RESEARCH TOOLS 2011

LECTURE 03

2011-Sep-06

Kurt Schwehr

http://schwehr.org

UNH CCOM/JHC

Wiki editing, Weather Demo, Command Line
Class 3: Weather Demo, Basic Command line, VM Ware Linux (DRAFT)

Introduction

Internet Relay Chat (IRC)

Start Firefox

Under "Tools," select "ChatZilla" at the bottom.

ChatZilla will use your login name (often referred to as you "shortname") by default. You can pick anything you like, but if you don't use something related to your name, people may have trouble which user in a chat is you. For this class, make sure to use the same short name as your CCOM account - please do not change your alias.

We need to now sign into an "IRC Server" that hosts chat rooms. There are many thousands of servers in the world and we will even setup on at CCOM later in the semester. For now, I have created an unofficial channel on http://freenode.net. Freenode supports free and opensource software and this class is primarily about just those topics. Type this command right into the bottom window of ChatZilla:

```
BEGIN_SRC sh
/attach irc.freenode.net
/join #unhresearchtools
END_SRC
```

** Say "hi" in the IRC channel
Editing Research Tools 2011

The 2011 Research Tools (ESCI 895–003) course taught by Kurt Schwehr

Auditors (official or otherwise) welcome. You will have to provide your own computer/laptop if not a registered student.

== Instructor ==

[[User:Schwehr|Kurt Schwehr]]

== Meeting Time and Location ==

- TTh 11:10–12:30
- Chase Ocean Engineering Rm 140 (Computer Classroom)

== Resources ==

- [[Researchtools.com.nh]] - Ubuntu Linux server (actually a virtual machine) for use during class projects

== Lectures ==

* 1 Introduction - topics that will be covered
* [[Lec 2 Research Tools 2011]] - Getting started - IRC via Firefox Chatzilla, the CCOM wiki, logging in to researchtools.ccom.nh
* Lecture 3 – 6-Sept-2011
* Lecture 4 – 8-Sept-2011 - VMWare, Short lecture for brown bag seminar

Please note that all contributions to CCOMwiki may be edited, altered, or removed by other contributors. If you do not want your writing to be edited mercilessly, then do not submit it here.

You are also promising us that you wrote this yourself, or copied it from a public domain or similar free resource (see CCOMwiki:Copyrights for details). Do not submit copyrighted work without permission!
It’s better for security not to let users put any HTML they want into pages. Just link to the map.
Manual:$wgRawHtml

HTML: $wgRawHtml

Allow raw, unchecked HTML in <html>...</html> sections.

Introduced in version: 1.3.4

Removed in version: still in use

Allowed values: (boolean)

Default value: false

Other settings: Alphabetical I By Function

Details

When $wgRawHtml = true; the wiki will allow you to insert raw unchecked HTML. However, you must embed your html within the <html>...</html> tags so that mediawiki can differentiate it.

⚠️ Warning: This is very dangerous on a publicly editable site, because it allows for arbitrary JavaScript code to be inserted, opening the door for session hijacking. Thus, you shouldn’t enable it unless you’ve restricted editing to trusted users only with $wgGroupPermissions (version 1.3.x and 1.4.x can use $wgWhitelistEdit). See Manual:Preventing access for more information on restricting write access.

💡 Note: This option does not affect how wikitext outside of <html>...</html> tags is handled.

💡 Note: If you have $wgRawHtml turned on and you notice some HTML being stripped out, you may need to turn off $wgUseTidy.

Is enabling raw HTML necessary?

Some HTML tags are permitted in wikitext, even with $wgRawHtml=false. See meta:Help:HTML in wikitext. The vast majority of fancy formatting seen on Wikimedia sites is achieved using these limited tags (e.g. tables with CSS style tags). If you can make do with these limitations (leave $wgRawHtml=false), your wiki will be more secure.

Also note that the "limited" wiki syntax is actually a deliberate design feature of wikis. It is a compact simplified markup which is easily understood even by non-technical users, easily visualised in diff displays, and discourages stylistic tinkering in favour of getting on with writing useful/interesting text.
Extension:Google Maps

Maps are a great way to present information, and the Google Maps Extension brings the coolness of interactive, annotated maps to your wiki. With Google Maps Extension, you can:

- Spice up articles with maps of anywhere in the world
- Marvel at detailed street maps and stunning satellite imagery
- Use the interactive editor's map to add markers and colored paths with wiki-fied captions
- Look up cities, addresses, and businesses with the built-in search engine

Google Maps Javascript API

Note: The Google Maps Javascript API Version 2 used by this extension has been officially deprecated as of May 19, 2010 by Google.

What does it do?

The extension hooks into your MediaWiki installation in two ways: first, it defines a `googlemap` tag that translates a special syntax into a map.
It's All Text! 1.6.4
by Christian Höltje (A.K.A. docwhat)

Edit textareas using an external editor, because it's all text!

Right click on a textarea, select "It's All Text!" and edit the text in the editor of your choice.

Alternatively, click on the edit buttons added for your convenience. Right click on the edit buttons for even more options, including preferences.

Continue to Download →

Enjoy this add-on?
The developer of this add-on asks that you help support its continued development by making a small contribution.

Contribute

$4.99 suggested
Demonstration - Processing weather data

For this section, you do not need to try to follow along. Just watch what I am doing. The goal is to give you a feeling for what we will be covering. I do not expect you to understand all the steps just yet. It will take us a number of classes to cover all of this material.

