RESEARCH TOOLS 2011
LECTURE 10

2011-Sept-29
Kurt Schwehr
http://schwehr.org

UNH CCOM/JHC

QGIS, bash script, Healy animated gif, ipython, matplotlib
Setup for today's class

You can do this one of 3 ways. You pick.

- Open a terminal and paste in the command
- Open a shell inside of emacs and paste in the command

```
mkdir -p ~/class/10
cd ~/class/10
wget http://vislab-ccom.unh.edu/~schwehr/rt/src/10-qgis-bash-python.org
```

Open the org file for lecture 10.

Creating today's work log entry

I will walk through creating an entry for today's research tools class. Follow along. I have pre-written some of the entry so I don't try to type it all in front of the class. You can include the text from another file into the current location in a buffer by doing `C-x i`

Who edited their bash alias?
/home/researchtools/class/10:
total used in directory 24 available 11681524
drwxr-xr-x 2 researchtools researchtools 4096 2011-09-29 09:15 .
drwxr-xr-x 4 researchtools researchtools 4096 2011-09-29 09:14 ..
-rw-r--r-- 1 researchtools researchtools 15081 2011-09-29 07:52 10-qgis-bash-python.org
It is important to reboot the virtual machine after doing an update. It is not strictly required and most updates do not need the Ubuntu virtual machine to reboot, but it is a good habit.

```bash
# BEGIN_SRC sh
sudo reboot
# END_SRC
```

* If the date of your virtual machine is way off

If the date of your virtual machine is way off from when "now" really is, use this command to jump the sense of time to get in sync with now. The "wilmot" computer is the UNH time server.

```bash
# BEGIN_SRC sh
sudo ntpdate ntp.ubuntu.com
# END_SRC
```

The output should look something like this, but with much bigger jumps.

```bash
# BEGIN_EXAMPLE
29 Sep 06:14:35 ntpdate[9861]: adjust time server 91.189.94.4 offset 0.045046 sec
# END_EXAMPLE
```

Today, we will get a brief view of Quantum GIS (QGIS), look at shell variables, create a "animated GIF" movie from the USCG Ice Breaker Healy's Aloftcon camera, and try out ipython with matplotlib.

* Setup for today's class

You can do this one of 3 ways. You pick.

- Open a terminal and paste in the command
- Open a shell inside of emacs and paste in the command

```bash
#BEGIN_SRC sh
mkdir -p ~/class/10
cd ~/class/10
wget http://vislab-ccom.unh.edu/~schwehr/rt/src/10-qgis-bash-python.org
#END_SRC
```

Open the org file for lecture 10.

* Creating today's work log entry

I will walk through creating an entry for today's research tools class. Follow along. I have pre-written some of the entry so I don't try to type it all in front of the class. You can include the text from another file into the current location in a buffer by doing \(C-x i=\)

* Who edited their `.bash_aliases`?

Did you watch all 4 YouTube videos? If you have not, you really need to watch them right away. Seriously!

http://www.youtube.com/playlist?list=PL7E11B34616530F5E

---

Class 10: QGIS, bash v...  researchtools@ubuntu  emacs23@ubuntu
mkdir -p ~/class/10

Open the org file for lecture 10.

* Creating today's work log entry

I will walk through creating an entry for today's research tools class. Follow along. I have pre-written some of the entry so I don't try to type it all in front of the class. You can include the text from another file into the current location in a buffer by doing =C-x i=

* Who edited their .bash\_alias?

- [X] export to mp3
- [X] export to m4a
- [X] export to ogg
- [X] upload to web to audio
- [X] update HEADER.org and run "make push"

became just over 1 hour. Used audacity for the editing.
mkdir -p ~/class/10
cd ~/class/10
wget http://vislab-ccom.unh.edu/~schwehr/rt/src/10-qgis-bash-python.org

Open the org file for lecture 10.

* Creating today's work log entry

I will walk through creating an entry for today's research tools class. Follow along. I have pre-written some of the entry so I don't try to type it all in front of the class. You can include the text from another file into the current location in a buffer by doing =C-x i=

* Who edited their .bash\_alias?

