From 10,000 ft to 1 km: Dictionaries

- What is a dictionary? By example
  - 152.3.140.1 is www.cs.duke.edu
  - 157.166.224.26 is cnn.com
  - 68.71.209.235 is espn.go.com
- A collection of (key,value) pairs
  - Look up a key, get an associated value
  - Update the value associated with a key
  - Insert a (key,value) pair
  - Loop over the keys, access pairs or value

A Python view of dictionaries

- A collection of (key,value) pairs that is similar syntactically to a list
  - A list can be accessed by index: a[3]
  - A dictionary can be accessed by key: d["cat"]
- The key in a dictionary must be immutable
  - Essentially because key converted to number and number used as index (to find value)
- Finding the value associated with a key is very fast
  - Essentially doesn't depend on # keys!

Python syntax for dictionaries

- Create a dictionary:
  - `d = {}`
  - `d = {"apple":3, "guava":37}`
  - `d = dict([("owen":62.5)("bob":73.9)]`
- Internal dictionaries in Python
  - Sometimes useful in meta-programming
  - `globals():`
  - `locals():`

Hangman internals: dictionaries/lists

- In the WordLoader.py file several idioms in use:
  - Global variables: maintain value over many function calls, e.g., from client code calling WordLoader methods
  - Cached values: avoid calculating the same thing more than once, store and retrieve the second time
  - Default parameters: so client code can override, but doesn't have to
- Which of these will you use in writing Hangman.py