```
#!/usr/local/bin/python
#clicker data munging, fall 2014

import csv

"""change to the appropriate path and filenames.  note, all files must be csv's
with commas not tabs or other delimiters"""
path = "/Users/bottk/Desktop/clicker_py/"

"""modify these field names to match the fields in the csvs"""
#file with all student records
infile = "class_list_all.csv"
firstField = "first_name"#csv header with first name
lastField = "last_name"#csv header with last name
courseField = "course"
sectionField = "section" #csv header with section ID
ID1 = "reed_id" #ID jo join to infile1
ID2 = "email" #ID to join to infile2

#most reliable file with clicker IDs
infile1 ="clickers_bookstore2.csv"
#firstField1 = ""#csv header with first name
#lastField1 = ""#csv header with last name
fullName1 = "fullname"
clickerField1 = "clickerID"#csv header with clicker ID
joinID1 = "reedID"

#least reliable file with clicker IDs.  this will only be used if clicker wasnt found
in inFile1
infile2 ="self_report_all.csv"
#firstField2 = ""#csv header with first name
#lastField2 = ""#csv header with last name
fullName2 = "Your name"
clickerField2 = "Your clicker #" #csv header with clicker ID
joinID2 = "Username"

#file to write errors to this file will have records that dont
#have an exact match in the "infile"  which is treated as a master list
errFile = "errors.csv"
outfile = "student_join.csv"

#list to store errors in
errList = []
#dictionary to store data in
dataDict = {}
rosetta = {}
#open class list file
inDict = csv.DictReader(open(path+infile, "rU"))
#iterate through class list and store email and reedID in rosetta
#and firstname, lastname, course, and section in main data dictionary for output
for row in inDict:
    #rosetta stores the 2 table keys, email ID and Reed ID.  this can be used to
```

```
    #find a reed ID for any email ID.  this is necessary since the main dataset is
keyed
    #by reedID and it is not easy to search by values(dicts are collection of key:
value pairs)
    rosetta[row[ID2]] = row[ID1]
    dataDict[row[ID1]] = {'first' : row[firstField],
                          'last' : row[lastField],
                          'course' : row[courseField],
                          'section' : row[sectionField],
                          'clicker' : ''}

#open second file
inDict1 = csv.DictReader(open(path+infile1, "rU"))
for row in inDict1:
    #check to see if reedID is in the data, if not output to the error file
    #why does the bookstore have a sold clicker but person isnt in class list?
    if row[joinID1] in dataDict:
        dataDict[row[joinID1]]['clicker'] = row[clickerField1]
    else:
        errList.append([row[fullName1], row[joinID1], row[clickerField1], infile1])
#open third file
inDict2 = csv.DictReader(open(path+infile2, "rU"))

for row in inDict2:
    #check if the ID has a match in the datadict(checking vs rosetta so i dont get
    #an error, but since rosetta is created from the master file and datadict is
    #created from the same it is the same thing.  use rosetta to find reedID that
mathes
    #the given email address
    if row[joinID2] in rosetta:
        #then check if the clicker ID is already stored
        if dataDict[rosetta[row[joinID2]]]['clicker'] == '':
            #then store clickerID if it is empty
            dataDict[rosetta[row[joinID2]]]['clicker'] = row[clickerField2]
        #if email/reed ID not found then write details to error list
    else:
        errList.append([row[fullName2].lower(), row[joinID2],row[clickerField2],
infile2])

#open error file for output
fout = open(path+errFile, 'wb')
errWriter = csv.writer(fout, delimiter=',',quotechar='"', quoting=csv.QUOTE_ALL)
#write header to error file
errWriter.writerow(('fullName', 'ReedIDorEmail','clicker','sourceFile'))
#iterate through errors list and write each row to error file
for item in errList:
    errWriter.writerow(item)
#close/write file
fout.close()

#open main file for output
fout2 = open(path+outfile, 'wb')
```

```
outWriter = csv.writer(fout2, delimiter=',',quotechar='"', quoting=csv.QUOTE_ALL)
#write header
outWriter.writerow(('reedID','first', 'last', 'course', 'section', 'clicker'))
#iterate through data and write data in a certain order.
for item in dataDict:
    #build tuple of data values to be written
    wrow = (item,
dataDict[item]['first'],dataDict[item]['last'],dataDict[item]['course'],dataDict[item]
['section'],dataDict[item]['clicker'])
    outWriter.writerow(wrow)
#close/write file
fout2.close()
```