- [ ] Make an entry for Geo0c. Make sure to tag with "teaching"
- [ ] Export the log entry and see how it looks
- [ ] Have the students create ~/class/10.
  - [ ] Put the org file in that directory.
  - Having the org file for the lecture somewhere else was too confusing
- [ ] Run through making journal entry
- [ ] Demo QGIS using the kml from lecture 9
- [ ] Show bash variables and how they are somewhat strange
- [ ] Walk through the syntax of the for loop in bash
- [ ] Using imagemagick convert to make an animated gif
- [ ] Get to python ASAP! Start people into =ipython -pylab=

---

researchtools-schwehr.org  Bot  L27 (Org)
* Sept 28, NH <2011-09-29 Thu>

** An entry

** another entry

* Sept 28, NH <2011-09-29 Thu>

** Edit the audio for lecture 9

Edited the audio file from the sansa clip: VORC012.WAV. 2 hours became just over 1 hour. Used audacity for the editing.

- [X] export to mp3
- [X] export to m4a
- [X] export to ogg
- [X] upload to web to audio
- [X] update HEADER.org and run "make push"

* Sept 29, CCOM, NH

- [ ] Make an entry for Geo0c. Make sure to tag with "teaching"
- [ ] Export the log entry and see how it looks
- [ ] Have the students create ~/class/10.
- [ ] Put the org file in that directory.

<table>
<thead>
<tr>
<th>August 2011</th>
<th>September 2011</th>
<th>October 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Su Mo Tu We Th Fr Sa</td>
<td>Su Mo Tu We Th Fr Sa</td>
<td>Su Mo Tu We Th Fr Sa</td>
</tr>
<tr>
<td>1 2 3 4 5 6</td>
<td>1 2 3 1 2 3</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>7 8 9 10 11 12 13</td>
<td>4 5 6 7 8 9 10</td>
<td>9 10 11 12 13 14 15</td>
</tr>
<tr>
<td>14 15 16 17 18 19 20</td>
<td>11 12 13 14 15 16 17</td>
<td>16 17 18 19 20 21 22</td>
</tr>
<tr>
<td>21 22 23 24 25 26 27</td>
<td>18 19 20 21 22 23 24</td>
<td>23 24 25 26 27 28 29</td>
</tr>
<tr>
<td>28 29 30 31</td>
<td>25 26 27 28 29</td>
<td>30 31</td>
</tr>
</tbody>
</table>

< Calendar | ? info / o other / | today | Thu, Sep 29, 2011 | >

** Date+time [2011-09-29]: => <2011-09-29 Thu>
** An entry

** another entry

* Sept 28, NH <2011-09-29 Thu>

** Edit the audio for lecture 9

Edited the audio file from the sansa clip: VORC012.WAV. 2 hours became just over 1 hour. Used *audacity* for the editing.

- [X] export to mp3
- [X] export to m4a
- [X] export to ogg
- [X] upload to web to *audio*
- [X] update HEADER.org and run "make push"

* Sept 29, CCOM, NH <2011-09-29 Thu>

- [ ] Make an entry for Geo0c. Make sure to tag with "teaching"
- [ ] Export the log entry and see how it looks
- [ ] Have the students create ~/class/10.
- [ ] Put the org file in that directory.
- Having the org file for the lecture somewhere else was too confusing
- [ ] Run through making journal entry
- [ ] Demo QGIS using the kml from lecture 9
- [ ] Show bash variables and how they are somewhat strange
- [ ] Walk through the syntax of the for loop in bash
- [ ] Using imagemagick convert to make an animated gif
- [ ] Get to python ASAP! Start people into -ipython -pylab-
3.2 Research Tools - QGIS, shell, ipython

Talked all about gravity

- [X] Make an entry for GeoOc. Make sure to tag with "teaching"
- [ ] Export the log entry and see how it looks
- [ ] Have the students create ~/.class/10.
  o [ ] Put the org file in that directory.
  o Having the org file for the lecture somewhere else was too confusing
- [ ] Run through making journal entry
- [ ] Demo QGIS using the kml from lecture 9
- [ ] Show bash variables and how they are somewhat strange
- [ ] Walk through the syntax of the for loop in bash
- [ ] Using imagemagick convert to make an animated gif
- [ ] Get to python ASAP! Start people into ipython -pylab

