Lecture 5

Standard Libraries
Nested Loops

Prof. Steven Ludtke
N410.07, sludtke@bcm.edu

Despair

(don't)

1) **"I'M NOT LEARNING ANYTHING, THE HOMEWORK IS TAKING FOREVER AND IS IMPOSSIBLE !"**

If you're spending (a lot of) time trying to figure out the homework, you're learning more than you think.

2) **"I'M GOING TO FAIL !"**

As long as you keep trying, and turn in your (even incomplete) attempts to figure it out, you will stay above the dreaded C.

3) Computers require precision.
Every character you type has meaning.
Separate syntax from concepts!

http://anandology.com/python-practice-book/index.html

Remember Algebra?

- $5a=15$
- $5a+5=15$
- $5a+3=12a-7$
- $5a+3=12b-7$ and $8b+6=3a+1$

Homework Review

- How do we represent the data ?
- Break the task into small pieces
- Code each of the pieces

Homework Review

- How do we represent the data?
  - balance - Balance at the end of each month
  - rate - monthly fractional interest rate
  - payment - amount of monthly payment
- Break the code into small pieces:
  - ask user for values
  - convert rate to monthly fraction (/1200)
  - if balance*rate > payment raise error
  - loop until balance <=0
    - compute interest = balance*rate
    - balance = balance + interest - payment
    - print values

Amortization

```
import sys
balance=float(raw_input("Amount of loan:"))
rate=float(raw_input("Rate as a %:"))/1200.0
payment=float(raw_input("Monthly payment:"))

if rate*balance>payment :
   print "Insufficient payment !"
   sys.exit(1)

month=1
while (balance>0):
   print month,")",balance,”+",rate*balance,"-",payment,"=",
balance+rate*\ balance-payment
   balance+=rate*balance-payment
   month+=1
```

List of Lists

```
X=[["a","b","c","d"],["e","f","g","h"],
["i","j","k","l"],["m","n","o","p"]]
```

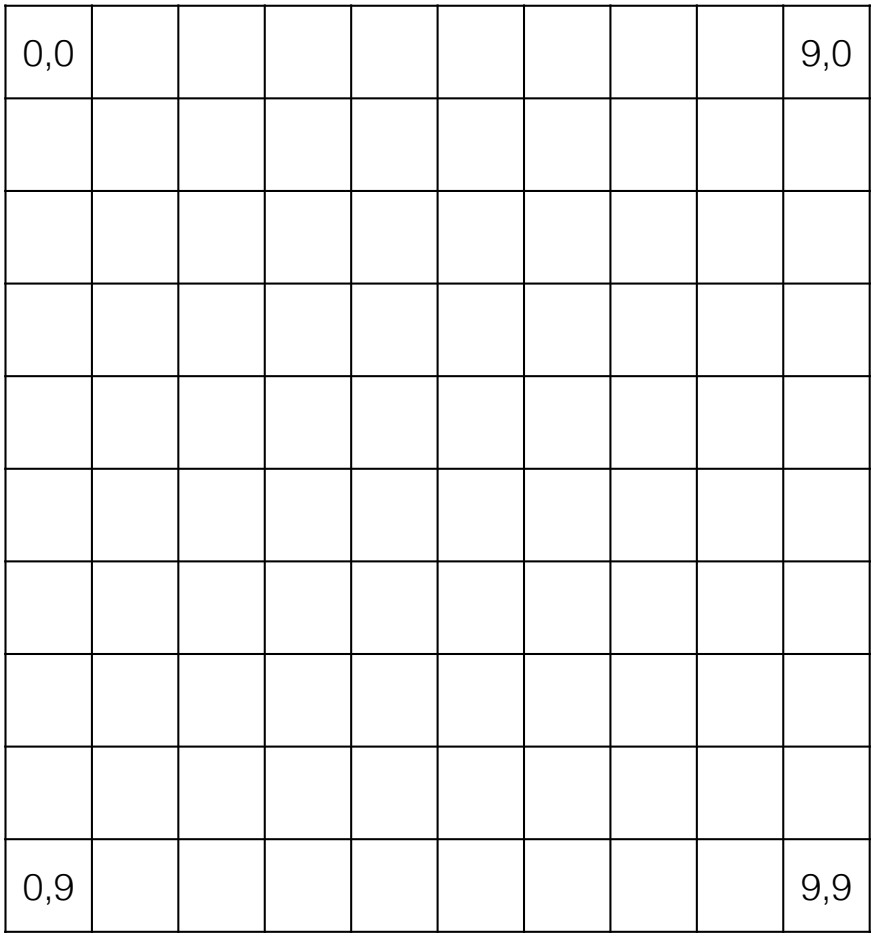
```
print X[1]
["e","f","g","h"]

print X[1][2]
g
```

