Python Challenge - PyPoll

Reference

Source Code

Unix Script - Showing running main.py from bash shell

Reference

GitHub

Source Code

```
# John M Warlop
# main.py
# UCSD Data Visualization Bootcamp
# Homework #3 - Python
# Challenge: PyPoll
import os
import re
import time
import csv
#
```

```
#
def preface():
    print("This program should be run in a directory where there is a directory called 'raw_data'.")
    print("In the raw_data directory should be the election results files.")
    print("The name of these files should be in the form of 'election_results_###.csv'.")
    print("The file type is csv(comma separated values). The ### denote the number of the file, ")
    print("for example:  'election_results_001.csv' or 'election_results_101.csv'.")
    print("The number should only be digits and there should be no two files with the same digits")
    print("Each file has one header row 'Voter ID', 'Country', 'Candidate'.  Following header row")
    print("each record has three fields of type: integer, text, text. For each record, only two comma's can")
    print("on each line.")
#
#
def askQ():
    print("Enter (y)es or (n)o")
    a1=input("Is this python script in a directory that contains a directory named 'raw_data'?")
    a2=input("Are election result files contained in the 'raw_data' directory?")
    a3=input("Are the election result files in the formate of: 'election_data_###.csv'?")
    if (a1,a2,a3) == ('y','y','y') or (a1,a2,a3)==('Y','Y','Y'):
        return True
    else:
        return False
#
#
```

```
def log_file(out_dir):
    print('x')









#
#
def get_file_paths(): #returns a list with full file paths to each election results file
    full_paths = []
    raw_data_path = os.path.join(os.getcwd(),'raw_data')
    flist = os.listdir(raw_data_path)
    for fname in flist:
        if re.match(r'election_data_\d+\.csv',fname):
            full_path = os.path.join(raw_data_path,fname)
            full_paths.append(full_path)
    return(full_paths)
#
#
def raw_data(): #check for raw data directory
    flist = os.listdir(os.getcwd())
    if not 'raw_data' in flist:
```

```
        print("raw_data directory not found")
        return False
    return True
#
#
def check_format(row):#Checks format of election_data_##.csv file
    if len(row) != 3:
        er_print("election_data_##.csv not properly formatted",False)
        return False
    return True
#
#
def er_print(msg,quit):
    print(msg)
    if quit:
        quit()

#
#
def info_print(msg,quit):
    print(msg)
    if quit:
        quit()
#
```

```
# Returns: A dictionary where keys are full file path. The value is a candidate_tally dictionary
def process_data(full_paths):#with open(udemy_csv, newline="") as csvfile:
    candidate_tally={}     #this is value of file_dict
    file_dict={}           #key is full path file name and value is candidate_tally
    for fname in full_paths:
        print("Processing file: "+fname)
        with open(fname,newline='') as csvfile:
            csvreader= csv.reader(csvfile, delimiter=",")
            c=0
            for row in csvreader:                          #Read each row/record
                if c == 0: #skip header row
                    c += 1
                    continue
                if not check_format(row):
                    csvfile.close()
                    quit()
                if not row[2] in candidate_tally:
                    candidate_tally[row[2]] = [1,row[0],row[1]] #Add 1st vote and voter ID and country
                else:
                    candidate_tally[row[2]][0] += 1 #Increment tally
                    candidate_tally[row[2]].append(row[0]) #Add Voter ID
                    candidate_tally[row[2]].append(row[1]) #Add County
            file_dict[fname] = candidate_tally #file_dict is dictionary of tally dictionaries
            candidate_tally = {}
```

```
        csvfile.close()
    return(file_dict)
#
#
def make_log_name(fname): #Makes log name of form election_data_##_20180213-134523.log
    timestr = time.strftime("%Y%m%d-%H%M%S") #time stamp
    m = re.search(r'(election_data_\d+\.csv)',fname)
    sum_fname = m.group(0)[0:-4]+"_"+timestr+".log"
    m = re.search(r'(.+raw_data/)',fname)
    sum_full_path = m.group(0)+sum_fname
    return(sum_full_path)
#
#
def write_summaries(file_dict):
    (total_votes,votes,results,name) =(0,0,[[]],'')
    timestr = time.strftime("%Y%m%d-%H%M%S")#time stamp for files and logs
    onames = [] #Output names
    for full_path in file_dict.keys():
        ofname = full_path[0:-4]+"_"+timestr+".txt"
        onames.append(ofname)
        for name in file_dict[full_path]:
            votes = file_dict[full_path][name][0]
            results.append(name)
            results.append(votes)
```

```
        total_votes = total_votes + votes
    results[0].append(total_votes) #results [ [ ],'name',votes,name,votes,END]
    total_votes = 0 #Reset for next file
    results.append('END')
results.append('FINAL')
oString = "\n\nElection Results\n\n----------------------\n"
i=1
j=0
winner = ['',0]
oString_list = []
while results[i] != 'FINAL' and len(results) >= 5:
    oString=oString+"Total Votes: "+str(results[0][j])+'\n'
    oString=oString+"----------------------\n"
    while results[i] != 'END':
        percent = (results[i+1]/results[0][j])*100.0
        if results[i+1] > winner[1]:
            winner[0]=results[i]
            winner[1]=results[i+1]
        oString=oString+"{0:<10}: {1:>3.2f}% ({2:d})\n".format(results[i],percent,results[i+1])
        i += 2
    oString=oString+"----------------------\n"
    oString=oString+"Winner: {0:s}\n".format(winner[0])+"----------------------\n\n\n\n"
    print(oString)
    oString_list.append(oString)
```

```
        oString = ''
        i += 1
        j += 1
        winner = ['',0]
    for name in onames:
        with open(name,"w") as f:
            f.write(oString_list.pop(0))


# insure_validity
# Description: Make sure program will run w/o error
def insure_validity(full_path):
    file_count = len(full_paths)
    if not file_count:
        er_print("No election_data_###.csv files found in raw_data directory",True)
    info_print("\n\n"+str(file_count)+" election_data_###.csv files found in raw_data directory",False)


# main
#
preface()
timestr = time.strftime("%Y%m%d-%H%M%S")#time stamp for files and logs
if not askQ(): quit()
if not raw_data(): quit()
```

```
full_paths = get_file_paths()
insure_validity(full_paths) #Make sure everything is coposetic before continueing
file_dict = process_data(full_paths)
write_summaries(file_dict)
```

