CS 4 this week

Building classes...

... vs. using the library

hw10pr2

Connect Four Board class

hw10pr3

file and dictionary classes

If I had a dictionary, I guess I could look up what it was!

Office hours ~ Thursday aft in LAC

Hw #10 due 4/15

CS 60?

2+ languages

Racket

Java

Runtime!

O( N )
O( N^2 )
O( 2^N )

Who is whoami?

How efficient is whoami?

CS 35?

8-10 libraries

8-10 libraries

Classes: DIY data

Class: a user-defined datatype

Object: data or a variable whose type is a class

OOP!

object-oriented programming
**Why classes?**

Python has no Connect-four datatype...

---

**Data design...**

(Data Members) What data do we need?

(Methods) What capabilities do we want?

---

**Our Board object, b**

(Data Members) What data do we need?

(Methods) What capabilities do we want?

---

... and now we can fix that!

---

How could we set ? to 'X' and ? to 'O'
class Board:
    """ a datatype representing a C4 board
    with an arbitrary number of rows and cols
    """
    def __init__(self, width, height):
        """ the constructor for objects of type Board """
        self.width = width
        self.height = height
        W = self.width
        H = self.height
        self.data = [ [' ']*W for row in range(H) ]

def __repr__(self):
    """ this method returns a string representation
    for an object of type Board """
    H = self.height
    W = self.width
    s = ''
    for r in range( H ):
        s += '|'  # what kind of loop will add the col #'s here?
        for c in range( W ):
            s += self.data[r][c] + '|'  
        s += 'n'
    s += (2*W+1)*'-'
    return s

Quiz

class Board:
    a C4 board
    def addMove(self, col, ox):
        """ buggy version! """
        H = self.height
        for row in range(0,H):
            if self.data[row][col] != ' ':
                self.data[row-1][col] = ox

(1) Run b2.addMove(3, 'O')

(2) Bugs! Can you fix them?!

Name(s) __________________________

Let's understand this allowsMove method...

If col is out-of-bounds or full, return False.
If it's in-bounds and not full, return True.
```python
class Board:
    def allowsMove(self, col):
        """ True if col is in-bounds + open False otherwise """
        H = self.height
        W = self.width
        D = self.data
        return False

b3.allowsMove(0) == True
b3.allowsMove(1) == True
b3.allowsMove(2) == False
b3.allowsMove(3) == False
b3.allowsMove(4) == True
b3.allowsMove(5) == True
b3.allowsMove(6) == False
b3.allowsMove(7) == False
```

```python
def winsFor(self, ox):
    """ does ox win? ""
    H = self.height
    W = self.width
    D = self.data
    for row in range(H):
        for col in range(W):
            if self.data[row][col] != ox:
                continue
            if row < H - 3:
                if all(self.data[row+i][col] == ox for i in range(4)):
                    return True
            if col < W - 3:
                if all(self.data[row][col+i] == ox for i in range(4)):
                    return True
            if row < H - 3 and col < W - 3:
                if all(self.data[row+i][col+i] == ox for i in range(4)):
                    return True
                if all(self.data[row+i][col-i] == ox for i in range(4)):
                    return True
    return False
```

```python
if b.winsFor( 'X' ) == True:
    rem = self.diff( d2 ) % 7
```

**Why objects and classes?**

**Elegance: Objects hide complexity!**

```text
if b.winsFor( 'X' ) == True:
    rem = self.diff( d2 ) % 7
```
CS 4 this week

Building classes...

hw10pr2

... vs. using the library

hw10pr3

files and the dictionary class

Connect Four Board class

Office hours ~ FRIDAY aft. in LAC

Algorithmic Authorship... ?

I like poptarts and 42 and spam.
Will I get spam and poptarts for the holidays? I like spam poptarts!

suppose this text represents my "style" ...

Algorithmic Authorship... !

I like poptarts and 42 and spam.
Will I get spam and poptarts for the holidays? I like spam poptarts!

suppose this text represents my "style" ...

Algorithmic authoring examples...

'Cause somethin' like he left knee and a harp," said he had to the whole school!
The shouting and then some strange and Mrs. "Well, I know Hagrid; they spotted handkerchief and get him get rid of course, had a gigantic beet with her," he knew what to all he's

Wanna live while we're cool, so tonight What a feeling to be doing what I wish I know we only met but it ain't hard to be nothing left The story of my life I'm watching her eyes smile you flip your eyes You don't know what makes you got stars, they're in the wire She said, "Can I got a feeling to be a dentist

Who's the original human author of each of these?

This is but ourselves. No, faith, My uncle! O royal bed of confession Of your rue for leave to nature; to this time I should weep for thy life is rotten before he is. have sworn 't. Or my blood. I have closely sent for nine; and unprofitable,

The Senators and the date of a written declaration that Purpose, they shall consist of nine States, shall not, when he shall have such Vacancies. The President pro tempore, in the Desire of a Qualification to the Speaker of the Senate. Article 6. When vacancies by the office upon probable
**Markov Models**

Techniques for modeling *any* sequence of natural data

*1st-order* Markov Model
(Defining property)

Each item depends *only* on the one immediately before it.

Lists are *sequential* containers:

\[
L = [ 47, 5, 47, 42 ]
\]

Dictionaries are *arbitrary* containers:

We need a new data structure!

(A new class...)

Dictionaries are *in*!

Dictionaries are *arbitrary* containers:

\[
zd = \{ \text{'rabbit':1999, 'ox':1997} \}
\]

Elements (or *values*) are looked up by a *key* starting anywhere you want! *Keys* don't have to be ints!

