A practical introduction to git and python
Workshop on Astronomical Data Analysis
Shaastra 2015

Poruri Sai Rahul

January 2015

git is a program(?) used for version control.

- to distribute a data set among a large number of people
- to accept changes or additions from contributors
- to keep track of the said changes.

git is not the only version control program. Mercurial (hg) and svg are other popular version control software. so, why are we learning about git? for no specific reason. In fact, we'd be happy if you try the other and tell us which fits best into your workflow.

data hosted on a central server, from which people can pull or download updates and to which people can push or upload changes. this server can be hosted on a personal computer or one can use commercial sites such as github to host their data. Why specifically github? Atlassian's bitbucket is another common host and there's no specific reason why we're using github here. I'm comfortable with it and I am talking about it. If you are interested, you can host your own ftp server that can act as the central or remote repository.

- git clone https://foo.bar/
- git pull https://foo.bar/ and then set the remote using.
- git remote add foobar https://foo.bar/
- git status to check what the status of your repository is, if there are any changes.
- *NOTE:* look at the hidden files.
- git add foo.bar, to add or prepare the file foo.bar to be committed.
- git commit -m 'foobar', to prepare the commit for pushing.
- git commit will open nano, asking you to make a comment.
- git commit -a -m 'foobar' to prepare all files being watched* to be pushed.
- git push remote, to push the commit to the remote location.

A brief introduction to vim.

- :w to save a file
- :q to quit
- :q! to quit file without saving
- :w! to quit after saving
- insert and review modes
- i to edit or insert additions or changes
- esc to revert to insert mode

- ver 2.7 and 3.3 are the most popular ones. we'll be working on python 2.7. for no specific reason. you are implored to explore 3.3
- the core of linux has dependencies in python. so never *ever* uninstall python completely
- use of virtual environment 'venv'
- package manager such as pip or easy_ install instead of apt-get install * or yum install *
- command line usage, file usage and using an IDE like pycharm, anaconda, canopy and ipython or ipython-notebook
- i'll be showing code in ipython notebooks but there's nothing special about it. It can split up code into multiple cells that can be run and checked independently. one can include plots in the notebook along with code and LaTeX

- print 'hello world'

-
  ```
  temp = 0
  for i in xrange(n):
  temp += 1
  print temp
  ```
- for i in xrange(len(vector)):

-
  ```
  import library as lib
  # include <stdio.h>
  ```
- library.module.submodule.function, called using function(foo,bar)

python and numpy arrays

- address array and an actual array in python lists compared to address location, counter and actual array in numpy arrays
- can contain anything compared to all of the numpy elements should have the same format, can be set
- list.append(item) compared to ?
- access starts from 0 to n-1 instead of 1 to n
- access to multi-dimensional arrays using array[i][j] or array[i,j]

use of convenience functions!