ENSF 592: Programming Fundamentals for Data Engineers

Yves Pauchard

Lecture 7: Files

September 27, 2019

Agenda

1. Key ideas in Ch 14
2. `with` statement
3. Error handling
4. Preparation for next lecture

Compound statement `with`

https://docs.python.org/3.7/reference/compound_stmts.html#the-with-statement

> The with statement is used to wrap the execution of a block with methods defined by a context manager.

```
with open('myfile.txt') as fin:
    s = fin.readline()
```

Error handling

Example from here: https://docs.python.org/3.7/tutorial/errors.html

Available exceptions: https://docs.python.org/3/library/exceptions.html

```
import math
import sys

try:
    f = open('myfile.txt')
    s = f.readline()
    i = int(s.strip())
    # math.sqrt(str(i)) #Type error
except OSError as err:
    print("OS error: {0}".format(err))
except ValueError:
    print("Could not convert data to an integer.")
except:
    print("Unexpected error:", sys.exc_info()[0])
    raise
```

Add error handling to sed.py

Get it from: http://greenteapress.com/thinkpython2/code/sed.py

Add error handling to sed.py

```
def sed(pattern, replace, source, dest):
    """Reads a source file and writes the destination file.

    In each line, replaces pattern with replace.

    pattern: string
    replace: string
    source: string filename
    dest: string filename
    """

    try:
        fin = open(source, 'r')
        fout = open(dest, 'w')

        for line in fin:
            line = line.replace(pattern, replace)
            fout.write(line)

    except OSError as err:
        print("OS error: {0}".format(err))
    except TypeError:
        print("Could not apply patterns, check variables pattern and replace.")
    finally:
        fin.close()
        fout.close()
```

For Fun: Get find_duplicates.py to work

md5sum

On macosx, `md5sum` is called `md5`

On windwos (and macosx) use python hashlib

```
# ref https://www.geeksforgeeks.org/md5-hash-python/
import hashlib

# initializing string
str = "GeeksforGeeks"

# encoding GeeksforGeeks using encode()
# then sending to md5()
result = hashlib.md5(str.encode())

# printing the equivalent hexadecimal value.
print("The hexadecimal equivalent of hash is : ", end ="")
print(result.hexdigest())
```

For Fun: Get find_duplicates.py to work

diff

macosx: `diff` is available on macosx.

windows: closest I could find is `FC`
https://stackoverflow.com/questions/6877238/what-is-the-windows-equivalent-of-the-diff-command

Preparation For Next Lab/Lecture

Read/follow Ch 15 and Ch 16 in Think Python 2e