Wk	Focus & Medium	Weekly Topic & Assignment
8 Oct 17 to 22	(1 of 5) **Goals** • dict = { } • list = [ ] • string = " " • tuple = (,) • set = set( ) • pd.DataFrame() • series = d = {'a': 1, 'b': 2, 'c': 3} ▪ ser = pd.Series(data=d, index=['a', 'b', 'c']) **What do you do if you need to see an object?**  **=> dir(myobject)** help(myanimal object) <F9>	**Congratulations!** In all sincerity, I am very proud of all of you. You have been incredible learning my 7 pillars of Python. As a team, you have completed experiencing Python's core data objects and being able to • bring any pd.read_csv('path') (comma separated values) or pd.read_xlsx() data from a spreadsheet application. • understand positional indexing • learned that axial data positioning is (0=row, 1=column) • understand how to pack, unpack, and read Python-packed data objects #========== [({...})],{"" : [ ] } #========== string data is in a dictionary {key:value} which is inside a tuple which is inside a list separated by a comma to another object which is a dictionary with a string for a key, and a list of its key values MOST IMPORTANTLY – "you" have the tools to decipher how any data is packed and figure out how to mix and mingle python objects to get data as needed. You have also worked with iterators, conditionals, and variables and can probably figure out how to transpose data if needed. ========== **dir(<myObject>) => displays its constructors, methods, and attributes** ['__class__', '__delattr__', '__dict__','__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__le__', '__lt__', '__module__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__','__setattr__', '__sizeof__','__str__', '__subclasshook__', '__weakref__', 'name', 'species',====> user defined attributes 'train'] help(<myobject> or <function> ------- | Data and other attributes defined here: | name = '' | species = '' ==> these are the attributes in our wk7 object | train = ''

Wk	Focus & Medium	Weekly Topic & Assignment
8 Oct 17 to 22	(2 of 5) wk8 code git	

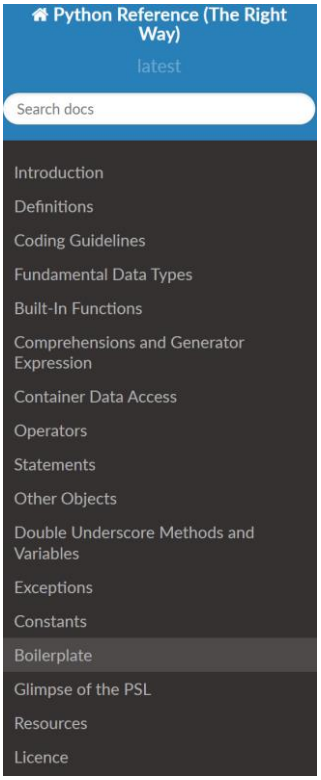
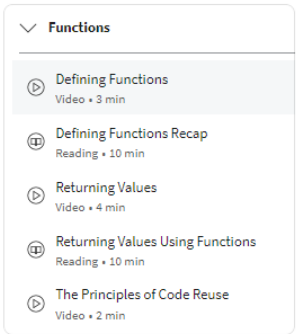
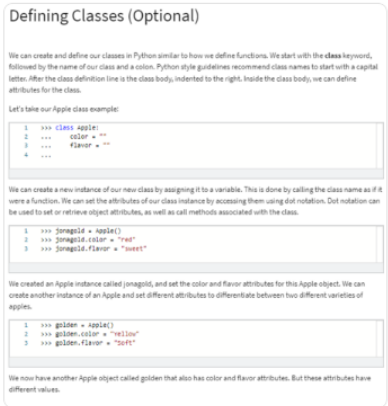
```
"""                                 it.304.wk8 (10/16-10/22/22)
Created on Sat Oct 15 13:56:24 2022
@author: 17574          b.hogan@snhu.edu
"""
#======================= > Week 8
#===== Classes - Week 8
#================
#========================

#=> Objective: use the following Classes example to make one of your own
#=> new function input("<message>") -> asks user for a value

#=====================================================================
# Part I:   Import libraries and source data
# Part II:  Draft an object with couple functions
# Part III: Creat a child object and run the function
# Part IV:  Run a report
#=====================================================================
  ''' CARLY! this is not boo scary!
      conditionals (below)are just a set of questions, often in your own
words.
      if you are stuck, set a timer and spend no more than 20 minutes.
      research says your better phoning or emailing a friend as anything
      after 20 minutes exceeds optimal learning    good luck to all !
~brian'''

#=> # Part I:
#=============================================================
#=> # Part I:  Import libraries and source data
#=============================================================
Import libraries + data
import pandas as pd                                #dataframe library
import numpy as np                                 #numeric library
import matplotlib.pyplot as plt                    #visualization library
import os                                          #operating system library
import sys                                         # sys.exit()
os.chdir('c:\\Users\\17574\\Desktop\\data_it304') #microsoft uses 2\\
df0 = pd.DataFrame()                               #explicitly set datafarme
df0 = pd.read_excel("data_shakes_corpus_v1.xlsx") #ETL method 2
df0.info()# RangeIndex: 37 entries,0 to 36, 4 col=> all data u need to
index
mydict = df0.to_dict()                             #df to dict
'''mydict_shakespeare => {'title':{},'script':{},'type':{},'ID:{} }'''
len(df0) #37                              lenth is always veritical by
default!

#=> # PART - DEtour - was best to add NEW INFO here
'''this will help you create a report for quiz end of wk...'''
#================================>
#=> Function idea and drive a bitchen camero data to excel
#================================
  # Use if, elif, else 'conditionals' to draft your questions based on data
  # Consider drafting 1-3 questions on an index card before coding
  # detail what information need to perform so you focus vs get stuck on
names
  # remember - objects are the actors and functions are their script

'''Fucntion ideas & examples:
   i) write a function to count total characters in a play or all plays!
   ii) use an iterator, count characters, and put in a list
   iii) use new lists to create a report or write back to excel using'''
```

