# step 1: ftpscan

- produce a program to handle all the arguments
- arguments are:
  - `-c` which counts the number of times a file is accessed
  - `-h` which displays a simple help line
- two arguments at present
  - could write the argument handling code ourself
  - better solution is to use getopts why?



# Arguments for ftpscan

- initially let us write the usage and imports:
- ```
  #!/usr/bin/python

  import sys, getopt

  def Usage ():
      print "ftp [-c][-h]"
      sys.exit(0)
  ```



# Arguments for ftpscan

- now lets collect the arguments:
- ```
  try:
      optlist, list = getopt.getopt(sys.argv[1:],
                                    ':ch')
  except getopt.GetoptError:
      Usage()
      sys.exit(2)
  ```



# Arguments for ftpscan

- now we process the arguments:
- ```
  for opt in optlist:
      print opt
      if opt[0] == '-h':
          Usage()
      if opt[0] == '-c':
          print "c option found"
  ```
  ```

# Arguments for ftpscan

- remember that `getopt` consists of two elements:
  - the first is a list of (option, value) pairs
  - the second is the list of program arguments left after the option list was stripped
- consider:

```
>>> import getopt
>>> args = '-a -b -cfoo -d bar a1 a2'.split()
>>> args
['-a', '-b', '-cfoo', '-d', 'bar', 'a1', 'a2']
>>> optlist, args = getopt.getopt(args, 'abc:d:')
>>> optlist
[('-a', ''), ('-b', ''), ('-c', 'foo'), \
 ('-d', 'bar')]
>>> args
['a1', 'a2']
```



# Long arguments

- using long option names is equally easy (although these are not needed for ftpscan!)

```
>>> s = '--condition=foo --testing --output-file \
        abc.def -x a1 a2'
>>> args = s.split()
>>> args
['--condition=foo', '--testing', '--output-file',
  'abc.def', '-x', 'a1', 'a2']
>>> optlist, args = getopt.getopt(args, 'x', [
...  'condition=', 'output-file=', 'testing'])
>>> optlist
[('--condition', 'foo'), ('--testing', ''),
 ('--output-file', 'abc.def'), ('-x', '')]
>>> args
['a1', 'a2']
```



# Step 2: ftpscan

- make the python program read the ftp log file a line at a time

```
#!/usr/bin/python

from sys import argv

def scanner(name, function):
    file = open(name, 'r')
    line = file.readline()
    while line:
        function(line)
        line = file.readline()
    file.close()

def processLine(line):
    print line

filename = 'ftplog.data'
scanner(filename, processLine)
```

- you will also need to download this test data file ftplog.data ⟨`ftplog.data`⟩.



# Step 2: ftpscan

- notice that this snippet of code is extremely useful :-)
- it provides a scanner function which scans a file
  - it reads the file a line at a time and
  - then calls the *function* to operate on each line
- currently it calls *processLine*
  - which simply prints the line

# ftpscan: step 3

- now alter the program to cut the line into *date*, *client machine*, *file*
  - display these component strings
    - think `split`
- here is a line (which is broken into several to fit the slide):
- 

```
Mon Nov 18 17:38:24 2002 1 \
g410-helpdesk.isd.glam.ac.uk 51200 \
/user/ftp/pub/m2/osi/lab1.tar b _ o a \
IEUser@ ftp 0 * c
```



# ftpscan: step 3

- using split write some code which prints the *date*, *client machine*, *file*



# ftpscan: step 3

- 

```
#!/usr/bin/python

import string

def processLine(line):
    words=string.split(line)
    print "date = ", words[:5]
    print "machine = ", words[-7]
    print "file = ", words[-5]

processLine("Mon Nov 18 17:38:24 2002 1 \
             g410-helpdesk.isd.glam.ac.uk \
             51200 \
             /user/ftp/pub/m2/osi/lab1.tar \
             b _ o a")
```