Author: Kurt Schwehr

Date: 2011-09-29 11:18:46 EDT

HTML generated by org-mode 7.4 in emacs 23
- [ ] Run through making journal entry
- [ ] Demo QGIS using the kml from lecture 9
- [ ] Show bash variables and how they are somewhat strange
- [ ] Walk through the syntax of the for loop in bash
- [ ] Using imagemagick convert to make an animated gif
- [ ] Get to python ASAP! Start people into -ipython -pylab-
class. Follow along. I have pre-written a script to type it all in front of the class to make it easier to work on.

* Who edited their .bash\_alias?

Did you watch all 4 YouTube videos? Send me some responses. Seriously. (Links)

http://www.youtube.com/playlist?list="

* Using QGIS to view a KML

Copy the kml and xy files from the lecture folder.

```bash
#BEGIN_SRC sh
cp ../*/09/2007-boston-construction.kml
cp ../*/09/2007-boston-construction.xy
ls -l
#END_SRC
```

If you do not have the lecture 09 results files, you can download them like this:

```bash
#BEGIN_SRC sh
bunzip2 *.bz2
#END_SRC
```

Start Quantum GIS (qgis):

Applications -> Science -> Quantum GIS

---

** 10-qgis-bash-python.org 9% L58 (Org)-------------

Auto-saving...done
Copy the kml and xy files from the lecture 09 directory.

```bash
#+BEGIN_SRC sh
cp ..//09/2007-boston-construction.kml .
cp ..//09/2007-boston-construction.xy .
ls -l
#+END_SRC
```

If you do not have the lecture 09 results files, you can download them like this:

```bash
#+BEGIN_SRC sh
bunzip2 *.bz2
#+END_SRC
```

Start Quantum GIS (`qgis`):

Applications -> Science -> Quantum GIS

---

Thursday, September 29, 11
Open up a shell inside of emacs by doing:

- Split the window: C-x 2
- Start the shell: M-x shell

```sh
# Set a variable
testing=123

# Print the variable
echo $testing
# 123

# Start a new bash shell inside the original one
bash
```
#BEGIN_SRC sh
# Set a variable
testing=123

# Print the variable
echo $testing
  # 123

# Start a new bash shell inside the original one
bash

---

Auto-saving... done
researchtools@ubuntu:$ testing="hello world"
researchtools@ubuntu:$ echo $testing
hello world
researchtools@ubuntu:$ bash
researchtools@ubuntu:$ echo $testing

researchtools@ubuntu:$ exit
exit
researchtools@ubuntu:$ export testing=123
researchtools@ubuntu:$ echo $EDITOR

researchtools@ubuntu:$ export EDITOR=emacs
researchtools@ubuntu:$ bash
researchtools@ubuntu:$ echo $testing
123
researchtools@ubuntu:$ less ~/.bashrc

U:***  *shell*  Bot L26  (Shell:run)

# See that "testing" is not set. If there is no variable, bash gives
# an empty string
echo $testing

# quit back to the main bash shell
exit

# Set testing to have a value that will be inherited
export testing="hello world"
bash

# Now see that the exported variable went through
echo $testing
# hello world
#END_SRC
# don't put duplicate lines in the history. See bash(1) for more options
# ... or force ignoredups and ignorespace
HISTCONTROL=ignoredups

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "$debian_chroot" ] && [ -r /etc/debian_chroot ]; then
debian_chroot=$(cat /etc/debian_chroot)
fi

# quit back to the main bash shell
exit

# Set testing to have a value that will be inherited
export testing="hello world"

bash

# Now see that the exported variable went through
echo $testing
# hello world
# uncomment for a colored prompt, if the terminal has the capability; turned # off by default to not distract the user: the focus in a terminal window # should be on the output of commands, not on the prompt
#force_color_prompt=yes

if [ -n "$force_color_prompt" ]; then
  if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then
    # We have color support; assume it's compliant with Ecma-48 # (ISO/IEC-6429). (Lack of such support is extremely rare, and such # a case would tend to support setf rather than setaf.)
color_prompt=yes
  else
    :q

researchtools@ubuntu:~/class/10$  
-U:::  *shell*  Bot L78  (Shell:run)------------------------

