

<p>Python là gì? Liệt kê một số lợi ích của Python.</p> <p>Python là một ngôn ngữ lập trình bậc cao, thông dịch và đa mục đích. Nó nổi tiếng với cú pháp đơn giản và dễ đọc, giúp việc học và sử dụng trở nên dễ dàng. Một số lợi ích của Python bao gồm:</p> <ul style="list-style-type: none"> - Dễ học và dễ đọc – cú pháp rõ ràng, gần giống ngôn ngữ tự nhiên. - Đa năng và chạy trên nhiều nền tảng – Windows, macOS, Linux, v.v. - Thư viện chuẩn và hệ sinh thái lớn – nhiều module tích hợp sẵn và gói bên thứ ba. - Hỗ trợ nhiều mô hình lập trình – lập trình thủ tục, hướng đối tượng, và hàm. - Cộng đồng mạnh mẽ – nhiều hướng dẫn, diễn đàn và tài nguyên học tập.
<p>Bạn có thể cho biết Python thuộc lập trình hướng đối tượng hay lập trình hàm không?</p> <p>Python là một ngôn ngữ lập trình đa mô hình, nghĩa là nó hỗ trợ cả lập trình hướng đối tượng (OOP) và lập trình hàm.</p> <ul style="list-style-type: none"> - Lập trình hướng đối tượng: Python cho phép bạn định nghĩa class, tạo object, và sử dụng kế thừa, đa hình. - Lập trình hàm: Python hỗ trợ hàm là đối tượng cấp 1, hàm bậc cao, map, filter, reduce, và biểu thức lambda. <p>Vì vậy, Python vừa hướng đối tượng vừa hỗ trợ lập trình hàm, tùy cách bạn viết mã.</p>
<p>Quy tắc nào điều khiển biến cục bộ (local) và biến toàn cục (global) trong Python?</p> <p>Trong Python, phạm vi biến quyết định nơi một biến có thể được truy cập:</p> <ol style="list-style-type: none"> 1. Biến cục bộ (local) được định nghĩa trong một hàm và chỉ có thể sử dụng bên trong hàm đó. Chúng được tạo khi hàm bắt đầu và bị hủy khi hàm kết thúc. 2. Biến toàn cục (global) được định nghĩa bên ngoài hàm và có thể truy cập ở bất kỳ đâu trong file. 3. Để thay đổi biến toàn cục trong hàm, bạn phải dùng từ khóa global.
<p>Bạn có thể giải thích slicing trong Python là gì không?</p> <p>Slicing trong Python là cách lấy một phần của một chuỗi, list hoặc tuple bằng cách chỉ định vị trí bắt đầu, vị trí kết thúc và bước nhảy (tùy chọn).</p>
<p>Namespace trong Python là gì?</p> <p>Namespace trong Python là một vùng chứa các tên (biến, hàm, class) và ánh xạ chúng tới các đối tượng. Nói cách khác, namespace giúp Python quản lý tất cả các tên biến và đối tượng tương ứng trong bộ nhớ.</p> <p>Các loại namespace:</p> <ol style="list-style-type: none"> 1. Local namespace – bên trong một hàm. 2. Global namespace – ở mức module. 3. Built-in namespace – các hàm và exception có sẵn của Python.
<p>"pass" trong Python dùng để làm gì?</p> <p>Trong Python, pass là một câu lệnh rỗng, không thực hiện bất kỳ hành động nào. Nó được dùng như chỗ giữ chỗ khi cú pháp yêu cầu một câu lệnh nhưng bạn chưa muốn thực hiện gì.</p> <p>"pass" giúp code chạy bình thường mà không báo lỗi, ngay cả khi hàm hoặc lớp chưa có nội dung.</p>

