

Rock and Paleomagnetism

SIO 247

Spring Quarter, 2016

Lecture 1

- What are rock and paleomagnetism?
- Applications
- How to take this class
- Python programming bootcamp

What is Rock magnetism?

- The study of the magnetic properties of rocks and minerals

What is Paleomagnetism?

- Application of magnetic measurements of rocks, minerals to solve geological problems
- Applied to archaeological artifacts, is called archaeomagnetism

Applications

- understanding geomagnetic field behavior in ancient times
- dating
- tectonic reconstructions
- magnetic fabrics of rocks (anisotropy)
- magnetic mineralogy, magnetic grain sizes, domain states

Why study the geomagnetic field?



- Acts as shield for solar and cosmic radiation
- controls production of cosmogenic nuclides (^{14}C , ^{10}Be ...)
- may play a role in nucleating clouds...
- navigation

How to take this class

- Attend each lecture
- Do the required reading in the textbook: Essentials of Paleomagnetism. This book is available free on line, or you can order one from me for \$30.
- Work the assignments after every lecture - they are due in one week. These are worth 40% of the grade
- There is one required field trip on April 16, 2016
- and a required lab project (this is worth 30% of the grade). For this I will need a copy of your Safety training certificate. Undergrads sign up as a volunteer with Monica Bailey (4th floor Ritter Hall) and take the online class.
- There will be a final project also worth 30% of the grade.

What you will learn in this class

- Basics of Rock and Paleomagnetism
- How to program in Python

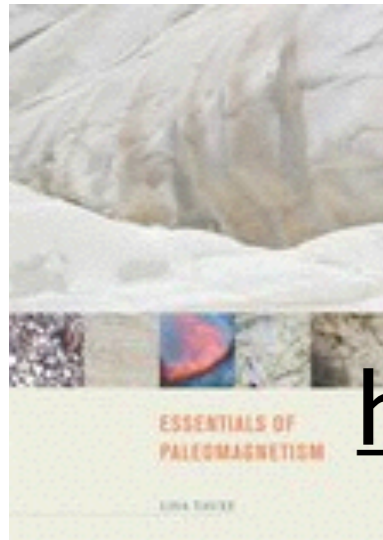
Why python?

- Flexible, freely available, cross platform
- Easy to learn and well documented
- lots of numerical, statistical and visualization packages
- Well supported and has a complete set of paleomagnetic software (PmagPy)

Getting started

- Reference materials
- Setting up your computer
 - installing essential software (Python, PmagPy)
 - survival Unix/Dos
- writing your first python notebook.

Reference Materials



Tauxe et al.,
Essentials of Paleomagnetism

<http://earthref.org/MAGIC/books/Tauxe/Essentials/>



PmagPy Cookbook

<http://earthref.org/PmagPy/>

Database

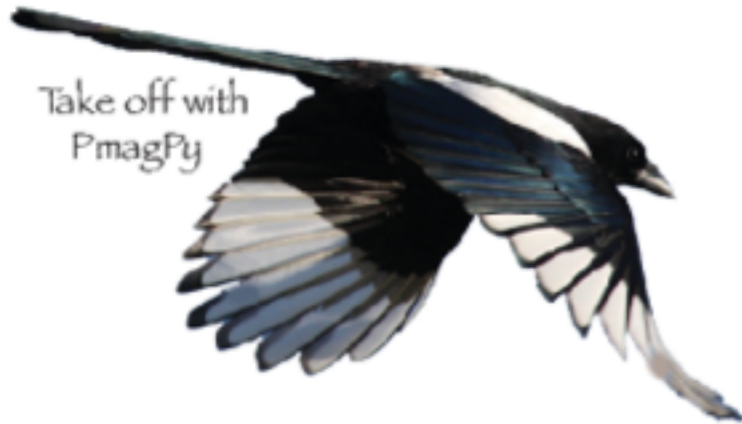
rock and paleomagnetic data
earthref.org/MagIC



Magnetics Information Consortium

- Databases
- Events
- Tools
- Publications
- Links

Contents



PmagPy Cookbook

March 11, 2016

Dear Reader,

This documentation is updated from that in the book entitled Essentials of Paleomagnetism by Tauxe et al. (2010). This cookbook was designed as a companion website to the book **Essentials of Paleomagnetism, 3rd Web Edition**. Chapter references to this companion book are, for example, "Essentials Chapter 1".

There are many chefs who contributed to this work, in particular, the MagIC Database Team (Cathy Constable, Anthony Koppers, Rupert Minnett, Nick Jarboe, Ron Shaar, and Lori Jonestrask). Nick Swanson-Hysell (UC Berkeley) contributed the demag_gui and Jupyter notebook documentation. The PmagPy project is supported by grants from the National Science Foundation.