This material has also been written up here on my blog:

http://schwehr.org/blog/archives/2011-09.html#e2011-09-05T19_50_00.txt

First, log into a Linux computer and setup a project directory.

```
# BEGIN_SRC sh
ssh researchtools.ccom.nh # or use putty
mkdir wx
cd wx # wx is short for "weather"
# END_SRC
```

Check out the weather data that is available over the CCOM internal network. socat is a network data helper. It can dump what it sees from a network connection, pass it to other programs, or forward the data.

```
# BEGIN_SRC sh
socat TCP4:datalogger1.ccom.nh:36000 - l head
# END_SRC
```

You should see some NMEA weather data.

```
$GWMW,140.3,R,1.8,N,A*2C,rccom-aimar,1315303659.98
$GMDA,30.0438,F,1.0174,B,15.6,C,--------,164.4,T,179.8,M,1.8,N,0.9,M*27,rccom-aimar,1315303660.12
$HCHDT,26.2,T*1F,rccom-aimar,1315303660.25
$GPZDA,100740,06,09,2011,00,00*47,rccom-aimar,1315303660.31
$WIMW,164.6,T,180.0,M,1.7,N,0.9,M*59,rccom-aimar,1315303660.4
$WIMW,138.6,T,1.7,N,A*2F,rccom-aimar,1315303660.45
$WIMW,140.3,R,1.7,N,A*23,rccom-aimar,1315303660.51
$PGPGG,100740,4308.1261,N,07056.3764,W,2.9,0.9,42.2,M,***0F,rccom-aimar,1315303660.63
$HCHDT,26.2,T*1F,rccom-aimar,1315303660.75
$GPVTG,303.8,T,319.2,M,0.1,N,0.1,K,D*27,rccom-aimar,1315303660.83
2011/09/06 06:07:40 socat[29595] E write1(1, 0x9d6b410, 53): Broken pipe
```

U:--- 03-basic-command-line.org  9% (57,0)  Hg:0 (Org yas Spc Fill)  12:23PM 0.94

Sunday, June 24, 12
09.05.2011 19:50

PYTHON DEVELOPMENT - HURRICANE IRENE
Today I gave a run through of a portion of what I aim to teach this semester in research tools. I wanted to make a demonstration of going from a sensor in the world, creating a parser for the data it produces, plotting up some results and releasing the code to the world. I'm using the CCOM weather station as an example. Andy and Ben got the Airmar PB150 setup quite a while ago. It spits out NMEA over a serial port at 4800 baud. I use my serial-logger script to read the serial port and rebroadcast the data over the internal network for anyone who is interested. Here is using socat to grab a few lines of the data:

```
socat TCP4:datalogger1:36000 - | head
$HCHDT,26.2,T*1F,rccom-airmar,1314661980.3
$GPVTG,275.1,T,290.5,N,0.1,N,0.1,K,D*29,rccom-airmar,1314661980.38
$GP2DA,235300,29,08,2011,00,00*4E,rccom-airmar,1314661980.45
$WIMNW,143.9,R,1.9,N,A*24,rccom-airmar,1314661980.52
$GPGGA,235300,4308.1252,N,07056.3764,W,2,9,0.9,37.2,M,,,*,08,rccom-airmar,1314661980.64
$WIMDA,30.0497,I,1.0176,B,17.8,C,,,,,,,,,167.2,T,182.6,M,1.9,N,1.0,M*2A,rccom-airmar,1314661980.79
$HCHDT,26.2,T*1F,rccom-airmar,1314661980.82
$WIMNW,167.2,T,182.6,M,1.9,N,1.0,M*5C,rccom-airmar,1314661980.97
$WIMNW,141.0,T,1.9,N,A*29,rccom-airmar,1314661981.02
```

The ",rccom-airmar,1314661980.97" is added by my serial-logger giving each line a station name and a UNIX UTC timestamp. Eric Raymond (ESR) has put together a very nice document on NMEA sentences: NMEA Revealed. It describes many of the sentences in common use. What do we have for contents? The unix "cut" command can pull out the "talker" + "sentence" part of each line. The -d specifies that the sort with "-u" for collapsing the output to the unique list of lines can get the job done:

```
egrep -v '^[#]' ccom-airmar-2011-08-28 | cut -d, -f1 | sort -u
$GPVGA
$GPFVG
$GPZDA
$HCHDT
$PNTZNT
$WIMDA
$WIMNW
$WIMNW
```

All of those messages (except my custom PNTZNT message for NTP clock status) are documented in ESR's NMEA Revealed.

To look at the weather from Hurricane Irene, we want to look at the MDA is listed as "Obsolete"
schwehr@researchtools:~$ `socat TCP4:datalogger1:36000 - | head -20`


http://en.wikipedia.org/wiki/Head_(Unix)
### GGA - Global Positioning System Fix Data

Time, Position and fix related data for a GPS receiver.

```
$--GGA,hhmmss.ss,1111.11,a,yyyy.yy,a,x,xx,xx,,x,M,x,x,M,x.x,xxxx*hh<CR><LF>
```