	0	1	2	3
0	a	b	c	d
1	e	f	g	h
2	i	j	k	l
3	m	n	o	p

How would you iterate over 2-dimensional data?

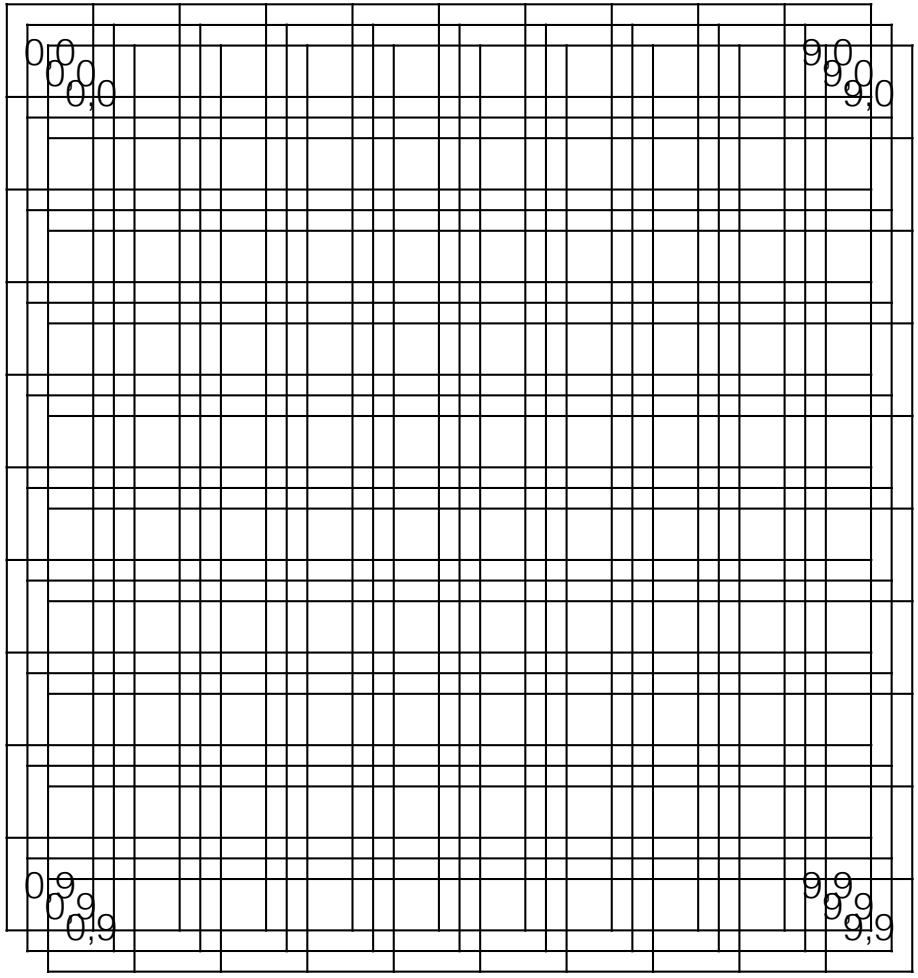
```
for a in range (100):
    y=a/10
    x=a%10
    print array[y][x]
```



How would you iterate over n-dimensional data?

```
for a in range (1000):
    z=a/100
    y=(a%100)/10
    x=(a%100)%10
    print array[z][y][x]
```

… (x10)