Lisa Tauxe
 Scripps Institution of Oceanography
 La Jolla, CA 92093
 March 11, 2016
<http://magician.ucsd.edu/~ltauxe/>

Chapter 1

Table of Chapters

- 1 Installing PmagPy
- 2 Pmag GUI Tutorial
- 3 MagIC GUI Tutorial
- 4 Survival computer skills
- 5 The PmagPy software package
- 6 The MagIC database and file formats
- 7 Introduction to Python Programming
- 8 Jupyter Notebooks
- 9 Troubleshooting
- Bibliography

Table of Contents

- 1 Installing PmagPy
 - 1.1 Standalone GUI download
 - 1.1.1 OSX Standalone download
 - 1.1.2 Windows Standalone download
 - 1.2 Full PmagPy install
 - 1.2.1 Getting Python for Mac users
 - 1.2.2 Getting Python for Windows users
 - 1.2.3 Installing PmagPy
 - 1.3 Pip PmagPy install (beta)
 - 1.3.1 Getting Python
 - 1.3.2 Installing PmagPy and PmagPy-cli
 - 1.3.3 Testing your installation
 - 1.3.4 Accessing example data files
 - 1.4 Next steps
- 2 Pmag GUI Tutorial
 - 2.1 Converting magnetometer files to MagIC format
 - 2.2 Optional: Calculate geographic / tilt-corrected direction
 - 2.3 Filling in the EarthRef data
 - 2.4 Demag GUI quick start
 - 2.5 Thellier GUI quick start
 - 2.6 Upload to the database
 - 2.7 Downloading data from MagIC
 - 2.8 Preparing for MagIC

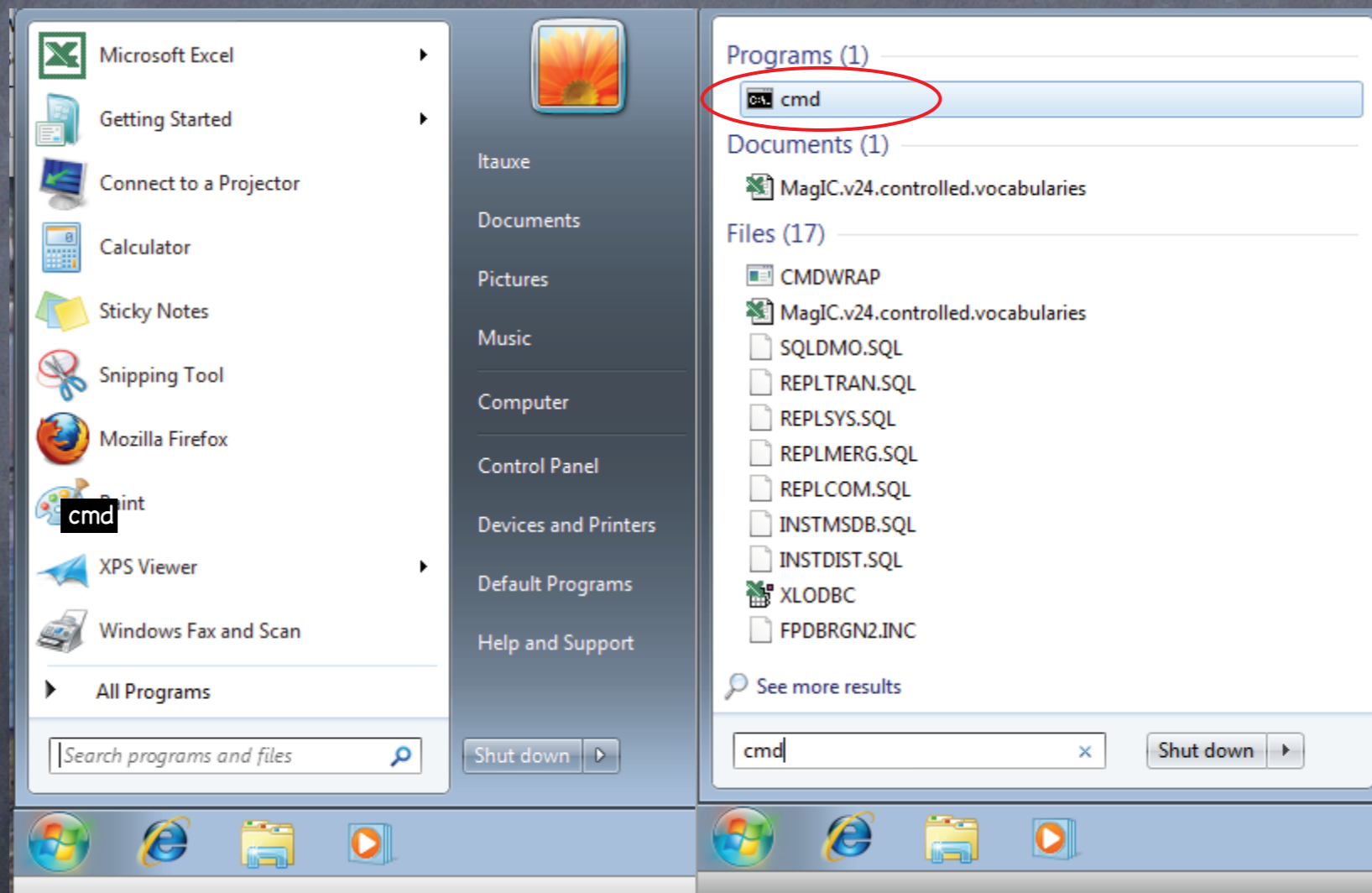
Setting up your computer

- Go to: <http://earthref.org/PmagPy/>
- Install the Full PmagPy instructions exactly
 - get the recommended Python version
 - install PmagPy following the instructions exactly.

