Back

Practice Quiz: Reading & Writing CSV Files

Practice Quiz • 10 min • 5 total points

Congratulations! You passed!

Grade received 80% **To pass** 80% or higher

Go to next item

1. We're working with a list of flowers and some information about each one. The create_file function writes this information to a CSV file. The contents_of_file function reads this file into records and returns the information in a nicely formatted block. Fill in the gaps of the contents_of_file function to turn the data in the CSV file into a dictionary using DictReader.

0 / 1 point

```python
import os
import csv

# Create a file with data in it
def create_file(filename):
  with open(filename, "w") as file:
    file.write("name,color,type\n")
    file.write("carnation,pink,annual\n")
    file.write("daffodil,yellow,perennial\n")
    file.write("iris,blue,perennial\n")
    file.write("poinsettia,red,perennial\n")
    file.write("sunflower,yellow,annual\n")

# Read the file contents and format the information about each row
def contents_of_file(filename):
  return_string = ""

  # Call the function to create the file
  create_file(filename)

  # Open the file
  with open(filename,'r') as csv_file:
    # Read the rows of the file into a dictionary
    reader = csv.DictReader(csv_file)
    # Process each item of the dictionary
    for row in reader:
      return_string += "a {} {} is {}\n".format(row["color"], row["name"], row["type"])
  return return_string

#Call the function
print(contents_of_file("flowers.csv"))
```

Run

Reset

```
a pink carnation is annual
a yellow daffodil is perennial
a blue iris is perennial
a red poinsettia is perennial
a yellow sunflower is annual
```

Incorrect

Something went wrong! Contact Coursera Support about this question!

2. Using the CSV file of flowers again, fill in the gaps of the contents_of_file function to process the data without turning it into a dictionary. How do you skip over the header record with the field names?

1 / 1 point

```python
import os
import csv

# Create a file with data in it
def create_file(filename):
  with open(filename, "w") as file:
    file.write("name,color,type\n")
    file.write("carnation,pink,annual\n")
    file.write("daffodil,yellow,perennial\n")
    file.write("iris,blue,perennial\n")
    file.write("poinsettia,red,perennial\n")
    file.write("sunflower,yellow,annual\n")

# Read the file contents and format the information about each row
def contents_of_file(filename):
  return_string = ""

  # Call the function to create the file
  create_file(filename)

  # Open the file
  with open(filename) as csv_file:
    # Read the rows of the file
    rows = csv.reader(csv_file)
    header = next(rows)
    # Process each row
    for row in rows:
      name, color, type = row
      # Format the return string for data rows only

      return_string += "a {} {} is {}\n".format(color,name,type)
  return return_string

#Call the function
print(contents_of_file("flowers.csv"))
```

Run

Reset

```
a pink carnation is annual
a yellow daffodil is perennial
a blue iris is perennial
a red poinsettia is perennial
a yellow sunflower is annual
```

Correct

You nailed it! Everything's coming up roses (pardon the pun!)

3. In order to use the writerows() function of DictWriter() to write a list of dictionaries to each line of a CSV file, what steps should we take? (Check all that apply) **1 / 1 point**

☑ Create an instance of the DictWriter() class

Correct
Excellent! We have to create a DictWriter() object instance to work with, and pass to it the fieldnames parameter defined as a list of keys.

☑ Write the fieldnames parameter into the first row using writeheader()

Correct
Nice work! The non-optional fieldnames parameter list values should be written to the first row.

☑ Open the csv file using *with open*

Correct
Good call! The CSV file has to be open before we can write to it.

☐ Import the OS module

4. Which of the following is true about unpacking values into variables when reading rows of a CSV file? (Check all that apply) **1 / 1 point**

☑ We need the same amount of variables as there are columns of data in the CSV

Correct
Awesome! We need to have the exact same amount of variables on the left side of the equals sign as the length of the sequence on the right side when unpacking rows into individual variables.

☑ Rows can be read using both csv.reader and csv.DictReader

Correct
Right on! Although they read the CSV rows into different datatypes, both csv.reader or csv.DictReader can be used to parse CSV files.

☑ An instance of the reader class must be created first

Correct
Nice job! We have to create an instance of the reader class we are using before we can parse the CSV file.

☐ The CSV file does not have to be explicitly opened

5. If we are analyzing a file's contents to correctly structure its data, what action are we performing on the file? **1 / 1 point**

- ○ Writing
- ○ Appending
- [✓ Parsing]
- ○ Reading

✓ **Correct**
Great work! Parsing a file means analyzing its contents to correctly structure the data. As long as we know what the data is, we can organize it in a way our script can use effectively.