**Field Number:**

1. Universal Time Coordinated (UTC)
2. Latitude
3. N or S (North or South)
4. Longitude
5. E or W (East or West)
6. GPS Quality Indicator,
   - 0 - fix not available,
   - 1 - GPS fix,
   - 2 - Differential GPS fix (values above 2 are 2.3 features)
   - 3 = PPS fix
   - 4 = Real Time Kinematic
   - 5 = Float RTK
   - 6 = estimated (dead reckoning)
   - 7 = Manual input mode
   - 8 = Simulation mode
7. Number of satellites in view, 00 - 12
8. Horizontal Dilution of precision (meters)
9. Antenna Altitude above/below mean-sea-level (geoid) (in meters)
10. Units of antenna altitude, meters
To look at the weather from Hurricane Irene, we want to look at the MDA is listed as "Obsolete" by ESR according to a NMEA 2009 doc, but that is the message we want to use. In python we could parse this by hand. Here is an example "Meteorological Composite" NMEA line:

```
$WMDA,30.0497,I,1.0176,B,17.8,C,,,,,,,,167.2,T,182.6,M,1.9,N,1.0,M*2A
```

Python makes it easy to do splits on strings and use any separator that we line. For example, we could do:

```python
fields = line.split(',
```

This would break apart each of the blocks. However, this doesn't scale well and does not tell us when a message is too corrupted to be usable data. I have written a large number of regular expressions in Python for NMEA sentences based on emails that I get from the USCG Healy.

I wanted to start turning those into a library that I could make usable by anyone. I created the nmea decoder package. I used mercurial (hg) for version control and uploaded it to bitbucket as (nmeadec). It's pure python and simpler (but less powerful) than gpsd. I really like the way that python's regular expression syntax lets you name the fields and retrieve a named dictionary when messages are decoded. You can find the regular expression for MDA here: nmeadec/raw.py - line 39. With nmeadec 0.1 written, I can now parse NMEA in Python like this:

```python
msg = nmeadec.decode(line)
```

The PasteScript package gave a helping hand for creating a basic python package. I did this from inside of a virtualenv to protect the system and fink python space.

```bash
virtualenv ve
cd ve
source bin/activate
mkdir src
paster create nmeadec
```

I answered a whole bunch of questions and it setup a simple package using setuptools/distribute.

Since you are not creating that package and might want to follow along, you can grab the package in src (and skip running the paster command to create a new project.

```bash
hg clone https://schwehr@bitbucket.org/schwehr/nmeadec
```

Because I set this up in a terminal using a virtualenv being active, then I can use this command to setup the package for development without funny python PATH hacks:

```bash
cd nmeadec
python setup.py develop
```
The following NMEA sentences have been designated "obsolete" in a publicly available NMEA document dated 2009.

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APA</td>
<td>Autopilot Sentence &quot;A&quot;</td>
</tr>
<tr>
<td>BER</td>
<td>Bearing &amp; Distance to Waypoint, Dead Reckoning, Rhumb Line</td>
</tr>
<tr>
<td>BPI</td>
<td>Bearing &amp; Distance to Point of Interest</td>
</tr>
<tr>
<td>DBK</td>
<td>Depth Below Keel</td>
</tr>
<tr>
<td>DBS</td>
<td>Depth Below Surface</td>
</tr>
<tr>
<td>DRU</td>
<td>Dual Doppler Auxiliary Data</td>
</tr>
<tr>
<td>GDA</td>
<td>Dead Reckoning Positions</td>
</tr>
<tr>
<td>GLA</td>
<td>Loran-C Positions</td>
</tr>
<tr>
<td>GOA</td>
<td>OMEGA Positions</td>
</tr>
<tr>
<td>GXA</td>
<td>TRANSIT Positions</td>
</tr>
<tr>
<td>GTD</td>
<td>Geographical Position, Loran-C TDs</td>
</tr>
<tr>
<td>GXA</td>
<td>TRANSIT Position</td>
</tr>
<tr>
<td>HCC</td>
<td>Compass Heading</td>
</tr>
<tr>
<td>HCD</td>
<td>Heading and Deviation</td>
</tr>
<tr>
<td>HDM</td>
<td>Heading, Magnetic</td>
</tr>
<tr>
<td>HDT</td>
<td>Heading, True</td>
</tr>
<tr>
<td>HVD</td>
<td>Magnetic Variation, Automatic</td>
</tr>
<tr>
<td>HVM</td>
<td>Magnetic Variation, Manually Set</td>
</tr>
<tr>
<td>IMA</td>
<td>Vessel Identification</td>
</tr>
<tr>
<td>MDA</td>
<td>Meteorological Composite</td>
</tr>
</tbody>
</table>
schwehr@researchtools:~/wx$ ls -l
total 163844
-rw-r--r-- 1 schwehr domain users 55808963 2011-09-06 06:05 ccom-airmar-2011-08-28
-rw-r--r-- 1 schwehr domain users 55979878 2011-09-06 06:05 ccom-airmar-2011-08-29
-rw-r--r-- 1 schwehr domain users 55977976 2011-09-06 06:05 ccom-airmar-2011-08-30
drwxr-xr-x 7 schwehr domain users 4096 2011-09-06 06:24 nmeadec

schwehr@researchtools:~/wx$ egrep -v '^[#]' ccom-airmar-2011-08-28 | cut -d, -f1 | sort -u

$GPAGA
$GPVTG
$GPZDA
$HCHDT
$PNTZNT
$WIMDA
$WIMWD
$WIMWV

schwehr@researchtools:~/wx$

#+BEGIN_SRC sh
pwd # make sure you are in the "wx" directory

# If you are unfamiliar with .bz2 files, ask the computer if it knows
file ccom*
ccom-airmar-2011-08-28.bz2: bzip2 compressed data, block size = 900k
ccom-airmar-2011-08-29.bz2: bzip2 compressed data, block size = 900k
ccom-airmar-2011-08-30.bz2: bzip2 compressed data, block size = 900k