<p>Bạn có thể giải thích unittest trong Python là gì không?</p> <p>"unittest" trong Python là một framework kiểm thử tích hợp sẵn, dùng để viết và chạy unit test — tức là kiểm tra từng phần nhỏ của chương trình (như hàm hoặc class) xem có hoạt động đúng không. Nó cung cấp các tính năng như test case, hàm setup/teardown, và các phương thức assert để kiểm tra kết quả mong đợi.</p> <p><i>"unittest là framework có sẵn của Python giúp kiểm tra từng phần nhỏ của chương trình hoạt động đúng hay không."</i></p>
<p>Chỉ số âm (negative indexes) trong Python là gì?</p> <p>Chỉ số âm (negative indexes) trong Python được dùng để truy cập các phần tử từ cuối của chuỗi, list hoặc tuple, thay vì từ đầu. Chỉ số -1 là phần tử cuối cùng, Chỉ số -2 là phần tử kế cuối, v.v. Tính năng này giúp truy cập phần tử cuối dễ dàng mà không cần biết độ dài danh sách.</p>
<p>What are ODBC modules in Python?</p> <p>ODBC (Open Database Connectivity) modules in Python are libraries that allow Python programs to connect and interact with databases using the ODBC standard. They provide a way to execute SQL queries, fetch data, and manage transactions from Python. Common Python ODBC modules:</p> <ol style="list-style-type: none"> 1. pyodbc – widely used for connecting to SQL Server, MySQL, Oracle, etc. 2. pypyodbc – pure Python alternative to pyodbc.
<p>Bạn sẽ gửi email từ một script Python như thế nào?</p> <p>Bạn có thể gửi email từ Python bằng module smtplib, cho phép kết nối tới SMTP server và gửi email. Thường kết hợp với module email để định dạng nội dung.</p>
<p>PEP 8 là gì và tầm quan trọng của nó là gì?</p> <p>PEP 8 là viết tắt của Python Enhancement Proposal 8. Nó là hướng dẫn chuẩn về phong cách viết code Python, quy định về cách đặt tên, căn lề, thụt lề, và nhiều yếu tố khác. Tầm quan trọng:</p> <ol style="list-style-type: none"> 1. Giúp code dễ đọc và đồng nhất trên các dự án. 2. Hỗ trợ làm việc nhóm nhờ tuân theo cùng một phong cách. 3. Tăng khả năng bảo trì và giảm lỗi. <p>Việc thụt lề, đặt tên và khoảng trắng đúng chuẩn giúp code dễ đọc hơn.</p>
<p>Những đặc điểm nổi bật (key features) của Python là gì?</p> <p>Python có nhiều đặc điểm nổi bật:</p> <ol style="list-style-type: none"> 1. Dễ học và dễ đọc – cú pháp đơn giản, gần gũi với ngôn ngữ tự nhiên. 2. Ngôn ngữ thông dịch – chạy code từng dòng. 3. Kiểu dữ liệu động – không cần khai báo kiểu biến. 4. Chạy đa nền tảng – Windows, macOS, Linux, v.v. 5. Thư viện chuẩn phong phú – nhiều module hỗ trợ các công việc khác nhau. 6. Hỗ trợ OOP và lập trình hàm – đa mô hình lập trình. 7. Mã nguồn mở – miễn phí, cộng đồng hỗ trợ mạnh.

Việc Python là ngôn ngữ dynamically typed có nghĩa là gì?

Python là dynamically typed có nghĩa là bạn không cần khai báo kiểu dữ liệu của biến. Trình thông dịch sẽ tự xác định kiểu tại thời gian chạy. Bạn cũng có thể thay đổi kiểu dữ liệu của biến trong quá trình thực thi.

Scope trong Python là gì?

Trong Python, scope (phạm vi biến) là khu vực trong code mà một biến có thể được truy cập. Nó quyết định biến nào có thể dùng ở đâu và biến nào được tham chiếu nếu có nhiều biến cùng tên.

Các loại scope:

1. Local scope – bên trong hàm.
2. Enclosing scope – trong hàm lồng nhau.
3. Global scope – ở mức module.
4. Built-in scope – các tên và hàm có sẵn của Python.

Làm thế nào để một script Python có thể thực thi (executable) trên hệ thống Unix?

Để một script Python có thể thực thi trên Unix:

1. Thêm shebang ở đầu file để chỉ định trình thông dịch Python:

```
#!/usr/bin/env python3
```

2. Cấp quyền thực thi cho file bằng lệnh chmod:

```
chmod +x script.py
```

3. Chạy script trực tiếp từ terminal:

```
./script.py
```

Điều này cho phép script chạy như một chương trình độc lập.

Docstring trong Python là gì?

Docstring trong Python là một chuỗi đặc biệt dùng để ghi chú (tài liệu) cho module, class hoặc function. Nó được viết bên trong dấu ba nháy (""" hoặc ''') ở đầu khối code và có thể truy cập qua thuộc tính `__doc__`.

