#### **PFTRotW**

- Review loops and lists (see Uppity.py again)
  - Loop over sequence: string, file, list, "other"
  - > Process each element, sometimes selectively
  - > Toward understanding the power of lists
- How do we debug?
  - > What do we do when we have code, but it's wrong?
  - > How do we minimize head-beating-against-wall?
  - > Mental model of code execution and Pythonic vocabulary
- The power of randomness

Compsci 06/101, Spring 2011

5.1

5.3

#### List methods

- In object oriented programming methods are functions that operate on an object
  - > Inspect or change the object
  - > Sometimes return values
- List methods that inspect a list
  - Search: count and index
- List methods that mutate by adding or removing
  - > append, insert, pop, remove
- List methods that re-arrange list
  - > reverse, sort

Compsci 06/101, Spring 2011

# Anatomy of a Python list

• Create list with brackets (values optional)

```
> s1 = []
> s2 = ["a", "b", "c"]
> s3 = list("123") #from an iterable
```

- Lists are mutable and iterable
  - > Append to list, change value stored at index

```
> s2[1] = 5, s2.append(77)
> for elem in list:
```

- Use function on lists: len, min, max, sum
  - > Operator: in

Compsci 06/101, Spring 2011

5.2

# Indexing a list

- Lists, like strings, start indexing with zero
  - > Strings are immutable, lists are mutable
- For some problems, looping by index useful
  - Use range function, range creates open-ended list
  - range(0,10), range(5,20), range(10,100,5)
  - > Advice/warning: in Python 3 range doesn't create list
- For some problems index and list useful
  - > Use for x,y in enumerate(list):idiom
  - Preview: tuple

Compsci 06/101, Spring 2011

5.4

## Making choices at random

- Why is making random choices useful?
  - ▶ How does modeling work? How does simulation work?
  - > Random v Pseudo-random, what's used?
  - > Online gambling?
- Python random module/library: import random
  - Methods we'll use: random.random(),
    random.randint(a,b), random.shuffle(seq),
    random.choice(seq), random.sample(seq,k),
    random.seed(x)
- How do we use a module?

Compsci 06/101, Spring 2011

5.5

### Compsci 6/101: Random debugging?!#

- The joys and rewards of writing code to solve a problem
  - > How do we know where to begin?
  - > How do we know we're making progress?
  - > How do we know when we're done?
- Make it run, make it right, (make it fast, small)
  - ▶ If we don't have a program that runs, can't make it right!
  - > Where to begin? Do something relevant to the problem
  - > Later you'll learn more about understanding design
- Once the program is running, how to fix mistakes?

Compsci 06/101, Spring 2011

5.7

## **Niklaus Wirth (Turing Award, 1984)**

- Designed and implemented several programming languages including Pascal, Modula-2, Oberon
- Wrote the paper that popularized the idea of step-wise refinement
  - > Iterative enhancement
  - Grow a working program



• Cranky or tasteful?

Simple, elegant solutions are more effective, but they are harder to find than complex ones, and they require more time which we too often believe to be unaffordable

Compsci 06/101, Spring 2011

5.6

# **Bug and Debug**

- software 'bug'
- Start small
  - **Easier** to cope
- Judicious 'print'
  - Debugger too



- Verify the approach being taken, test small, test frequently
  - ➤ How do you 'prove' your code works?

Compsci 06/101, Spring 2011

5.8