# Uncompress the data. "**" matches any text
bunzip2 ccom*.bz2

file ccom-airmar-2011-08-**
ccom-airmar-2011-08-28: ASCII English text
To look at the weather from Hurricane Irene, we want to look at the MDA is listed as "Obsolete" by ESR according to a NMEA 2009 doc, but that is the message we want to use. In python we could parse this by hard. Here is an example "Meteorological Composite" NMEA line:

```
$WMDA,30.0497,I,1.0176,B,17.8,C,,167.2,T,182.6,M,1.9,N,1.0,M*2A
```

Python makes it easy to do splits on strings and use any separator that we line. For example, we could do:

```
fields = line.split(',,')
```

This would break apart each of the blocks. However, this doesn't scale well and does not tell us when a message is too corrupted to be usable data. I have written a large number of **regular expressions** in Python for NMEA sentences based on emails that I get from the USCG Healy.

I wanted to start turning those into a library that I could make usable by anyone. I created the nmea decoder package. I used mercurial (hg) for version control and uploaded it to bitbucket as *(nmeadec)*. It's pure python and simpler (but less powerful) than gpds. I really like the way that python's regular expression syntax lets you name the fields and retrieve a named dictionary when messages are decoded. You can find the regular expression for MDA here: [nmeadec/raw.py - line 39](http://example.com). With nmeadec 0.1 written, I can now parse NMEA in Python like this:

```
msg = nmeadec.decode(line)
```

The [PasteScript](http://example.com) package gave a helping hand for creating a basic python package. I did this from inside of a virtualenv to protect the system and fink python space.

```
virtualenv ve
cd ve
source bin/activate
mkdir src
paster create nmeadec
```

I answered a whole bunch of questions and it setup a simple package using setuptools/distribute.

Since you are not creating that package and might want to follow along, you can grab the package in src (and skip running the paster command to create a new project).

```
hg clone https://schwehr@bitbucket.org/schwehr/nmeadec
```

Because I set this up in a terminal using a virtualenv being active, then I can use this command to setup the package for development without funny python PATH hacks:

```
cd nmeadec
python setup.py develop
```
Now, we need to pull out the data. I created a little module called "process_wx.py". It let's you down sample the data there were more than 86,000 MDA messages in a day.

```python
from __future__ import print_function
import nmeadec

def get_wx(filename, nth=None):
    pres = []
    speed = []
    timestamps = []
    mda_count = 0  # for handling the nth MDA entry

    for line in file(filename):
        try:
            msg = nmeadec.decode(line)
        except:
            continue

        try:
            if msg['sentence'] != 'MDA': continue
        except:
            print('trouble:', line, msg)

        mda_count += 1
        if nth is not None and mda_count % nth != 1:
            continue  # skip all but the nth. start with first

        #print (msg['pressure_bars'], msg['wind_speed_ms'])
        pres.append(msg['pressure_bars'])
        speed.append(msg['wind_speed_ms'])
        timestamps.append(float(line.split(',')[1]))

    return { 'pres': pres, 'speed': speed, 'timestamps': timestamps }
```

We can then use that in ipython to see how it works:

```bash
ipython -pylab  # Ask for ipython to preload lots
import process_wx
data = process_wx.get_wx('ccom-aimar-2011-08-28')
data.keys()
['timestamps', 'speed', 'pres']
len(data['timestamps'])
86361
data = process_wx.get_wx('ccom-aimar-2011-08-28', nth=10)
len(data['timestamps'])
8637
```
Now, we need to pull out the data. I created a little module called "process_wx.py". It let's you down sample the data there were more than 86,000 MDA messages in a day.

```python
from __future__ import print_function
import nmade

def get_wx(filename, nth=None):
    pres = []
    speed = []
    timestamps = []
    mda_count = 0 # for handling the nth MDA entry

    for line in file(filename):
        try:
            msg = nmade.decode(line)
            except:
                continue

        try:
            if msg['sentence'] != 'MDA': continue
        except:
            print ('trouble:', line, msg)

        mda_count += 1
        if nth is not None and mda_count % nth != 1:
            continue # skip all but the nth. start with first

        #print (msg['pressure_bars'], msg['wind_speed_ms'])
        pres.append(msg['pressure_bars'])
        speed.append(msg['wind_speed_ms'])
        timestamps.append(float(line.split(',')[1][-1]))

    return {'pres': pres, 'speed': speed, 'timestamps': timestamps}
```

We can then use that in ipython to see how it works:

```
ipython -pylab # Ask for ipython to preload lots
import process_wx
data = process_wx.get_wx('ccom-airmar-2011-08-28')
data.keys()
['timestamps', 'speed', 'pres']
len(data['timestamps'])
86361
data = process_wx.get_wx('ccom-airmar-2011-08-28', nth=10)
len(data['timestamps'])
8637
```
Now to load 3 days:

```python
import process_wx
from numpy import array

# explicit:
days = []
days.append( process_wx.get_wx('ccom-airmar-2011-08-27', nth=10) )
days.append( process_wx.get_wx('ccom-airmar-2011-08-28', nth=10) )
days.append( process_wx.get_wx('ccom-airmar-2011-08-29', nth=10) )

# Does the same as the above, but in one line with "list comprehensions"
days = [ process_wx.get_wx('ccom-airmar-2011-08-\'+str(day), nth=10) for day in (27, 28, 29) ]

# We then have to get the pressure, temperature, and timestamps for the 3 days and combine them
# This is pulling out a few too many tricks in one line!
pres = array ( sum( [ day['pres'] for day in days ], [ ] ) )
speed = array ( sum( [ day['speed'] for day in days ], [ ] ) )
timestamps = array ( sum( [ day['timestamps'] for day in days ], [ ] ) )
```