8 Oct 17 to 22	(3 of 5) **wk8 code git**	

```
mylist = []                 #so this could be a function to count
characters
for i in mydict['title'].values(): mylist.append(len(i))
print(mylist, type(mylist), sum(mylist))

#use the new objects and variables to creat a dictionary
myNewDict = {sum(mylist): mylist}
        #   or {'play-1':[<TitleTotalWords>,<ScriptTotalWords>]}

print(myNewDict)
print(type(myNewDict))
myDF = pd.DataFrame.from_dict(myNewDict)#function create a pandas.DF from
dict
#myDF.info()                                #check it out

''' Send to excel or view here - will review in class'''
mywriter = pd.ExcelWriter('myoutput.xlsx') #create object that writes out
myDF.to_excel(mywriter)
mywriter.save()
myDF                                #Excel will look exact same !

#=> # Part II:
#============================================================
#=> # Part II:  Draft an object with couple functions
    # We are training with .self notation. write self.<attribute or
variable>
    # are inherent, or part of our instantiated children objects
#============================================================

'''START - HIGHlight all of class and hit F9 from lines 93 to 150 '''

class shakespeare_minion:           #this defines the parent object
    pass
    name = ""
    perform_work = 0                #yes,no switch so could exit terminal
    total_plays_not_read = len(df0) #use an object vs. hardcode a value
    total_plays_read = 0            #increment so you know how much work
done
    num_plays_work_now = 0          # countdown tracker based on user input

    '''Function-1: ask user how many plays to read'''
    def how_much_work_master(self):
                        #int() function here ensures user response encoded as
a #
        perform_work = int(input("Enter greater than 0 to run program =>
"))
        if perform_work <= 0:
            sys.exit()       #On/off switch so can exit program in terminal

        if perform_work > 0:    #NEW - ask user a question with input()
            self.num_plays_work_now = \
                int(input("Enter how many plays you will read today?=> "))
        perform_work = 0    #set back to zero as 1x trigger
```