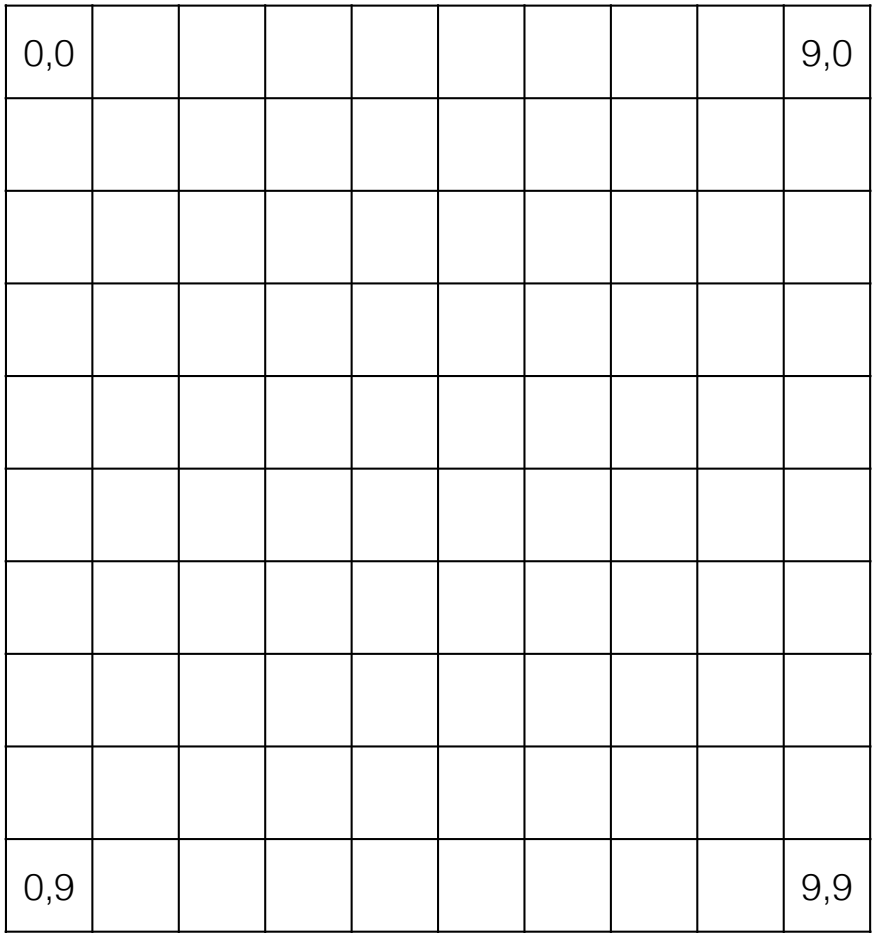


Nested Loops

(the right answer)

- a loop inside a loop

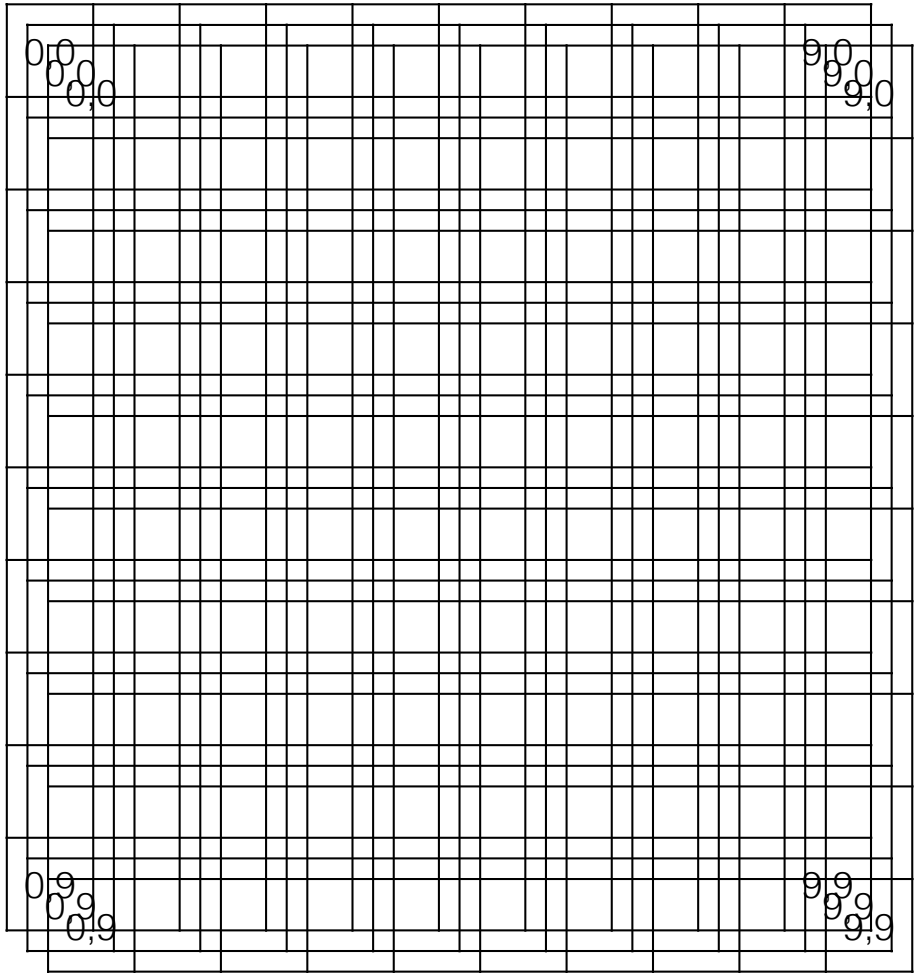
```
for y in range(10):
  for x in range(10):
    print array[y][x]
```



