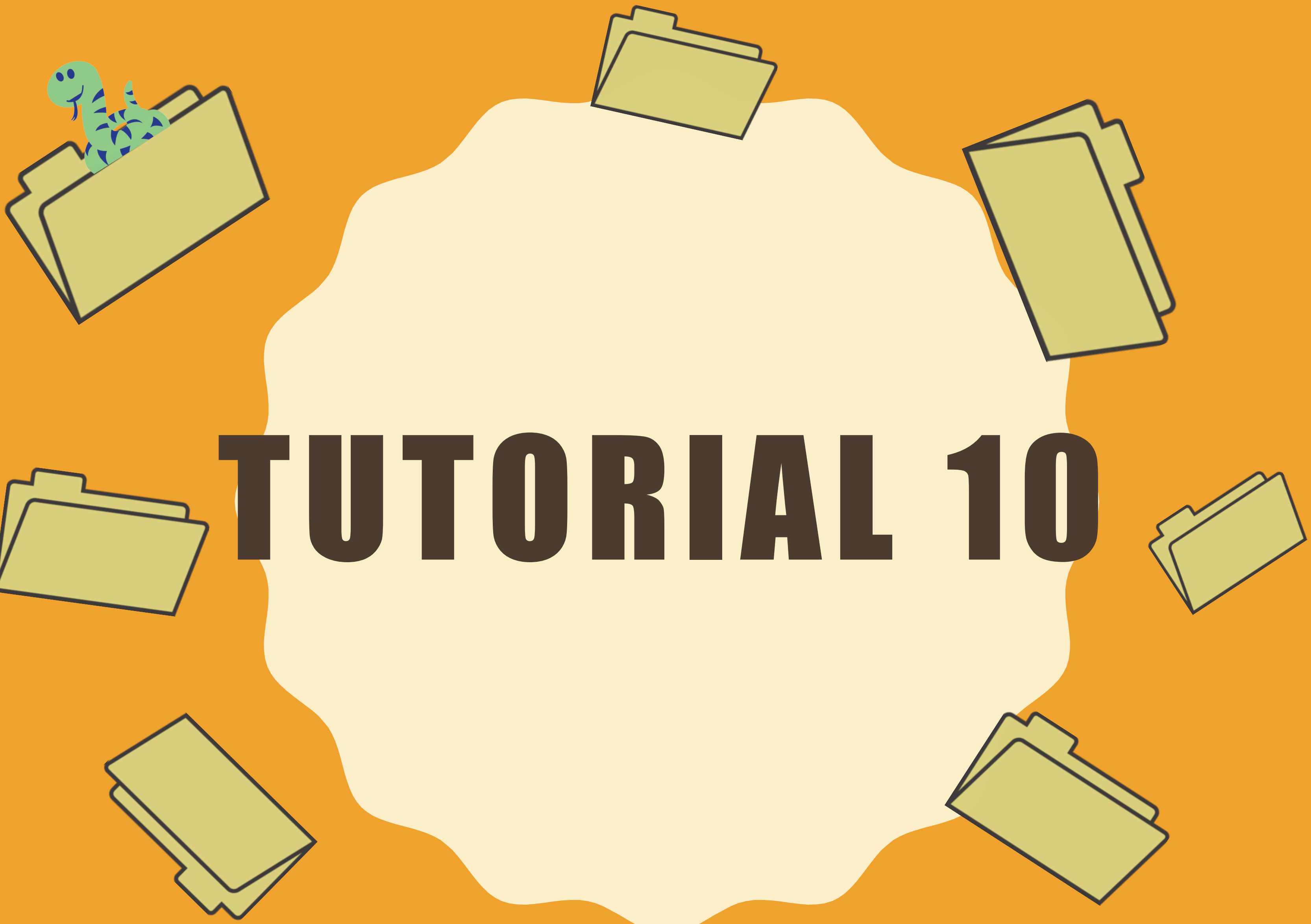
TUTORIAL 10

READING AND WRITING FILES



REMINDER

- Assignment 8 will be due on April 3rd, the last day of classes.
  - We recommend that you start this assignment early, so you have more time to seek help.
- Stay tuned to Piazza for information about the development of the last few weeks of the course.

QUICK REVIEW - CLASSES

- Allows related information to be grouped together
- We'll use `__init__`, `__repr__`, and `__eq__` with the class
- Class methods
  - First parameter is always `self`
  - Class methods are called using the same dot notation as the string and list methods
  - Like other functions, may return values or print information

REVIEW - FILES

Find – (same folder)	

Open	`read_file = open(filename, ‘r’) or open(filename)`
	# opens the file named `filename` for reading
	`write_file = open(filename, ‘w’)`
	# creates the file named `filename` for writing

Read	`read_file.readline()  -> Str`
	# reads one line, includes newline character
	`read_file.readlines() -> (listof Str)`
	# read all lines at once, each string terminates with newline character

Write	`write_file.write(Str)`
	# write one string
	`write_file.writelines((listof Str))`
	# write all strings in the list at once

Close	`read_file.close()`
	`write_file.close()`
	# If you forget to close a file after writing, you may lose some data