# See that "testing" is not set. If there is no variable, bash gives # an empty string
echo $testing

# quit back to the main bash shell
exit

# Set testing to have a value that will be inherited
export testing="hello world"

bash

# Now see that the exported variable went through
echo $testing
# hello world
#++END_SRC

****  10-qgis-bash-python.org  26% L129  (Org)------------------------

Kill buffer (default *shell*):
# See that "testing" is an empty string
echo $Testing

# quit back to the main exit
exit

# Set testing to have a different value with
export testing="hello world"

bash

# Now see that the environment variable $testing

echo $Testing
hello world

#+END_SRC

* Creating a script

How can we use a variable to store one image every hour for the 2011 set of images for this experiment?

- [http://mgds.ldeo.columbia.edu/healy/reports/alof](http://mgds.ldeo.columbia.edu/healy/reports/alof)

Open emacs open a file

#+BEGIN_SRC sh
for hour in 01 02 03 04 05 06; do
    echo $hour
done

---

USCGC HEALY SCIENCE SUPPORT ALOFTCON WEBCAM:

- [20110929-1501.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-1501.jpg)
- [20110929-1401.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-1401.jpg)
- [20110929-1301.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-1301.jpg)
- [20110929-1201.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-1201.jpg)
- [20110929-1101.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-1101.jpg)
- [20110929-1001.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-1001.jpg)
- [20110929-0901.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-0901.jpg)
- [20110929-0801.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-0801.jpg)
- [20110929-0701.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-0701.jpg)
- [20110929-0601.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-0601.jpg)
- [20110929-0501.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-0501.jpg)
- [20110929-0401.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-0401.jpg)
- [20110929-0301.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-0301.jpg)
- [20110929-0201.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-0201.jpg)
- [20110929-0101.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-0101.jpg)
- [20110929-0001.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110929-0001.jpg)
- [20110928-2301.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110928-2301.jpg)
- [20110928-2201.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110928-2201.jpg)
- [20110928-2101.jpg](http://mgds.ldeo.columbia.edu/healy/reports/alof/20110928-2101.jpg)
See that "testing" is an empty string
exit

Set testing to have
export testing="hello world"
bash

Now see that the export
echo $testing
hello world
+#END_SRC

* Creating a script

How can we use a variable one image every hour for 2011 set of images for

- http://mgds.ldeo.columbia.edu

Open emacs open a file

#BEGIN_SRC sh
for hour in {01..03..06} do
    echo $hour
done

---

10-qgis-bash-python
* Creating a script

How can we use a variable to help out? What if we want to download one image every hour from one day on the USCGC Healy? Here is the 2011 set of images for the Healy:


Open emacs open a file in emacs called `~/class/10/healy.bash` and start typing:

```sh
# BEGIN_SRC sh
for hour in 01 02 03 04 05 06 07
do
echo $hour
done
# END_SRC
```
* Creating a script

How can we use a variable to help out? What if we want to download one image every hour from one day on the USCGC Healy? Here is the 2011 set of images for the Healy:


Open emacs open a file in emacs called `~/class/10/healy.bash` and start typing:

```bash
#+BEGIN_SRC sh
for hour in 01 02 03 04 05 06 07
do
echo $hour
done
#+END_SRC
```

---

**10-qgis-bash-python.org** 28% L147 (Org)
* Creating a script

How can we use a variable to help out? What if we want to download one image every hour from one day on the USCGC Healy? Here is the 2011 set of images for the Healy:


