

## Outline for October 26, 2023

**Reading:** §7

**Assignments:** Homework 2, due October 26, 2023

---

1. Files
  - (a) What is a file?
  - (b) What can you do with it? (For example, read, write, append)
  - (c) Types of files (text, binary)
2. File Input and Output for text files
  - (a) Opening and closing: `open(filename, mode), close()`
  - (b) Reading: `readline(), readlines(), read(), read(n)`
  - (c) Writing: `write(str), writelines(list)`
3. Exception `EOFError` — input function encounters an end of file
4. Examples
  - (a) Print out a named file [`fileio1.py`]
  - (b) Print out a named file and prepend line numbers [`fileio2.py`]
  - (c) Store the output in `filename.lst` [`fileio3.py`]
5. Examples
  - (a) Put lines in a file in random order [`randlines.py`]
  - (b) Read in a list of words from a file, then search it as requested; similar to linear search program [`search-1.py`]
  - (c) Now see how many words you checked total [`search-1c.py`]
6. Dictionary
  - (a) Collection of key-value pairs
7. Creating dictionaries
  - (a) Using `d = {}`
  - (b) Using `d = dict()`
8. Methods for dictionaries
  - (a) `k in D`: True if dictionary `D` has key `k`; else False
  - (b) `D.keys()`: list of keys in `D`
  - (c) `D.values()`: list of values in `D`
  - (d) `D.items()`: list of tuples (key, value) in `D`
  - (e) `D.get(k, d)`: if key `k` in `D`, return associated value; else return `d`
  - (f) `del D[k]`: delete tuple with key `k` from `D`
  - (g) `D.clear()`: delete all entries in `D`
9. Example: memos
  - (a) Remember how slowly the recursive Fibonacci number program `rfib.py` ran? Here is a faster recursive version that uses memos [`rfibmemo.py`]