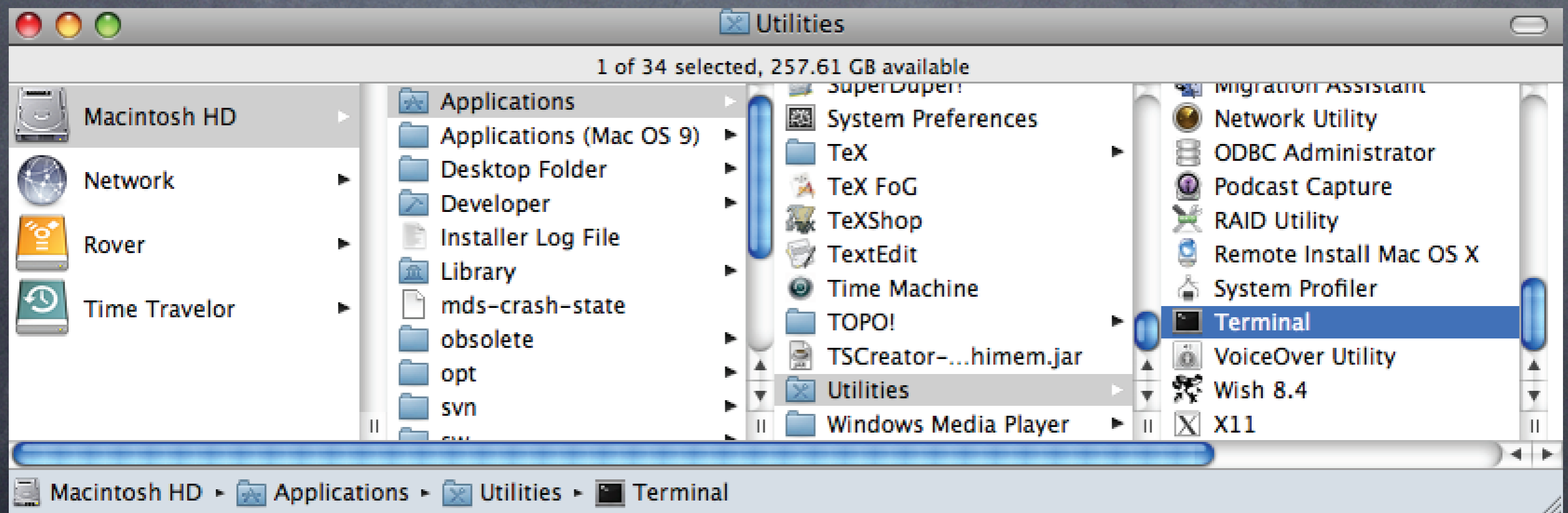
Survival Unix (Dos)

- Consult Chapter 4 of the PmagPy cookbook to learn the following:
 - Find your terminal window (command line)
 - Understand your file system
 - Learn a few essential commands

Find your command line (Windows)