`__init__` trong Python là gì?

Trong Python, `__init__` là phương thức đặc biệt (constructor) của một class, được gọi tự động khi một đối tượng của class được tạo. Nó dùng để khởi tạo các thuộc tính của đối tượng.

List và Tuple trong Python là gì?

Trong Python:

- List (danh sách) là tập hợp có thứ tự và có thể thay đổi. Bạn có thể sửa, thêm hoặc xóa phần tử. List được tạo bằng dấu ngoặc vuông [].
- Tuple (bộ giá trị) là tập hợp có thứ tự nhưng không thể thay đổi. Khi tạo xong, các phần tử không thể chỉnh sửa. Tuple được tạo bằng dấu ngoặc tròn ().

Sự khác nhau giữa Array và List trong Python là gì?

Trong Python, list có thể chứa các phần tử khác kiểu, rất linh hoạt và có sẵn, trong khi array (từ module array) chỉ chứa các phần tử cùng kiểu, ít linh hoạt hơn nhưng tiết kiệm bộ nhớ và nhanh hơn khi xử lý dữ liệu số lớn.

Từ khóa self trong Python được dùng để làm gì?
Trong Python, self là tham chiếu tới chính đối tượng hiện tại của lớp. Nó được dùng để truy cập thuộc tính và phương thức của đối tượng từ bên trong lớp. Mỗi phương thức của đối tượng phải có self làm tham số đầu tiên.
Hai loại vòng lặp chính (loop statements) trong Python là gì?
Hai loại vòng lặp chính trong Python là vòng lặp for và vòng lặp while. - for loop dùng để lặp qua một chuỗi, list, tuple hoặc range. - while loop thực hiện một khối code liên tục khi điều kiện còn đúng.
Decorator trong Python là gì?
Decorator trong Python là một hàm dùng để thay đổi hoặc mở rộng chức năng của một hàm hoặc phương thức khác mà không cần sửa code gốc. Decorator thường dùng cho logging, kiểm soát truy cập, đo thời gian chạy hoặc caching. Cú pháp áp dụng là @tên_decorator phía trên định nghĩa hàm.
Những kiểu dữ liệu (built-in types) có sẵn trong Python là gì?
Python cung cấp nhiều kiểu dữ liệu có sẵn (built-in types). Các kiểu chính bao gồm: 1. Kiểu số: int, float, complex 2. Kiểu tuần tự (sequence): list, tuple, range 3. Kiểu văn bản: str 4. Kiểu tập hợp (set): set, frozenset 5. Kiểu ánh xạ (mapping): dict 6. Kiểu Boolean: bool 7. Kiểu nhị phân: bytes, bytearray, memoryview Những kiểu này giúp bạn lưu trữ và xử lý dữ liệu theo nhiều dạng khác nhau.
Làm thế nào để phân biệt tệp .py và .pyc trong Python?
Trong Python: Tệp .py là file mã nguồn, chứa code Python do lập trình viên viết. Tệp .pyc là file bytecode đã biên dịch, được Python tạo tự động khi import tệp .py. Nó chứa phiên bản biên dịch không phụ thuộc nền tảng, giúp chạy chương trình nhanh hơn. "Tệp .py là mã nguồn, còn .pyc là bytecode đã biên dịch do Python tạo ra."
Làm thế nào để tạo một hàm (function) trong Python?
Trong Python, bạn tạo một hàm (function) bằng từ khóa def, theo sau là tên hàm, dấu ngoặc cho tham số, và dấu hai chấm. Nội dung hàm được thụt vào. Có thể trả về giá trị bằng câu lệnh return. "Tạo hàm trong Python bằng def, viết code bên trong và có thể dùng return để trả về giá trị."
Một hàm trong Python trả về giá trị như thế nào?
Trong Python, một hàm trả về giá trị bằng câu lệnh return. Khi hàm gặp return, nó gửi giá trị được chỉ định về cho nơi gọi hàm. Nếu không dùng return, hàm sẽ trả về None mặc định.

