Information Infrastructure II

INFO I211 – Spring 2014 – Sections 18530 & 22519

Lecture 12 – 2014.02.24 & 2014.02.25

Instructor:
Mitja Hmeljak,
http://mypage.iu.edu/~mitja
mitja@indiana.edu

Recap: Python Standard Library

Documentation about the Python Standard Library:

http://docs.python.org/2.6/library/

A comprehensive guide to the built-in modules in 2.6 Python.

Recap: Modules

Every Python file is a module...

Modules contain groups of related functions

- **math** contains a lot of useful mathematical code
- Modules can also contain classes or constants
  - **math.pi**

Normally when we import a module we then have to say **modulename.methodname()**

Recap: Ways to Import Pyhon Modules:

```
import mathFunc
mathFunc.summation([1,2,3])

import mathFunc as mf
mf.summation([1,2,3])

from mathFunc import *
summation([1,2,3])
# not recommended: potential name clash
```

The **os** and **sys** modules: working with Files & Directories

```
import os
myDirPath = os.getcwd()
path = myDirPath + "/testfile.py"
```

Remove the path from a filename:

```
    os.path.basename(path)    ->
    "testfile.py"
```

Join parts of a path (OS independent: solves the "/" vs. "\" issue)

```
    os.path.join(myDirPath, "testfile", ".py")
```

Dynamic File Names

We can create file names dynamically, i.e. decide how to name a file when the program is running (filenames and directory names don't need to be hardcoded anymore) :

```
import os

myDirPath = os.getcwd()

for i in range (3):
    f = open(os.path.join(myDirPath,"test" + str(i) + ".txt"),"w")
    f.write("Test")
    f.close()
```

Reading from CSV Files

CSV – Comma Separated Values

Text file with data stored as a table, with commas between columns:

```
Jim,25,1.68,75.3,Black,Brown
Alice,31,1.72,61.6,Orange,Green
Sam,51,1.87,94.2,Blond,Black
```

Reading from CSV Files

Reading a CSV:

Manually read it line by line, split the lines by "**,**" – and you are done...

… except that some CSV files may have **"** or **'** around the values, or different amounts of whitespace, or even use a different separator: "**;**" instead of "**,**" etc.

So we can use the **csv** module to read the files, and let it do the work...

CSV Reader

```
import os
import csv

myDirPath = os.getcwd()

f = open(os.path.join(myDirPath,"people.csv"), "r")
read = csv.reader(f)

for row in read:
    print row
```

`people.csv` is in the Resources → SampleCode folder on Oncourse

Dates & Time Module (datetime)

Dates can be constructed and formatted:

Try this with a few different dates.

```
import datetime

now = datetime.date.today()
print now

now = datetime.date(2000, 1, 1)
print (now.strftime("%m-%d-%y. %d %b %Y
is a \    %A on the %d day of %B"))
```

Dates & Time Module (datetime)

Dates support calendar arithmetic:

Try this with your own birthday.

```
import datetime

now = datetime.date.today()
birthday = datetime.date(1981, 2, 29)
age = now - birthday

print "I am", age.days, "days old."
print "Which is", age.days / 365, "years."
```

When to use the Python Standard Library?

Don't reinvent the wheel...

Check to see if the standard library already does what you need.

But you should be – capable – of implementing many of the simpler methods from it on your own.

Reconnecting to where we left at Lecture 5...

Towards a Distributed Application: Connecting to the Web

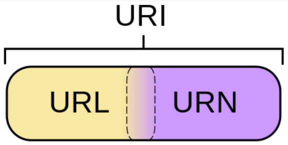
Python can do many things over the network:

- Emails
- FTP, SSH
  - Transferring files
- HTTP
  - Retrieving web pages

How can we use HTTP in Python to connect to web pages from our programs?

Reconnecting to where we left at Lecture 5...

Connecting to the Web: Uniform Resource Locators



URL – Uniform Resource Locator

- Web address (a special case of a U.R.Identifier)
- Ex: http://www.python.org
- Ex: ftp://ftp.ncbi.nlm.nih.gov/

urllib – *a Python 2.x library*

- For opening a connection to a URL & reading contents
- Web contents are just like file contents...

Reconnecting to where we left at Lecture 5...

Connecting to the Web from Python

Try this code – open pico *on* silo.soic.indiana.edu, *in your* ~/cgi-pub/ *directory –*

```
import urllib

webConnection= urllib.urlopen("http://www.google.com/")

# it supports all of the read methods for files:
lines = webConnection.readlines()
print lines

# don't forget to close the connection:
webConnection.close()
```

Reconnecting to where we left at Lecture 5...

