

Unix Tutorial
for
FreeSurfer Users

Ender Konukoglu

What is Unix/Linux?

- An operating system
(like Windows and OS X)
- Linux is the free, modifiable, and redistributable version of Unix
- Why use it?

What is Unix/Linux?

- An operating system
(like Windows and OS X)
- Linux is the free, modifiable, and redistributable version of Unix
- Why use it?
 - power to write many scripts with many commands to work with lots of data
 - to use computer resources on the network efficiently, such as clusters

Getting Started

Communicate with operating system through a “shell” or terminal window.

For Linux:

Double click Terminal icon on Desktop

For Macs:

Double click on hard drive

Applications > Utilities > X11 (double click)

Applications > Utilities > Terminal

Warm Up

Type:

date

and hit enter.

```
[astevens@gate ~]$
```

Warm Up

Type:

date

and hit enter. Should see

Mon Apr 6 8:05:24 EDT 2009

Warm Up

Type:

date

and hit enter. Should see

Mon Apr 6 8:05:24 EDT 2009

Type:

cal

and hit enter.

Warm Up

Type:

date

and hit enter. Should see

Mon Apr 6 8:05:24 EDT 2009

Type:

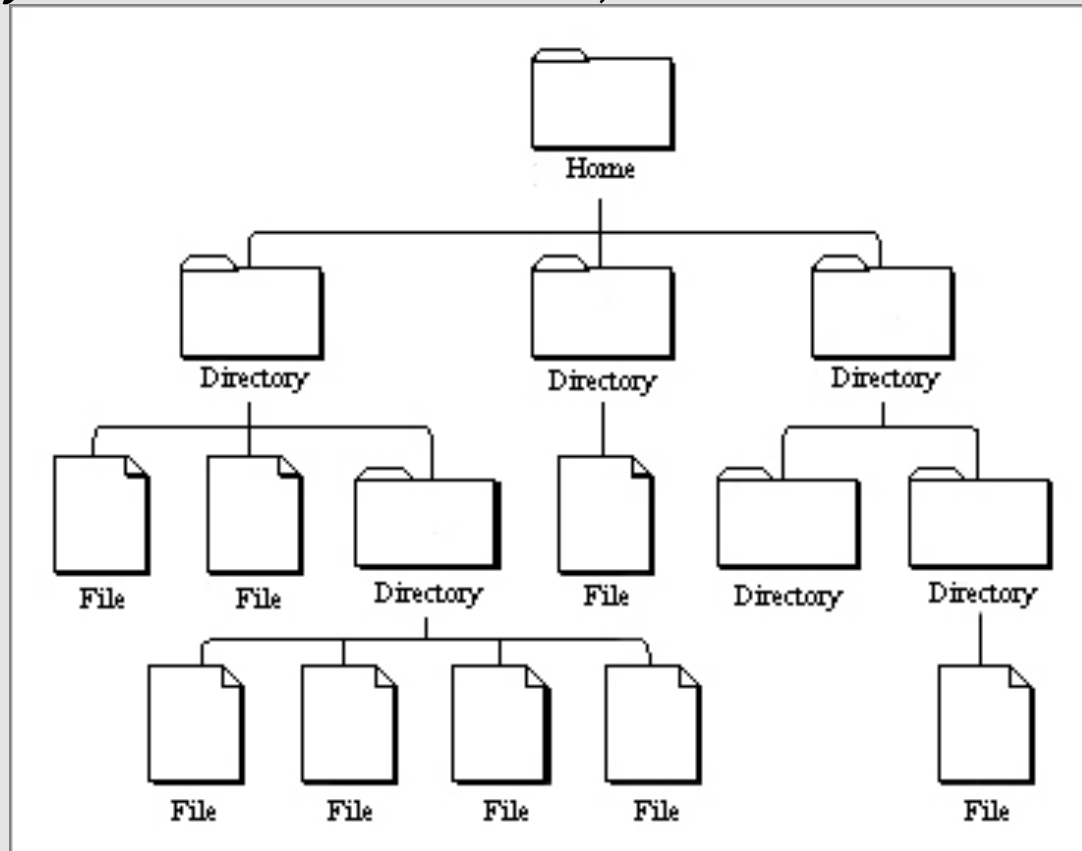
cal

and hit enter. Should see

April 2009
Su Mo Tu We Th Fr Sa
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30

