Objectives

- Reading from files
- Writing to files

Midway Check: Parts of an Algorithm

- Primitive operations
  - What data you have, what you can do to the data
- Naming
  - Identify things we're using
- Sequence of operations
- Conditionals
  - Handle special cases
- Repetition/Loops
- Subroutines
  - Call, reuse similar techniques

- Which of these have we covered?
- How do we implement them in Python?

Midway Check: Parts of an Algorithm

- Primitive operations
  - What **data** you have, what you **can do** to the data
- Naming
  - Identify things we're using
- Sequence of operations
- Conditionals
  - Handle special cases
- Repetition/Loops
- Subroutines
  - Call, reuse similar techniques

where most of the rest of the semester focuses

Working toward no longer *primitive*

Review

- What is the major difference between strings and lists?
  - What are the implications of that difference?

Review: Lists vs. Strings

- Strings are **immutable**
  - Can't be mutated?
  - Err, can't be modified/ changed
- Lists are **mutable**
  - Can be changed
  - Changes how we call/use methods

Implications:
- Think of `list` variables as ***pointing*** to the list
- Assigning a list to another variable does ***not* make a copy** of the list
- list methods ***modify*** the list on which the method was called
  → Don't return a copy of the object, modified
- When you pass a list into a function, you ***can modify*** the list

Review

- Why should we care about files?

Files

- Conceptually, a file is a **sequence** of data stored in memory
- To use a file in a Python script, create an object of type `file`
  - `file` is a *data type*

Built-in function
"constructs" a `file` object

  - `<varname> = open(<filename>,<mode>)`
    - `<filename>`: `string`
    - `<mode>`: `string`, "r" for read, "w" for write, "a" for append (and others)
  - `Ex: dataFile = open( "years.dat", "r" )`

Common File Methods

Method Name	Functionality
`read()`	Read all the content from the file, returned as a string object
`readline()`	Read next line from file, returned as a string object (which includes the "\n"). If it returns "", then you've reached the end of the file
`write(string)`	Write a string to the file
`close()`	Close the file. Must close the file after done reading from/writing to a file

Reading from a File

- Examples of reading from a file using file methods
  - Show file: `data/famous_pairs.txt`

Typically use `.dat` or `.txt` file extension to name files containing data or text

- `file_read.py` (using `read`())
  - How is what Python printed different than the file's content?
  - How to fix?
- Using `readline`()

Reading from a File

- Recall that a file is a ***sequence*** of data
- Can use a `for` loop to iterate through a file

A *line* (of type `str`) from the file (includes \n)

`file` object

```
for line in dataFile:
    print(line)
```

  - Read as: for each line in the file, do something

`for_file_read.py`

Data Types of Loop Variables

What are the data types of the loop variable **x**?

```
myString = "some string"
dataFile = open("datafile.dat", "r")

for x in range(len(myString)):
     # loop body …

for x in myString:
     # loop body …

for x in dataFile:
     # loop body …
```


Data Types of Loop Variables

What are the data types of the loop variable **x**?

```
myString = "some string"
dataFile = open("datafile.dat", "r")

for x in range(len(myString)):
     # loop body …

for x in myString:
     # loop body …

for x in dataFile:
     # loop body …
```

integer

string → single characters

string → line (include \n)

Problem: Searching a File

- We want to search a file for some term. We want to know *which lines* of the file contain that term and a *count* of the number of lines that contained that term

Writing to a File

- Create a file object in **write** mode:
  - `myFile = open("myfile.txt", "w")`
- Example: create a file from user input
  - `file_write.py`

What happens if you execute the program again with different user input?

Handling Numeric Data

- We have been dealing with reading and writing *strings* so far
  - Read from a file: get a string
  - Write to file: use a string
- What do we need to do to **read numbers** from a file?
- How can we **write numbers** to a file?

Handling Numeric Data

- We have been dealing with reading and writing *strings* so far
  - Read from a file: get a string
  - Write to file: use a string
- What do we need to do to **read numbers** from a file?
  - Cast as a numeric type, e.g., `int` or `float`
- How can we **write numbers** to a file?
  - Cast number as a `str`

Broader Issue: Cryptography

Chase Lindsey Margaret Parker	Ben Harris Joseph Mary-Frances	Andrew Anna Chas Findley	Alison Ian Kalady Ryan	Jordan Lizzie Olivia Robert

Broader Issues Discussion

- What is cryptography?
  - Why is cryptography useful?
- Debate is often summarized as "privacy vs security"
  - What does this mean? How does it relate to the recent case (Apple vs FBI) referred to in the article?
- Why are computer scientists involved in politics?
  - Has this class informed your politics?
- Who is Alan Turing?

Duffie and Hellman

- Technique still used in protocols (like SSL today)