Connecting to the Web from Python

Let's write it out as a file that we can then open in our browser:

```
import urllib
webConnection = urllib.urlopen("http://www.google.com/")

# it supports all of the read methods for files:
lines = webConnection.readlines()

f = open("page.html", "w")
for line in lines:
    line = line.decode("utf-8")  # renders the lines in a compatible encoding
    f.write(line)

f.close()
print "All done. Open page.html in your browser."
webConnection.close()      # don't forget: close the connection
```

Reconnecting to where we left at Lecture 5...

File from Web (Group Work)

Write a Python function called *getContent* which takes one argument, *url* and outputs the *content* of the page at *url* into a *file with the same name*. Ask the user to input an URL. Save the page to a file named "index.html" if the URL does not have a basename (e.g. if the *basename* is "" or "/"). Save the output file in the same directory as the *.py* program you're running.

Hint: to obtain a valid *name* for the file to write out, we need to extract the URL's *basename* thus:

```
import os
filename = os.path.basename(url)
```

Hint 2:

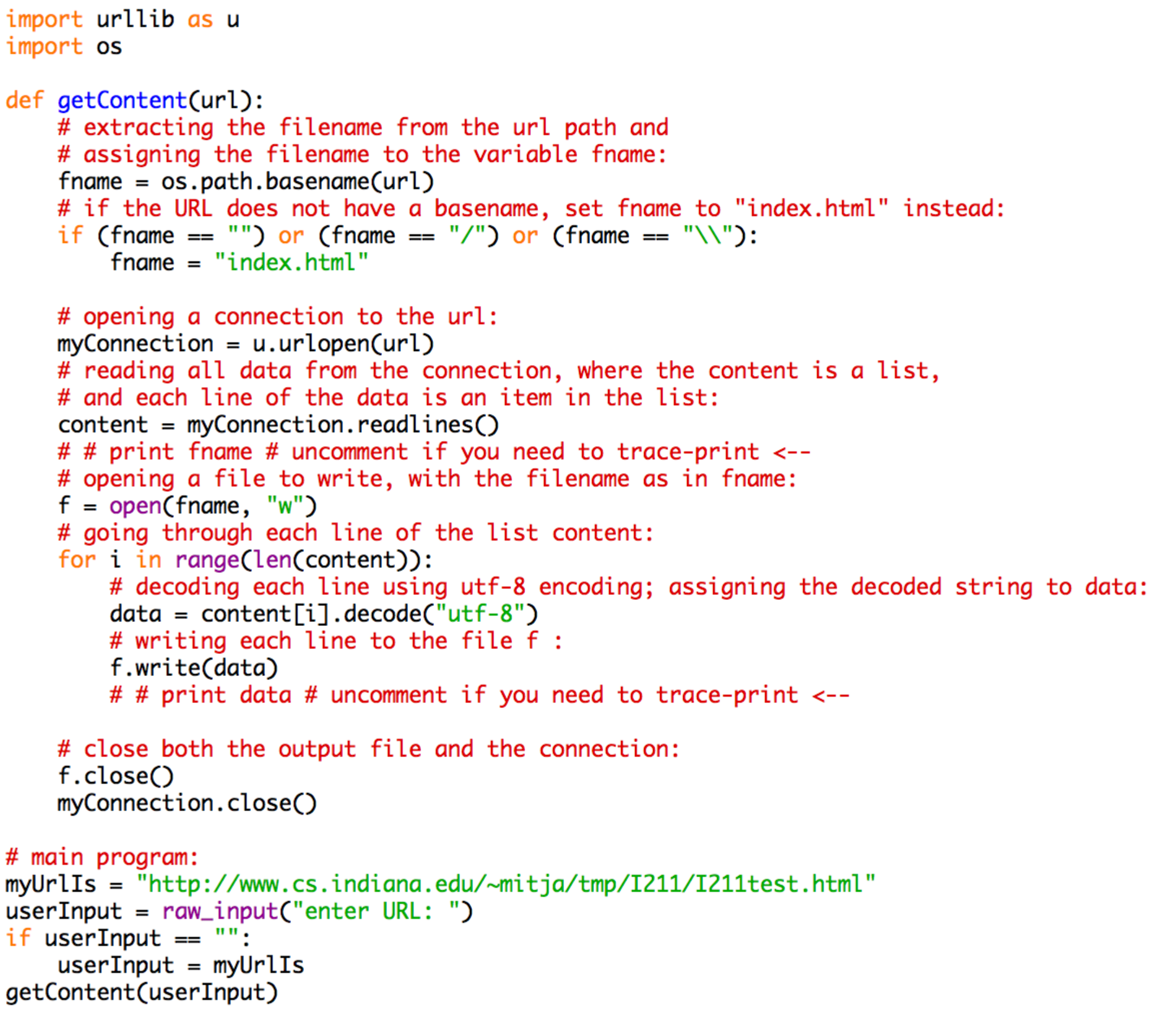
If the URL is "http://www.cs.indiana.edu/~mitja/tmp/I211/I211test.html" then the resulting base name is "I211test.html" ...