Open emacs open a file in emacs called `~/class/10/healy.bash` and start typing:

```bash
#BEGIN_SRC sh
for hour in 01 02 03 04 05 06 07
do
  echo $hour
done
#END_SRC
```

```org
for hour in 01 02 03 04 05 06 07
do
  echo $hour
done
```
```bash
researchtools@ubuntu:~$ ls -l
total 3084
-rw-r--r-- 1 researchtools researchtools 15081 2011-09-29 11:51 #10-qgis-bash-python.org#
-rw-r--r-- 1 researchtools researchtools 15081 2011-09-29 07:52 10-qgis-bash-python.org
-rw-r--r-- 1 researchtools researchtools 26259 2011-09-29 11:26 2007-boston-construction.kml
-rw-r--r-- 1 researchtools researchtools 3093687 2011-09-29 11:26 2007-boston-construction.xy
researchtools@ubuntu:~$ ls -l
total 3088
-rw-r--r-- 1 researchtools researchtools 15081 2011-09-29 11:51 #10-qgis-bash-python.org#
-rw-r--r-- 1 researchtools researchtools 15081 2011-09-29 07:52 10-qgis-bash-python.org
-rw-r--r-- 1 researchtools researchtools 26259 2011-09-29 11:26 2007-boston-construction.kml
-rw-r--r-- 1 researchtools researchtools 3093687 2011-09-29 11:26 2007-boston-construction.xy
-rw-r--r-- 1 researchtools researchtools 55 2011-09-29 11:53 healy.bash
researchtools@ubuntu:~$ source healy.bash
```

```
10-qgis-bash-python
for hour in 01 02 03 04 05 06 07
do
echo $hour
done
```

```
-U:---- healy.bash All L5 (Shell-script[bash])----------------
Wrote /home/researchtools/class/10/healy.bash
```
Now we can try to construct a curl command in the echo.

```sh
for hour in 01 02 03 04 05 06 07
do
echo curl -0 http://mgds.ldeo.columbia.edu/healy/reports/alofcon/2011/20110928-$hour01.jpeg
done
```

Try it and you should see the follow, but since we are using the echo

```bash
for hour in 01 02 03 04 05 06 07
do
  echo curl -0 http://mgds.ldeo.columbia.edu/healy/reports/alofcon/2011/20110928-$hour01.jpeg
done
```
Now we can try to construct:

```bash
#!/bin/bash

for hour in 01 02 03 04 05 06 07
  do
  done
```

Try it and you should see:

```
10-ggis-bash-python
```

```bash
for hour in 01 02 03 04 05 06 07
  do
  done
```
Now we can try to construct a curl command in the echo.

```sh
#BEGIN_SRC sh
for hour in 01 02 03 04 05 06 07
do
echo curl -O http://mgds.ldeo.columbia.edu/healy/reports/aloftcon/2011/20110928-\$(hour)01.jpeg
done
#END_SRC

Try it and you should see the follow, but since we are using the echo

```
*** 10-qgis-bash-python.org 33% L175 (Org)
for hour in 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23
do
curl -O http://mgds.ldeo.columbia.edu/healy/reports/aloftcon/2011/20110928-\$(hour)01.jpeg
done
```
Now we can try to construct

```bash
#!/bin/sh
for hour in 01 02 03 04 05 06 07
  do
  done
```

Try it and you should see:

```
--- 10-ggis-bash-python
```

```bash
for hour in 01 02 03 04 05 06 07
  do
```

```bash
Wrote /home/researchtools/class/10/healy.bash
```
Now we can try to construct:

```bash
#BEGIN_SRC sh
for hour in 01 02 03 04
    do
        echo curl -o http://mgds
    done
#END_SRC
```

Try it and you should see:

```bash
-- END-EXAMPLE --
```
Run it! You should now have all the images down. Time to make an "animated gif" using imagemagick/graphicsmagick.

```bash
convert -delay 100 -loop 0 *.jpeg healy-20110928-day-animation.gif
```

View that animated gif movie!!!
Run it! You should now have all the images down. Time to make an "animated gif" using imagemagick/graphicsmagick.

```
convert -delay 100 -loop 0 *.jpeg healy-20110928-day-animation.gif
```

View that animated gif movie!!

