import analyzer, explorer, data, interface

from pprint import pprint

def get\_interface\_choices\_dict():

"""

Returns a dict that has interface choice keys and values containing

an inner dict specifying the function the choice should call and

a description to display to the user.

Changing the returned dict is the main API to alter the program's

main menu behavior. Pass dict to `interface.present\_choices()` to

display main menu and call the returned function in response to user

input.

"""

# Since this dict contains function names, wrapping it in a

# function avoids circular imports.

# Note that this dict is returned without first assigning it a name.

return {

"1" : {

"function" : explorer.browse\_rows,

"description" : "Browse rows of data"

},

"2" : {

"function" : explorer.print\_random\_row,

"description" : "Print random row"

},

"3" : {

"function" : explorer.print\_specific\_row,

"description" : "Print specific row"

},

"4" : {

"function" : data.get\_new\_csv\_data\_dict,

"description" : "Change data file"

},

"5" : {

"function" : data.get\_selected\_columns\_list\_in\_dict,

"description" : "Select columns to print"

},

"6" : {

"function" : interface.get\_column\_type\_dict,

"description" : "Select column data types"

},

"7" : {

"function" : analyzer.new\_function,

"description" : "Get counts and percentages of categorical values"

}

}

def main(filename\_str = 'ischools-clean.csv'):

"""

Main entrypoint of program. Loads data from file specified by

`filename\_str` as dict returned from `data.get\_new\_csv\_data\_dict`.

Main loop of program repeatedly presents user with main menu choices

returned by `get\_interface\_choices\_dict`, calls the function corresponding

to user's input with `csv\_data\_dict` as argument, and updates that dict if

the function returns a dict.

All functions named in `interface\_choices\_dict` should accept keyword arguments

(i.e. include a `\*\*kwargs` parameter) and return either `None` or a dictionary

with some subset of the keys in `csv\_data\_dict`.

"""

print("Welcome to the CSV Explorer and Analyzer\n")

interface\_choices\_dict = get\_interface\_choices\_dict()

csv\_data\_dict = data.get\_new\_csv\_data\_dict(new\_data\_file=filename\_str, extra\_info = True, ask\_user = False)

#pprint(csv\_data\_dict)

while True:

user\_func = interface.present\_choices(interface\_choices\_dict)

returned\_val = user\_func(\*\*csv\_data\_dict)

if returned\_val == None:

continue

else:

# The function returned a value; update arguments dict passed to future functions

assert isinstance(returned\_val, dict), "Any returned value should be a dict"

csv\_data\_dict.update(returned\_val)

if \_\_name\_\_ == "\_\_main\_\_":

# Call the `main` function only if this file is the one being executed.

# Allows other files to import from it without circular imports.

main()