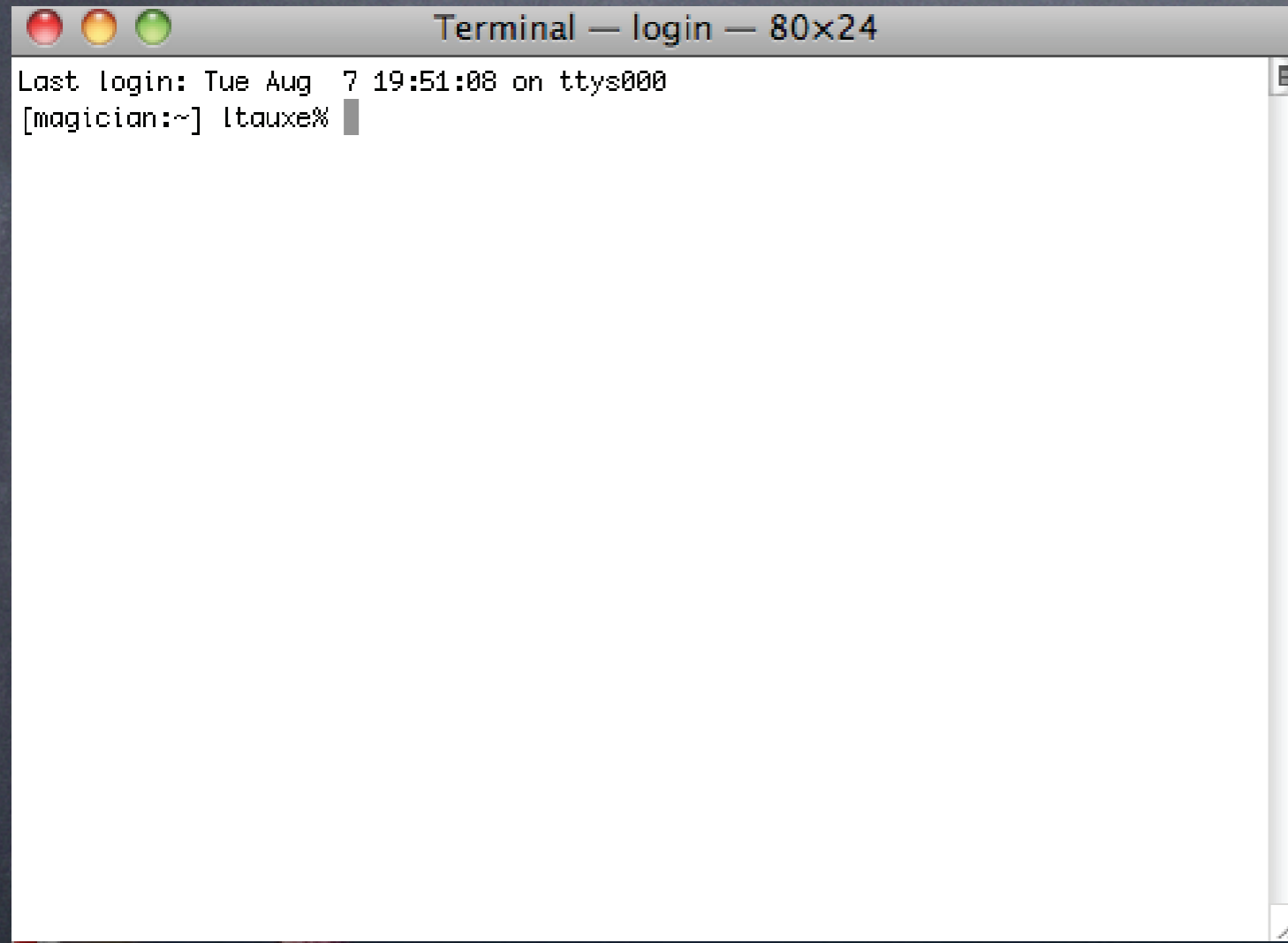


Find your command line (Mac OS)



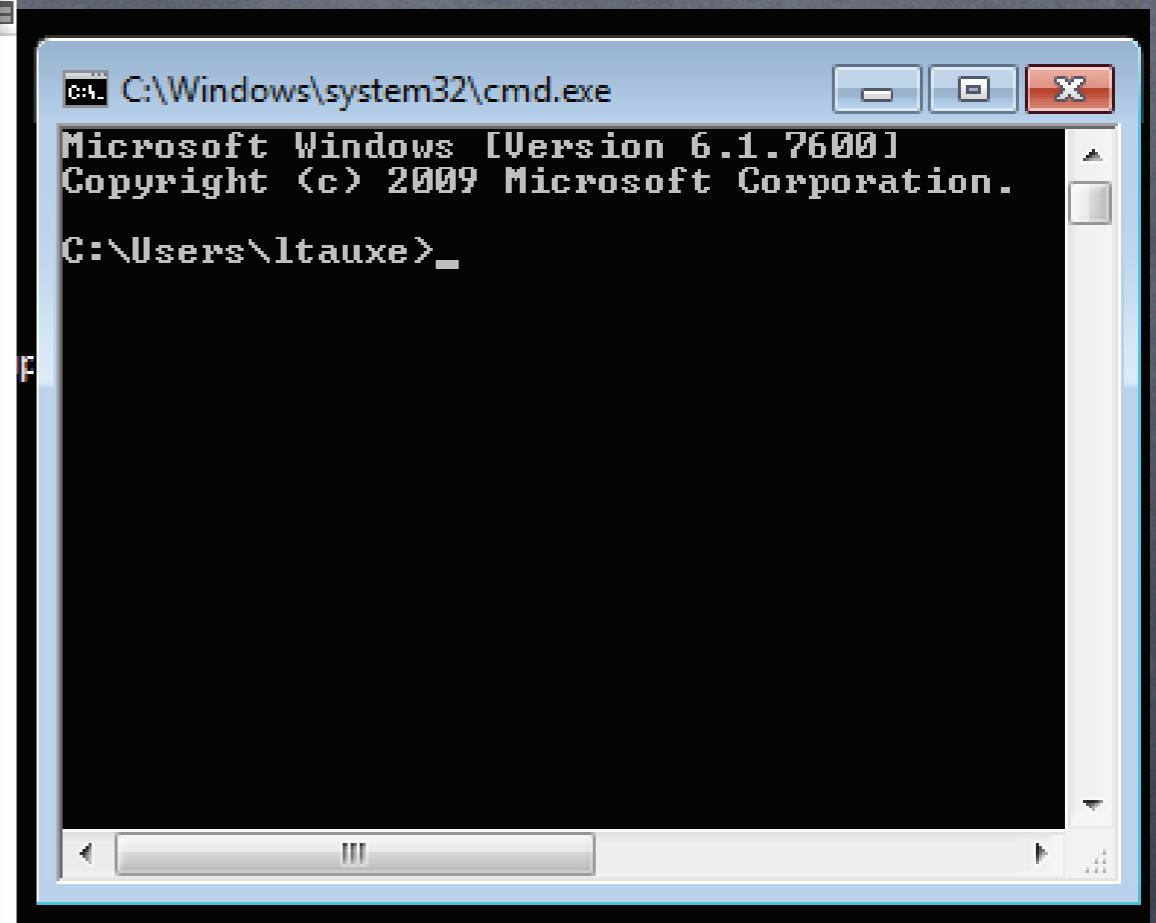
Type commands on the command line
followed by 'return' key.

