Lesson 4: Writing to Text Files

In this lesson, students will learn how to write and append to text files using Python. The lesson begins with a bell-ringer activity where students reflect on the power of words and the impact of writing. The teacher then introduces the concept of text files and their importance in storing and retrieving data. Through direct instruction, students will learn how to open a text file in write mode, write content to the file, and close the file. Guided practice exercises and challenges will allow students to apply their knowledge in pairs or small groups. For independent practice, students will individually write a short story or personal reflection in a new text file. The lesson concludes with an exit ticket to assess students' understanding and a recap of the main points covered.

## **Materials:**

- Computers or laptops with text editors

- Internet access

- Handouts with exercises and challenges

## **Objectives:**

By the end of this lesson, students will be able to:

- Write to text files using Python

- Append to text files using Python

## **Bell-Ringer Activity (5 minutes):**

1. Display a quote on the board: "The pen is mightier than the sword."

2. Ask students to reflect on the quote and write a short paragraph explaining what it means to them. Encourage them to think about the power of words and how writing can have a lasting impact.

## **Introduction (10 minutes):**

1. Begin by reviewing the concept of text files and their importance in storing and retrieving data.

2. Explain that in this lesson, students will learn how to write and append to text files using Python.

3. Discuss the significance of being able to write and append to text files, such as creating logs, saving user input, or updating existing files.

4. Emphasize the importance of clear and coherent writing when working with text files.

## **Direct Instruction (20 minutes):**

1. Demonstrate how to open a text file in write mode using Python's `open()` function.

2. Show students how to write content to the file using the `write()` method.

3. Explain the difference between writing to a new file and overwriting an existing file.

4. Provide examples of different scenarios where writing to a text file can be useful.

5. Demonstrate how to close the file after writing to it.

## **Guided Practice (20 minutes):**

1. Divide students into pairs or small groups.

2. Distribute handouts with exercises and challenges related to writing to text files.

3. Instruct students to work together to complete the exercises, using Python to write to text files.

4. Circulate the classroom to provide assistance and answer any questions.

## **Independent Practice (25 minutes):**

1. Assign a writing task to each student individually.

2. Instruct students to create a new text file and write a short story or a personal reflection.

3. Encourage students to use descriptive language and demonstrate their understanding of writing to text files.

4. Remind students to save and close their files after completing the task.

## **Exit Ticket (5 minutes):**

1. Ask students to write a brief summary of what they have learned about writing to text files.

2. Collect the exit tickets to assess students' understanding of the topic.

## **Closure (5 minutes):**

1. Recap the main points covered in the lesson, emphasizing the importance of clear and coherent writing when working with text files.

2. Encourage students to continue practicing their skills in writing to text files and explore other ways to utilize this knowledge.

3. Thank the students for their participation and remind them of the next lesson's topic.

## **Common Core Standards:**

- CCSS.ELA-LITERACY.W.9-10.4: Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.

- CCSS.ELA-LITERACY.W.9-10.5: Develop and strengthen writing as needed by planning, revising, editing, rewriting, or trying a new approach.