<p>Lệnh nào được dùng để xóa (delete) file trong Python?</p> <p>Trong Python, bạn có thể xóa file bằng module os hoặc pathlib. Dùng os.remove(): Dùng pathlib.Path.unlink():</p>
<p>Module trong Python là gì?</p> <p>Module trong Python là một file chứa code Python, như hàm, lớp hoặc biến, có thể import và sử dụng trong chương trình khác. Module giúp tổ chức code và tái sử dụng hiệu quả.</p>
<p>Python PATH là gì?</p> <p>Trong Python, Python PATH là danh sách các thư mục mà trình thông dịch sẽ tìm kiếm khi bạn import module hoặc package. Nó được lưu trong biến sys.path. Bạn cũng có thể thêm thư mục của riêng mình vào PATH nếu cần.</p>
<p>Package trong Python là gì?</p> <p>Trong Python, package là tập hợp các module được tổ chức trong thư mục. Một package có file đặc biệt __init__.py (có thể rỗng) để Python nhận biết thư mục là package. Package giúp tổ chức các module liên quan thành một cấu trúc duy nhất.</p> <p>"Package là thư mục chứa các module Python và file __init__.py để tổ chức chúng."</p>
<p>Làm thế nào để tạo một module trong Python?</p> <p>Trong Python, module đơn giản là một file Python với đuôi .py chứa các hàm, biến hoặc lớp. Bạn có thể import module đó trong file khác để sử dụng.</p> <p>"Module là file Python chứa hàm hoặc lớp có thể import vào file khác."</p>
<p>Lambda trong Python là gì?</p> <p>Trong Python, lambda là một hàm vô danh, tức là hàm không có tên. Nó có thể nhận nhiều tham số nhưng chỉ có một biểu thức. Lambda thường dùng cho các phép toán ngắn gọn hoặc khi truyền hàm làm đối số cho các hàm khác như map(), filter() hay sorted().</p> <p>"Lambda là hàm vô danh trong Python, có một biểu thức và nhận nhiều tham số."</p>
<p>Làm thế nào để quản lý bộ nhớ (memory management) trong Python?</p> <p>Bộ nhớ trong Python được quản lý tự động thông qua các cơ chế:</p> <ol style="list-style-type: none"> 1. Đếm tham chiếu (reference counting): Mỗi đối tượng theo dõi số lượng tham chiếu trỏ tới nó. Khi số tham chiếu về 0, bộ nhớ có thể được giải phóng. 2. Garbage collection (thu gom rác): Python có trình thu gom rác để phát hiện và xóa các đối tượng không còn dùng, đặc biệt là các tham chiếu vòng. 3. Cấp phát bộ nhớ động: Các đối tượng được cấp phát từ private heap do trình quản lý bộ nhớ của Python kiểm soát. <p>Trong hầu hết trường hợp, lập trình viên không cần cấp phát hay giải phóng bộ nhớ thủ công.</p>

<p>Keyword trong Python là gì?</p> <p>Trong Python, keyword (từ khóa) là từ được giữ riêng, có ý nghĩa đặc biệt trong ngôn ngữ. Chúng không thể dùng làm tên biến, hàm hay định danh khác. Keywords định nghĩa cú pháp và cấu trúc của Python.</p> <p>Một số keyword: if, else, for, while, def, class, import, return, try, except</p> <p>"Keywords là từ khóa có ý nghĩa đặc biệt trong Python, không dùng làm tên biến hay hàm được."</p>
<p>Python có phải là ngôn ngữ phân biệt chữ hoa – chữ thường (case-sensitive) không?</p> <p>Đúng, Python là ngôn ngữ phân biệt chữ hoa – chữ thường (case-sensitive). Điều này có nghĩa là các định danh như tên biến, tên hàm, tên lớp được phân biệt chữ hoa và chữ thường. Ví dụ, myVar và myvar được coi là hai định danh khác nhau.</p> <p>"Đúng, Python phân biệt chữ hoa chữ thường, tên biến và hàm khác nhau nếu khác chữ hoa/chữ thường."</p>
<p>Literal trong Python là gì?</p> <p>Trong Python, literal (giá trị hằng) là giá trị cố định được viết trực tiếp trong code. Literal biểu diễn dữ liệu thuộc một kiểu nhất định như số, chuỗi, hoặc giá trị Boolean.</p> <p>Ví dụ về literals:</p> <ul style="list-style-type: none"> - Số: 10, 3.14 - Chuỗi: "Hello", 'Python' - Boolean: True, False
<p>Type Conversion trong Python là gì?</p> <p>In Python, type conversion (also called type casting) is the process of converting a value from one data type to another. Python provides built-in functions for type conversion like int(), float(), str(), list(), etc.</p> <p>"Type conversion là chuyển giá trị từ kiểu dữ liệu này sang kiểu khác bằng các hàm như int(), float(), str()."</p>
<p>Theo bạn, hàm dir() trong Python được dùng để làm gì?</p> <p>Trong Python, hàm dir() được dùng để liệt kê tất cả thuộc tính và phương thức của một đối tượng, module hoặc lớp. Nó giúp khám phá những gì đối tượng đó chứa.</p> <p>"Hàm dir() hiển thị tất cả thuộc tính và phương thức của một đối tượng, lớp hoặc module."</p>
<p>Làm thế nào để xóa giá trị khỏi mảng (array) trong Python?</p> <p>Trong Python, nếu dùng module array, bạn có thể xóa giá trị bằng remove() hoặc pop():</p> <ul style="list-style-type: none"> - remove(value) xóa giá trị đầu tiên xuất hiện trong mảng. - pop(index) xóa giá trị ở vị trí chỉ định (mặc định là phần tử cuối).