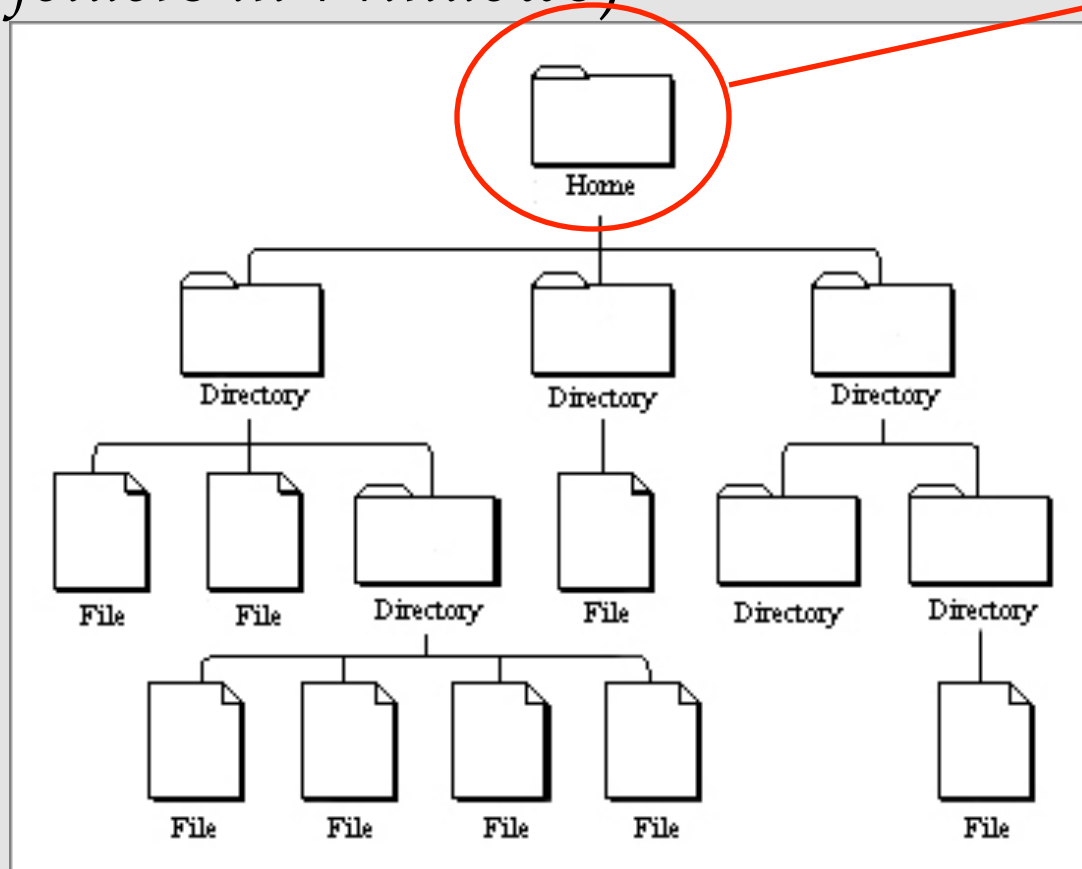
Directories

- Unix uses a hierarchical file system
(think folders in Windows)



Directories

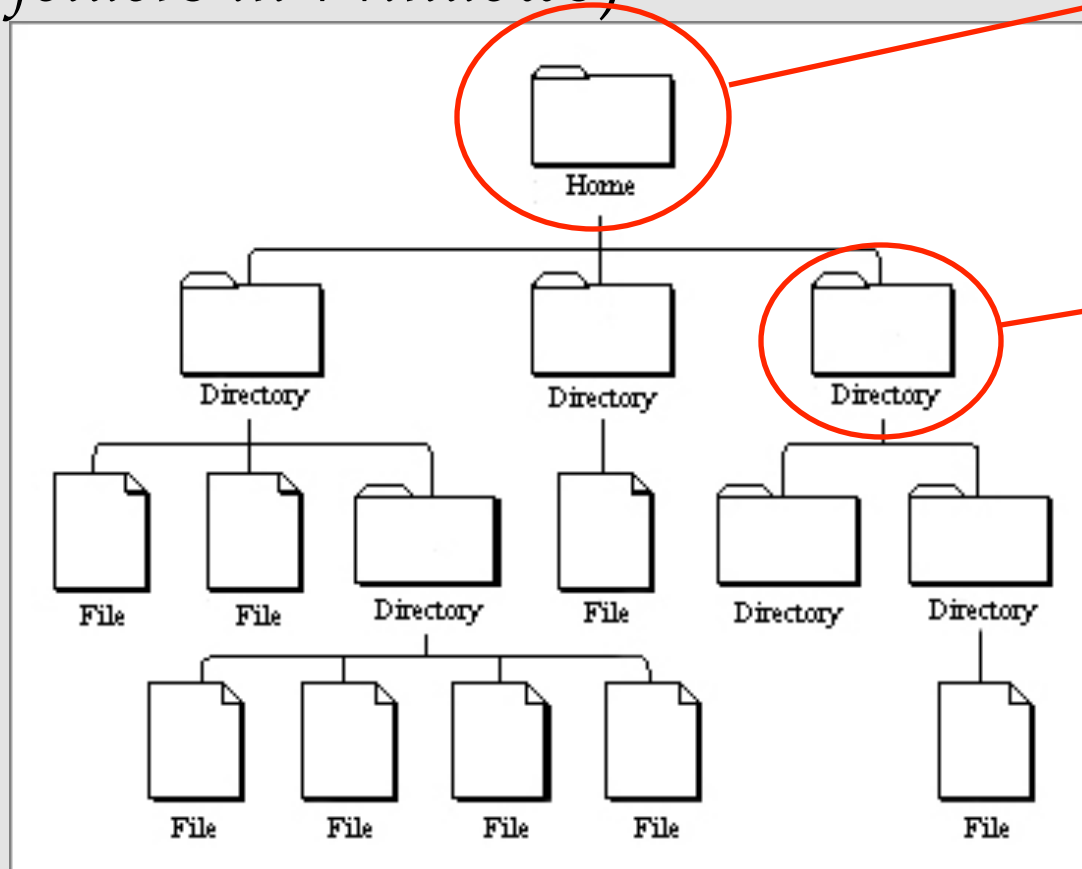
- Unix uses a hierarchical file system
(think folders in Windows)



Home is like
“My Computer”

Directories

- Unix uses a hierarchical file system
(think folders in Windows)