```bash
```
You can now see that the permissions of the files have been changed.

We can turn that script into a proper script. Add the following:

```bash
#!/bin/bash
```

Now make the script executable for all users:

```bash
chmod +x healy.bash
```

You can also put that in the same line:

```bash
#!/bin/bash
chmod +x healy.bash
```

Now change directory and list the contents:

```bash
ls -l
```

Use `chmod` to modify permissions:

```bash
chmod +x ./healy.bash
```

Use `ls -l` to list the contents:

```bash
ls -l
```

Introduction to Python:

It is time to get away from the shell script. We will be using iPython.

Start up iPython:

```bash
http://ipython.org/ipython
```

Starting up iPython:

The very first time you start up your account. You will:

```bash
10-qgis-bash-python
```

10-qgis-bash-python
#!/bin/bash

for hour in 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23
do
curl -0 http://mgds.ldeo.columbia.edu/healy/reports/aloftcon/2011/20110928-${hour}01.jpeg
done

You can now see that the Healy pulled into Dutch Harbor yesterday!

We can turn that script that we are running with `=source=` into a proper script. Add this to the script at the first line:

```bash
#!/bin/bash
```

Now make the script "executable" with `chmod`. "Change mod" sets the permissions for a file and one of those permissions is "x" for executable. You can also put that convert command inside the script as the last line.

```bash
chmod +x /home/researchtools/class/10/healy.bash
```
Now make the script "export Permissions for a file"
You can also put that on

```bash
#+BEGIN_SRC sh
chmod +x healy.bash
./healy.bash
#+END_SRC
```

* Introduction to Python

It is time to get away
We will be using iPython

http://ipython.org/ipython

** Starting up iPython

The very first time you use your account. You will need to start iPython with the "-pylab" flag.

In a terminal, run this:

```bash
#+BEGIN_SRC sh
ipython -pylab
#+END_SRC
```

You will see something
Now make the script ".bashrc" 
permissions for a file.
You can also put that in:

```
# BEGIN_SRC sh
chmod +x healy.bash
./healy.bash
# END_SRC

* Introduction to IPython

It is time to get away.
We will be using ipython.

http://ipython.org/ipython

** Starting up IPython

The very first time you start up your account. You will first run IPython with the 
"-pylab" flag.

In a terminal, run this:

```
# BEGIN_SRC sh
ipython -pylab
# END_SRC

You will see something like this:

```
10-qgis-bash-pyth
```
Now make the script "executable" permissions for a file.
You can also put that down:

```bash
# BEGIN_SRC sh
chmod +x healy.bash
./healy.bash
# END_SRC
```

*Introduction to Python*

It is time to get away

We will be using *ipython*

http://ipython.org/ipython

**Starting up *ipython**

The very first time you run *ipython* you will need to set up your account. You will be prompted to install the "-pylab" environment.

In a terminal, run this:

```bash
# BEGIN_SRC sh
ipython -pylab
# END_SRC
```

You will see something like this:

```
Welcome to pylab, a matplotlib-based Python environment.
For more information, type 'help(pylab)'.
```

```
In [1]: exit
Out[1]: Type exit() to exit.
```

```
In [2]: exit()
Do you really want to exit ([y]/n)?
```

---

**SUCCESSFUL INSTALLATION**

Please read the sections 'Initial Configuration' and 'Quick Tips' in the IPython manual (there are both HTML and PDF versions supplied with the distribution) to make sure that your system environment is properly configured to take advantage of IPython's features.

Important note: the configuration system has changed! The old system is still in place, but its settings may be partly overridden by the settings in "~/.ipython/ipy_user_conf.py" config file. Please take a look at the file if some of the new settings bother you.

Please press <RETURN> to start IPython.

```
Python 2.7.1+ (r271:86832, Apr 11 2011, 18:05:24)
Type "copyright", "credits" or "license" for more information.
```

```
IPython 0.10.1 -- An enhanced Interactive Python.
%quickref -> Quick reference.
%help -> Python's own help system.
%object? -> Details about 'object'. %object also works, ?? prints more.
```

Welcome to pylab, a matplotlib-based Python environment.
For more information, type 'help(pylab)'.
Now, if you start ipython

** Looking around with

Start ipython