List trong Python được lưu trữ trong bộ nhớ như thế nào?

Trong Python, một list là một danh sách động (dynamic array), có thể chứa các phần tử thuộc bất kỳ kiểu dữ liệu nào. Python không lưu trực tiếp giá trị trong list mà lưu các con trỏ tới các object trong bộ nhớ.

List là một object chứa một mảng các con trỏ, kích thước hiện tại (size) và dung lượng dự trữ (capacity). Khi cập nhật phần tử tại vị trí X, Python chỉ thay đổi con trỏ tại vị trí đó để trỏ tới object mới, các phần tử còn lại không thay đổi.

Khi thêm phần tử bằng append, Python sẽ kiểm tra dung lượng còn đủ không: nếu đủ, thêm con trỏ vào mảng; nếu không, Python sẽ cấp phát một mảng mới lớn hơn, copy các con trỏ cũ sang, rồi thêm phần tử mới. Vì vậy, append() trung bình là $O(1)$, nghĩa là thêm phần tử vào cuối list thường mất thời gian hằng số, nhưng thỉnh thoảng phải mở rộng bộ nhớ nên mất $O(n)$ thời gian cho lần append đó.

Tuple trong Python được lưu trữ trong bộ nhớ như thế nào?

Tuple trong Python là một danh sách bất biến (immutable sequence), có thể chứa các phần tử thuộc bất kỳ kiểu dữ liệu nào. Python không lưu trực tiếp giá trị trong tuple mà lưu các con trỏ tới các object trong bộ nhớ. Tuple là một object chứa một mảng các con trỏ, cùng với thông tin về kích thước (size).

Vì tuple bất biến, nên sau khi tạo:

- Kích thước không thay đổi, và mảng con trỏ cố định.

- Khi truy cập phần tử tại vị trí X, Python chỉ đọc con trỏ tại vị trí đó để lấy object tương ứng, các phần tử khác không bị ảnh hưởng.

- Không có cơ chế mở rộng như list, nên tuple không thể append, insert hay xóa phần tử.

Do cấu trúc cố định và không cần quản lý dung lượng dự trữ, tuple nhẹ hơn list, tiết kiệm bộ nhớ hơn, và truy cập phần tử nhanh hơn.

Ngoài ra, nếu tất cả phần tử trong tuple đều immutable, tuple có thể hash được và dùng làm key của dict hoặc element của set.

How are list, tuple, set, array, and dictionary stored in memory in Python?

List, tuple, set, array và dictionary trong Python được lưu trữ trong bộ nhớ như thế nào?

Một hàm trong Python có thể không có câu lệnh return không? Và điều đó có hợp lệ không?

Đúng, một hàm trong Python có thể không có câu lệnh return và điều này vẫn hợp lệ. Nếu không có return, hàm sẽ tự động trả về None theo mặc định.

Khi nào nên dùng dấu ngoặc kép ba (triple quotes) làm ký tự phân tách trong Python?

Trong Python, dấu ngoặc kép ba (""" hoặc ''') được dùng khi bạn muốn:

1. Tạo chuỗi nhiều dòng trải dài trên nhiều dòng.

2. Viết docstring để ghi chú cho module, class hoặc hàm.

Vai trò chính của phương thức __init__ là gì? Hãy đưa ra một ví dụ minh họa bằng code.