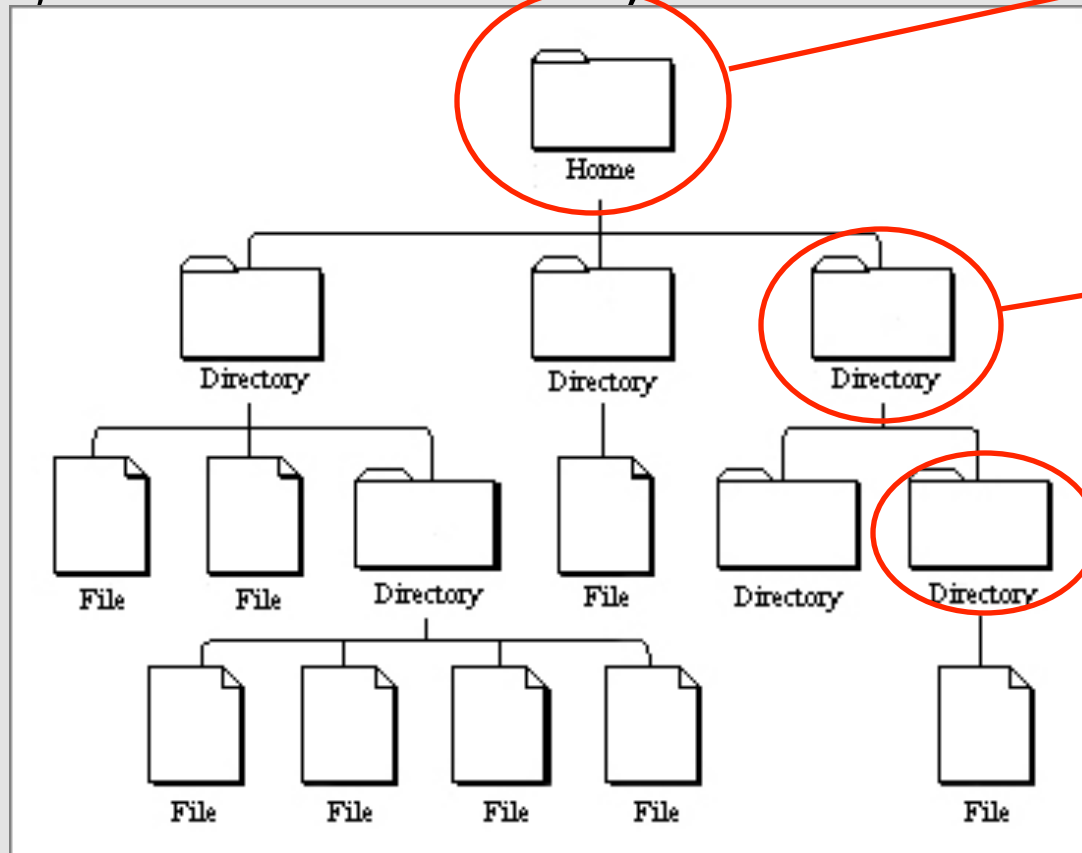


Home is like
"My Computer"

Like "My
Documents"

Directories

- Unix uses a hierarchical file system
(think folders in Windows)



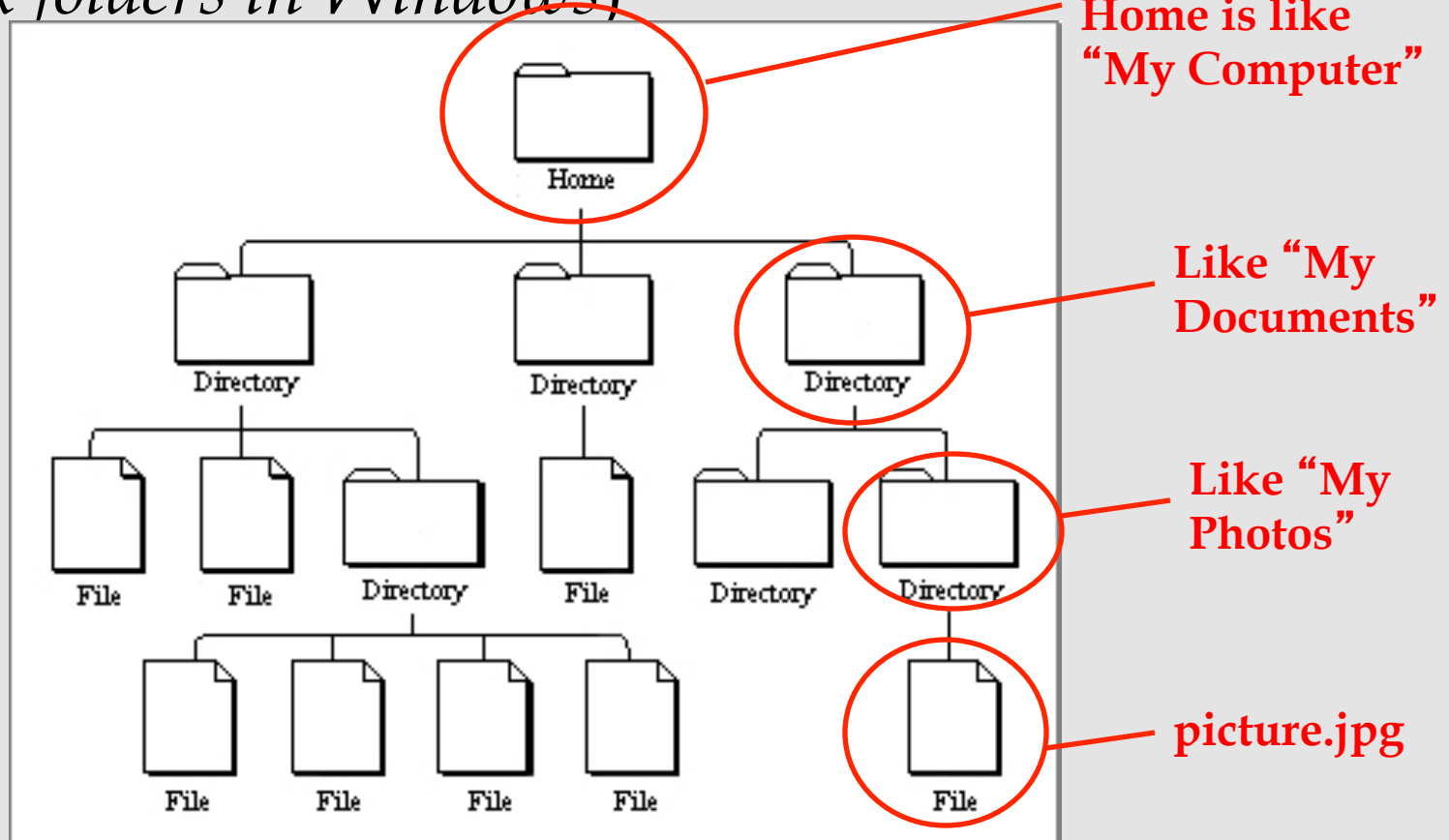
Home is like
"My Computer"

Like "My
Documents"

Like "My
Photos"

Directories

- Unix uses a hierarchical file system
(*think folders in Windows*)



Location

Type:

pwd

and hit enter. Should see

/home/nmrclass

OR

/Users/YourName

shows “present working directory” or current location as a *path*

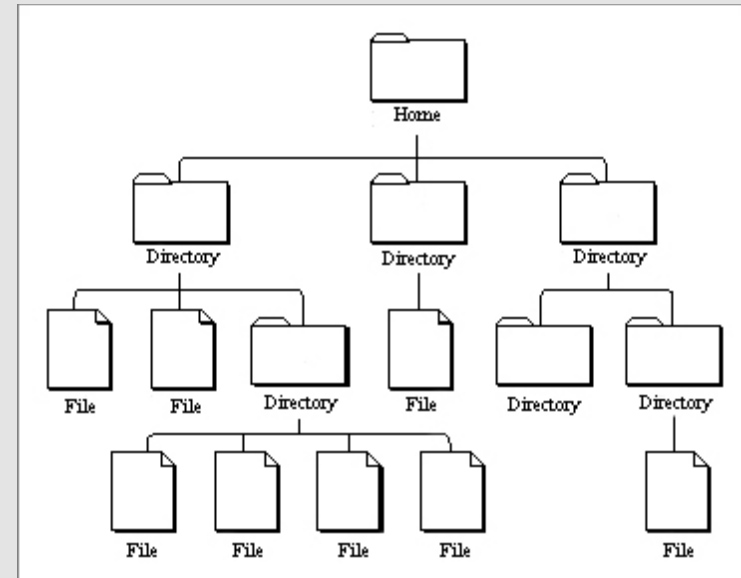
similar to:

