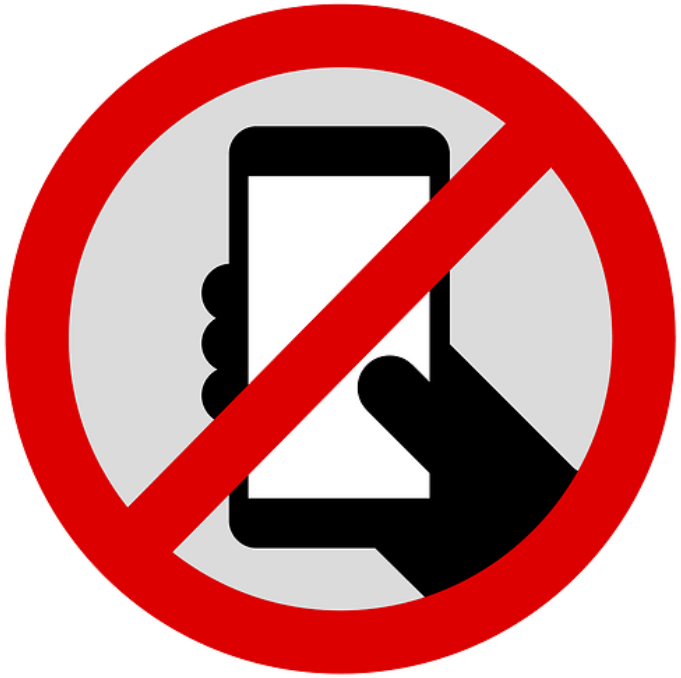
Information Infrastructure II

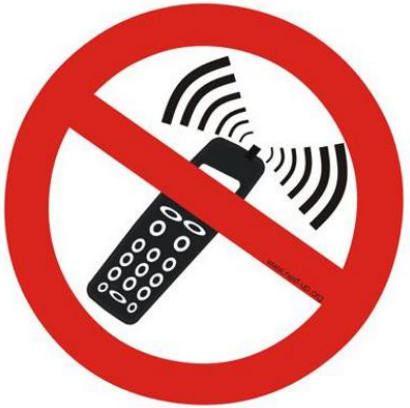
INFO I211 – Spring 2014 – Sections 18530 & 22719

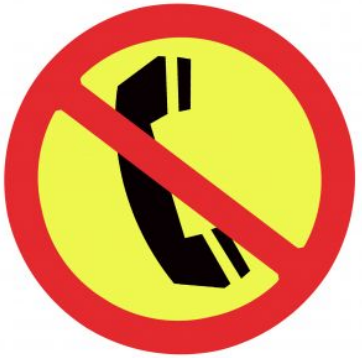
Lecture 13 – 2014.02.26 & 2014.02.27

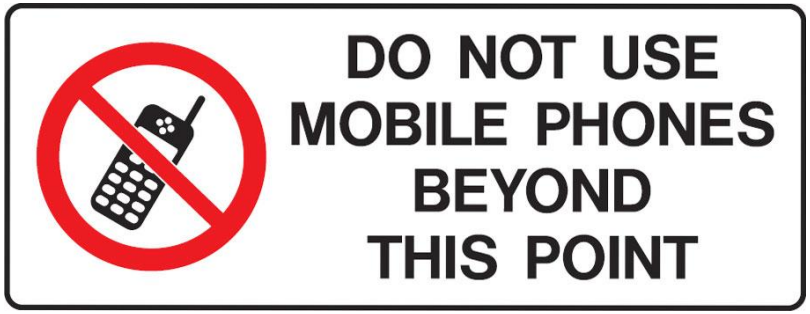
Instructor:
Mitja Hmeljak,
http://mypage.iu.edu/~mitja
mitja@indiana.edu

I211 course policy reminder:
smartphone/tablet/cellphone/chat/SMS/text... you know you don't need it









I211 course policy reminder:
Laptop and Device Use Policy at Lab & Lecture times

Unless specifically instructed, cellphones, mobile devices, tablets &c. will not be necessary.

Kindly switch them off (or put them in "airplane mode") and refrain from using them during lecture & lab time, unless specifically requested by instructors.

Unrequested use of mobile devices during lab & lecture times will result in *missing attendance for that entire lab or lecture period* (with the consequences outlined in the course syllabus).