Vai trò chính của phương thức __init__ trong Python là khởi tạo các thuộc tính của lớp khi một đối tượng được tạo ra. Nó được gọi tự động khi một instance mới của lớp được khởi tạo.

Làm thế nào để chuyển một chuỗi (string) sang chữ thường (lowercase) trong Python?

Trong Python, bạn có thể chuyển một chuỗi sang chữ thường bằng phương thức lower() của chuỗi.

How do you use the split method in Python?

In Python, the split() method is used to divide a string into a list of substrings based on a separator. By default, it splits by whitespace.

Khối Try (Try Block) trong Python là gì?

Trong Python, khối try dùng để bao quanh đoạn code có thể phát sinh lỗi (exception). Nếu lỗi xảy ra trong try, nó có thể bắt và xử lý trong khối except. Try giúp chương trình không bị dừng khi có lỗi.

<p>Generator trong Python là gì?</p> <p>Trong Python, generator là hàm trả về một iterator và phát sinh (yield) từng giá trị một bằng từ khóa yield thay vì return. Generator giúp xử lý dữ liệu lớn hiệu quả hơn vì không lưu toàn bộ giá trị trong bộ nhớ, mà tạo ra từng giá trị khi cần.</p> <p>"Generator là hàm dùng yield để tạo giá trị từng cái một, giúp tiết kiệm bộ nhớ."</p>
<p>Làm thế nào để truy cập một module viết bằng Python từ ngôn ngữ C?</p> <p>Một module Python có thể được truy cập từ ngôn ngữ C bằng cách nhúng trình thông dịch Python (Python interpreter) vào chương trình C. Việc này thực hiện thông qua Python/C API, có trong file tiêu đề Python.h. Bạn khởi tạo trình thông dịch, import module Python, rồi gọi hàm của nó qua API.</p> <p>"Có thể truy cập module Python từ C bằng cách nhúng trình thông dịch Python và dùng Python/C API."</p>
<p>Làm thế nào để đảo ngược (reverse) một list trong Python?</p> <p>Trong Python, có thể đảo ngược list bằng vài cách:</p> <ol style="list-style-type: none"> 1. Dùng phương thức reverse() — đảo trực tiếp trên list gốc. 2. Dùng cú pháp cắt (slicing) — list[::-1] tạo một bản sao đảo ngược.
<p>Có những cách nào để kết hợp (combine) các DataFrame trong Python?</p> <p>Trong Python (dùng thư viện pandas), có nhiều cách để kết hợp các DataFrame tùy vào mục đích:</p> <ol style="list-style-type: none"> 1. concat() – Ghép các DataFrame theo hàng hoặc theo cột. 2. merge() – Kết hợp dựa trên các cột hoặc khóa chung, tương tự lệnh JOIN trong SQL. 3. join() – Kết hợp dựa trên chỉ số (index). <p>"Có thể kết hợp DataFrame bằng concat() để ghép, merge() để nối theo cột chung, và join() để nối theo index."</p>
<p>PIP là gì trong Python?</p> <p>PIP là trình cài đặt package cho Python. Nó cho phép bạn cài đặt, nâng cấp và quản lý các package Python từ Python Package Index (PyPI) hoặc các kho khác. Đây là công cụ chuẩn để thêm thư viện bên thứ ba vào môi trường Python.</p> <p>"PIP là trình cài đặt package cho Python, dùng để cài đặt và quản lý thư viện bên ngoài."</p>
<p>Tại sao lại sử dụng finalize trong Python?</p> <p>Trong Python, finalize (từ module weakref) được dùng để đăng ký một hàm callback sẽ được gọi khi một đối tượng sắp bị garbage collection. Nó hữu ích để dọn dẹp tài nguyên như đóng file hoặc kết nối mạng khi đối tượng không còn cần nữa, mà không phải dựa vào __del__().</p> <p>"finalize dùng để đăng ký callback dọn dẹp khi một đối tượng sắp bị garbage collection."</p>

Phân biệt giữa override và new modifiers.

Sự khác biệt giữa override và new (thường trong C#; Python không có từ khóa này nhưng ý tưởng tương tự):

1. Override:

- Khi lớp con muốn cung cấp triển khai mới cho một phương thức đã định nghĩa trong lớp cha.
- Đảm bảo hành vi đa hình, phương thức lớp con được gọi ngay cả khi tham chiếu là lớp cha.