Mac

A screenshot of a Mac Terminal window. The title bar reads "Terminal — login — 80x24". The terminal content shows "Last login: Tue Aug 7 19:51:08 on ttys000" followed by a prompt "[magician:~] ltauxe%". A cursor is visible at the end of the prompt.

```
Terminal — login — 80x24
Last login: Tue Aug 7 19:51:08 on ttys000
[magician:~] ltauxe%
```

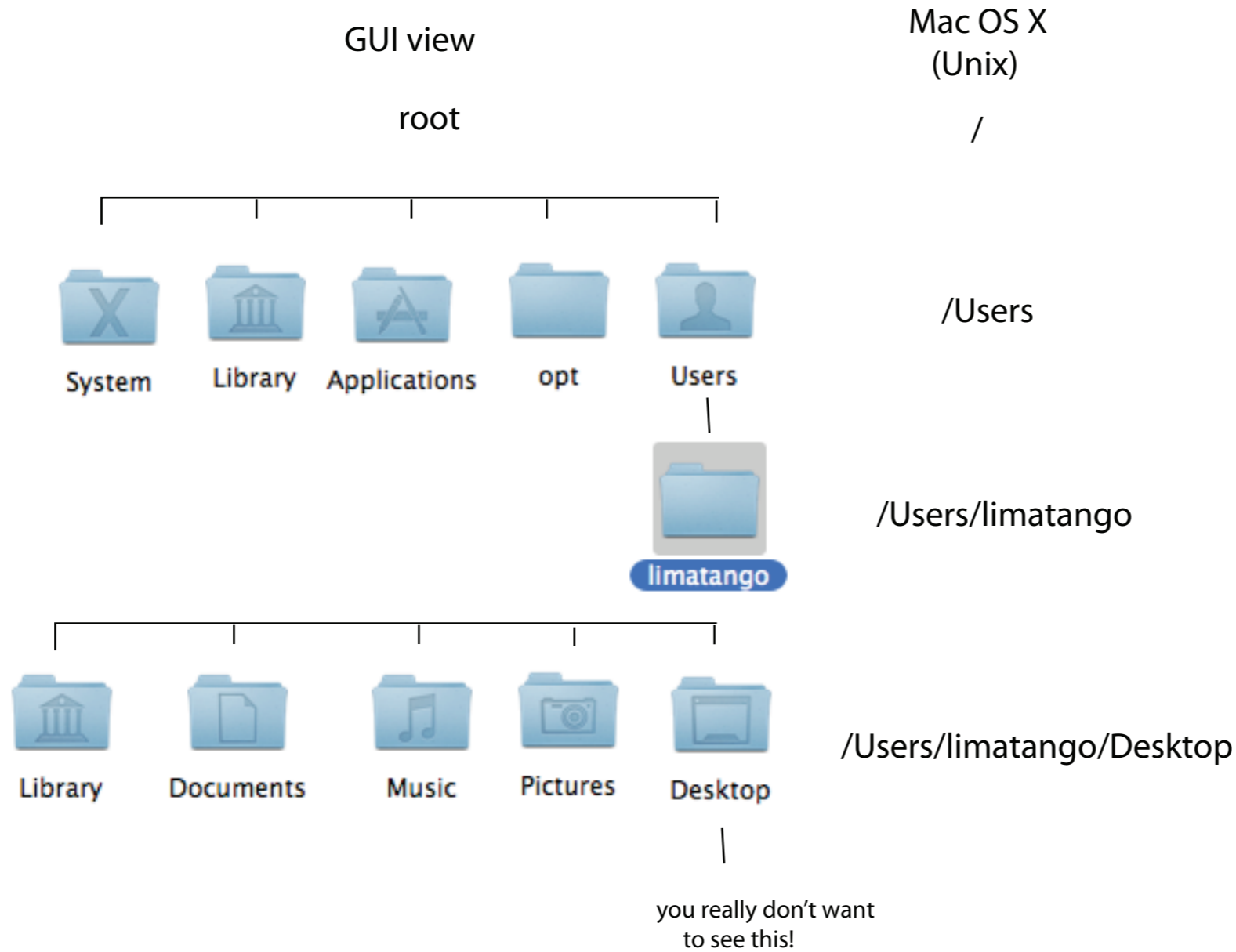
PC

A screenshot of a Windows Command Prompt window. The title bar reads "C:\Windows\system32\cmd.exe". The terminal content shows the Microsoft Windows version and copyright information, followed by the prompt "C:\Users\ltauxe>". A cursor is visible at the end of the prompt.

