

CS1 Lecture 19

Mar. 1, 2017

- If you didn't get Exam 1 back, get it in disc. section today or from me after that.
- HW 5 available this afternoon, due Mar. 10

Today

- more on Ch11 – dictionaries *very important!*

(last time) Chapter 11: Dictionaries

- Dictionaries are:
 - collections of key – value pairs
- Similar to but importantly different from lists
 - think of lists as *ordered* collection of key-value pairs, where the keys are integers 0, 1, 2, ...
 - with dictionaries, the collection is *unordered* but the cool thing is that the *keys can be any immutable values*
 - *E.g.* create dictionary numlegs
 - `>>> numlegs = { 'frog': 4, 'human': 2, 'ant':6, 'dog':4}`
- one important feature of dictionaries is that they provide *very fast* access to values associated with keys despite being more flexible than lists

(last time) Dictionaries

- create: { k1:v1, k2:v2, ...}
- empty dictionary: {}
- retrieve value: dict[key]
- modify (or insert) value for key: dict[key]=value
- len(d)
- d.keys()
- d.values()
- k in d
- del d[k]
- for key in dict:
- d.get(key, defaultVal) when you don't want possible KeyError for d[key]

Dictionaries

Consider the wordNeighborsInfo program of HW3

The job of generateAndSaveAllNeighborLists was to return a list of two-element lists:

```
# Thus for list with words: rat, dog, cat, ran, sat, moon, soon
# this function returns:
# [['rat', ['cat', 'ran', 'sat']],
#  ['dog', []],
#  ['cat', ['rat', 'sat']],
#  ['ran', ['rat']],
#  ['sat', ['rat', 'cat']],
#  ['moon', ['soon']],
#  ['soon', ['moon']]]
```

But this is better done as a dictionary! {'rat': ['cat', 'ran', 'sat'], ...}

Another dictionary example

- Text file with info about people – name, birth year, favorite color, weight, home city, home country
 - Read and store in dictionary
 - Name as key
 - Subdictionary (and sub-sub-dictionary) for other properties

```
{'birthyear': 1980,  
  'favcolor': 'red',  
  ...,  
  'home': {'city': 'Tokyo', 'country': 'Japan'}}
```
 - Add simple password handling, storing “hash” in dict
- Files: [ppldata.py](#), [people.text](#)*

Related news

- Last year's Turing Award winners:
<https://www.theguardian.com/science/2016/mar/01/turing-award-whitfield-diffie-martin-hellman-online-commerce>
- <http://amturing.acm.org/byyear.cfm>
- Remember primeDivisorsOf problem? Important for cryptography. Internet security depends hugely on the fact that there is no known way to find factors of very large numbers quickly

A few little exercises

- Given a list of numbers, find the pair with greatest difference
- Given a list of numbers find the pair with smallest difference
- Given a list of numbers and a target number (call it k), find two numbers (if they exist) in the list that sum to k

`lec19exercises.py` has solutions for first two. Has slow (and not completely correct) solution for third one. Can you think of a much faster solution using dictionaries?

Next Time

- Third exercise from previous slide
- Global variables, tuples (Ch 12), variable length and keyword function parameters
- HW5