Unix Script - Showing running main.py from bash shell

```
Xenon:PyPoll jwarlop$ cat typescript
Script started on Fri Mar  9 14:00:45 2018
bash-3.2$ ls
README.md       main.py         raw_data        typescript
bash-3.2$ ls raw_data
README.md                               election_data_1_20180309-134527.txt
election_data_2_20180309-134527.txt
election_data_1.csv                     election_data_2.csv
bash-3.2$ python main.py
This program should be run in a directory where there is a directory called 'raw_data'.
In the raw_data directory should be the election results files.
The name of these files should be in the form of 'election_results_###.csv'.
The file type is csv(comma separated values). The ### denote the number of the file,
for example:  'election_results_001.csv' or 'election_results_101.csv'.
The number should only be digits and there should be no two files with the same digits
Each file has one header row 'Voter ID', 'Country', 'Candidate'.  Following header row
```

```
each record has three fields of type: integer, text, text. For each record, only two comma's can
on each line.
Enter (y)es or (n)o
Is this python script in a directory that contains a directory named 'raw_data'?y
Are election result files contained in the 'raw_data' directory?y
Are the election result files in the formate of: 'election_data_###.csv'?y


2 election_data_###.csv files found in raw_data directory
Processing file: /Users/jwarlop/dev/DataViz/python-challenge/PyPoll/raw_data/election_data_1.csv
Processing file: /Users/jwarlop/dev/DataViz/python-challenge/PyPoll/raw_data/election_data_2.csv


Election Results

----------------------
Total Votes: 803000
----------------------
Vestal    : 48.00% (385440)
Torres    : 44.00% (353320)
Seth      : 5.00% (40150)
Cordin    : 3.00% (24090)
----------------------
Winner: Vestal
```

```
---------------------




Total Votes: 3521001
---------------------
Khan      : 63.00% (2218231)
Correy    : 20.00% (704200)
Li        : 14.00% (492940)
O'Tooley  : 3.00% (105630)
---------------------
Winner: Khan
---------------------




bash-3.2$ ls raw_data
README.md                               election_data_1_20180309-134527.txt     election_data_2.csv
election_data_2_20180309-140113.txt
election_data_1.csv                     election_data_1_20180309-140113.txt
election_data_2_20180309-134527.txt
bash-3.2$ cat raw_data/election_data_1_20180309-140113.txt
```

```
Election Results

---------------------
Total Votes: 803000
---------------------
Vestal    : 48.00% (385440)
Torres    : 44.00% (353320)
Seth      : 5.00% (40150)
Cordin    : 3.00% (24090)
---------------------
Winner: Vestal
---------------------



bash-3.2$ cat raw_data/election_data_2_20180309-140113.txt
Total Votes: 3521001
---------------------
Khan      : 63.00% (2218231)
Correy    : 20.00% (704200)
Li        : 14.00% (492940)
O'Tooley  : 3.00% (105630)
```

```
----------------------
Winner: Khan
----------------------



bash-3.2$ pwd
/Users/jwarlop/dev/DataViz/python-challenge/PyPoll
bash-3.2$ ls
README.md       main.py         raw_data        typescript
bash-3.2$ exit

Script done on Fri Mar  9 14:02:33 2018
```