IDS 400
Programming for Data Science in Business

Schedule (tentative)

Date	Lecture Number	Topics
08/24	Lecture 1	Introduction
08/31	Lecture 2	Basic
09/07	Lecture 3	Condition
09/14	Lecture 4	Loop
09/21	Lecture 5	String + **Quiz 1**
09/28	Lecture 6	Type
10/05	Lecture 7	Function
10/12	Lecture 8	File + **Quiz 2**
10/19	Lecture 9	Pandas
10/26	Lecture 10	Numpy
11/02	Lecture 11	Machine Learning
11/09	Lecture 12	Visualization
11/16	Lecture 13	Web Scraping & Deep Learning
11/23	***Thanksgiving***	***No lecture***
11/30	**Final presentation**	In class presentation
12/05	**Project submission due**	

25 multichoice questions (No coding questions) on Blackboard.

For This Class

- Files and Modules

File processing

- A text file is a computer file that only contains text and has no special formatting such as bold text, italic text, images, etc. Text files are identified with the **.txt** file extension.

Open 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.*
- Similar to “File -> Open” in a Word Processor.

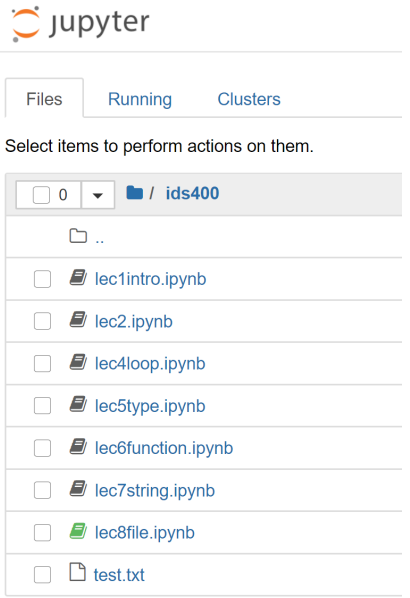
Open()

- Syntax: `fle_handler_variable = open(filename,mode)`
- Returns a handle use to manipulate the file.
- **filename** is a string (a string variable or a string constant)
- **mode** is optional and should be '**r**' if we are planning reading the file; and '**w**' if we are going write to the file.

```
fh = open('test.txt','r')
print(fh)
```

```
<_io.TextIOWrapper name='test.txt' mode='r' encoding='cp1252'>
```

- The ***open()*** method opens the file and returns a *TextIOWrapper* object but **does not read the files content**.



Python file mode

Modes	Description
r	Opens a file for reading only.
rb	Opens a file for reading only in binary format.
r+	Opens a file for both reading and writing.
rb+	Opens a file for both reading and writing in binary format.
w	Opens a file for writing only. Overwrites the file if the file exists. A new file is created if the file does not exist.
wb	Opens a file for writing only in binary format. Overwrites the file if the file exists. A new file is created if the file does not exist.

Python file mode

Modes	Description
w+	Opens a file for both writing and reading. Overwrites the existing file if the file exists. A new file is created if the file does not exist.
wb+	Opens a file for both writing and reading in binary format. Overwrites the existing file if the file exists. A new file is created if the file does not exist.
a	Opens a file for appending. A new file is created if the file does not exist.
ab	Opens a file for appending in binary format. A new file is created if the file does not exist.
a+	Opens a file for both appending and reading. A new file is created if the file does not exist.
ab+	Opens a file for both appending and reading in binary format. A new file is created if the file does not exist.

* rb+ does not create the file from scratch
* wb+ does create the file from scratch

Jupyter working directory

- If you use Google Colab →

```
from google.colab import drive
drive.mount("/content/drive")
```

```
%cd "/content/drive/MyDrive/Ids400"
```

```
/content/drive/MyDrive/Ids400
```

- When files are missing

```
fh = open('test1.txt','r') # there is no 'test1.txt' file
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-2-61bd367b39d2> in <module>
----> 1 fh = open('test1.txt','r') # there is no 'test1.txt' file

FileNotFoundError: [Errno 2] No such file or directory: 'test1.txt'
```

- Change working directory in Jupyter

```
import os
os.getcwd() #check current working directory

os.chdir(' ') # put your new desired working directory in ''

os.getcwd() # check the new working directory
```

File handler 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('test.txt','r')
for line in fh:
    print(line)
```

```
IDS 400 Python

Hello
```

Read the whole file

- We can read the whole file(newlines and all) into a single string.

```
fh = open('test.txt')
inp = fh.read()
print(inp)
```

```
IDS 400 Python
Hello
```

```
print(len(inp))
```

```
20
```

```
print(inp[14])
```

```
print(inp[13])
```

```
n
```

Read file into a list

- We can use *readlines()* to get a list.
- Each element in the list is a line.

```
fh = open('test.txt')
lines = fh.readlines()
print(len(lines))
print(lines[:2])
```

```
2
['IDS 400 Python\n', 'Hello']
```

File write

- The *write()* method writes any string to an open file.
- The write() method does not add a newline character ('\n') to the end of the string.

```
fh = open('test.txt', 'w')
#lines = fh.readlines()
fh.write('Python is great\nI like Python')
fh.close()
```

```
fh = open('test.txt','r')
for line in fh:
    print(line)
```

```
Python is great

I like Python
```

Other file operations

- Python *os* module provides methods that help you perform file-processing operations, such as renaming and deleting files.
- To use this module, you need to import it first and then you can call any related functions.
  - import os
  - os.mkdir("newdir")
  - os.remove(file_name)
  - os.listdir(path)
  - os.rename(current_file, new_file)

```
os.mkdir("/content/drive/MyDrive/Ids400/new")
```

```
os.remove("test_1.txt")
```

```
os.listdir("/content/drive/MyDrive/Ids400/")
```

```
os.rename("new", "new_2")
```

```
os.getcwd()
```

```
'/content/drive/MyDrive/Ids400'
```

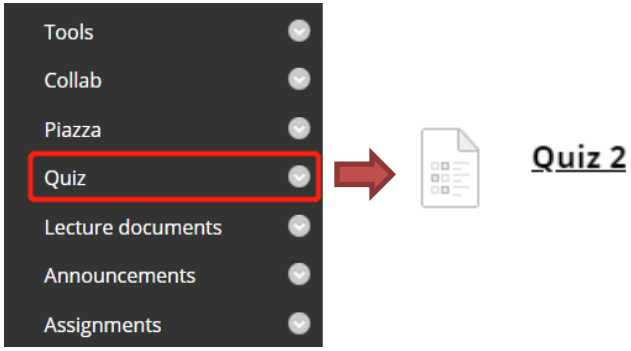
```
with open("HW4/stock.txt", "r") as file:
  contents = file.readlines()
  print(contents)
```

```
['coke,10\n', 'juice,5\n', 'milk,13']
```


More os functions: https://docs.python.org/3.3/library/os.html

Lab *File*

Quiz 2



- **25 multichoice questions**
- **60 minutes**
- You only have ONE attempt to do the quiz.
- This quiz is forced to complete. Once started, this test must be completed in one sitting. Do not leave/refresh/close the test before clicking "*Save and Submit*".
- There is a 60 minutes timer. Warnings appear when half the time, 5 minutes, 1 minute, and 30 seconds remain (The timer does not appear during this test). Test will save and submit automatically when time expires.
- You must have your camera on during the quiz.
- You can change your previous answers before submission.
- Please do not share your voice during the quiz. If you have a question, please send a private message at chat panel.
- Once you submit, your score will be available immediately.