Laptop computer use at lecture and lab times, and STC lab computer use at lab times, *is intended solely for course-related work, e.g. programming assignments etc.*

Connecting to the Web from Python: why?

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

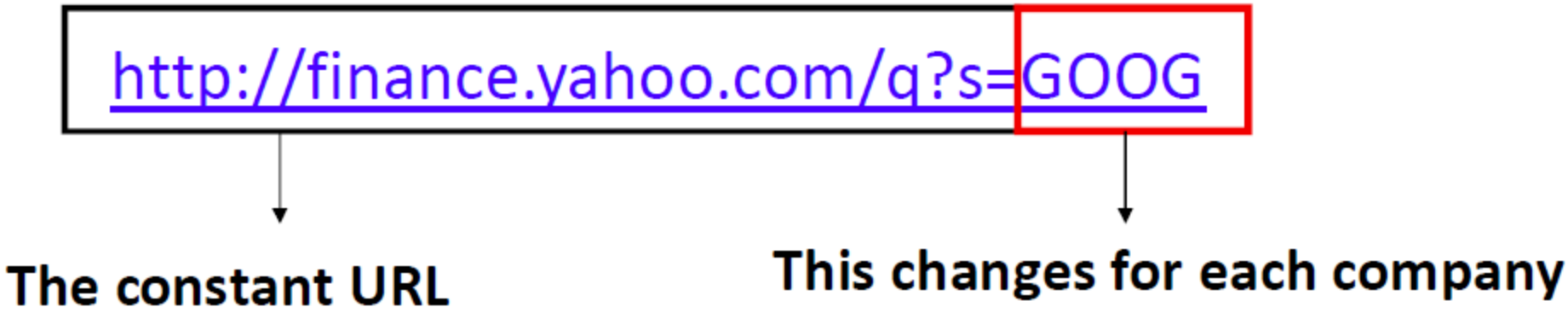
http://finance.yahoo.com/q?s=GOOG

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:



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()
```

Connecting to the Web

We want some specific things:

- Last Trade
- Change
- Date

We could extract these directly from the pages source…

- But that would be unpleasant....
  - using the View Page Source in the browser...

Connecting to the Web

Fortunately, Yahoo provides the same data in CSV format, precisely to access from programs like ours..

http://quote.yahoo.com/d/quotes.csv?s=GOOG&f=sl1d1t1c1ohgvj1pp2owern&e=.csv

(if you copy-paste this URL, beware of spurious formatting characters and strange encodings.
if you type this URL manually, beware of typos...
correctly entered URLs of this kind should produce results as below)

We get this:

"GOOG",
1023.3899,"11/5/2013","10:38am",-2.7161,1020.35,1023.74,1017.42,336875,341.9B,1026.106,"-0.26%",1020.35,"636.00 - 1041.52",36.746,27.92,"Google Inc."

Connecting to the Web

The data returned matches the webpage:

- Symbol, last trade, time, change
- Open, high, low, volume
- Market cap, previous close, previous change
- Year low, year high, EPS, PE Ratio

You can read this line of data as a string and split it...

Last Trade (Group Work)

Write code to print out the last trade for Google, and add:
co = ["GOOG", "AMZN", "MSFT", "EBAY"]

baseurl = "http://quote.yahoo.com/d/quotes.csv?
s=GOOG&f=sl1d1t1c1ohgvj1pp2owern&e=.csv "

 # open the URL
 # read content (hint: print it. What do you get? A list?)

 # decode content (hint: print it. What you get? A string?)

 # split the data into a list (hint: *stringvariable.split("separator")*)
 # from the list, get the required data (last trade value) and print it

 # Output:
 # Last Trade for GOOG is xxx.xx (Today)

Last Trades (Solution)

