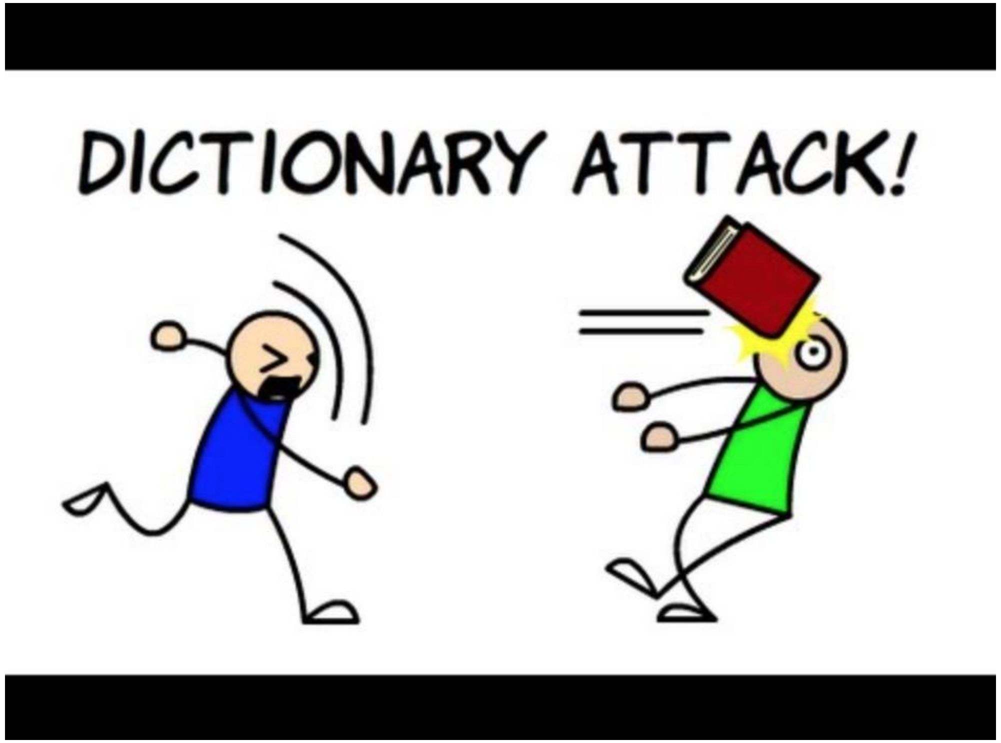
CSc 110, Autumn 2017

Lecture 31: Dictionaries

Adapted from slides by Marty Stepp and Stuart Reges



Exercise

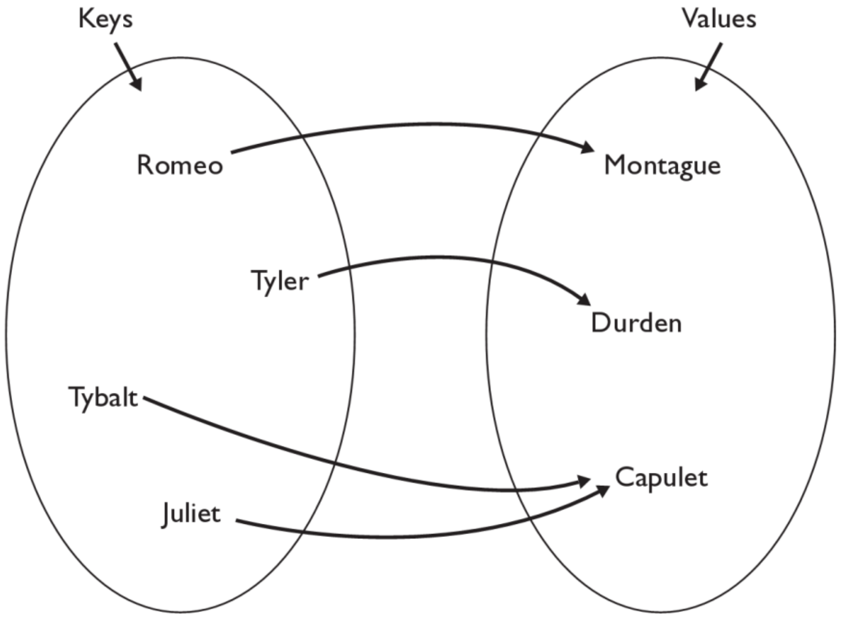
- Write a program to count the number of occurrences of each unique word in a large text file (e.g. *Moby Dick*).
  - Allow the user to type a word and report how many times that word appeared in the book.
  - Report all words that appeared in the book at least 500 times.

- What structure is appropriate for this problem?

Dictionaries

- **dictionary**: Holds a set of unique *keys* and a collection of *values*, where each key is associated with one value.
  - a.k.a. "map", "associative array", "hash"

- basic dictionary operations:
  - Add a mapping from a key to a value.
  - Retrieve a value mapped to a key.
  - Remove a given key and its mapped value.



Dictionary operations

`items()`	return a new view of the dictionary’s items ((key, value) pairs)
`pop(`**key**`)`	removes any existing mapping for the given key and returns it (error if key not found)
`popitem()`	removes and returns an arbitrary (key, value) pair (error if empty)
`keys()`	returns the dictionary's keys
`values()`	returns the dictionary's values

You can also use `in`, `len()`, etc.

Looping over a set or dictionary?

- You must use a `for element in` structure loop
  - needed because sets have no indexes; can't `get` element `i`

Example:

```
for item in a:
    print(item)
```

Outputs:

```
the
happy
hello
```

items, keys and values

- items function returns tuples of each key-value pair
  - can loop over the keys in a for loop

```
ages = {}
ages["Merlin"] = 4
ages["Chester"] = 2
ages["Purrcival"] = 12
for cat, age in ages.items():
    print(cat + " -> " + str(age))
```

- values function returns all values in the dictionary
  - no easy way to get from a value to its associated key(s)
- keys function returns all keys in the dictionary

Exercise

- Use word counts to figure out if a document is positive or negative
  - Count all of the positive words and count all of the negative words.
  - Whichever count is bigger is the sentiment of the document.
- How do we know which words are positive and which are negative?

Exercise

Consider the following function:

```
    def mystery(list1, list2):
        result = {}
        for i in range(0, len(list1)):
            result[list1[i]] = list2[i]
            result[list2[i]] = list1[i]
        return result
```

What is returned after calls with the following parameters?

```
list1: [b, l, u, e]          list2:[s, p, o, t]
```

dictionary returned:____

```
list1:[k, e, e, p]          list2: [s, a, f, e]
```

dictionary returned:____

```
list1:[s, o, b, e, r]     list2: [b, o, o, k, s]
```

dictionary returned:____