

Lecture 1.4

Python Arrays and List (Review)

Agenda

- 1. Introduction to Arrays in Python
- 2. The array module
- 3. Introduction to numpy
- 4. Operations on Arrays
- 5. Arrays vs Lists
- 6. Practice Exercises

Recap: Python Lists

- Lists are dynamic, ordered collections.
- They can hold heterogeneous data (mix of int, string, etc.).
- Highly versatile but can be memory intensive.
- Operations: append, insert, pop, remove.

Introduction to Arrays in Python

- Unlike C/C++, Python does not have a built-in static Array data structure.
- Python lists are often used as arrays, but they are not strictly arrays.
- An array is a collection of items stored at contiguous memory locations.
- Items in a strict array must be of the same data type.

The 'array' Module

- Python provides an 'array' module to create C-style arrays.
- Arrays require all elements to be of the same type.
- Requires importing the module: 'import array'.
- Defined by a typecode (e.g., 'i' for integers, 'f' for floats).

Example: Using the array Module

- `import array as arr`
- `a = arr.array('i', [1, 2, 3])`
- `print(a)` # Output: `array('i', [1, 2, 3])`

Basic Operations on Arrays (array module)

- Accessing: `'a[0]'` returns the first element.
- Insertion: `'a.insert(1, 4)'` inserts 4 at index 1.
- Appending: `'a.append(5)'` adds 5 to the end.
- Removal: `'a.remove(2)'` removes the first occurrence of 2.

Why use NumPy?

- The standard array module is quite limited.
- NumPy (Numerical Python) is the standard for numerical arrays in Python.
- Provides multidimensional array objects (ndarrays).
- Offers fast mathematical operations on arrays.

Importing NumPy

- NumPy is a third-party library, typically installed via pip.
- Command: `'pip install numpy'`
- Standard import alias: `'import numpy as np'`
- Most Python data scientists use `'np'` universally.

Creating NumPy Arrays

- `import numpy as np`
- `arr = np.array([1, 2, 3, 4, 5])`
- `print(arr)`
- `print(type(arr))` # Output: `class 'numpy.ndarray'`

NumPy Multidimensional Arrays

- NumPy easily supports 2D, 3D, and n-dimensional arrays.
- Example of 2D: `'np.array([[1, 2], [3, 4]])'`
- Useful for matrices and images.

Operations on NumPy Arrays

- Element-wise addition: `'arr + 2'` adds 2 to every element.
- Element-wise multiplication: `'arr * 2'`.
- Array addition: `'arr1 + arr2'` adds corresponding elements.
- These operations are vectorized and extremely fast.

Common NumPy Array Methods

- `arr.shape`: Returns a tuple representing dimensions.
- `arr.size`: Total number of elements.
- `arr.mean()`: Computes the average.
- `arr.max()`: Finds the maximum value.

Arrays vs Lists: The Core Differences

- Data Types: Lists hold heterogeneous data; Arrays hold homogeneous data.
- Memory: Arrays are more memory-efficient.
- Performance: Operations on NumPy arrays are much faster.
- Flexibility: Lists can easily grow and shrink in size.

Arrays vs Lists: Comparison Table

- Feature — List — NumPy Array
- Data Type — Heterogeneous — Homogeneous
- Memory — High overhead — Contiguous, low overhead
- Math Ops — Requires loops — Built-in element-wise (fast)
- Size — Dynamic — Fixed size upon creation

When to Use Which?

- Use Lists when: Dealing with mixed data types, frequently adding/removing items.
- Use array module when: Need memory efficiency for basic data but no math operations.
- Use NumPy when: Doing numerical analysis, matrices, machine learning, or processing large datasets.

Practice Exercise 1

- Question: Create a NumPy array with values 10 to 50.
- Hint: Use `np.arange()`
- Solution:
- `arr = np.arange(10, 51)`

Practice Exercise 2

- Question: Reverse the array created in Exercise 1.
- Solution:
- `reversed_arr = arr[::-1]`

Summary

- Python does not have built-in strict arrays.
- The 'array' module provides basic C-style arrays.
- NumPy provides powerful, efficient multi-dimensional arrays.
- NumPy arrays are the standard for numerical operations.
- Lists are flexible but arrays are faster for math.

Next Lecture Preview

- Unit 2: Basics of Object-Oriented Programming
- Classes and Objects
- Constructors in Python
- Methods: Instance, Class, Static