/MyComputer/MyDocuments/MyPhotos

Opening a Directory

- Not double clicking
- Type command to “open”

```
[astevens@gate ~]$ █
```



- Commands to open files will differ

Navigating Directories

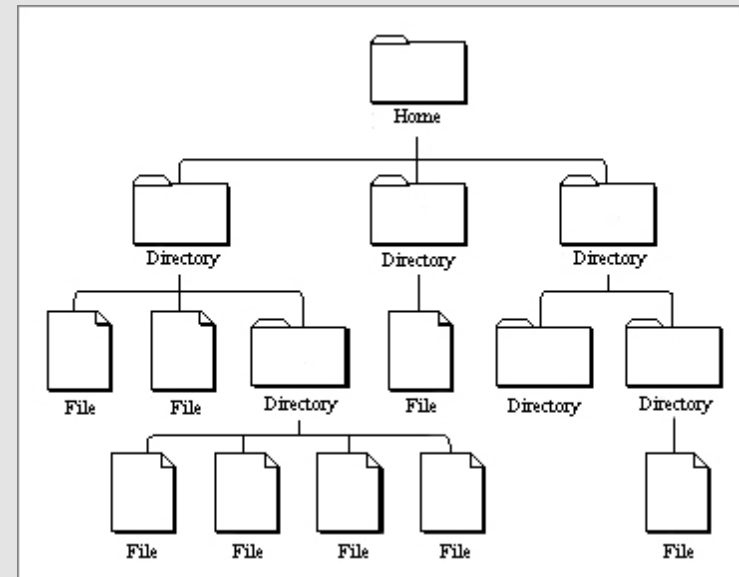
cd _

“change directory”: move into a folder

ls

“list”: see contents of directory

ls _



Navigating Directories

cd _

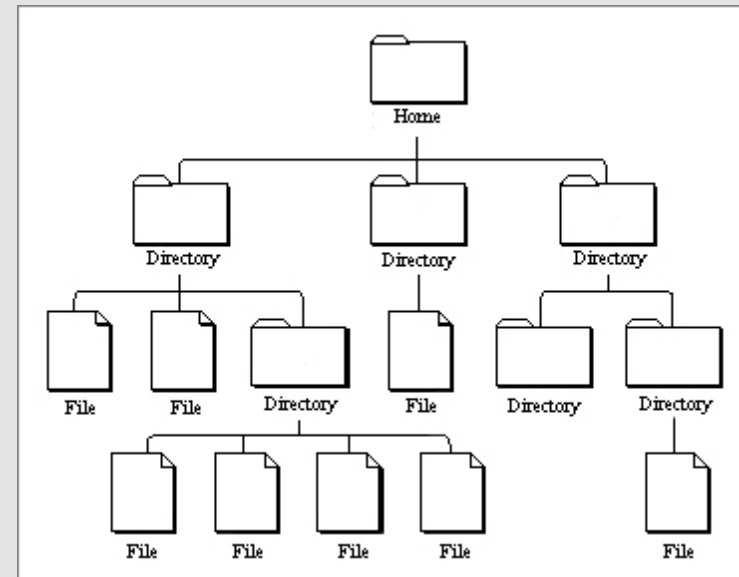
“change directory”: move into a folder

ls

“list”: see contents of directory

ls _

```
Desktop matlab tmp TUTORIAL_DATA
```



Anatomy of a Command

```
command -option1 -option2 file
```

```
command --help
```

Anatomy of a Command

```
command -option1 -option2 file
```

```
command --help
```

