Lesson 8: List Methods and Functions

In this lesson, students will learn about advanced list methods and functions in Python. They will explore how to traverse a list using a for loop, and they will learn how to use list methods such as append(), insert(), remove(), pop(), and sort(). Additionally, students will have the opportunity to practice creating their own functions that return lists. The lesson will include a bell-ringer activity, direct instruction, guided practice, independent practice, and an exit ticket to assess student understanding.

## **Objectives:**

- Students will be able to traverse a list of elements.

- Students will be able to use list methods to manipulate and modify lists.

- Students will be able to create functions that return lists.

- Students will be able to import and use custom-built functions.

## **Materials:**

- Computers with Python IDE installed

- Projector or whiteboard

- Deck of cards (optional)

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

1. Display the following code on the projector or whiteboard:

```python

numbers = [1, 2, 3, 4, 5]

for num in numbers:

print(num)

```

2. Ask the students to predict the output of the code.

3. Allow a few students to share their predictions.

4. Run the code and discuss the output as a class.

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

1. Explain to the students that in the previous lesson, they learned about basic list operations and how to access elements in a list.

2. Inform them that in this lesson, they will explore more advanced list methods and learn how to traverse a list using a for loop.

3. Emphasize that these concepts are important for manipulating and working with lists in programming.

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

1. Display the following code on the projector or whiteboard:

```python

fruits = ['apple', 'banana', 'orange']

fruits.append('grape')

print(fruits)

```

2. Explain that the `append()` method is used to add an element to the end of a list.

3. Run the code and discuss the output as a class.

4. Display the following code:

```python

numbers = [1, 2, 3, 4, 5]

numbers.insert(2, 10)

print(numbers)

```

5. Explain that the `insert()` method is used to insert an element at a specific index in a list.

6. Run the code and discuss the output as a class.

7. Display the following code:

```python

fruits = ['apple', 'banana', 'orange']

fruits.remove('banana')

print(fruits)

```

8. Explain that the `remove()` method is used to remove a specific element from a list.

9. Run the code and discuss the output as a class.

10. Display the following code:

```python

numbers = [1, 2, 3, 4, 5]

numbers.pop()

print(numbers)

```

11. Explain that the `pop()` method is used to remove the last element from a list.

12. Run the code and discuss the output as a class.

13. Display the following code:

```python

numbers = [1, 2, 3, 4, 5]

numbers.sort()

print(numbers)

```

14. Explain that the `sort()` method is used to sort the elements in a list in ascending order.

15. Run the code and discuss the output as a class.

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

1. Divide the students into pairs.

2. Instruct each pair to create a deck of cards using a list.

3. Each card should be represented as a string with the format "rank of suit" (e.g., "Ace of Spades").

4. Once the decks are created, instruct the pairs to perform the following list operations and methods on their decks:

- Append a new card to the deck.

- Insert a card at a specific index.

- Remove a specific card from the deck.

- Sort the deck in ascending order.

- Print the final deck.

5. Circulate the room to provide assistance and guidance as needed.

## **Independent Practice (30 minutes):**

1. Instruct the students to individually create a function that takes a list of numbers as input and returns a new list containing only the even numbers from the input list.

2. Encourage the students to use a for loop and the modulus operator (%) to determine if a number is even.

3. Once the students have completed their functions, instruct them to test their functions by passing different lists of numbers as input.

4. Circulate the room to provide assistance and guidance as needed.

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

1. Distribute the exit ticket to the students.

2. Instruct them to answer the questions on the exit ticket individually.

3. Collect the exit tickets before the end of the class.

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

1. Review the main concepts covered in the lesson: list methods, traversing a list using a for loop, and creating functions that return lists.

2. Ask the students to share one thing they learned or found interesting during the lesson.

3. Address any remaining questions or concerns from the students.

4. Remind the students to practice using list methods and functions on their own to reinforce their understanding.

5. Thank the students for their participation and effort in the lesson.

**Common Core Standards:**

- None applicable for Computer Science