What's *zd*'s data here?

Now I see the *key* to dictionaries' *value*...

Elements (or *values*) are looked up by their *location*, or *index*, starting from 0.

List elements are looked up by their *location*, or *index*, starting from 0.

Dictionaries are *arbitrary* containers:

\[
\]

*zd*'s data here?

Is *dragon* a key in *z*?

Is 1969 in *z*['dragon']?

Is 'dragon' in *z*?
```
LoW = [ 'spam', 'spam', 'poptarts', 'spam' ]

Oldenborg's menu!

vc_print(LoW)
vc_print("a.txt")
```

```python
d = {}
for w in LoW:
    if w not in d:
        d[w] = 1
    else:
        d[w] += 1

final d

files...

```python
f = open( 'a.txt' )
open the file and calls it f

text = f.read()
reads the whole file into the string text

f.close()
closes the file (optional)

text
'I like poptarts and 42 and spam.\nWill I

LoW = text.split()
[ 'I', 'like', 'poptarts', ... ]
text.split() returns a list of each "word"

```python
def word_count( filename ):

```python
f = open( filename )
file handling

text = f.read()
f.close()

LoW = text.split()

print("There are",len(LoW),"words")
```

What if we wanted the number of different words in the file?

This would be the author's vocabulary count, instead of the total word count.

Files...

```
f = open( 'a.txt' )
opens the file and calls it f

text = f.read()
reads the whole file into the string text

f.close()
closes the file (optional)

text
'I like poptarts and 42 and spam.\nWill I

LoW = text.split()
[ 'I', 'like', 'poptarts', ... ]
text.split() returns a list of each "word"

```python
```
def word_count( filename ):

```python
f = open( filename )
file handling

text = f.read()
f.close()

LoW = text.split()

print("There are",len(LoW),"words")
```

Vocabulary, anyone?

Shakespeare used 31,534 different words -- and a
grand total of 884,647 words, counting repetitions
across all of his works....

http://www.math.cudenver.edu/~wbriggs/qr/shakespeare.html

Shakespearean coinages

gust
besmirk
unreal
superscript
watchdog
swagger
affined
rooky
attasked
out-villained
successful
unsuccessful

Who? with what word?

There's also one contemporary British
author in the Oxford English Dictionary...

http://www.pathguy.com/shakeswo.htm
http://www.shakespeare-online.com/biography/wordsinvented.html
from filename import defaultdict

def vocab_count( filename ):
    f = open( filename )
    text = f.read()
    f.close()

    LoW = text.split()
    print "There are", len(LoW), "words."
    d = {}  
    Tracking the number of occurrences of each word with a dictionary, d.
    for w in LoW:
        if w not in d:
            d[w] = 1
        else:
            d[w] += 1

    print "There are", len(d), "distinct words."
    return d  
    # return d for later use by other code...

Markov Models can be generative!

A key benefit of Markov Models is that they can generate feasible data!

Original file:

I like poptarts and 42 and spam. Will I get spam and poptarts for the holidays? I like spam poptarts!

d = create_model('hpwhich.txt')
d = create_model('randj.txt')
d = create_model('oneD.txt')
d = create_model('a.txt')
gt(d,250)

d = create_model('hpwhich.txt')
d = create_model('randj.txt')
d = create_model('oneD.txt')
d = create_model('a.txt')
gt(d,250)

d = create_model('hpwhich.txt')
d = create_model('randj.txt')
d = create_model('oneD.txt')
d = create_model('a.txt')
gt(d,250)

Our Markov Model

Try it!

Markov Model

A dictionary!

{  
  '$' : ['I', 'Will', 'I'],  
  'I' : ['like', 'get', 'like'],  
  'like' : ['spam.'],  
  'poptarts' : ['and', 'for'],  
  'and' : ['42', 'spam.', 'poptarts'],  
  '42' : ['and'],  
  'Will' : ['I'],  
  'the' : ['and', 'poptarts!'],  
  'get' : ['spam'],  
  'for' : ['the']  
}

d = create_model('hpwhich.txt')
d = create_model('randj.txt')
d = create_model('oneD.txt')
d = create_model('a.txt')
gt(d,250)

Markov-modeling's algorithm

LoW  ['I','like','spam.','I','eat','poptarts!']

pw

nw

d = {}

pw = '

for nw in LoW:
    if pw not in d:
        d[pw] = [nw]
    else:
        d[pw] += [nw]

pw = ________

d = create_model('hpwhich.txt')
d = create_model('randj.txt')
d = create_model('oneD.txt')
d = create_model('a.txt')
gt(d,250)

$d$'s final form (without quotes)

$: [I, I]$  
$I$ : ['like', 'eat']  
like : ['spam.']  
$eat$ : ['poptarts!']
Model creation:
1) start with previous word, pw as '$'
2) for each next word, nw, in the list of words, add it in ...
3) then change pw to nw ...
   (a) except if nw[-1] was punctuation: change pw to...

Generating text:
1) start with pw as the '$' string
2) choose a nw that follows pw, at random.
3) print nw, (the comma continues on the same line)
4) pw gets set to either nw or '$'
   or if nw[-1] was punctuation: change pw to...

Have Python write your papers for you...  
Have a worry-free weekend!  
... you're still the author!
class Board:

def addMove(self, col, ox):
    """ buggy version! """
    H = self.height
    for row in range(0,H):
        if self.data[row][col] != ' ':
            self.data[row-1][col] = ox

(1) Run \texttt{b2.addMove(3,'O')} \\

(2) \textbf{Bugs!} Can you fix them?!

Name(s) ____________________________