Try:

```
ls --help
```

Directory Contents

- List contents of directory you are in

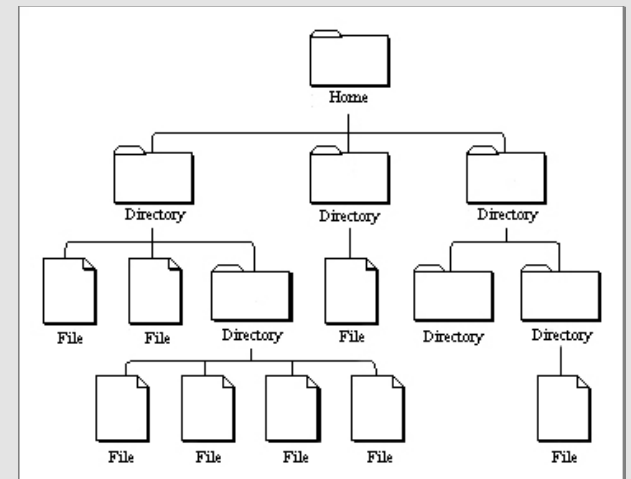
ls

lists names of directories/files

ls -a

ls -l

ls -lrt



Directory Contents

- List contents of directory you are in

ls

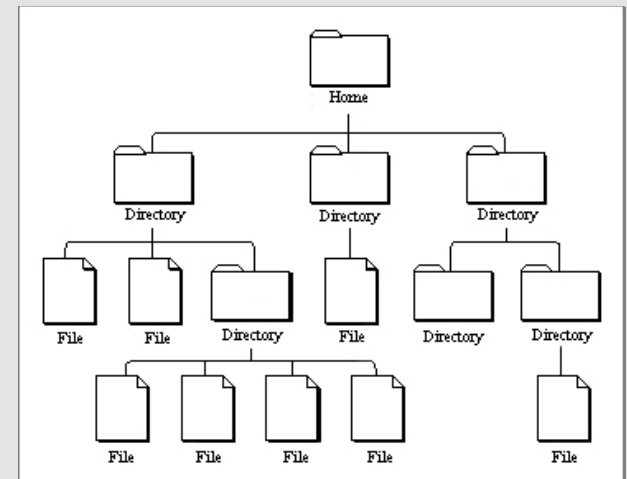
lists names of directories/files

ls -a

lists hidden files too

ls -l

ls -lrt



.xdebug_tkmedit

.cshrc

.alias

Directory Contents

- List contents of directory you are in

ls

lists names of directories/files

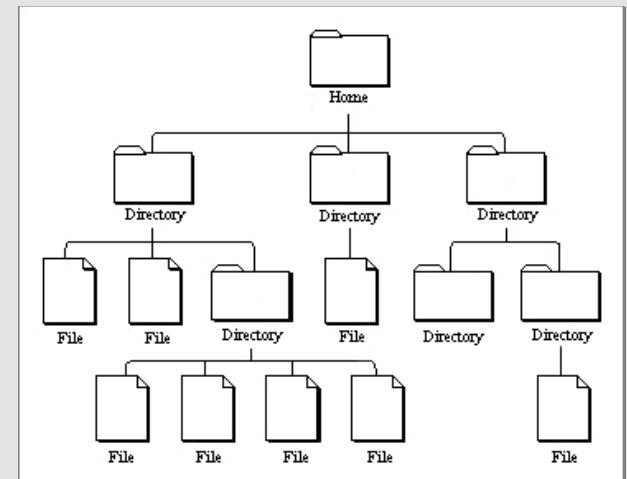
ls -a

lists hidden files too

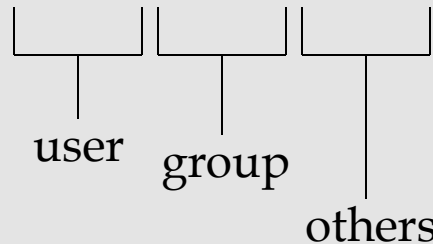
ls -l

lists file details

ls -lrt



drwxrwx---



Directory Contents

- List contents of directory you are in

ls

lists names of directories/files

ls -a

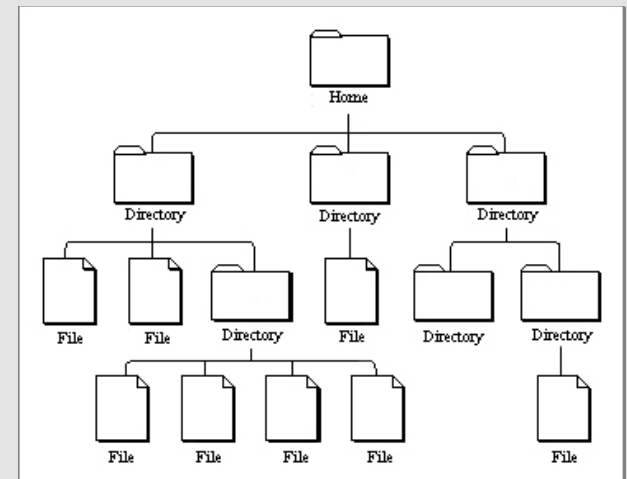
lists hidden files too

ls -l

lists file details

ls -lrt

lists recent files last



Save Some Time

Filename Completion

ls Des

hit Tab key

should see

ls Desktop

hit enter

History

hit ↑ key

history

should see

ls Desktop

should see the recent commands

Changing Directories

```
mkdir practice
```

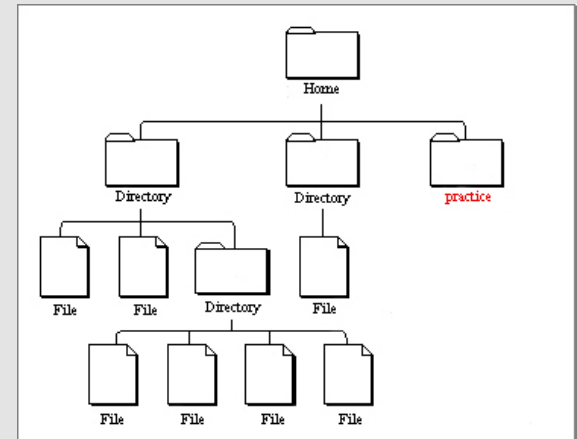
makes a new directory “practice”

```
ls -lrt
```

```
pwd
```

should see

```
/home/nmrclass
```



```
cd practice
```

changes to directory “practice”

```
pwd
```

should see

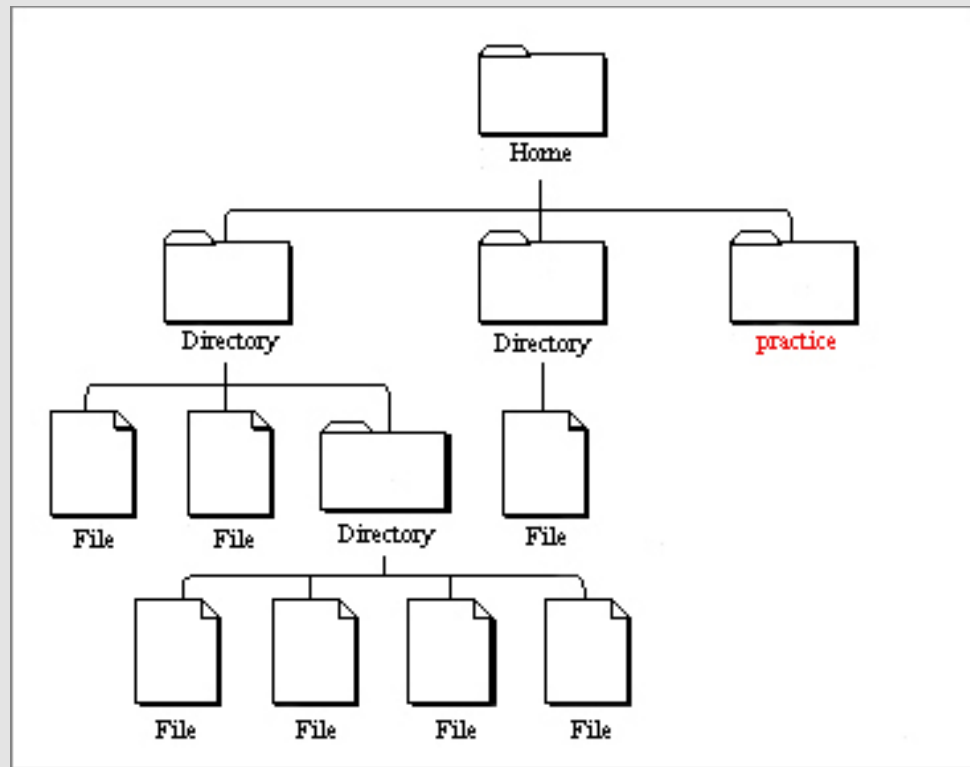
```
/home/nmrclass/practice
```

```
ls
```

should see

Nothing!