We now have the data loaded and it’s time to take a look at it:

```
min(data['speed']), max(data['speed'])
(0.0, 12.4)
min(data['pres']), max(data['pres'])
(0.9837000000000002, 1.0201)
average(data['speed'])
1.52199
average(data['pres'])
1.0084
median(data['speed'])
1.0
median(data['pres'])
1.013650
```
And finally, we would like to make a plot of these parameters. There are several plotting packages for python. Probably the most flexible and powerful is matplotlib. It is very similar to plotting in matlab.

```python
# Top plot
subplot(211)
ylabel('Pressure (bar)')
xlabel('')

# Turn off labels for the xaxis
ax=gca()
ax.xaxis_date()
old_xfmt = ax.xaxis.get_major_formatter()
xfmt=DateFormatter('')
ax.xaxis.set_major_formatter(xfmt)
title('Hurricane Irene, 2011')
plot (data['dates'], data['pres'])

# Bottom plot
subplot(212)
xlabel('UTC time')
ylabel('Wind speed (m/s)')

# Label x-axis by Hour:Minute
xticks( rotation=25 )
subplots_adjust(bottom=0.2)
ax=gca()
ax.xaxis_date()
xfmt=DateFormatter('%H:%M')
ax.xaxis.set_major_formatter(xfmt)

# 30.6 (meters / second) = 68.5 mph
plot (data['dates'], data['speed'])
title('')
```

I used GraphicsMagick (fork of ImageMagick) to resize the image to have a width of 600 pixels. Yes, I could have set the output size in matplotlib.

`convert -resize 600 ~/Desktop/raw-fig.png final-figure.png`
Check for common characters to confuse. It is easy to replace a "1" (number one) with an "l" (Lima) or vice versa if the fonts you have in your terminal and web browser make those two characters. Make sure you are using the right quote character (e.g. ",", ', or ` are all different).

Another pair of characters that is sometime trouble are the Ø (zero) and 0 (Oscar).

Note that the pipe character is a vertical bar: "|". This character is sometimes two vertical dashes. On US keyboards it is located between the delete and return/enter keys and is the shift of "\".

**What to do if you get stuck?

Before we get into the commands, we need to talk about what to do if things get stuck. If you mistype a command and it just sits there doing nothing, you should first try holding down the "control" key and hitting the "C" key. This sends a "break" or "kill" message to the program. This is often written as "Ctrl-C" or "C-c". Here is a command that hangs. I then use Ctrl-C to get out of it. The bash shell responds with a "^C" and gives a prompt again.

```bash
#+BEGIN_EXAMPLE
egrep some-string
^C
#+END_EXAMPLE
```

If the command really gets stuck and does not respond to the Ctrl-C, you can close the terminal window and open a new window. Later on, you will learn fancier techniques for controlling programs (also known as processes), but this will work for now.
$pwd  # Print Working Directory or "Where am I?"
/home/CCOMNH/schwehr

$ mkdir example

$ cd example

$ ls

$ ls -l

total 0

$ ls -a

..  
.

# .  is the current directory

# ..  is the parent directory or "up"

$ ls -la

total 8

drwxr-xr-x 2 schwehr domain users 4096 2012-06-24 16:52 .

drwxr-xr-x 14 schwehr domain users 4096 2012-06-24 16:52 ..

current directory is /home/CCOMNH/schwehr

mkdir example

cd example

ls

ls -l

ls -a

# . is the current directory
# .. is the parent directory or "up"

ls -la

ls

ls -l

ls -a
schwehr@researchtools:~$ pwd  # Print Working Directory or "Where am I?"
/home/CCOMNH/schwehr
schwehr@researchtools:~$ mkdir example
schwehr@researchtools:~$ cd example
schwehr@researchtools:~/example$ ls
schwehr@researchtools:~/example$ ls -l
total 0
schwehr@researchtools:~/example$ ls -a
..  