```
import urllib

baseurlPart1 = "http://quote.yahoo.com/d/quotes.csv?s="
baseurlPart2 = "&f=sl1d1t1c1ohgvj1pp2owern&e=.csv"
co = ["GOOG", "AMZN", "MSFT", "EBAY"]

for company in co:
    url = baseurlPart1 + company + baseurlPart2
    # print "url = ", url
    webConnection = urllib.urlopen (url)
    pageContent = webConnection.readlines()
    # print "pageContent = ", pageContent
    # get the page's first line (at index 0) and decode it:
    data = pageContent[0].decode("utf-8")
    # print "data = ", data
    # split the first line (a string) into items (a list):
    items = data.split(",")
    print "Last trade for",company, "is", items[1]
    webConnection.close()
```

urllib module

Home reading: the documentation for **urllib** — extensible library for opening URLs in Python 2.6, on:

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

Connecting to CGI

CGI – Common Gateway Interface

Standard way for a web server to

1. pass a user's request to an application and
2. receive data back to forward to the user

A CGI call:

- is requested as a URL
- takes arguments like a function

Connecting to CGI

Construct a proper URL and connect

General form is:

URLtoCGI?arg1=value1&arg2=value2&arg3=value3

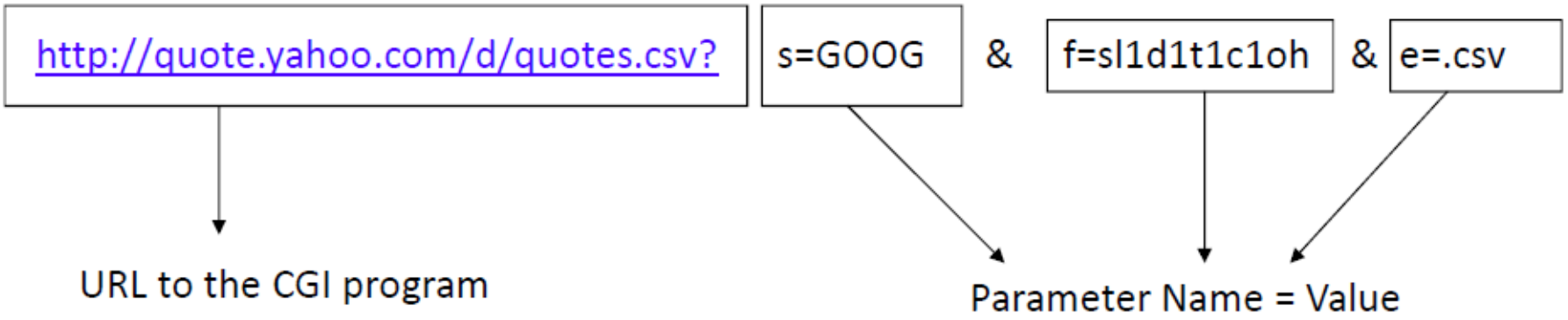
These both involve CGI...

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

http://quote.yahoo.com/d/quotes.csv?s=GOOG&f=sl1d1t1c1ohgvj1pp2owern&e=.csv

Connecting to CGI

Calling CGI is like calling a function with arguments:



By using string manipulation, we can automate access to this application.

Connecting to CGI

Restaurant locator:

http://www.bostonmarket.com/locations

http://www.bostonmarket.com/locations?page=locator&state=IN

http://www.bostonmarket.com/locations?page=locator&state=IN&city=Indianapolis

Connecting to CGI

For URLs, HTTP supports A-Z, a-z, 0-9, /, : and .

It also supports many special characters, like ~ and space

- However, these may need to be converted into special character encodings, for all http sites, browsers, etc. to understand them
- Just like \n and \t are understood to be special characters for printing
- For URLs, this is called **quoting**.

Connecting to CGI

Special characters are converted into hexadecimal codes (base 16 numbers)

A single character is represented by a single code, even if that code uses multiple numbers or letters. Ex: C8 is hex for 200 base 10.

To get the integer code for a character:

ord("A") -> 65 #ord stands for 'ordinal'

ord("~") -> 126 #these numbers are base 10

Connecting to CGI

So, to quote a URL, we:

1. Identify a special character **char = “~”**
2. Get its integer code **c = ord(char)**
3. Convert the code to hex
   1. **h = hex(c)[2:]** -> **“7e”** **#we remove the “0x”**
   2. **h = "%X" % c** **-> “7E”#second method**
4. Need to add a “%” in front of hex codes

Why quoting?

To form a standard URL that will always work

which also answers the question *'What are those "%20" codes in* URLs*?'*

http://www.blooberry.com/indexdot/html/topics/urlencoding.htm