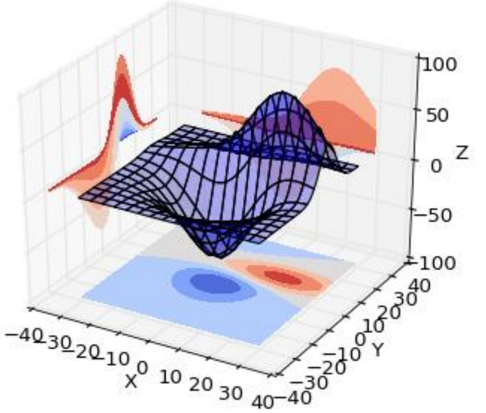
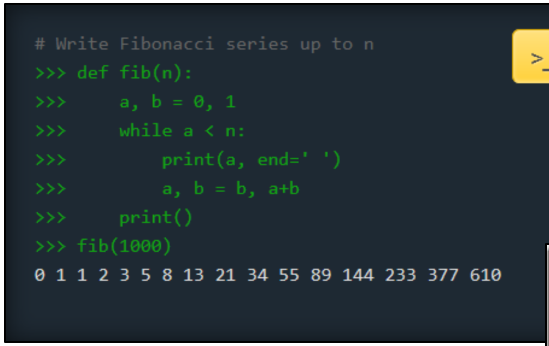
Introduction to Scientific Computing
Meeting 1

python™ git software carpentry

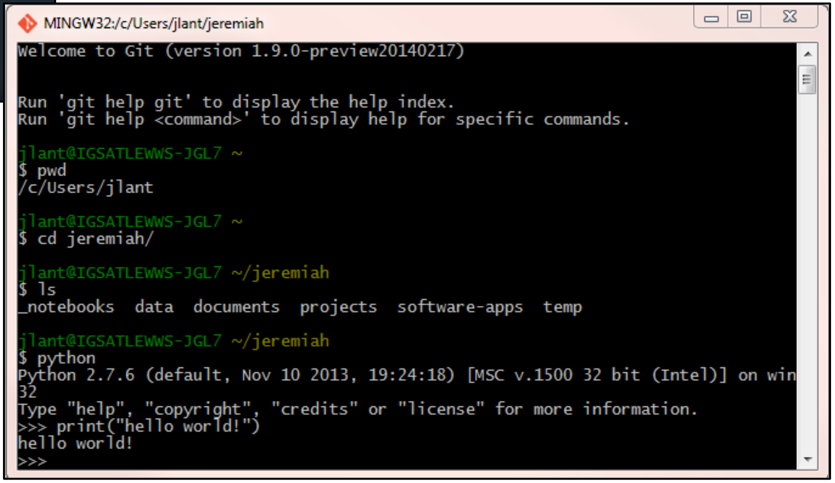
github SOCIAL CODING

Bitbucket

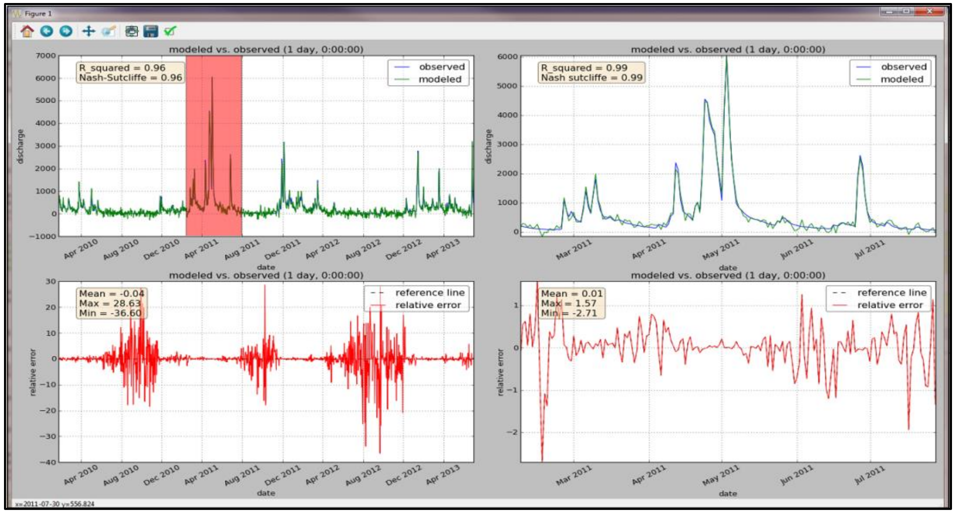




```
# Write Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
```



```
Welcome to Git (version 1.9.0-preview20140217)


Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

jlant@IGSATLEWWS-JGL7 ~
$ pwd
/c/Users/jlant

jlant@IGSATLEWWS-JGL7 ~
$ cd jeremiah/

jlant@IGSATLEWWS-JGL7 ~/jeremiah
$ ls
_notebooks  data  documents  projects  software-apps  temp

jlant@IGSATLEWWS-JGL7 ~/jeremiah
$ python
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hello world!")
hello world!
>>>
```



Jeremiah Lant, Hydrologist
USGS Kentucky Water Science Center
jlant@usgs.gov

Meeting Outline

- General introductions
- Overall objective
- General overview of the topics to be covered
- Meeting structure
- Recommended software
- A few recommended resources
- Quick look at a Python project
- Quick demo of Git Bash shell
- Questions

General Introductions

- Brief bio
- Interest in attending and participating in these meetings?
- What would you like to learn from these meetings?
- Other interests/hobbies.

Overall Objective

- Teach, guide, and build a group that has interest in learning basic scientific computing skills, learning how to program, and learning tools and techniques to help become more computationally efficient.

General Overview of Topics

1. **Unix shell** – with **Git Bash** for learning basic commands/language to communicate with a computer; basis for learning how to program.
2. **Version control** – with **Git**, **GitHub**, **Bitbucket** for learning how to track changes, reproduce, and share work efficiently.
3. **Programming** – with **Python** for learning how to program, and to write programs in a modular and testable way.
   - scientific libraries; **numpy** and **matplotlib** for learning how to work with arrays and matrices and how to create plots.

Meeting Structure

- Discuss objective(s) for the meeting.
  - ~ 1 - 2 minutes
- Give introduction to topic
  - ~ 5 – 10 minutes
- Watch a related brief video that covers the topic.
  - ~ 10 minutes
- Work through hands-on examples together.
  - ~ 20 – 30 minutes
- Discuss topic covered and answer any outstanding questions.
  - ~ 5 minutes

Recommended Software

- Notepad++ (version 6.5.5) - text editor
  - http://notepad-plus-plus.org/
- Git and Git Bash (Version 1.9.0 or 1.9.1) - version control software and bash shell for Windows
  - http://git-scm.com/
- Python (Version 2.7.6) – programming language
  - https://www.python.org/

A Few Recommended Resources

- Software Carpentry, Greg Wilson
  - http://software-carpentry.org/
- Learn Python the Hard Way, Zed Shaw
  - http://learnpythonthehardway.org/
- Pro Git, Scott Chacon
  - http://git-scm.com/book

Quick Look at a Python Project

- https://github.com/jlant-usgs/nwispy
- http://ky.water.usgs.gov/usgs/projects/jlant_program_code/nwispy/html/gallery.html#command-line-interface

Quick Demo of Git Bash Shell

```
$ echo “hello world”
$ whoami
$ pwd
$ ls
$ which python
$ python
>>> print(“hello world!”)
>>> exit()
$ start notepad++
$ start notepad++ test.txt
$ rm test.txt
```