'''

Python Homework with Chipotle data

https://github.com/TheUpshot/chipotle

'''

'''

BASIC LEVEL

PART 1: Read in the file with csv.reader() and store it in an object called 'file\_nested\_list'.

Hint: This is a TSV file, and csv.reader() needs to be told how to handle it.

https://docs.python.org/2/library/csv.html

'''

import csv

# specify that the delimiter is a tab character

with open('chipotle.tsv', mode='rU') as f:

file\_nested\_list = [row for row in csv.reader(f, delimiter='\t')]

'''

BASIC LEVEL

PART 2: Separate 'file\_nested\_list' into the 'header' and the 'data'.

'''

header = file\_nested\_list[0]

data = file\_nested\_list[1:]

'''

INTERMEDIATE LEVEL

PART 3: Calculate the average price of an order.

Hint: Examine the data to see if the 'quantity' column is relevant to this calculation.

Hint: Think carefully about the simplest way to do this!

'''

# count the number of unique order\_id's

# note: you could assume this is 1834 since that's the maximum order\_id, but it's best to check

num\_orders = len(set([row[0] for row in data])) # 1834

# create a list of prices

# note: ignore the 'quantity' column because the 'item\_price' takes quantity into account

prices = [float(row[4][1:-1]) for row in data] # strip the dollar sign and trailing space

# calculate the average price of an order and round to 2 digits

round(sum(prices) / num\_orders, 2) # $18.81

'''

INTERMEDIATE LEVEL

PART 4: Create a list (or set) of all unique sodas and soft drinks that they sell.

Note: Just look for 'Canned Soda' and 'Canned Soft Drink', and ignore other drinks like 'Izze'.

'''

# if 'item\_name' includes 'Canned', append 'choice\_description' to 'sodas' list

sodas = []

for row in data:

if 'Canned' in row[2]:

sodas.append(row[3][1:-1]) # strip the brackets

# equivalent list comprehension (using an 'if' condition)

sodas = [row[3][1:-1] for row in data if 'Canned' in row[2]]

# create a set of unique sodas

unique\_sodas = set(sodas)

'''

ADVANCED LEVEL

PART 5: Calculate the average number of toppings per burrito.

Note: Let's ignore the 'quantity' column to simplify this task.

Hint: Think carefully about the easiest way to count the number of toppings!

'''

# keep a running total of burritos and toppings

burrito\_count = 0

topping\_count = 0

# calculate number of toppings by counting the commas and adding 1

# note: x += 1 is equivalent to x = x + 1

for row in data:

if 'Burrito' in row[2]:

burrito\_count += 1

topping\_count += (row[3].count(',') + 1)

# calculate the average topping count and round to 2 digits

round(topping\_count / float(burrito\_count), 2) # 5.40

'''

ADVANCED LEVEL

PART 6: Create a dictionary in which the keys represent chip orders and

the values represent the total number of orders.

Expected output: {'Chips and Roasted Chili-Corn Salsa': 18, ... }

Note: Please take the 'quantity' column into account!

Optional: Learn how to use 'defaultdict' to simplify your code.

'''

# start with an empty dictionary

chips = {}

# if chip order is not in dictionary, then add a new key/value pair

# if chip order is already in dictionary, then update the value for that key

for row in data:

if 'Chips' in row[2]:

if row[2] not in chips:

chips[row[2]] = int(row[1]) # this is a new key, so create key/value pair

else:

chips[row[2]] += int(row[1]) # this is an existing key, so add to the value

# defaultdict saves you the trouble of checking whether a key already exists

from collections import defaultdict

dchips = defaultdict(int)

for row in data:

if 'Chips' in row[2]:

dchips[row[2]] += int(row[1])

'''

BONUS: Think of a question about this data that interests you, and then answer it!

'''