

Practice Quiz: Reading & Writing CSV Files

PUNTOS TOTALES DE 5

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 puntos

```
1 import os
2 import csv
3
4 # Create a file with data in it
5 def create_file(filename):
6     with open(filename, "w") as file:
7         file.write("name,color,type\n")
8         file.write("carnation,pink,annual\n")
9         file.write("daffodil,yellow,perennial\n")
10        file.write("iris,blue,perennial\n")
11        file.write("poinsettia,red,perennial\n")
12        file.write("sunflower,yellow,annual\n")
13
14 # Read the file contents and format the information about each row
15 def contents_of_file(filename):
16     return_string = ""
17
18     # Call the function to create the file
19     create_file(filename)
20
21     # Open the file
22     with open(filename) as file:
23         # Read the rows of the file into a dictionary
24         f = csv.DictReader(file)
25         # Process each item of the dictionary
26         for row in f:
27             return_string += "a {} {} is {}\n".format(row["color"], row["name"], row["type"])
28     return return_string
29
30 #Call the function
31 print(contents_of_file("flowers.csv"))
```

Ejecutar

Restablecer

```
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
```

! Incorrecto

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 puntos

```
1  import os
2  import csv
3
4  # Create a file with data in it
5  def create_file(filename):
6      with open(filename, "w") as file:
7          file.write("name,color,type\n")
8          file.write("carnation,pink,annual\n")
9          file.write("daffodil,yellow,perennial\n")
10         file.write("iris,blue,perennial\n")
11         file.write("poinsettia,red,perennial\n")
12         file.write("sunflower,yellow,annual\n")
13
14     # Read the file contents and format the information about each row
15     def contents_of_file(filename):
16         return_string = ""
17
18         # Call the function to create the file
19         create_file(filename)
20
21         # Open the file
22         with open(filename) as file:
23             # discard/ignore header row
24             file.readline()
25             # Read the rows of the file
26             rows = csv.reader(file)
27             # Process each row
28             for row in rows:
29                 name,color,type = row
30                 # Format the return string for data rows only
31                 return_string += "a {} {} is {}\n".format(color, name, type)
32         return return_string
33
34     #Call the function
35     print(contents_of_file("flowers.csv"))
36
```

[Ejecutar](#)[Restablecer](#)

```
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
```

**Correcto**

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 puntos**

☒ Create an instance of the `DictWriter()` class

✓ **Correcto**

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()`

✓ **Correcto**

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

☒ Open the csv file using *with open*

✓ **Correcto**

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 puntos**

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

✓ **Correcto**

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`

✓ **Correcto**

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



Correcto

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 puntos

☐ Writing

☐ Appending

☒ Parsing

☐ Reading



Correcto

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.