```bash
#+BEGIN_SRC sh
ipython -pylab
#+END_SRC

ipython provides some
are *not* available in

```bash
#+BEGIN_EXAMPLE
In [1]: ls
10-qgis-bash-python.org# 20110928-0701.jpeg 20110928-1701.jpeg
10-qgis-bash-python.org 20110928-0801.jpeg 20110928-1801.jpeg
2007-boston-construction.kml 20110928-0901.jpeg 20110928-1901.jpeg
20110928-0101.jpeg 20110928-2101.jpeg
20110928-0201.jpeg 20110928-2201.jpeg
20110928-0301.jpeg 20110928-2301.jpeg
20110928-0401.jpeg
20110928-0501.jpeg
20110928-0601.jpeg
20110928-0701.jpeg

In [2]: cd ..
/home/researchtools/class

In [3]: pwd
/home/researchtools/class/

In [4]: ls
09/ 10/

In [5]: ls
20110928-1601.jpeg

In [6]: cd 10
/home/researchtools/class/10

In [7]:
```

** An initial plot

Before we dig into the

```bash
#+BEGIN_SRC sh
python
#+END_SRC

Auto-saving...done
```
** An initial plot

Before we dig into the details, let’s not worry about how the data looks. All the things we want to do can be done with a numpy.loadtxt:

```python
In [7]: numpy.loadtxt('2007-boston-construction.xy')
```

But really, we want to save the x and y into their own variables so that we can plot x versus y.

```python
x, y = numpy.loadtxt('2007-boston-construction.xy', delimiter=',', unpack=True)
```
In [2]: pwd
Out[2]: '/home/researchtools'

** An initial plot

Before we dig into the data, let's look at how the columns are aligned. There is no key to map the columns, so don't worry about how the values are sorted, just that we have the same number of rows.

```python
import numpy as np

# BEGIN EXAMPLE
x, y = np.loadtxt('2007-boston-construction.xy', delimiter=',', unpack=True)
# END EXAMPLE
```

But really, we want to save the `x` and `y` into their own variables so that we can plot `x` versus `y`. 

```python
In [8]: x, y = np.loadtxt('2007-boston-construction.xy', delimiter=',', unpack=True)
```

```
ValueError: invalid literal for float(): -70.501456667,42.1006833333
```
**An initial plot**

Before we dig into the code, let's not worry about how this code is set up.

```
# BEGIN EXAMPLE
numpy.loadtxt('2007-boston-construction.xy', delimiter=',',)
```

But really, we want to save the x and y into their own variables so that we can plot x versus y.

```
# BEGIN_SRC python
x, y = numpy.loadtxt('2007-boston-construction.xy', delimiter=',', unpack=True)
# END_SRC
```
But really, we want to make sure that we can plot x versus y.

```
In [8]: numpy.loadtxt('2007-boston-construction.xy',delimiter=',')
Out[8]:
array([[-70.50145667, 42.10068333],
       [-70.50164667, 42.101755 ],
       [-70.501845 , 42.10287667],
       ...,
       [-70.97004 , 42.24342833],
       [-70.969975, 42.243611  ],
       [-70.970045, 42.24345833]])
```

```
In [9]: x,y = numpy.loadtxt('2007-boston-construction.xy',delimiter=',', unpack=True)
```

Now make a plot of x vs y.

```
In [10]: len(x)
Out[10]: 119194
```

```
In [11]: plot(x,y)
```

```
file:/home/username/figures/10-matplotlib-first.png
```

* Updating Ubuntu

*NOTE:* please do this at the end of class before you leave.

It is a good idea to keep your computer up-to-date with patches. The same goes with your Ubuntu virtual machine. Open a normal terminal and run these commands one at a time. Remember that the `researchtools` account password is `rt2011vm`.

```
In [7]: sudo apt-get update
In [8]: sudo apt-get upgrade
```
But really, we want to see what we can plot x versus y.

```python
# BEGIN_SRC python
x, y = numpy.loadtxt('2007.dat')
# END_SRC

Now make a plot of x versus y.
```

```python
# BEGIN_SRC python
plot(x, y)
# END_SRC

file:./figures/10-matplotlib-first.png
```

* Updating Ubuntu

*NOTE:* please do this at the end of class before you leave.

It is a good idea to keep your computer up-to-date with patches. The same goes with your Ubuntu virtual machine. Open a normal terminal and run these commands one at a time. Remember that the researchtools account password is "rt2011vm".

```sh
# BEGIN_SRC sh
sudo apt-get update
sudo apt-get upgrade
# END_SRC
```