Changing Directories



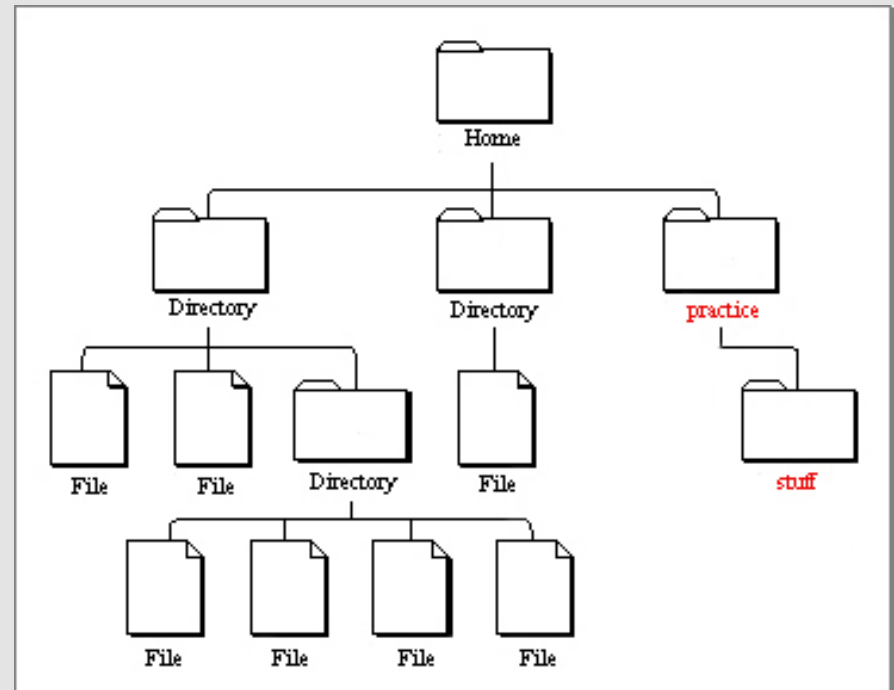
Changing Directories

```
mkdir stuff
```

makes folder “stuff” inside practice

```
ls
```

should see “stuff”



Using an Editor

```
emacs mynotes.txt
```

If using a Mac:

```
pico mynotes.txt
```

Type: I could write a script in here.

File > Save (Buffer)

Ctrl-x if using pico
and then 'Y' and enter

File > Exit emacs

```
ls
```

should see “mynotes.txt”

Using an Editor

```
gedit mynotes.txt
```

If using a Mac:

```
pico mynotes.txt
```

Type: I could write a script in here.

File > Save (Buffer)

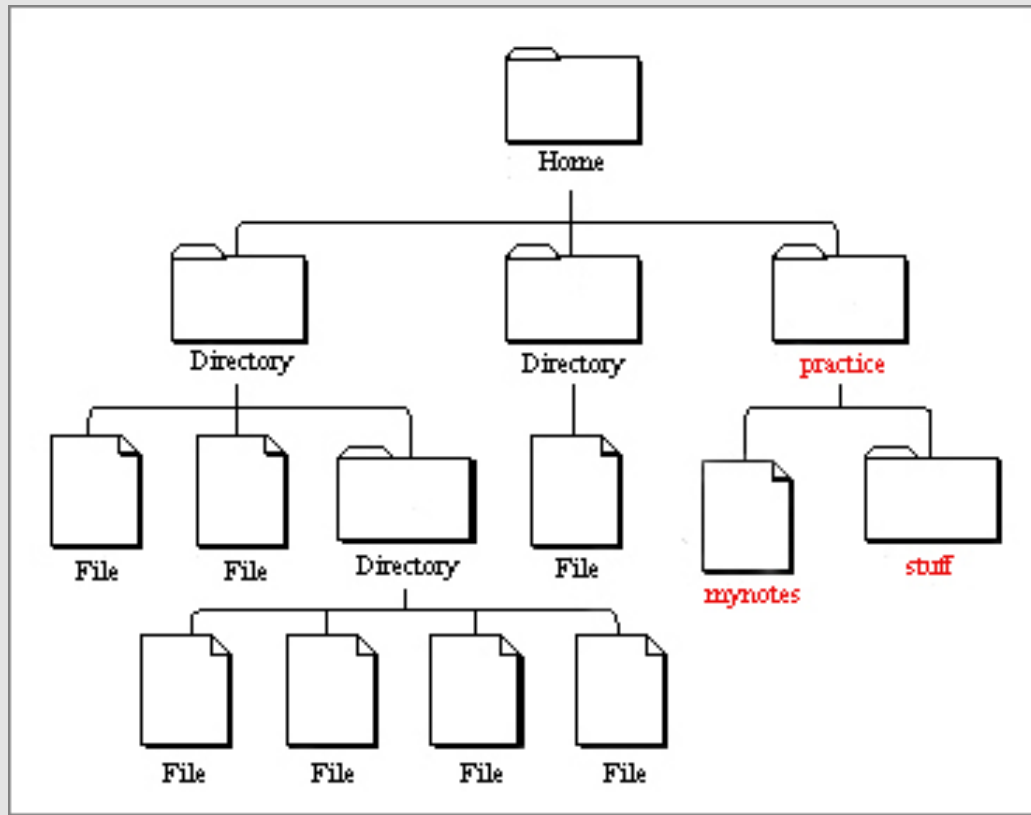
Ctrl-x if using pico
and then 'Y' and enter

File > Exit emacs

```
ls
```

should see “mynotes.txt”

