TASK

IO Operations - Output

Visit our website

Introduction

WELCOME TO THE IO OPERATIONS - OUTPUT TASK!

Until now, the Python code you've been writing has only received input in one manner and has only displayed output in one way – you type input using the keyboard and its results are displayed on the console. But what if you want to read information from a file on your computer and write that information to another file? This process is called file **I/O** (the "I/O" stands for "input/output") and Python has some built-in functions that handle this for you.

A note from the
Hyperion Team

Explore the following **five types of software development** to gain insight into different tech roles.

1. Front-End Web Development

2. Product Management

3. Site Reliability Engineer

4. Machine Learning Developer

5. Natural Language Processing Developer

WRITING DATA TO A TEXT FILE

First of all, let's recap file access modes. Last time we focussed on reading, and this time we are going to focus on writing.

Mode	Description
r	Opens the file for reading only. Throws I/O error if the file does not exist.
r+	Opens the file for both reading and writing. Throws I/O error if the file does not exist.
w	Opens the file for writing only. It will create a file if the file

	doesn't exist. For the existing files, the data is overwritten.
w+	Opens the file for reading and writing. It will create a file if the file doesn't exist. For the existing files, the data is overwritten.
a	Opens the file for writing only, creating the file if it doesn't exist. Any data being written will be inserted at the end (appended, thus **a**), after the existing data.
a+	Opens the file for reading and writing, creating the file if it isn't there. Any data being written will be inserted at the end (appended, thus **a**), after the existing data.

Now, let's see how to create a new text file and write data to it using the **w** access mode:

```
name = input("Enter name: ")

with open('output.txt', 'w') as f:
    f.write(name+"\n")
```

We create a new file called **output.txt** (it doesn't exist yet) in write mode. Python will automatically create this file in the directory/folder that our program is in. We ask the user for their name. When they enter it, it is stored as a string in the variable called **name**. You then use the **write()** method in order to write to a file. The final line of code above will write the string value stored in the variable called **name** and a newline (**\n**) to the file that has been opened.

You must run this Python file for the file **output.txt** to be created with the output generated by this program in it.

We can write to the file again, and the current contents of the file will not be overwritten. Instead, it will be written on the 2nd line of the text file:

```
f.write("My name is on the line above in this text file.")
```

However, if you open the file again, the existing contents will be overwritten. This is important to remember when editing files.

Don't forget to close the file if you're not using **with**/**as**! (For this reason, it is generally better to stick to using **with**/**as**.)

```
open_file.close()
```

Instructions

Read and run the accompanying example file(s) provided before doing the practical task to become more comfortable with the concepts covered in this task.

Practical Task

Follow these steps:

- Create a file called **student_register.py**
- Write a program that allows a user to register students for an exam venue.
- First, ask the user how many students are registering.
- Create a *for loop* that runs for that number of students.
- Each time the loop runs the program should ask the user to enter the next student ID number.
- Write each of the ID numbers to a text file called **reg_form.txt**
- Include a dotted line after each student ID because this document will be used as an attendance register, which the students will sign when they arrive at the exam venue.

Rate us
Share your thoughts

Hyperion strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved, or think we've done a good job?

Click here to share your thoughts anonymously.