Walk through previous lectures

Recall: reading files

```
fobject = open("filename", "w")    # write
fobject = open("filename", "a")    # append
```

- opens file for write (deletes any previous contents), or
- opens file for append (new data is placed after previous data)

```
fobject.close()  # close an open file object
```

Line based file processing

- f.readline()

Returns the next line in the file or a blank string if
There are no more lines

- f.readlines()

Returns a list of lines in the file.

One line at a time

```
fo = open("filename")
for line in fo:
    # process line
```

Example: Process CSV files output from Excel

Tuple

```
tuple_name = (value, value, ..., value)
```

- A way of packing multiple values into a variable

```
>>> x = 3
>>> y = -5
>>> p = (x, y, 42)
>>> p
(3, -5, 42)
```

```
name, name, ..., name = tuple_name
```

- Unpacking a tuple‘s contents in to multiple variables

```
>>> a, b, c = p
>>> a
3
>>> b
-5
>>> c
42
```

Using Tuples

- Useful for storing multi-dimensional data (eg- (x,y) points)

```
>>> p = (42, 39)
```

- Useful for returning more than one value

```
>>> def slope ((x1,y1), (x2, y2)):
...     return (y2 – y1) /(x2 – x1)
... p1 = (2, 5)
... p2 = (4, 11)
... slope(p1, p2)
3
```

Dictionaries

- Hash tables, "associative arrays"

```
d = {"duck": "eend", "water": "water"}
```

- Lookup:

```
d["duck"] -> "eend"
d["back"] # raises KeyError exception
```

- Delete, insert, overwrite:

```
del d["water"] # {"duck": "eend", "back": "rug"}
d["back"] = "rug" # {"duck": "eend", "back": "rug"}
d["duck"] = "duik" # {"duck": "duik", "back": "rug"}
```

Dictionaries

- Keys, values, items:
```
d.keys() -> ["duck", "back"]
d.values() -> ["duik", "rug"]
d.items() -> [("duck","duik"), ("back","rug")]
```
- Presence check:
```
d.has_key("duck") -> 1; d.has_key("spam") -> 0
```
- Values of any type; keys almost any
```
{"name":"Guido", "age":43, ("hello","world"):1,
  42:"yes", "flag": ["red","white","blue"]}
```

Dictionaries

- Keys must be **immutable**:
  - numbers, strings, tuples of immutables
    - these cannot be changed after creation
  - reason is *hashing* (fast lookup technique)
  - **not** lists or other dictionaries
    - these types of objects can be changed "in place"
  - no restrictions on values

- Keys will be listed in **arbitrary order**
  - again, because of hashing

to be continued...