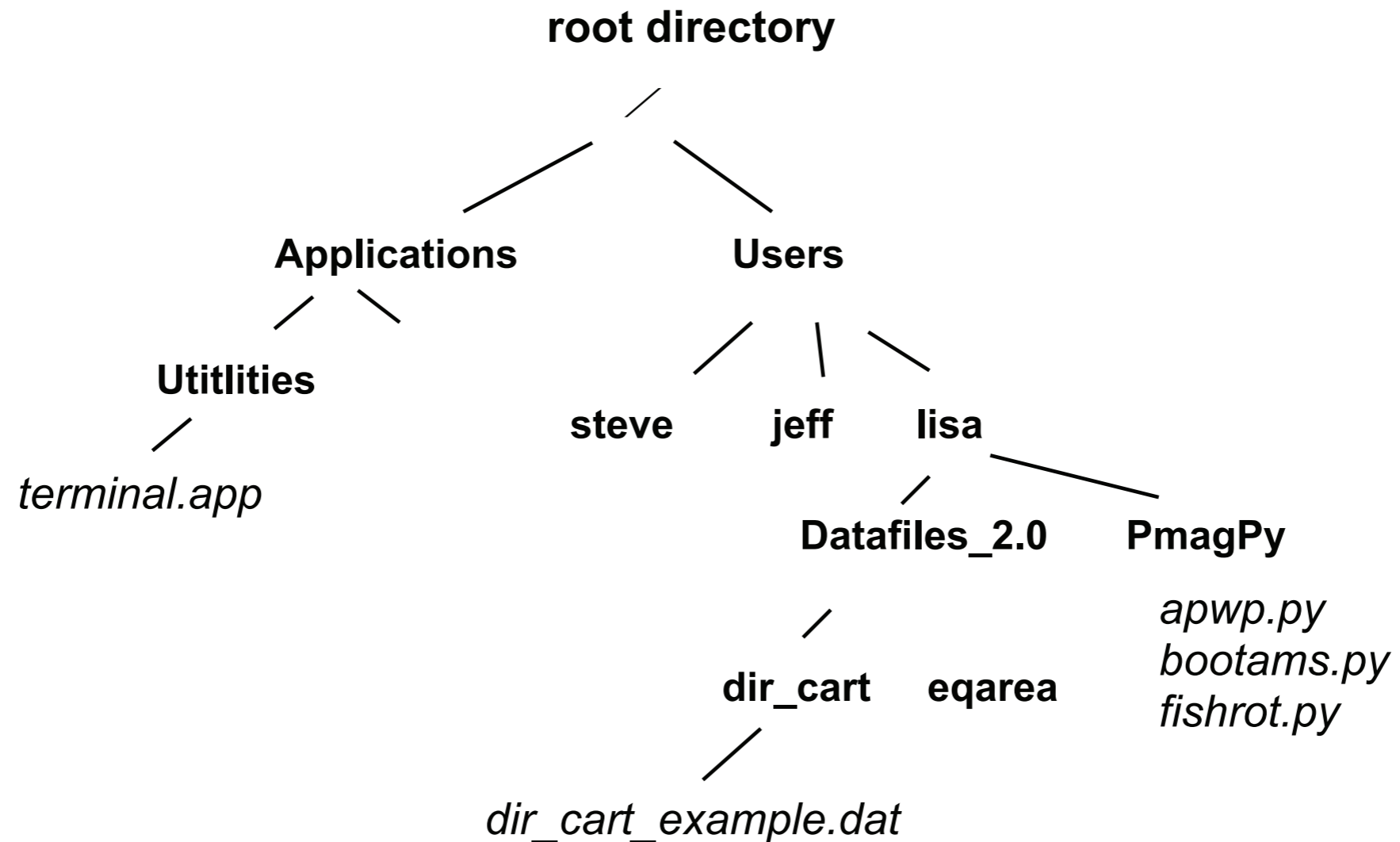
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.