Using an Editor



Copying files

cp

is the copy command

cp --help

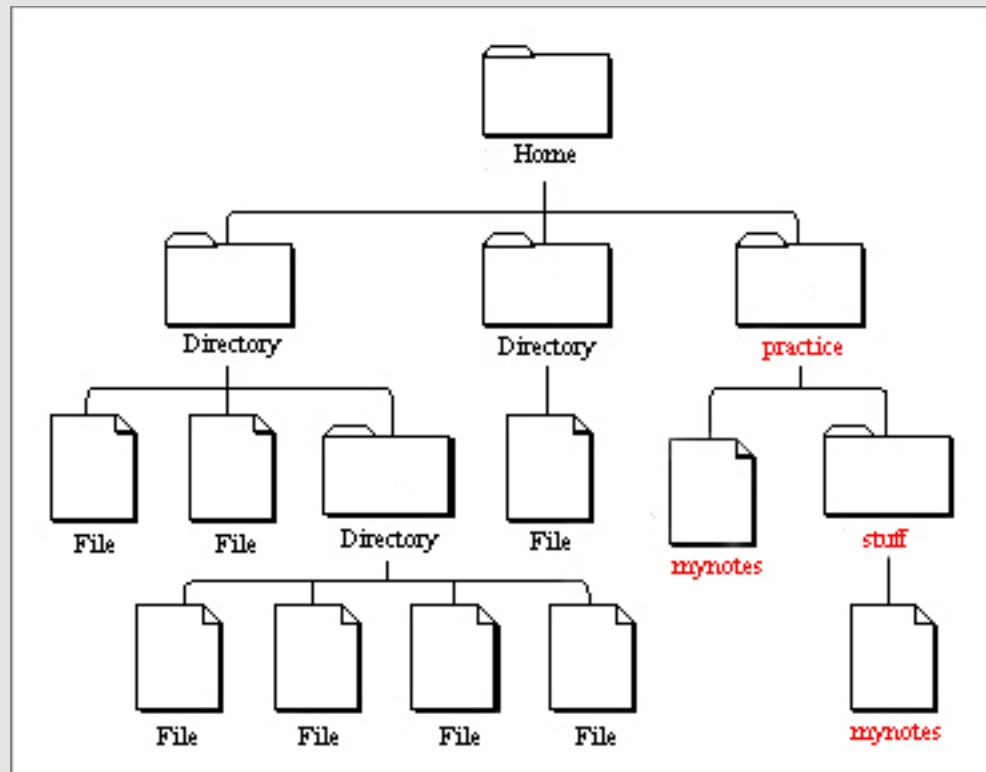
learn all the options or “arguments”

cp mynotes.txt stuff

cd stuff

ls

more mynotes.txt



Copying files

cp

is the copy command

cp --help

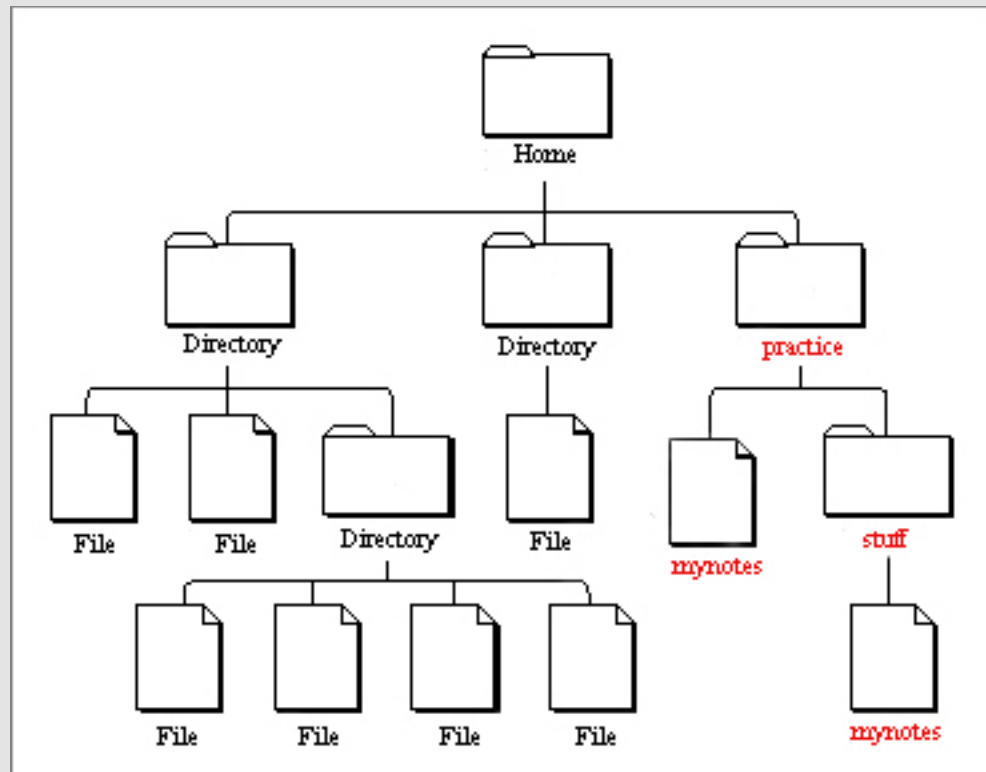
learn all the options or “arguments”

```
cp mynotes.txt stuff
```

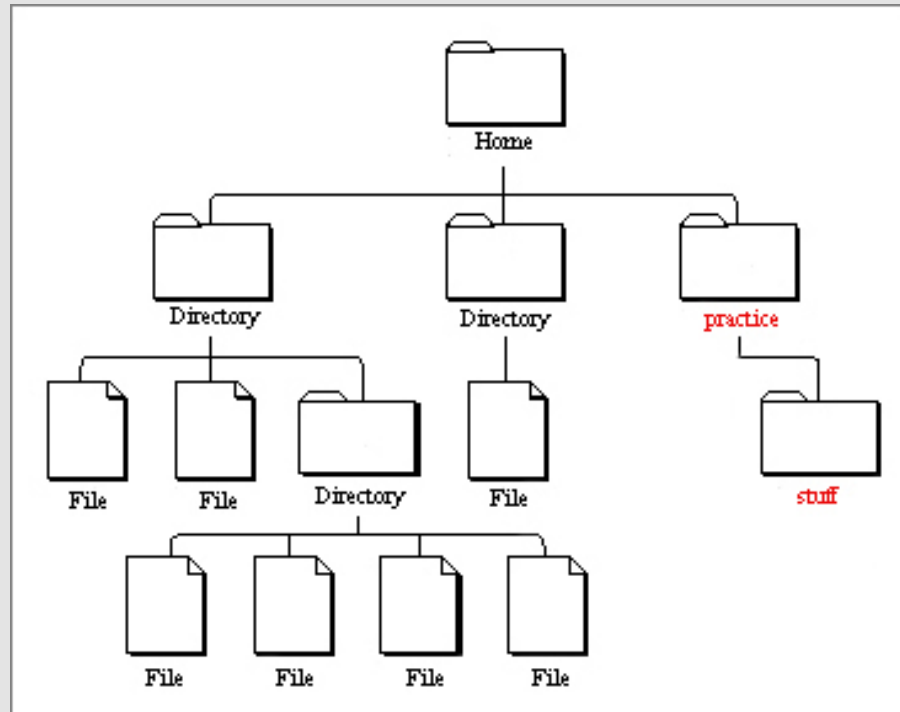
```
cd stuff
```

```
ls
```

```
less mynotes.txt
```