8 Oct 17 to 22	(4 of 5) **wk8 code git**	

```
    '''Function-2: have minions completed what they said they would do?'''
    def do_work_and_report_status(self):

    #0) for transactions, here would be some kind of wait time to do work


    #1) condition 1 - Did we complete total work yet?
        if self.num_plays_work_now <= 0:
            #after test, then increment/decrement associated variables
            self.total_plays_not_read = self.total_plays_not_read - 1
            self.num_plays_work_now = self.num_plays_work_now - 1
            total_plays_read =+1 #another way to increment variables

            return "Master! {} is done. I finished {} plays today.". \
                            format(self.name,self.total_plays_read)

    #2) condition 2 - Still doing daily work ?
        elif self.num_plays_work_now > 0:
            #after test, then increment/decrement associated variables
            self.total_plays_not_read = self.total_plays_not_read - 1
            self.total_plays_read = self.total_plays_read +1
            self.num_plays_work_now = self.num_plays_work_now -1
            total_plays_read =+1        #another way to increment variables

    #3) condition 3 - this is a NESTED loop b/c now you either no more work
    #       or you report what you have left to do in this batch
            if self.num_plays_work_now == 0:
                return "Master I have {} plays left to read AND no more
work.\
                            I am 100% done for today so start over!".\
                                        format(self.total_plays_not_read)

            else:
                return "Master I read {} plays today and have {} more plays
\
                to do in this most egregiousness and unjust batch.".\
                    format(self.total_plays_read,self.num_plays_work_now)


'''END HERE - HIGHlight all of class to define full object'''

#==============================================================
# Part III: Creat a child object and run the function
#==============================================================

# IIIa: ask user number plays to ready & run the transaction
'''Run these 3 lines together! - This starts to queue up total work'''
minion = shakespeare_minion()
minion.name = "Toothless Harold"
minion.how_much_work_master()         #ask user how much to do!

#=================
'''====>Now run a transaction, that is read a play.
        this program runs these transactions manually.
        The final little program we make will run them all at once.'''

#================= select all 4 lines - keep running to run out of work!
print(minion.do_work_and_report_status())
print(minion.total_plays_not_read)
print(minion.num_plays_work_now)
print(minion.total_plays_read)
```

Wk	Focus & Medium	Weekly Topic & Assignment
8 Oct 17 to 22	(5 of 5) **Blog discussion of pros \ cons searching for information to help learn clases** **cliff**-notes, even this 'ok' site is there to sell and grab your computer data. Always use a vpn! Sometimes you need a quick answer. But – when you start something new spend time on your cheat sheet and find quality resources so in times of need you can make it happen. **mit student gem**  	**b.hogan** 11:06 AM autobots.304 - I am not against the internet for training and checking things out but everyone remember what Jackson said in class, "often I can get distracted with other things." This is a significant challenge for your generation and learning what is good, bad, and ugly infomation is usually usually easy for 'ugly' information and questionable for all else. For instance, this article is decent for what we are doing for week 8 as you get busy with classes. The examples are informative, complete, and meaningful. I would be comfortable putting as a syllabus reference BUT it isn't a quality academic reference. https://www.toptal.com/python/python-class-attributes-an-overly-thorough-guide (edited) **Toptal Engineering Blog** Python Class Attributes: An Overly Thorough Guide Python class attributes can lead to elegant code, as well as frustrating bugs. In this guide, we will outline specific use-cases for attributes, properties, variables, objects and more. (87 kB)  A MUCH better way is to go onto cousera and find a class that is similar to what you are learning whne you need to, ah - do things more quickly lets say. So it took me 2 minutes to find on Coursera "crash course in python." Here is what they ahve in week 2 - anything look familar to the zipper? image.png  Sure, its costs $39 a month to have access but for Week 8 you are now writing your own functions while learning your own python objects. For our class purposes this is nice supplemental information but I am also giving you evcerything you need, or at least I hop so, to be productive. As promised, we will review such deft research approaches in November. I keep hitting this nail because I don't want you to experience what I did, that is not going outside, while working harder and not smarter for parts of my coding and analysis work life. for this week - here is what the google coursera class has for class attributes like we reviewed yesterday for our farm animal names and species. It is nothing special, and I like my better. image.png  Thank you for reading! my purpose is to keep hammering this so you spend more time performing quality work
	example ok website https://www.toptal.com/python/python-class-attributes-an-overly-thorough-guide	