Every Boilermaker Engineer Codes: 101
Entry-Level Programming in Python

Lecture 08b

Dr. John H. Cole
<jhcole@purdue.edu>

PURDUE
UNIVERSITY®

COLLEGE OF ENGINEERING

Spring 2021

Part II

File IO

Table of Contents

Opening Files

`open(file)`

`open(file, 'r')` opens a text file in read-only mode.

- returns a file object (contains information about the file)
- 'r' in the second argument specifies read mode
- raises `FileNotFoundError` if the file does not exist

Terminal

```
>>> fo = open('spam.txt')
>>> fo.readable()
True
>>> fo.writable()
False
>>> fo.close()
```

Closing Files

`fo.close()` closes a file

- the number of open files is limited by the OS
- files should be closed when they are no longer needed
- failure to close a file causes a resource leak

Terminal

```
>>> fo = open('spam.txt')
>>> fo.readable()
True
>>> fo.writable()
False
>>> fo.close()
```

Using a Context Manager

```
with open(file) as fo:
```

- `with` is a context manager
- opens the file and sets `fo` to the file object
- automatically closes the file when the `with` block finishes
- use `with` instead of using `open` and `close` directly

Terminal

```
>>> with open('spam.txt') as fo:
...     fo.closed
...
False
>>> fo.closed
True
```

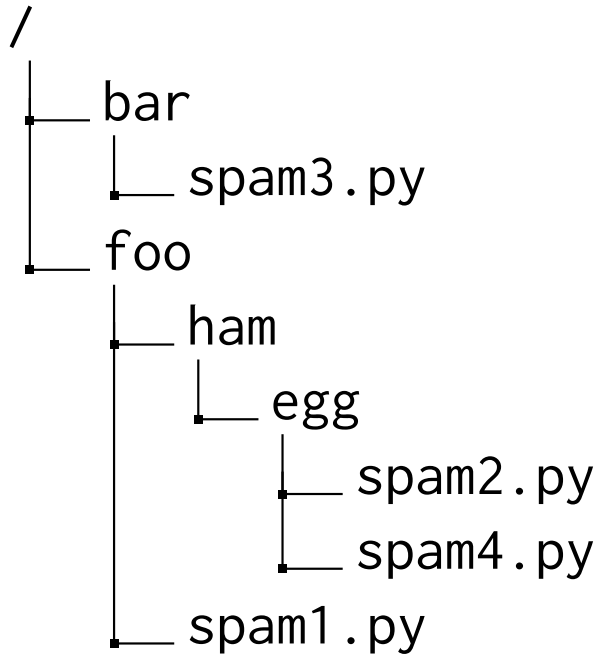
File Names

`file` a file path string

- must include the file extension (e.g. `.txt`)
- can specify a path to other directories
- absolute paths start with '/'
- relative paths are relative to the directory in which python ran

valid `file` strings from the `foo` directory

- `open('spam1.py')`
- `open('ham/egg/spam2.py')`
- `open('../bar/spam3.py')`
- `open('/foo/ham/egg/spam4.py')`



fo.read()

`fo.read()` returns the entire file as a string

Editor - cast.txt

```
1 John Cleese
2 Terry Gilliam
3 Eric Idle
4 Michael Palin
5 Graham Chapman
6 Terry Jones
```

Editor - read_all.py

```
1 with open('cast.txt') as fo:
2     everything = fo.read()
3 print(everything)
```

Terminal

```
$ python read_all.py
John Cleese
Terry Gilliam
Eric Idle
Michael Palin
Graham Chapman
Terry Jones
```

fo.read() – CONTINUED

- newline characters are included in the returned string

Terminal

```
>>> with open('spam.txt') as fo:
...     s = fo.read()
...
>>> s
'Spam\nSpam\negg\nSpam\n'
```

Editor - spam.txt

```
1 Spam
2 Spam
3 egg
4 Spam
```

fo.readlines()