...however, if the URL is "http://www.google.com/"

then the resulting basename is empty, and the filename needs to be changed to "index.html".

Reconnecting to where we left at Lecture 5...

File from Web (Group Work) solution



```
import urllib as u
import os

def getContent(url):
    # extracting the filename from the url path and
    # assigning the filename to the variable fname:
    fname = os.path.basename(url)
    # if the URL does not have a basename, set fname to "index.html" instead:
    if (fname == "") or (fname == "/") or (fname == "\\"):
        fname = "index.html"

    # opening a connection to the url:
    myConnection = u.urlopen(url)
    # reading all data from the connection, where the content is a list,
    # and each line of the data is an item in the list:
    content = myConnection.readlines()
    # # print fname # uncomment if you need to trace-print <--
    # opening a file to write, with the filename as in fname:
    f = open(fname, "w")
    # going through each line of the list content:
    for i in range(len(content)):
        # decoding each line using utf-8 encoding; assigning the decoded string to data:
        data = content[i].decode("utf-8")
        # writing each line to the file f :
        f.write(data)
        # # print data # uncomment if you need to trace-print <--

    # close both the output file and the connection:
    f.close()
    myConnection.close()

# main program:
myUrlIs = "http://www.cs.indiana.edu/~mitja/tmp/I211/I211test.html"
userInput = raw_input("enter URL: ")
if userInput == "":
    userInput = myUrlIs
getContent(userInput)
```

Connecting to the Web from Python: why?

Yahoo! Finance provides a stock quote service. For example, this is a link to Google stock:

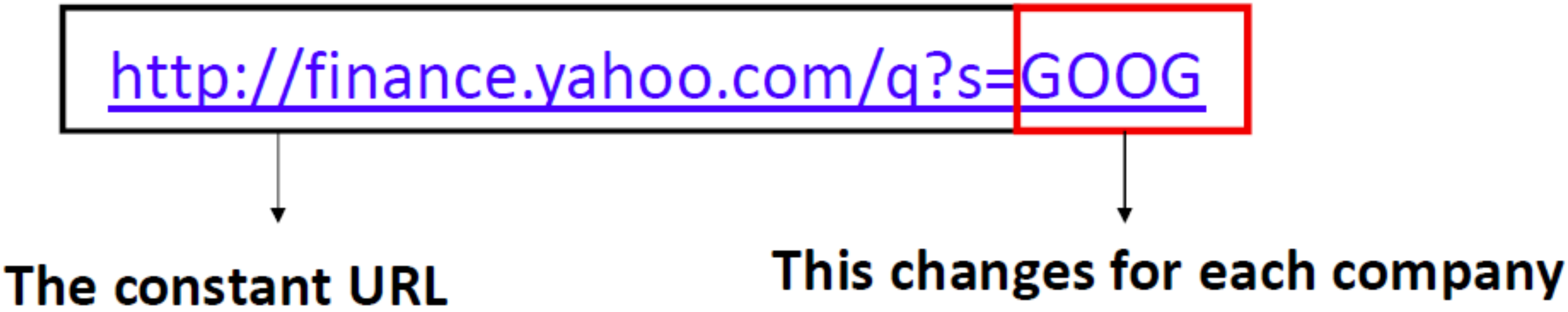
http://finance.yahoo.com/q?s=GOOG

This provides a lot of information about Google's stock, but...

- what if we want to track many different stocks?
- what if we want to download information for later analysis?
- what if we want to track the stock value automatically over a period of time?

Connecting to the Web: what?

We can use Python to read this data. The URL is:



http://finance.yahoo.com/q?s=GOOG

The constant URL

This changes for each company

Connecting to the Web from Python: how?

```
# open connections to urls, then retrieve content from web pages.

import urllib

url = "http://finance.yahoo.com/q?s="  #  <-- constant part
co = ["GOOG", "AMZN", "MSFT"]          #  <-- part that changes

for company in co:

    webConnection = urllib.urlopen (url + company)
    lines = webConnection.readlines()

    #        (here do something with the content)

    webConnection.close()
```