Changing Directories



```
ls ..
```

```
ls ../../
```

*shows one directory up
goes up two!*

can also do (but don't):

```
cd ..
```

```
cd ../../
```

```
pwd
```

should see

```
/home/nmrclass/practice/stuff
```

Copying / Moving files

Could also use do:

```
cp mynotes.txt myothernotes.txt
```

```
mv myothernotes.txt hernotes.txt
```

```
mv hernotes.txt ..
```

Removing Files

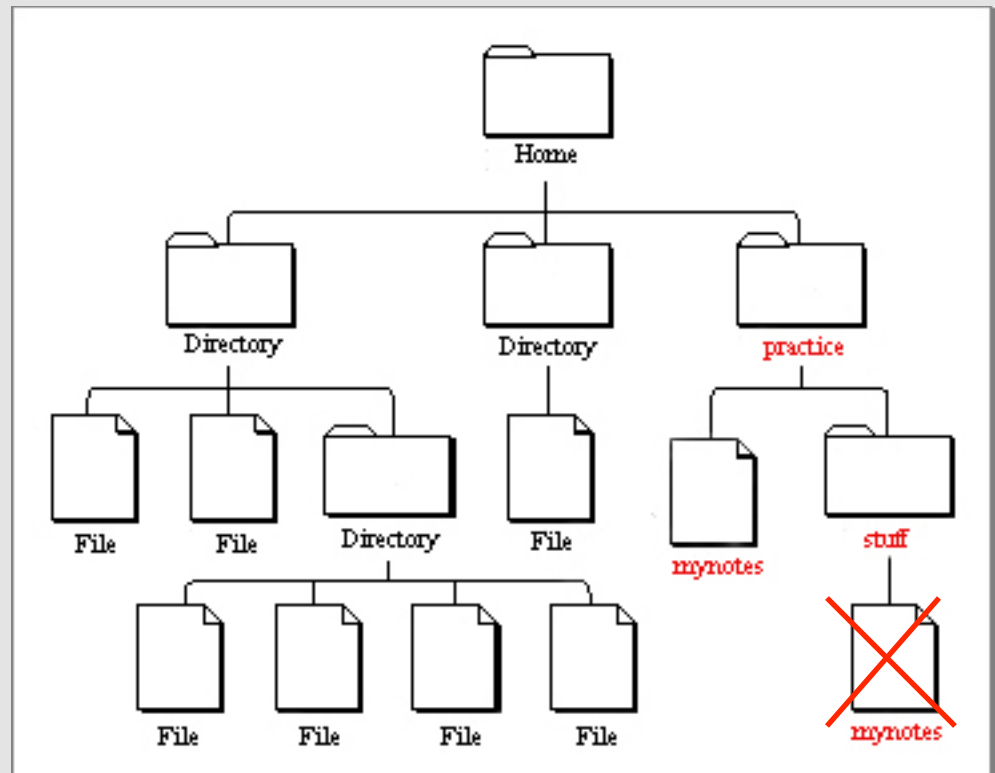
pwd

should be in "stuff"

ls

rm mynotes.txt

ls



Review

- no double clicking
- mkdir
- ls
- cd
- pwd
- emacs, gedit, vi, pico
- cp
- rm

Using FreeSurfer

With FreeSurfer, certain variables must be set in order to use it correctly:

FREESURFER_HOME

tell Operating System where FreeSurfer is

SUBJECTS_DIR

tell FreeSurfer where data is

Required Variables

```
setenv SUBJECTS_DIR /path/to/data
```

To go to location of your data:

```
cd $SUBJECTS_DIR
```

\$ means take the value of the variable

Required Variables

```
setenv SUBJECTS_DIR /path/to/data
```

To go to location of your data:

```
cd $SUBJECTS_DIR  
    aka  
cd /path/to/data
```

\$ means take the value of the variable

\$

How 'echo' works:

```
echo It is called football not soccer.
```

To set a variable:

```
setenv TEST_VARIABLE yourfirstname
```

To check what a variable is set to:

```
echo $TEST_VARIABLE
```


Required Variables

With FreeSurfer, certain variables must be set in order to use it correctly:

FREESURFER_HOME

tell Operating System where FreeSurfer is

SUBJECTS_DIR

tell FreeSurfer where data is

```
echo $FREESURFER_HOME
```

← To check variables
↓

```
echo $SUBJECTS_DIR
```

Visualization Tools

- For FreeSurfer visualization tools:

```
freeview volume1 volume2 -f surface1 surface2
```

```
tksurfer subj001 lh inflated
```

```
tkmedit subj001 T1.mgz -aux wm.mgz
```

```
VisualizationTool subject data_file -aux other_data_file
```

```
VisualizationTool subject hemisphere data_file
```

More Help

[http://surfer.nmr.mgh.harvard.edu/
fswiki/FsTutorial/
CommandLineNavigation](http://surfer.nmr.mgh.harvard.edu/fswiki/FsTutorial/CommandLineNavigation)

Links on Wiki under “Unix Tutorial”

Glossary of Unix commands

The End

Good Luck!

Writing Scripts

- Putting multiple commands together.
- Automatically running a sequence of commands
- Example: create a file named `my_first_script.csh` with the contents:

```
#!/bin/csh  
  
setenv name FreeSurfer  
echo I love $name !
```

```
% chmod u+x my_first_script.csh  
% ./my_first_script.csh
```

Loops in Scripts

- You can create loops, if statements, ...
- Example: create a file named `my_first_loop.csh` with the contents:

```
#!/bin/csh

foreach name ('Bill Murray' 'Wes Anderson')
    echo I love $name !
end
```

Inputs to Scripts

- You can create take inputs from the command line
- Example: create a file named `my_first_IO.csh` with the contents:

```
#!/bin/csh
```

```
echo The $0 command is called with $#argv parameters
```

```
echo parameter 1 is $1
```

```
echo parameter 2 is $2
```