

Tóm tắt mô-đun 4: Làm việc với dữ liệu bằng Python

Chúc mừng! Bạn đã hoàn thành mô-đun này. Tại thời điểm này, bạn biết rằng:

- Python sử dụng hàm `open()` và cho phép bạn đọc và ghi tệp, cung cấp quyền truy cập vào nội dung trong tệp để đọc. Nó cũng cho phép ghi đè lên nó để ghi và chỉ định chế độ tệp (ví dụ: `r` để đọc, `w` để viết, `a` để nối thêm).
 - Để đọc một tệp, Python sử dụng một hàm mở cùng với `r`.
 - Python sử dụng hàm **open with** để đọc và xử lý một thuộc tính file, nghĩa là từ `open` đến `close`.
 - Trong Python, bạn sử dụng phương thức **mở** để chỉnh sửa hoặc ghi đè lên một tệp.
 - Để viết một tệp, Python sử dụng hàm **mở** cùng với `w`.
 - Trong Python, "`a`" chỉ ra rằng chương trình đã gắn vào tệp.
 - Trong Python, "`\n`" biểu thị rằng mã sẽ bắt đầu trên một dòng mới.
 - Python sử dụng nhiều phương pháp khác nhau để in các dòng từ các thuộc tính.
- Pandas là một thư viện Python mạnh mẽ để thao tác và phân tích dữ liệu, cung cấp các cấu trúc và chức năng dữ liệu để làm việc với dữ liệu có cấu trúc như khung dữ liệu và chuỗi.
 - Bạn nhập tệp (gấu trúc) bằng cách sử dụng lệnh nhập theo sau là tên tệp.
 - Trong Python, bạn sử dụng lệnh **as** để cung cấp tên ngắn hơn cho tệp.
 - Trong Pandas, bạn sử dụng khung dữ liệu (`df`) để chỉ định các tệp để đọc.
 - DataFrames bao gồm các hàng và cột.
 - Bạn có thể tạo DataFrame mới bằng cách sử dụng cột hoặc cột của một DataFrame cụ thể.
 - Chúng tôi có thể làm việc với dữ liệu trong DataFrames và lưu kết quả ở các định dạng khác nhau.
 - Trong Python, bạn sử dụng phương pháp **Unique** để xác định các phần tử duy nhất trong một cột của DataFrames.
 - Bạn sử dụng toán tử bất đẳng thức cùng với `df` để gán giá trị Boolean cho cột đã chọn trong DataFrames.
 - Bạn lưu khung dữ liệu mới dưới dạng khung dữ liệu khác, có thể chứa các giá trị từ khung dữ liệu cũ hơn.
- NumPy là một thư viện Python cho các hoạt động số và ma trận, cung cấp các đối tượng mảng đa chiều và nhiều hàm toán học để làm việc với dữ liệu một cách hiệu quả.
 - NumPy là cơ sở cho Pandas.
 - Một mảng NumPy hoặc mảng ND tương tự như một danh sách, thường có kích thước cố định với cùng một loại phần tử.
- Mảng NumPy một chiều là một chuỗi các phần tử tuyến tính có trục đơn, giống như danh sách truyền thống, nhưng được tối ưu hóa cho các tính toán số và hoạt động mảng.
 - Bạn có thể truy cập các phần tử trong NumPy bằng cách sử dụng chỉ mục.
 - Bạn sử dụng thuộc tính **dtype** để lấy kiểu dữ liệu của các phần tử mảng.
 - Bạn sử dụng **ndsize** và **ndim** để có được kích thước và kích thước của mảng, tương ứng.
 - Bạn có thể sử dụng các phương pháp lặp chỉ mục và cắt lát trong NumPy.
 - Bổ sung vector là các hoạt động được sử dụng rộng rãi trong Python.
 - Biểu diễn phép cộng vector với các đoạn thẳng hoặc mũi tên là hữu ích.
 - Mã NumPy hoạt động nhanh hơn nhiều, rất hữu ích với nhiều dữ liệu.
 - Bạn thực hiện phép trừ vector bằng cách thay thế dấu cộng bằng dấu âm.
 - Nhân một mảng với vô hướng trong Python đòi hỏi phải nhân mỗi phần tử của mảng với giá trị vô hướng, dẫn đến một mảng mới trong đó mỗi phần tử chia tỷ lệ cho vô hướng.
 - Tích Hadamard đề cập đến phép nhân phần tử khôn ngoan của hai mảng có cùng hình dạng, dẫn đến một mảng mới trong đó mỗi phần tử là tích của các phần tử tương ứng trong mảng đầu vào.
 - Tích chấm trong Python là tổng các tích phần tử khôn ngoan của hai mảng, thường được sử dụng cho các phép toán vector và ma trận để tìm kết quả vô hướng của việc nhân các phần tử tương ứng và tính tổng chúng.
 - Khi làm việc với NumPy, người ta thường sử dụng các thư viện như Matplotlib để tạo đồ thị và trực quan hóa từ dữ liệu số được lưu trữ trong mảng NumPy.
- Mảng NumPy hai chiều là một cấu trúc giống như lưới với các hàng và cột phù hợp để biểu diễn dữ liệu dưới dạng ma trận hoặc bảng để tính toán số.
 - Trong NumPy, "hình dạng" đề cập đến kích thước của một mảng (số hàng và cột), cho biết kích thước và cấu trúc của nó.
 - Bạn sử dụng thuộc tính "size" để có được kích thước của một mảng.
 - Bạn sử dụng các thuộc tính hình chữ nhật để truy cập các phần tử khác nhau trong một mảng.
 - Bạn sử dụng vô hướng để nhân các phần tử trong NumPy.

Chuyển đến mục tiếp theo

✓ Hoàn thành