C:\Users\ltauxe>
```

File system basics



The Unix View of things



Text editors

- Canopy has it's own
- TextWrangler (free)
- Notepad
- My favorite is `vi`
- My son insists on
emacs

Essential commands

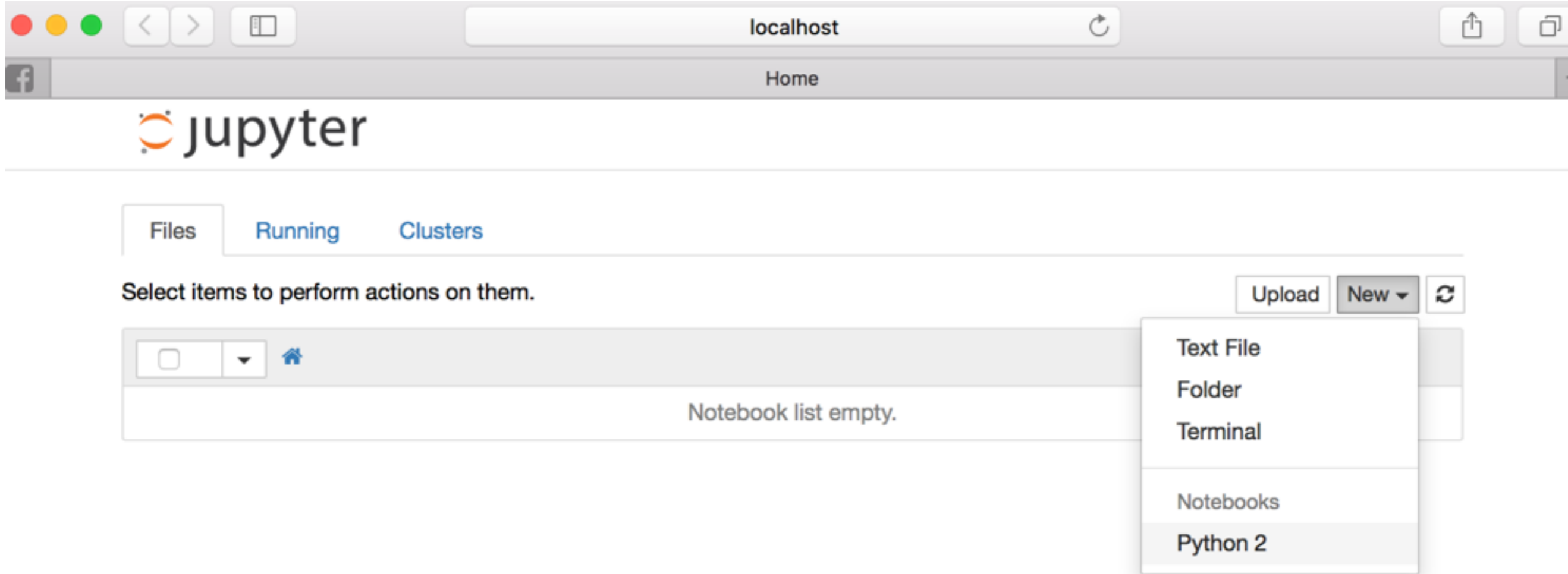
- Make a new directory: use `mkdir`
- List contents of directory: use `ls` (or `dir` in Dos)
- List the contents of a file: use `cat` (or `type` in Dos).
- Concept of redirect (`<` or `>`) and pipe (`|`)
- Moving around in the file system: use the command `cd` (change directory). short cuts: home (just `cd`), up one (`cd ../`)

Python programming bootcamp - Pt I

- Firing up a jupyter notebook
- First look at python
- First look at the numerical package NumPy

Using jupyter notebooks

- make a directory on your computer called "Homework" or something. Use a combination of `cd` and `mkdir` to do this.
- `cd` into your new Homework directory
- `type: jupyter notebook`



Click on the 'new' button and create a Python 2 notebook



New Notebook

Open...

Make a Copy...

Rename...

Save and Checkpoint

Revert to Checkpoint

Print Preview

Download as

Trusted Notebook

Close and Halt

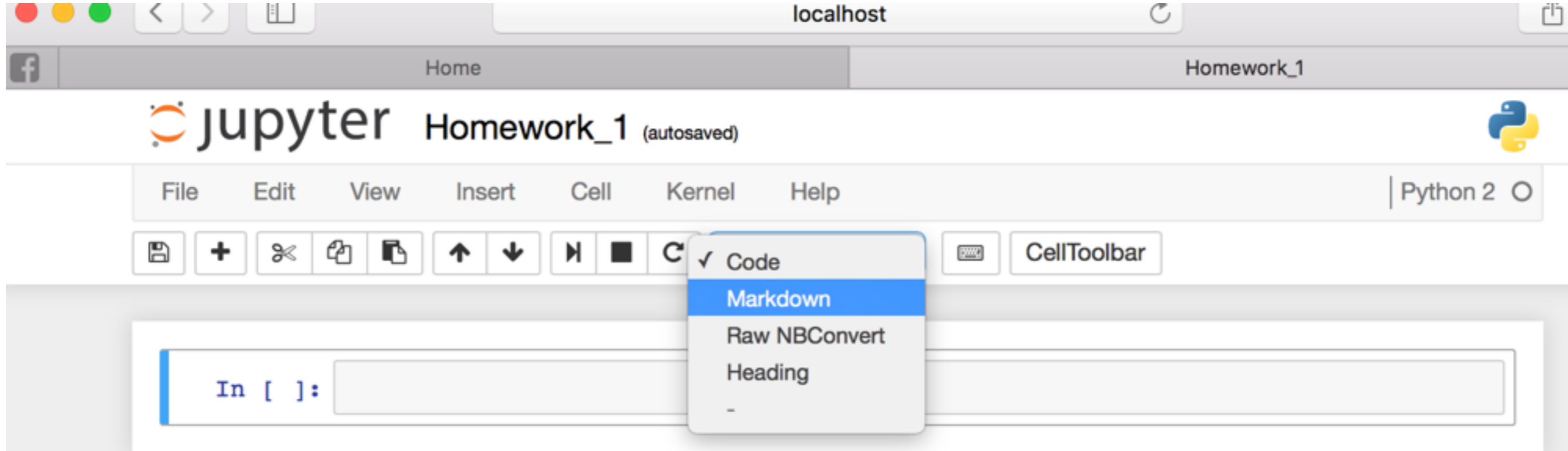


Code



CellToolbar

Rename and save your notebook.
Use a name of the form:
YOURLASTNAME_HW_1
(you don't need the file type
- it will be ipynb)



There are two basic 'cells' in the notebook:

Code: for writing python code
and

Markdown: for typesetting notes.

Change the first cell to 'Markdown' and type some notes.

To typeset them, click on the run key or run the cell under 'Cell'



File Edit View Insert Cell Kernel Help Python 2

Save Add Cut Copy Paste Up Down Run Stop Refresh Code CellToolbar

This notebook will introduce the basic concept of Jupyter notebooks.

In []:

As soon as you 'run' a cell, a new code cell is inserted below.

Now we can try some simple programming.

