Python Basics: List and Nested List

Lists in Python

In Python, a list is a versatile and mutable collection of elements. Lists are defined by square brackets (`[]`) and can contain items of different data types, such as integers, floats, strings, or even other lists.

Creating Lists

```
# Creating an empty list
empty_list = []

# Creating a list with elements
numbers = [1, 2, 3, 4, 5]
names = ['Alice', 'Bob', 'Charlie']

# Lists can contain mixed data types
mixed_list = [1, 'hello', 3.14, True]
```

Accessing Elements

You can access elements in a list using index notation. Indexing starts from 0 for the first element.

```
first_element = numbers[0]
second_element = names[1]
```

Modifying Lists

Lists are mutable, meaning you can change their elements after creation.

```
# Modifying an element
numbers[0] = 10

# Adding an element to the end of the list
numbers.append(6)

# Removing an element by value
names.remove('Bob')

# Removing an element by index
del numbers[2]
```

Nested Lists

A nested list is a list within another list. This allows for the creation of more complex data structures.

```
nested_list = [[1, 2, 3], ['a', 'b', 'c'], [True, False]]
```

Accessing elements in a nested list involves using multiple indices.

```
first_element = nested_list[0][0]  # Accessing the first element of the first inner list
```

Data Analysis and Pandas

Introduction to Data Analysis

Data analysis involves inspecting, cleaning, transforming, and modeling data to discover useful information, draw conclusions, and support decision-making.

Pandas Library

Pandas is a powerful open-source data analysis and manipulation library for Python. It provides data structures like Series and DataFrame, along with a wide range of functions for efficiently manipulating large datasets.

Installing Pandas

You can install Pandas using:

```
pip install pandas
```

Basic Pandas Concepts

Series

A one-dimensional labeled array that can hold any data type.

```
import pandas as pd

# Creating a Series
s = pd.Series([1, 3, 5, np.nan, 6, 8])
```

DataFrame

A two-dimensional table with rows and columns, similar to a spreadsheet

```
# Creating a DataFrame from a dictionary
data = {‘Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 35],
        'City': ['New York', 'San Francisco', 'Los Angeles']}

df = pd.DataFrame(data)

# Creating a DataFrame from a dictionary
data = {‘Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 35],
        'City': ['New York', 'San Francisco', 'Los Angeles']}
```

```
df = pd.DataFrame(data)
```

Reading and Writing Data

Pandas supports reading data from various file formats, including CSV, Excel, and SQL databases.

```
# Reading a CSV file
df = pd.read_csv('data.csv')

# Writing to a CSV file
df.to_csv('output.csv', index=False)
```

These are the basics of working with lists in Python and performing data analysis using the Pandas library. As you delve deeper into these topics, you'll discover more advanced techniques and functionalities to enhance your programming and data analysis skills.