How would you iterate over n-dimensional data?

```
for z in range(10):
 for y in range(10):
  for x in range(10):
   print array[z][y][x]
```

… (x10)



Loop flow

- Continue/break/else - flow of a loop

```
for i in range(20):
    if i==5 : continue
    if i>17 : break
    print i
else: print "done"
```

- continue - skip the rest of the current iteration
- break - immediately exit the loop entirely
- else - only if the loop finishes without a break

import

- import module
- from module import name
- from module import *
- import module as othername

A Few Standard Libraries

- sys - System-specific parameters
- os - Operating system functions
- string - String manipulation
- time - Delays, formatting time
- datetime - Manipulate dates/times
- pprint - Pretty printing
- urllib2 - Easy web access

urllib2

Web Scraping:

```
import urllib2

f=urllib2.urlopen("http://blake.bcm.edu/dl/test.html")

for i in f: print i
```

HTML

- http://www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html
- Declarative language
- HTML is a type of XML, XHTML obeys XML rules more completely
- Python HTMLParser module
- ‘commands’ in HTML are denoted by
  <command option=value option=value>text</command>

HTML

<HTML>

<HEAD><TITLE>My Page</TITLE></HEAD>

<BODY>

<H3>Hi Everyone</H3>

<P>This is really just some test text to demonstrate how HTML works. I can do interesting things like italicize make text bold, or even both together. ta da

</BODY>

</HTML>

File Manipulation

- os.getcwd() - the current working directory (folder)
- os.chdir() - change the current working directory
- os.listdir - Lists files in a particular folder
- os.stat - info about a file
- os.rename - rename (mv) a file
- os.mkdir - create a folder
- os.remove - delete a file
- os.rmdir - remove a directory
- os.system - execute a command (mostly mac/linux)

PyPi

- http://pypi.python.org
- Note that many packages also have installers available for Windows
- easy_install
  - Comes with Mac
  - May be in a package called python_setuptools on linux
- pip
  - included as part of Python 2.7.9 and later
  - not standard on mac (2.7.6 on Mavericks)
  - Supports uninstallation, otherwise pretty comparable

Homework 3

- Install BioPython on your computer, and make sure you can import it successfully before lab on Monday.

http://biopython.org/DIST/docs/install/Installation.html

- Send me (sludtke@bcm.edu, not Amanda) what you plan to do for your class project (even if you already sent it in with the last homework) by next Friday.

1) Write a program to print a power table (y^x), with nice formatting, with both axes going from 2 to 10, and 5 significant figures if you elect to use scientific notation for some values. e.g. -

	2	3	4	5	...
2	4	8	16	32	
3	9	27	81	243	
4	16	64	256	1024	
5	25	125	625	3125	
...					

Homework 3

2) Write a program that can read 2-column text files containing numbers. Assume the 2 numbers on each line are X and Y coordinates. Compute and print the average X and Y values (the center of mass of the set of points). The numbers on each line may be separated by any whitespace (any number of spaces or tabs).

eg-

1 2

2 3.2

2.9 3.9

4.1 5.0

…