schwehr@researchtools:~/example$  # . is the current directory
schwehr@researchtools:~/example$  # .. is the parent directory or "up"
schwehr@researchtools:~/example$ ls -la
total 8
drwxr-xr-x 2 schwehr domain users 4096 2012-06-24 16:52 .
```
```
schwehr@researchtools:~$ pwd  # Print Working Directory or "Where am I?"
/home/CCOMNH/schwehr
schwehr@researchtools:~$ mkdir example
schwehr@researchtools:~$ cd example
schwehr@researchtools:~/example$ ls
schwehr@researchtools:~/example$ ls -l
total 0
schwehr@researchtools:~/example$ ls -a
.
..
schwehr@researchtools:~/example$ # . is the current directory
schwehr@researchtools:~/example$ # .. is the parent directory or "up"
schwehr@researchtools:~/example$ ls -la
total 8
drwxr-xr-x 2 schwehr domain users 4096 2012-06-24 16:52 .
drwxr-xr-x 14 schwehr domain users 4096 2012-06-24 16:52 ..
schwehr@researchtools:~/example$
schwehr@researchtools:~$ pwd # Print Working Directory or "Where am I?"
/home/CCOMNH/schwehr
schwehr@researchtools:~$ mkdir example
schwehr@researchtools:~$ cd example
schwehr@researchtools:~/example$ ls
schwehr@researchtools:~/example$ ls -l
```
total 0
```
schwehr@researchtools:~/example$ ls -a
```
.
.. ...
```
schwehr@researchtools:~/example$ # . is the current directory
schwehr@researchtools:~/example$ # .. is the parent directory or "up"
schwehr@researchtools:~/example$ ls -la
```
total 8
drwxr-xr-x 2 schwehr domain users 4096 2012-06-24 16:52 .
drwxr-xr-x 14 schwehr domain users 4096 2012-06-24 16:52 ..
```
schwehr@researchtools:~/example$
schwehr@researchtools:~$ pwd  # Print Working Directory or "Where am I?"
/home/CCOMNH/schwehr
schwehr@researchtools:~$ mkdir example
schwehr@researchtools:~$ cd example
schwehr@researchtools:~/example$ ls
schwehr@researchtools:~/example$ ls -l
    total 0
schwehr@researchtools:~/example$ ls -a
  .  ...

  schwehr@researchtools:~/example$ # . is the current directory
  schwehr@researchtools:~/example$ # .. is the parent directory or "up"
  schwehr@researchtools:~/example$ ls -la
    total 8
drwxr-xr-x  2 Schwehr domain users 4096 2012-06-24 16:52 .
drwxr-xr-x 14 Schwehr domain users 4096 2012-06-24 16:52 ..
schwehr@researchtools:~/example$
schwehr@researchtools:~$ pwd  # Print Working Directory or "Where am I?"
/home/CCOMNH/schwehr
schwehr@researchtools:~$ mkdir example
schwehr@researchtools:~$ cd example
schwehr@researchtools:~/example$ ls
schwehr@researchtools:~/example$ ls -l
  total 0
schwehr@researchtools:~/example$ ls -a
  .  ..
schwehr@researchtools:~/example$ # . is the current directory
schwehr@researchtools:~/example$ # .. is the parent directory or "up"
schwehr@researchtools:~/example$ ls -la
  total 8
drwxr-xr-x 2 schwehr domain users 4096 2012-06-24 16:52 .
drwxr-xr-x 14 schwehr domain users 4096 2012-06-24 16:52 ..
schwehr@researchtools:~$ pwd # Print Working Directory or "Where am I?"
/home/CCOMNH/schwehr
schwehr@researchtools:~$ mkdir example
schwehr@researchtools:~$ cd example
schwehr@researchtools:~/example$ ls
schwehr@researchtools:~/example$ ls -l
 total 0
schwehr@researchtools:~/example$ ls -a
 ..
 ..
schwehr@researchtools:~/example$ # . is the current directory
schwehr@researchtools:~/example$ # .. is the parent directory or "up"
schwehr@researchtools:~/example$ ls -la
 total 8
 drwxr-xr-x 2 schwehr domain users 4096 2012-06-24 16:52 ..
 drwxr-xr-x 14 schwehr domain users 4096 2012-06-24 16:52 ..
schwehr@researchtools:~/example$
schwehr@researchtools:~/example$ cd ..
schwehr@researchtools:~$ # ~ or tilde is a reference to your "home directory"
schwehr@researchtools:~$ ls ~
a-folder another-folder anothership away example hw myship2 wx
schwehr@researchtools:~$ echo ~
/home/CCOMNH/schwehr
schwehr@researchtools:~$ ls ~schwehr
a-folder another-folder anothership away example hw myship2 wx
schwehr@researchtools:~$ ls ~jchadwick/
schwehr@researchtools:~$ ls -a ~jchadwick/
 .. .bash_history .bash_logout .bashrc .cache .irssi .profile .viminfo
schwehr@researchtools:~$ ls -a ~schwehr
 . another-folder .bash_aliases .bash_logout .emacs.d hw .lessht .ssh
 .. anothership .bash_aliases~ .bashrc example .ipython myship2 wx
a-folder away .bash_history .cache .gnupg .irssi .profile
schwehr@researchtools:~$ ls ~sc # now press tab after the sc
ls: cannot access ~sc: No such file or directory
schwehr@researchtools:~$ ls ~schwehr/
a-folder another-folder anothership away example hw myship2 wx
schwehr@researchtools:~$
schwehr@researchtools:~$ alias ls
alias ls='ls --color=auto'
schwehr@researchtools:~$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort.

Mandatory arguments to long options are mandatory for short options too.
-a, --all       do not ignore entries starting with .
-A, --almost-all do not list implied . and ..
         --author       with -l, print the author of each file
         --escape       print C-style escapes for non-graphic characters
         --block-size=SIZE use SIZE-byte blocks. See SIZE format below
-b, --escape    print C-style escapes for non-graphic characters
         --block-size=SIZE use SIZE-byte blocks. See SIZE format below
-B, --ignore-backups do not list implied entries ending with ~
-c         with -lt: sort by, and show, ctime (time of last
         modification of file status information)
         with -l: show ctime and sort by name
         otherwise: sort by ctime
-C         list entries by columns
         --color=[WHEN] colorize the output. WHEN defaults to `always'
         or can be `never' or `auto'. More info below
-d, --directory list directory entries instead of contents,
         and do not dereference symbolic links
-D, --dired    generate output designed for Emacs’ dired mode
-f         do not sort, enable -aU, disable -ls --color
schwehr@researchtools:~$ echo

