Jin-Soo Kim
(jinsoo.kim@snu.ac.kr)

Systems Software &
Architecture Lab.

Seoul National University

Jan. 6 – 17, 2020

Python for Data Analytics

Files

File Processing

- A text file can be thought of as a sequence of lines

```
The First Book of Moses:  Called Genesis

1:1 In the beginning God created the heaven and the earth.
1:2 And the earth was without form, and void; and darkness was upon the face of the deep.
And the Spirit of God moved upon the face of the waters.
1:3 And God said, Let there be light: and there was light.
1:4 And God saw the light, that it was good: and God divided the light from the darkness.
1:5 And God called the light Day, and the darkness he called Night. And the evening and the
morning were the first day.
1:6 And God said, Let there be a firmament in the midst of the waters, and let it divide
the waters from the waters.
1:7 And God made the firmament, and divided the waters which were under the firmament from
the waters which were above the firmament: and it was so.
1:8 And God called the firmament Heaven. And the evening and the morning were the second
day.
1:9 And God said, Let the waters under the heaven be gathered together unto one place, and
let the dry land appear: and it was so.
```

Opening a File

- Before we can read the contents of the file, we must tell Python which file we are going to work with and what we will be doing with the file

- This is done with the `open()` function

- `open()` returns a "file handle" – a variable used to perform operations on the file

Using open( )

- *fh* = open(*filename*, *mode*)
  - Creates a Python file object, which serves as a link to a file residing on your machine
  - You can read or write file by calling the returned file object's methods
  - *Filename* is a string (pathname)
  - *mode* is optional: `'r'` to open for text input (default), `'w'` to create and open for text output, `'a'` to open for appending text to the end

```
>>> fh = open('genesis.txt')
>>> print(type(fh))
<class '_io.TextIOWrapper'>
```

The Newline Character

- We use a special character called the "newline" to indicate when a line ends
- We represent it as \n in strings
- Newline is still one character – not two

```
>>> msg = 'Hello\nWorld!'
>>> msg
'Hello\nWorld!'
>>> print(msg)
Hello
World!
>>> msg = 'X\nY'
>>> print(msg)
X
Y
>>> len(msg)
3
```

File Processing

- A text file has newlines at the end of each line

```
The First Book of Moses:  Called Genesis\n
\n
1:1 In the beginning God created the heaven and the earth.
1:2 And the earth was without form, and void; and darkness was upon the face of the deep.
And the Spirit of God moved upon the face of the waters.\n
1:3 And God said, Let there be light: and there was light.\n
1:4 And God saw the light, that it was good: and God divided the light from the darkness.\n
1:5 And God called the light Day, and the darkness he called Night. And the evening and the
morning were the first day.\n
1:6 And God said, Let there be a firmament in the midst of the waters, and let it divide
the waters from the waters.\n
1:7 And God made the firmament, and divided the waters which were under the firmament from
the waters which were above the firmament: and it was so.\n
1:8 And God called the firmament Heaven. And the evening and the morning were the second
day.\n
1:9 And God said, Let the waters under the heaven be gathered together unto one place, and
let the dry land appear: and it was so.\n
```

File Handle as a Sequence

- A file handle open for read can be treated as a sequence of strings where each line in the file is a string in the sequence
- We can use the `for` statement to iterate through a sequence
- Remember – a sequence is an ordered set

```
fh = open('genesis.txt')
for line in fh:
    print(line)
```

Counting Lines in a File

- Open a file read-only
- Use a for loop to read each line
- Count the lines and print out the number of lines

```
# open.py
count = 0
fh = open('genesis.txt')
for line in fh:
    count = count + 1
print('Line count:', count)


$ python open.py
Line count: 1221
```

Reading the Whole File

- *fh*.`read()`
  - Read the while file (newlines and all) into a single string

```
>>> fh = open('genesis.txt')
>>> contents = fh.read()
>>> print(len(contents))
206951
>>> print(contents[:20])
The First Book of Mo
>>> print(contents[-20:])
 a coffin in Egypt.
```

Searching Through a File

- *str*.`startswith()`
  - Put an if statement in our for loop to only print lines that meet some criteria

```
fh = open('genesis.txt')
for line in fh:
    if line.startswith('1:'):
        print(line)
```

Blank Lines?

- Each line from the file has a newline at the end
- The `print` statement adds a newline to each line

```
1:1 In the beginning God created the heaven and the earth.\n
\n
1:2 And the earth was without form, and void; and darkness was upon the face of the
deep. And the Spirit of God moved upon the face of the waters.\n
\n
1:3 And God said, Let there be light: and there was light.\n
\n
1:4 And God saw the light, that it was good: and God divided the light from the
darkness.\n
\n
```

Searching Through a File (Revised)

- *str*.`rstrip()`
  - Strip the whitespace from the right-hand side of the string
  - Whitespace: blank(`' '`), tab(`'\t'`), newline(`'\n'`), etc.

```
fh = open('genesis.txt')
for line in fh:
    if line.startswith('1:'):
        line = line.rstrip()
        print(line)
```

Skipping with Continue

- Skip a line by using the `continue` statement
- *str*.`isdigit()`
  - Return `True` if all characters in the string are digits

```
fh = open('genesis.txt')
for line in fh:
    if not line[0].isdigit():
        continue
    line = line.rstrip()
    print(line)
```

Using `in` to Select Lines

- Use an `in` operator to look for a certain substring in a line

```
fh = open('genesis.txt')
for line in fh:
    if not line[0].isdigit():
        continue
    if not line.startswith('1:'):
        continue
    if 'heaven' in line:
        line = line.rstrip()
        print(line)
```

Splitting a String

- *str*.split(*sep*, *maxsplit*)
  - Split a string into a list of words (*sep* is the separator with the default value ' ')

```
fh = open('genesis.txt')
line_cnt, word_cnt, byte_cnt = 0, 0, 0
for line in fh:
    line_cnt += 1
    byte_cnt += len(line)
    words = line.split()
    word_cnt += len(words)
print(line_cnt, word_cnt, byte_cnt)
```

When Files are Missing

```
>>> fh = open('nofile')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
FileNotFoundError: [Errno 2] No such file or
directory: 'nofile'
```

Handling Bad File Names

```
fname = input('Enter a file name: ')
try:
    fh = open(fname)
except:
    print('File not found:', fname)
    quit()

count = 0
for line in fh:
    count += 1
print('There are', count, 'lines in', fname)
```