Disk Files – Obtaining Data in Large Quantities

Getting Data

Three steps to getting data from a file:

1) Open the *filename*
   - Returns a *file handle*

2) *Read* from the file handle
   - Extract the contents as text strings

3) *Close* the file handle
   - Tidy things up when done

Saving Output To a File

Similar process to reading

1) Open a filename for *writing*
   - Again returns a *file handle*

2) *Print* or *write* to the file handle
   - Can be like printing to the console

3) *Close* the file handle when done
   - Necessary to verify that contents are saved to disk

Open a File for Reading or Writing

- Start with the file's *name*
  - Just a text string
- Open the file for *reading*, *writing*, or *appending*
  - Get back a *file handle*
- Perform all operations using the file handle

Example: Copy a File All at Once

```
"""
Created on Sun Apr 18 11:35:24 2021

@author: bobmon
"""

def main(argv=[__name__]):
    if len(argv) == 2:
        filename = argv[1]
    else:
        filename = input('File name? ')
        if len(filename) == 0:
            return

    handle1 = open(filename, 'r')
    contents = handle1.read()    # grab the whole thing at once
    handle1.close()

    handle2 = open( 'copy.txt', 'w')
    handle2.write(contents)      # write everything to a new file
    handle2.close()

if __name__ == '__main__':
    import sys
    sys.exit(main(sys.argv))
```

Reading From a File

- Three ways to do it:
  - handle**.read()** – Read the entire file contents into a single (long) string
  - handle**.readlines()** – Read the entire file into a list of separate lines, each ended by a newline '\n'
  - handle**.readline()** – Read only one line of text
    - Can be repeated to read successive lines of text, or followed by .readlines() to read the rest of the file
    - Can be used to read an initial "header" line of text

Example: Reading Input Data

- Read a keyword from the first line of a file
- Read the remaining lines
  - Count the number of occurrences of the keyword in each line
- Report the total number of occurrences, and the average number of occurrences per line

  - file: **https://montcs.bloomu.edu/Readings/zombies.key.txt**

Writing To a File

Two approaches available

- handle.**write()** – method of the file handle
  - » Writes a single text string
    - The counterpart to read()

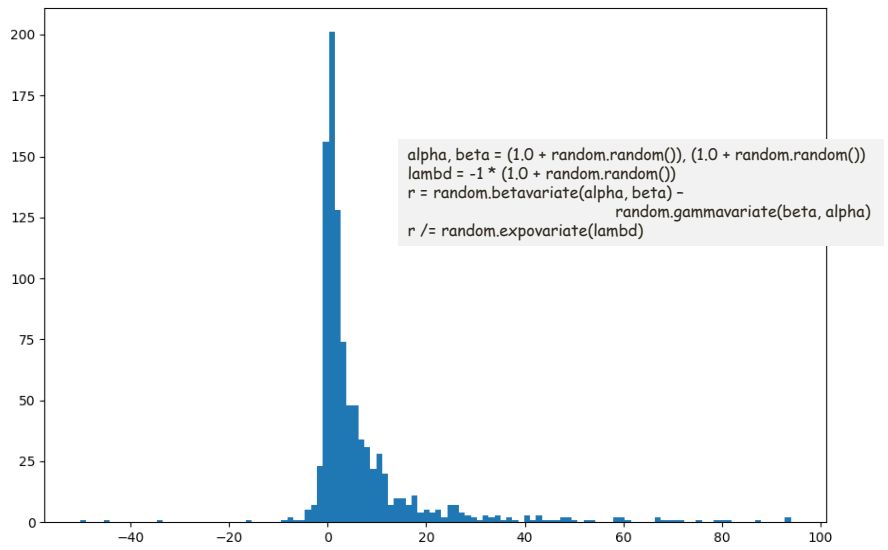
- **print()** – independent function
  - » Use as usual, but send the output to the file handle
    - Add "file=handle" argument

- write() is more efficient, print() is more versatile

Example: Write Random Numbers To a File

- Get a filename from the user
  - command-line argument, or input() function
- Get a (largish) integer *n* from the user
  - command-line argument, or input() function
- Generate *n* random values, and write them to the file in different ways
  - One value per line; or all values on same line
  - Using .write(); or using print()

A Nonuniform, Asymmetric Distribution



Controlling the Output

- Build strings a number at a time
- When string is long enough, start new line

```
import random

def main(argv=[__name__]):
    if len(argv) == 3:
        fn = argv[1]
        n = int(float(argv[2]))
    else:
        return

    linelength = 72

    lines = []
    line = ''
    for i in range(n):
        alpha, beta = (1.0 + random.random()), (1.0 + random.random())
        lambd = -1 * (1.0 + random.random())
        r = random.betavariate(alpha, beta) - random.gammavariate(beta, alpha)
        r /= random.expovariate(lambd)
        rstr = str(r)
        if len(line) + len(rstr) > linelength:
            lines.append(line)
            line = ''
        line += ' ' + rstr
    lines.append(line)
    lines.append('')

    with open(fn, 'w') as h:
        h.write('\n'.join(lines))
#--------
```

The "with" Statement

- The basic approach is:
  - open
  - use
  - close
- The "with" statement modifies the "open" step, and does the "close" implicitly
  - Example:

```
with  open("filename", "r") as handle:
    # do stuff with the file
    # "close" happens when this block is done
```

Example: Read a File, Reverse It, Write

```
infile = 'foo-in.txt'
outfile = 'foo-out.txt'

ifhandle = open(infile, 'r')
contents = ifhandle.read()
ifhandle.close()
rc = ''.join([c for c in reversed(contents)]) + ['']
ofhandle = open(outfile, 'w')
ofhandle.write(rc)
ofhandle.close()
```

...becomes...

```
with open(infile, 'r') as inhandle:
    contents = inhandle.read()
rc = ''.join([c for c in reversed(contents)]) + ['']
with open(outfile, 'w') as outhandle:
    outhandle.write(rc)
```

Example 2: Capitalize Lines of a File

```
ifhandle = open(infile, 'r')
ofhandle = open(outfile, 'w')
for line in ifhandle.readlines():
    line = line.title()
    ofhandle.write(line)
ifhandle.close()
ofhandle.close()
```

...becomes...

```
with open(infile, 'r') as inhandle:
    with open(outfile, 'w') as outhandle:
        for line in ifhandle.readlines():
            line = line.title()
            ofhandle.write(line)
```