schwehr@researchtools:~$ echo prints without doing anything
prints without doing anything

schwehr@researchtools:~$ echo hello world
hello world

schwehr@researchtools:~$ echo ~
/home/CCOMNH/schwehr

schwehr@researchtools:~$ echo ~jchadwick
/home/CCOMNH/jchadwick

schwehr@researchtools:~$ echo ~s
~sbenton  ~speech-dispatcher  ~ssuh  ~swright
~schwehr/  ~sranaweera  ~stephens  ~sync/
~sdenney  ~sree  ~sthein/  ~sys/
~semmed  ~srinivas  ~studenttest  ~syslog
~shachak  ~ssauth  ~svanhorn  ~ward
~slee  ~ssharma  ~swineberg
~soaresrosa  ~sshd/  

schwehr@researchtools:~$ echo ~g
~games/  ~glennm  ~gnats  ~gretchen  ~guest
~george  ~gmasetti/  ~greenaway  ~grice  ~guillermo
~ghostjewelfish  ~gmitchell/  ~greenlaw  ~gtibor

schwehr@researchtools:~$ echo ~g # pressing tab 2 times gives you all the matches
schwehr@researchtools:~$ ls ~schwehr/example
schwehr@researchtools:~$ echo ~schwehr/example
/home/CCOMNH/schwehr/example
schwehr@researchtools:~$ # press the "up arrow" to scroll back through history
schwehr@researchtools:~$ history | tail
1086 echo
1087 echo prints without doing anything
1088 echo hello world
1089 echo ~
1090 echo ~jchadwick
1091 clear
1092 ls ~schwehr/example
1093 echo ~schwehr/example
1094 # press the "up arrow" to scroll back through history
1095 history | tail
schwehr@researchtools:~$
schwehr@researchtools:~$ echo $HISTSIZE
1000
schwehr@researchtools:~$ history | head -5
  102  n
  103  ls -l
  104  ls 1*
  105  ls *0
  106  ls 1*2
schwehr@researchtools:~$ history | tail -5
  1098  echo $HISTSIZE
  1099  clear
  1100  echo $HISTSIZE
  1101  history | head -5
  1102  history | tail -5
schwehr@researchtools:~$
schwehr@researchtools:~$ man
What manual page do you want?
schwehr@researchtools:~$ # RTFM - Read the F* Manual
schwehr@researchtools:~$ # http://en.wikipedia.org/wiki/RTFM
schwehr@researchtools:~$ man df
NAME
df - report file system disk space usage

SYNOPSIS
df [OPTION]... [FILE]...

DESCRIPTION
This manual page documents the GNU version of df. df displays the amount of disk space available on the file system containing each file name argument. If no file name is given, the space available on all currently mounted file systems is shown. Disk space is shown in 1K blocks by default, unless the environment variable POSIXLY_CORRECT is set, in which case 512-byte blocks are used.

If an argument is the absolute file name of a disk device node containing a mounted file system, df shows the space available on that file system rather than on the file system containing the device node (which is always the root file system). This version of df cannot show the space available on unmounted file systems, because on most kinds of systems doing so requires very nonportable intimate knowledge of file system structures.

OPTIONS
Show information about the file system on which each FILE resides, or all file systems by default.

Manual page df(1) line 1

You are in a “pager.” Use the “q” key to quit out of the manual page. The space bar gets the next page.
AUTHOR

Written by Torbjorn Granlund, David MacKenzie, and Paul Eggert.

REPORTING BUGS

Report df bugs to bug-coreutils@gnu.org
GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
General help using GNU software: <http://www.gnu.org/gethelp/>
Report df translation bugs to <http://translationproject.org/team/>

COPYRIGHT

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

The full documentation for df is maintained as a Texinfo manual. If the info and df programs are properly installed at your site, the command

    info coreutils 'df invocation'

should give you access to the complete manual.

GNU coreutils 8.5           February 2011
Manual page df(1) line 83/109 (END)
schwehr@researchtools:~$ man
What manual page do you want?
schwehr@researchtools:~$ # RTFM - Read the F* Manual
schwehr@researchtools:~$ # http://en.wikipedia.org/wiki/RTFM
schwehr@researchtools:~$ man df
schwehr@researchtools:~$ man -k sort # apropos
alphasort (3)  - scan a directory for matching entries
apt-sortpkgs (1) - Utility to sort package index files
bsearch (3)  - binary search of a sorted array
bunzip2 (1)  - a block-sorting file compressor, v1.0.4
bzip2 (1)   - a block-sorting file compressor, v1.0.4
comm (1)   - compare two sorted files line by line
qsort (3)   - sorts an array
sort (1)   - sort lines of text files
sort-dctrl (1) - sort Debian control files
texindex (1) - sort Texinfo index files
tsrt (1)   - perform topological sort
versionsort (3) - scan a directory for matching entries
winop (3blt) - Perform assorted window operations
schwehr@researchtools:~$ man sort
NAME
sort - sort lines of text files

SYNOPSIS
sort [OPTION]... [FILE]...
sort [OPTION]... --files0-from=F

DESCRIPTION
Write sorted concatenation of all FILE(s) to standard output.