2. New:

- Khi lớp con định nghĩa phương thức cùng tên với lớp cha nhưng không override.
- Ẩn phương thức lớp cha, phương thức lớp cha được gọi nếu tham chiếu là kiểu lớp cha.

Trong Python, làm thế nào để tạo một class rỗng (empty class)?

Trong Python, để tạo một class rỗng, bạn định nghĩa class bằng từ khóa class và dùng pass để chỉ rằng class không có nội dung.

Bạn có nghĩ rằng có thể gọi lớp cha (parent class) mà không cần tạo instance của nó không?

Trong Python, bạn có thể gọi phương thức hoặc truy cập thuộc tính của lớp cha mà không cần tạo instance nếu phương thức đó là class method (@classmethod) hoặc static method (@staticmethod). Với phương thức bình thường, bạn cần tạo instance của lớp cha.

"Bạn có thể gọi lớp cha mà không tạo instance nếu phương thức là static hoặc class method."

Có những cách nào để truy cập các thành phần (members) của lớp cha (parent) trong lớp con (child)?

Trong Python, các thành phần của lớp cha (thuộc tính hoặc phương thức) có thể được truy cập trong lớp con bằng nhiều cách:

1. Dùng super() – Gọi phương thức hoặc truy cập thuộc tính của lớp cha một cách an toàn và linh hoạt.
2. Dùng trực tiếp tên lớp cha – Gọi phương thức hoặc truy cập thuộc tính một cách rõ ràng.
3. Kế thừa trực tiếp – Nếu thành phần là public, lớp con có thể dùng trực tiếp.

Pandas trong Python là gì?

Pandas là thư viện Python dùng để xử lý và phân tích dữ liệu. Nó cung cấp các cấu trúc dữ liệu như DataFrame và Series, giúp xử lý dữ liệu dạng bảng hoặc dữ liệu theo thời gian, thực hiện lọc, tổng hợp và các phép toán thống kê một cách hiệu quả.

"Pandas là thư viện Python để xử lý và phân tích dữ liệu, dùng các cấu trúc DataFrame và Series"

NumPy là gì?

NumPy (Numerical Python) là thư viện Python cho tính toán số học. Nó cung cấp mảng đa chiều hiệu suất cao (ndarray) và các hàm để thực hiện các phép toán toán học, logic và thống kê một cách hiệu quả. NumPy được dùng nhiều trong tính toán khoa học, phân tích dữ liệu và học máy.

"NumPy là thư viện Python cho tính toán số học, với mảng hiệu suất cao và các hàm toán học."

Tại sao NumPy lại được ưu tiên sử dụng hơn Python lists?

NumPy được ưu tiên hơn Python list vì:

1. Hiệu năng: Mảng NumPy được cài đặt bằng C, nhanh hơn nhiều khi tính toán số học trên dữ liệu lớn.
2. Tiết kiệm bộ nhớ: NumPy dùng khối bộ nhớ liên tục và kiểu dữ liệu đồng nhất, tốn ít bộ nhớ hơn list.
3. Phép toán vector hóa: Cho phép thao tác từng phần tử mà không cần dùng vòng lặp, code gọn và nhanh hơn.
4. Hàm toán học phong phú: Cung cấp nhiều hàm toán, thống kê và đại số tuyến tính sẵn có.

"NumPy nhanh hơn, tiết kiệm bộ nhớ, hỗ trợ phép toán vector hóa và nhiều hàm toán học hơn Python list."

Làm thế nào để tải dữ liệu từ tệp văn bản (text file) một cách hiệu quả trong Python?

Trong Python, dữ liệu từ tệp văn bản có thể được tải hiệu quả tùy thuộc vào kích thước và định dạng:

1. Dùng with open() – Đọc tệp an toàn và tự động đóng tệp.
2. Dùng readlines() hoặc lặp từng dòng – Hiệu quả cho tệp lớn, tránh tải toàn bộ vào bộ nhớ.
3. Dùng pandas.read_csv() – Với dữ liệu có cấu trúc (CSV hoặc tab), nhanh và tiện lợi.
4. Dùng numpy.loadtxt() – Với dữ liệu số, hiệu quả và trả về mảng NumPy.

Reindexing trong Pandas là gì?

