Objectives

- Algorithm review
- Introduction to Files

Review: Functions

```
CONSTANT = 12    Where does program start "doing stuff"?

def main():
    first = input("Enter the first number: ")
    second = input("Enter the second number: ")
    computedVal = myFunction(first, second)
    print "The answer is", computedVal

def myFunction(x, y):
    result = x*x + y*y + CONSTANT
    return result

main()
```

What variables can function "see" here? What vars can't it see?

Review: Why Functions?

- Organize code
- Easier to read
- Easier to change
- Easier to reuse

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?

Parts of an Algorithm

- Primitive operations   *where most of the rest of the semester focuses*
  - What **data** you have, what you **can do** to the data
- Naming   *No longer primitive*
  - Identify things we're using
- Sequence of operations
- Conditionals
  - Handle special cases
- Repetition/Loops
- Subroutines
  - Call, reuse similar techniques

Review

- The data type of the loop variable depends on what's after **in**

```
string = "some string"

for x in xrange(len(string)):
    # loop body …

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

What is the data type of the loop variable **x**?

1

Review

- The data type of the loop variable depends on what's after **in**

```
string = "some string"

for x in xrange(len(string)):        Integer
     # loop body …

for x in string:                     String
     # loop body …
```

Sources of Input to Program

- User input
  - Slow if need to enter a lot of data
  - Error-prone
    - User enters the wrong value!
  - What if want to run again after program gets modified?

Sources of Input to Program

- Text files
  - Enter data once into a file, save it, and reuse it
  - Good for large amounts of data
  - Programs can use files to *communicate*
 - Need to be able to *read from* and *write to* files
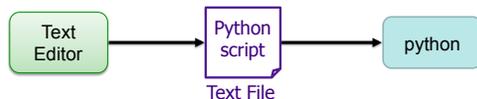
---
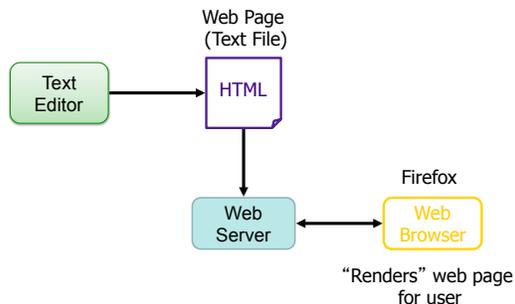
Example Use of Files

Example Use of Text File as Input
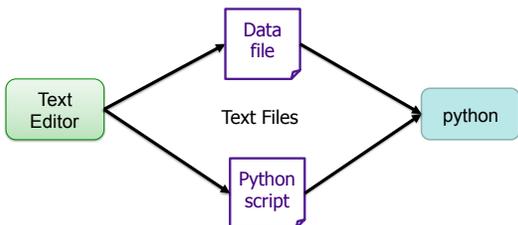
Updated Wheel of Fortune

- Uses a file of puzzles
  - Puzzles no longer appear directly in program
  - Can modify puzzle file to get different puzzles

```python
def displayPuzzle(puzzle):
    displayedPuzzle = ""
    for char in puzzle:
        if char.isalpha():
            displayedPuzzle += "_"
        else:
            displayedPuzzle += char
    return displayedPuzzle
```

2

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*

  > **constructor** - "constructs" a file object

  - `<varname> = file(<filename>,<mode>)`
    - `<filename>` : `string`
    - `<mode>`: `string`, either "r" for read or "w" for write
  - Ex: `dataFile = file( "years.dat", "r")`

Mar 7, 2011   Sprenkle - CSCI111   13

Common File Methods

Method Name	Functionality
`read()`	Read the entire content from the file, *returned* as a string object
`readline()`	Read one 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

Mar 7, 2011   Sprenkle - CSCI111   14

Reading from a File

- Examples of reading from a file using file methods

  > Show file: `data/years.dat`

  > Typically use `.dat` or `.txt` file extension for files containing data/text

- `file_read.py` (using `read()`)
  > How is what Python printed different than the file's content?
  > How to fix?
- `file_read2.py` (using `readline()`)

Mar 7, 2011   Sprenkle - CSCI111   15

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   `file` object

  ```
  for line in dataFile:
      print line
  ```

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

Mar 7, 2011   Sprenkle - CSCI111   `file_read3.py`   16

Data Types of Loop Variables

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

```
string = "some string"
dataFile = file("years.dat", "r")

for x in xrange(len(string)):
    # loop body …

for x in string:
    # loop body …

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

Mar 7, 2011   Sprenkle - CSCI111   17

Data Types of Loop Variables

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

```
string = "some string"
dataFile = file("years.dat", "r")

for x in xrange(len(string)):          integer
    # loop body …

for x in string:          string → single
    # loop body …                   characters

for x in dataFile:          string → line
    # loop body …           (include \n)
```

Mar 7, 2011   Sprenkle - CSCI111   18

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

Problem: Searching a File

- This time, we want to ignore all lines that begin with "#" (a.k.a., the line is a comment)
  - Assume comments are at the beginning of the line
  - Why would we have comments in a data file?
    - `data/years2.dat`
  - How can we revise the previous solution to do this?

Writing to a File

- Create a file object in **write** mode:
  - `myFile = file("years.txt", "w")`

- Example: create a file from user input
  - `file_write.py`

  > What happens if 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`

Problem: Temperature Data

- **Given**: data file that contains the daily high temperatures for last year at one location
  - Data file contains one temperature per line
  - Example: `data/florida.dat`
- **Problem**: What is the average high temperature (to 2 decimal places) for the location?

  Rule of Thumb: Always look at data file before processing it

This Week

- Tuesday
  - Lab – functions, files
- Wednesday
  - Files, Lists
- Friday
  - Advanced Lists
  - Broader Issue: Digital Humanities