WRITING A FILE

- To write to a file you must include the string `“w”` in your call to the file function, `open(filename, “w”):`
- If `filename` does **not** exist, a new file with that name will be created
- If `filename` does exist, its previous contents will be erased when opened

DESIGN RECIPE

- Purpose: include details about what is read from a file or written to a file
- Effects:
  - Reads a file `filename`
  - Writes to a file called `filename`
- Examples
- Testing

  `check.set_file_exact(actual, expected)`

  - Comparing the contents in the file created (`actual`) to a created existing file (`expected`)

QUESTION 1

Write the Python function `list_tables` that reads the data from a text file, `seating_in`, which contains the seating arrangement for the event, and returns a list of the tables everyone will be sitting at

- You may assume that the file contains one line for each table with everyone's name separated by a single space and no spaces inside someone's name:
- Example, if banquet.txt contains

  Jughead Bret Mr.Weatherbee
  Archie Veronica Cheryl
  Betty Kevin

  Then list_tables(banquet.txt) =>
  [["Jughead", "Bret", "Mr.Weatherbee"]["Archie", "Veronica", "Cheryl"]
  ["Betty", "Kevin"]]

QUESTION 2

Write the Python function `fancy_tables` that consumes a `(listof (listof Str))`, `tables`, representing the seating plan for the guests and writes to a file `seating_plan` in the following way:

- There is a title for each table: "Table X Seating Plan" where X is the table number (it's index in the list plus one), then followed by each individual guest's name on separate lines, followed by a blank line

EXAMPLE

- Example, if banquet_tables = [[“Jughead”, “Bret”][“Archie”, “Veronica”]]

  Then calling fancy_tables(banquet_tables, “banquet.txt”) writes the following to banquet.txt:

  ```
  Table 1 Seating Plan
  Jughead
  Bret

  Table 2 Seating Plan
  Archie
  Veronica
  ```

CD CLASS

```
class CD:
   '''
   Fields: artist (Str), duration (Nat), songs (listof Str)
   where
   * artist is the name of the performer of the album
   * duration is the length of the CD in seconds
   * songs contains the names of the songs on the CD.
   '''

   def __init__(self, artist, dur, los):
       #Initializes a CD
       self.artist = artist
       self.duration = dur
       self.songs = los
```

```python
def __repr__(self):
     '''
    The method represents a CD
    [format]:
       Artist Name: XXX
       Duration: XXX
       Songs: XXX
     '''
    songstr = ", ".join(self.songs)
    return "Artist Name: {0}\nDuration: {1}\nSongs:
           {2}”.format\
           (self.artist, self.duration, songstr)”


def __eq__(self, other):
   '''
    The method checks if two CDs are the same. '''
    return isinstance(other, CD) \
           and self.artist == other.artist \
           and self.duration == other.duration \
           and self.songs == other.songs
```

QUESTION 3 – MAKE_CD_DICT

Write the Python function `make_cd_dict` that reads the data from a text file, `fname`, and returns a dictionary of CD objects containing the data from the file, with keys equal to the album title of the CD.

```
{album title: CD(artist, duration, [listof songs])}
```

- You may assume that each CD has a unique album title.
- The files will be formatted in the following way:

  name of artist, album title of CD, length in seconds, song titles

  (separated by a comma and at least one space)

EXAMPLE FILE

Comment:

# name of artist, name of CD, length in seconds, song titles

File:

Yes, Close to the Edge, 2266, Close to the Edge, And You and I, Siberian Khatru
Various, Disney Mix, 912, You'll Be in My Heart, Kiss the Girl, Circle of Life, I'll Make a Man Out of You, Whole New World, Go the Distance
Pink Floyd, Wish You Were Here (Side 1), 1268, Shine On You Crazy Diamond (1-5), Welcome to the Machine
Jonathan Coulton, Code Monkey, 203, Code Monkey
The Beatles, Abbey Road, 255, Mean Mr. Mustard, Polythene Pam, She Came in Through the Bathroom Window

QUESTION 4 – OUTPUT_CD_INFO

Write the Python function `output_CD_info` that consumes a dictionary of CD objects, `cd_dict` (like the one returned in Q1) and a string, `name`.

- If the string is not a key in the dictionary, the function should do nothing.
- If the string is a key in the dictionary, the function should create a text file named `Artist - Title.txt`, where Artist and Title relate to the CD found at `cd_dict[name]`.
- The text file should contain (each on its own line)
  - the title of the CD in uppercase,
  - then the artist,
  - then the duration of the CD (formatted like in question 1),
  - then a blank line, and
  - then the index of the song with name of each track in the CD.

EXAMPLE - "VARIOUS - DISNEY MIX.TXT"

```
d3 = {"Disney Mix": CD("Various", 912, ["You'll Be in My
Heart", "Kiss the Girl", "Circle of Life", "I'll Make a Man
Out of You", "Whole New World", "Go the Distance”])}
```

```
>> output_CD_info(d3, "Disney Mix")
```

```
DISNEY MIX
Various
912

1.  You'll Be in My Heart
2.  Kiss the Girl
3.  Circle of Life
4.  I'll Make a Man Out of You
5.  Whole New World
6.  Go the Distance
```