```
In [2]: a=2
print 'a= ',a
b=3
print 'b= ',b
c=a+b
print 'c= ',c
c+=1
print 'now c= ',c
```

```
a= 2
b= 3
c= 5
now c= 6
```

Type in the code block above in your notebook.
when you click on the 'run' button, the text
below it appears. If there is a typo, you will hear about it!

```
In [3]: a=2
print 'a= 'a
b=3
print 'b= ',b
c=a+b
print 'c= ',c
c+=1
print 'now c= ',c
```

```
File "<ipython-input-3-8e8f21c06e19>", line 2
```

```
    print 'a= 'a
           ^
```

```
SyntaxError: invalid syntax
```

- Variable names can be defined on the fly and are composed of case sensitive alphanumeric characters, including '-' and '_'.
- + adds, - subtracts, * multiplies, / divides, \% gives the remainder, ** raises to the power
- These two are fun: += and -=. They add to and subtract from respectively.
- Parentheses determine order of operation (as in any reasonable programming language).
- For math functions, we will use the NumPy module (a module is a collection of functions).

A first look at NumPy

```
In [1]: import numpy as np  
np.pi
```

```
Out[1]: 3.141592653589793
```

```
In [2]: np.sqrt(4.0)
```

```
Out[2]: 2.0
```

```
In [4]: a,b,c=2,-12,16 # note all on one line and that # means comment follows.  
(-b+np.sqrt(b**2-4.*a*c))/(2.*a) # quadratic equation
```

```
Out[4]: 4.0
```

```
In [5]: y=np.sin(np.pi/6.)  
y
```

```
Out[5]: 0.49999999999999994
```

Here is a (partial) list of some useful NumPy functions:

<code>absolute(x)</code>	absolute value
<code>arccos(x)</code>	arccosine
<code>arcsin(x)</code>	arcsine
<code>arctan(x)</code>	arctangent
<code>arctan2(y,x)</code>	arctangent of y/x in correct quadrant (**very useful!)
<code>cos(x)</code>	cosine
<code>cosh(x)</code>	hyperbolic cosine
<code>exp(x)</code>	exponential
<code>log(x)</code>	natural logarithm
<code>log10(x)</code>	base 10 log
<code>sin(x)</code>	sine
<code>sinh(x)</code>	hyperbolic sine
<code>sqrt(x)</code>	square root
<code>tan(x)</code>	tangent
<code>tanh(x)</code>	hyperbolic tangent

Note that in the trigonometric functions, the argument is in RADIANS!. You can convert from degrees to radians by multiplying by: `numpy.pi/180..` Also notice how these functions have parentheses, as opposed to `numpy.pi` which has none. The difference is that these take arguments, while `numpy.pi` just returns the value of π .

Assignment

- Go to: <http://earthref.org/PmagPy/> and install the Full PmagPy instructions exactly
- Read Chapters 1, 4 and 8 of the cookbook – install software and find your command line (terminal window).
- Download the file Tauxe_HW_1.ipynb from the class website (remove any .txt termination and make sure it is .ipynb)
- Open it in with jupyter notebook and play with it.
- Make sure you can create and save your OWN Jupyter notebook with both 'markdown' and 'code' cells.
- Turn in this and all other homework assignments as a zip file with the .ipynb file and all other files required to run it. email the zip file with the name: YOURLASTNAME_HW_X.zip
- The homework is due by class time on Friday! Start now so we can debug any problems now.