`fo.readlines()` returns a list of the lines in the file

- newline characters are included in each item

Editor - spam.txt

```
1 Spam
2 Spam
3 egg
4 Spam
```

Terminal

```
>>> with open('spam.txt') as fo:
...     l = fo.readlines()
...
>>> l
['Spam\n', 'Spam\n', 'egg\n', 'Spam\n']
```

Reading Line-by-line

- a `for` loop can process a file line-by-line

Editor - spam.txt

```
1 Spam
2 Spam
3 egg
4 Spam
```

Terminal

```
>>> l = []
>>> with open('spam.txt') as fo:
...     for line in fo:
...         l.append(line.rstrip())
...
>>> l
['Spam', 'Spam', 'egg', 'Spam']
```

Write Mode

`open(file, 'w')` opens a file in write mode

- returns a file object (contains information about the file)
- 'w' in the second argument specifies write mode
- creates a new file if it does not exist
- truncates the file if it already exist

Terminal

```
>>> fo = open('egg.txt', 'w')
>>> fo.readable()
False
>>> fo.writable()
True
```

fo.write(s)

`fo.write(s)` writes the string `s` to the file `fo`

- newline characters are *not* automatically added

Editor - write_spam1.py

```
1 s = 'Spam'
2 with open('spam1.txt', 'w') as fo:
3     fo.write(s)
4     fo.write(s)
5     fo.write(s)
```

Editor - spam1.txt

```
1 SpamSpamSpam
```

Terminal

```
$ python write_spam1.py
```

fo.write(s) – CONTINUED

`fo.write(s)` writes the string `s` to the file `fo`

- newline characters are *not* automatically added

Editor - write_spam2.py

```
1 s = 'Spam\n'
2 with open('spam2.txt', 'w') as fo:
3     fo.write(s)
4     fo.write(s)
5     fo.write(s)
```

Terminal

```
$ python write_spam2.py
```

Editor - spam2.txt

```
1 Spam
2 Spam
3 Spam
```

fo.writelines(l)

`fo.writelines(l)` writes the items in list `l` to the file `fo`

- newline characters are *not* automatically added

Editor - write_spam3.py

```
1 l = ['Spam', 'Spam', 'Spam']
2 with open('spam3.txt', 'w') as fo:
3     fo.writelines(l)
```

Terminal

```
$ python write_spam3.py
```

Editor - spam3.txt

```
1 SpamSpamSpam
```

fo.writelines(l) – CONTINUED

`fo.writelines(l)` writes the items in list `l` to the file `fo`

- newline characters are *not* automatically added

Editor - write_spam4.py

```
1 l = ['Spam\n', 'Spam\n', 'Spam\n']
2 with open('spam4.txt', 'w') as fo:
3     fo.writelines(l)
```

Terminal

```
$ python write_spam4.py
```

Editor - spam4.txt

```
1 Spam
2 Spam
3 Spam
```

Append Mode

`open(file, 'a')` opens a file in append mode

- returns a file object (contains information about the file)
- 'a' in the second argument specifies append mode
- creates a new file if it does not exist
- adds to the file if it already exist

Terminal

```
>>> fo = open('egg.txt', 'a')
>>> fo.readable()
False
>>> fo.writable()
True
```

Appending

Editor - append_devs.py

```
1 name = input('Name: ')
2 with open('devs.txt', 'a') as fo:
3     fo.write(name + '\n')
```

Terminal

```
$ python append_devs.py
Name: Ada Lovelace
$ python append_devs.py
Name: Grace Hopper
$ python append_devs.py
Name: Klara Dan von Neumann
```

Editor - devs.txt

```
1 Ada Lovelace
```

Editor - devs.txt

```
1 Ada Lovelace
2 Grace Hopper
```

Editor - devs.txt

```
1 Ada Lovelace
2 Grace Hopper
3 Klara Dan von Neumann
```


Thanks for watching!