt1
  - ls virt_env/virt1
- No preinstalled packages:
  - virtualenv virt_env/virt1 –no-site-packages
- Loading the virtual environment:
  - source virt_env/virt1/bin/activate
- Deactivating:
  - deactivate
- Listing all installed Python packages:
  - Yolk –l (if is not installed, run: “sudo easy_install yolk” first)
- Now you can install:
  - easy_install pylons
  - easy_install SqlAlchemy

2-4- CURL Tutorial …

- If you're developing a Webapp and want to test the back end before implementing a stupid front end, use curl.
- **curl** is a tool to transfer data from or to a server, using one of the supported protocols (HTTP, HTTPS, FTP, FTPS, TFTP, DICT, TELNET, LDAP or FILE). The command is designed to work without user interaction.
- If you write a PHP program that is designed to go and fetch a webpage from the World Wide Web you soon find out that you are not allowed to because of a sensible restriction placed on the use of fopen(), simplexml_load_file and the like.
- http://www.yilmazhuseyin.com/blog/dev/curl-tutorial-examples-usage/
- http://httpkit.com/resources/HTTP-from-the-Command-Line/
- http://www.thegeekstuff.com/2012/04/curl-examples/
- https://gist.github.com/caspyin/2288960
- Using curl in php code:
  - http://www.tuxradar.com/practicalphp/15/10/2

2-4- CURL Tutorial …

- Make a GET request without any data:
  - curl http://simplebits.com
  - curl –request GET ‘http://simplebits.com’

- curl -X PUT -H 'Content-Type: application/json' -d '{"firstName":"Kris", "lastName":"Jordan"}' echo.httpkit.com

References

All or some parts of this presentation were duplicated / changed from the References, thanks to all the authors:

1. http://www.learnpython.org/
2. http://docs.python.org/2/tutorial/
3. http://www.tutorialspoint.com/python/python_pdf_version.htm
4. http://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1#awesm=~osS3icyMU6Qmnu
5. http://readwrite.com/2013/10/02/github-for-beginners-part-2#awesm=~osSc6v8G31otYZ
6. http://rogerdudler.github.io/git-guide/
7. http://marklodato.github.io/visual-git-guide/index-en.html
8. https://www.kernel.org/pub/software/scm/git/docs/everyday.html
9. http://www.virtualenv.org/en/latest/virtualenv.html#installation
10. http://simononsoftware.com/virtualenv-tutorial/
11. http://www.yilmazhuseyin.com/blog/dev/curl-tutorial-examples-usage/
12. http://httpkit.com/resources/HTTP-from-the-Command-Line/
13. http://www.thegeekstuff.com/2012/04/curl-examples/
14. https://gist.github.com/caspyin/2288960
15. http://www.tuxradar.com/practicalphp/15/10/2
16. http://www.tutorialspoint.com/unix_commands/curl.htm
17. http://wiki.dreamhost.com/CURL_PHP_tutorial