Mandatory arguments to long options are mandatory for short options too. Ordering options:

-b, --ignore-leading-blanks
  ignore leading blanks

-d, --dictionary-order
  consider only blanks and alphanumerical characters

-f, --ignore-case
  fold lower case to upper case characters

-g, --general-numeric-sort

Manual page sort(1) line 1
schwehr@researchtools:~/example$ cd
schwehr@researchtools:~$ touch 1 2 3
schwehr@researchtools:~$ ls -l
total 24
-rw-r---r-- 1 schwehr domain users 0 2012-06-24 17:24 1
-rw-r---r-- 1 schwehr domain users 0 2012-06-24 17:24 2
-rw-r---r-- 1 schwehr domain users 0 2012-06-24 17:24 3
drwxr-xr-x 2 schwehr domain users 4096 2012-06-23 14:44 a-folder
drwxr-xr-x 2 schwehr domain users 4096 2012-06-23 14:44 another-folder
-rw-r---r-- 1 schwehr domain users 0 2012-06-23 14:41 anothership
drwxr-xr-x 2 schwehr domain users 4096 2012-06-23 14:32 away
drwxr-xr-x 2 schwehr domain users 4096 2012-06-23 14:52 example
drwx------- 4 schwehr domain users 4096 2011-10-02 15:40 hw
-rw-r---r-- 1 schwehr domain users 0 2012-06-23 14:39 myship2
drwxr-xr-x 3 schwehr domain users 4096 2011-09-06 06:28 wx
schwehr@researchtools:~$ rm 1 2 3
rm: remove regular empty file `1'? y
rm: remove regular empty file `2'? y
rm: remove regular empty file `3'? y
schwehr@researchtools:~$ alias rm
alias rm='rm -i'
schwehr@researchtools:~$ touch 1 2 3
schwehr@researchtools:~$ rm -i 1 2 3
rm: remove regular empty file `1'? y
rm: remove regular empty file `2'? n
rm: remove regular empty file `3'? y
schwehr@researchtools:~$ touch 1 2 3 4 5 6 7 8 9 10 100 11 12 13
schwehr@researchtools:~$ ls -d *
a-folder anothership example myship2
10 11 13 3 5 7 9 another-folder away hw wx
schwehr@researchtools:~$ # Don't do "rm *" DANGER!
schwehr@researchtools:~$ ls 1*
1 10 100 11 12 13
schwehr@researchtools:~$ ls *0
10 100
schwehr@researchtools:~$ ls 1*0
10 100
schwehr@researchtools:~$ ls 1*3
13
schwehr@researchtools:~$ touch foo.tar
schwehr@researchtools:~$ ls *.tar
foo.tar
schwehr@researchtools:~$ ls *.txt
ls: cannot access *.txt: No such file or directory
schwehr@researchtools:~$ ls ?
1 2 3 4 5 6 7 8 9
schwehr@researchtools:~$ ls ??
10 11 12 13

hw:
02 03
schwehr@researchtools:~$ ls ??
10 11 12 13

hw:
02 03

wx:
ccom-airmar-2011-08-28  ccom-airmar-2011-08-29  ccom-airmar-2011-08-30  nmeadec

schwehr@researchtools:~$ ls -d ??
10 11 12 13  hw  wx

schwehr@researchtools:~$ ls -dl ??
-rw-r--r-- 1 schwehr domain users 0 2012-06-24 17:27 10
-rw-r--r-- 1 schwehr domain users 0 2012-06-24 17:27 11
-rw-r--r-- 1 schwehr domain users 0 2012-06-24 17:27 12
-rw-r--r-- 1 schwehr domain users 0 2012-06-24 17:27 13
drwx------ 4 schwehr domain users 4096 2011-10-02 15:40  hw
drwxr-xr-x 3 schwehr domain users 4096 2011-09-06 06:28  wx

schwehr@researchtools:~$ ls -d 1?
10 11 12 13

schwehr@researchtools:~$ ls -d ?3
13

schwehr@researchtools:~$
schwehr@researchtools:~$ ls [2-5]
2 3 4 5
schwehr@researchtools:~$ ls -d [a-j]*
a-folder another-folder anothership away example foo.tar hw
schwehr@researchtools:~$ ls -ld [m-z]*  # anything that starts with m through z
-rw-r--r-- 1 schwehr domain users 0 2012-06-23 14:39 myship2
drwxr-xr-x 3 schwehr domain users 4096 2011-09-06 06:28 wx
schwehr@researchtools:~$
# anything with exactly two letters
ls ??
# 10 11 12 13
# the letter "1" followed by any single
ls 1?
# 10 11 12 13
++END_SRC

You can get fancier by using square brackets of characters or ranges by putting a dash. It's best to just see some examples.

++BEGIN_SRC sh
# List files that are one character of the number 2 through 5
ls [2-5]
# 2 3 4 5

# List files that start with 1 and have a 1 or 3 following.
ls 1[13]
# 11 13

# Combine the * and [] to ask for any file ending in 1 or 3
ls *[13]
# 1 11 13 3

# Here we are using a special system directory for an example using a
# range of alphabetical characters (x, y, & z).
# Please do not worry about what these files are
ls /sbin/*[x-z]
#/sbin/fscck.minix /sbin/getty /sbin/iwspy /sbin/mkfs.minix /sbin/pam_tally
++END_SRC

* Using a Virtual Machine (VM) version of Ubuntu Linux

Before we go any more into the shell, it is worth taking the time to show you how to be able to do this type of thing on your own computer and not always have to log into researchtools.com.mh.