Trong Pandas, reindexing là thay đổi nhãn hàng hoặc cột của một DataFrame hoặc Series sang tập nhãn mới. Điều này có thể dùng để thêm, xóa hoặc sắp xếp lại thứ tự các hàng/cột. Khi nhãn mới được thêm, giá trị thiếu sẽ được điền NaN theo mặc định hoặc giá trị bạn chỉ định.

"Reindexing trong Pandas thay đổi nhãn hàng/cột, thêm hoặc sắp xếp lại và điền giá trị cho các mục thiếu."

Làm thế nào để sao chép (copy) một đối tượng (object) trong Python?

Trong Python, bạn có thể sao chép một đối tượng bằng hai cách chính:

1. Shallow copy – Tạo đối tượng mới, nhưng các đối tượng lồng bên trong vẫn được chia sẻ. Dùng copy.copy().
2. Deep copy – Tạo đối tượng mới với tất cả các đối tượng lồng bên trong được sao chép đệ quy. Dùng copy.deepcopy().

"Dùng copy.copy() cho shallow copy và copy.deepcopy() cho deep copy."

Shallow copy và deep copy trong Python khác nhau như thế nào?

Sự khác nhau giữa shallow copy và deep copy trong Python:

1. Shallow copy:
 - Tạo đối tượng mới, nhưng các đối tượng lồng bên trong vẫn được chia sẻ với bản gốc.
 - Thay đổi các đối tượng lồng sẽ ảnh hưởng cả bản gốc lẫn bản sao.
 - Dùng copy.copy().
2. Deep copy:
 - Tạo một bản sao hoàn toàn độc lập, bao gồm tất cả các đối tượng lồng bên trong đệ quy.
 - Thay đổi ở bản sao không ảnh hưởng đến đối tượng gốc.
 - Dùng copy.deepcopy().

"Shallow copy chỉ sao chép đối tượng cấp trên, chia sẻ các đối tượng lồng, còn deep copy sao chép toàn bộ đệ quy, tạo đối tượng độc lập."

Pickling là gì?

Trong Python, pickling là quá trình chuyển một đối tượng Python thành luồng byte để có thể lưu vào file hoặc gửi qua mạng. Quá trình ngược lại, chuyển byte stream thành đối tượng Python, gọi là unpickling. Pickling thường dùng để lưu trữ đối tượng Python.

"Pickling là chuyển đối tượng Python thành byte stream để lưu trữ hoặc gửi đi; unpickling phục hồi lại đối tượng."

Bạn định nghĩa Unpickling trong Python như thế nào?

Trong Python, unpickling là quá trình chuyển một byte stream đã được pickling trở lại thành đối tượng Python ban đầu. Đây là quá trình ngược với pickling và dùng để tải lại đối tượng đã lưu hoặc đã truyền đi trước đó.

"Unpickling chuyển byte stream đã pickling thành đối tượng Python ban đầu."

Hàm nào được sử dụng để thực hiện Pickling và Unpickling?

Trong Python, module pickle cung cấp các hàm để pickling và unpickling:

1. Pickling (chuẩn hóa): Dùng `pickle.dump()` để ghi đối tượng vào file hoặc `pickle.dumps()` để tạo byte stream.
2. Unpickling (phục hồi): Dùng `pickle.load()` để đọc đối tượng từ file hoặc `pickle.loads()` từ byte stream.

Kể tên một số kiểu chuyển đổi dữ liệu (Type Conversion) trong Python.

Trong Python, Type Conversion là chuyển đổi biến từ kiểu dữ liệu này sang kiểu dữ liệu khác. Có hai loại chính:

1. Chuyển đổi ngầm định (Implicit Type Conversion): Python tự động chuyển đổi kiểu dữ liệu, ví dụ int sang float.
2. Chuyển đổi tường minh (Explicit Type Conversion): Bạn chuyển đổi thủ công bằng các hàm sẵn có:
 - `int()` – chuyển sang số nguyên
 - `float()` – chuyển sang số thực
 - `str()` – chuyển sang chuỗi
 - `list()` – chuyển sang list
 - `tuple()` – chuyển sang tuple
 - `dict()` – chuyển sang dictionary

Hãy đưa ra một ví dụ về hàm lambda.

Hàm lambda trong Python là hàm ẩn danh, viết gọn trong một dòng và dùng từ khóa lambda.

"Hàm lambda là hàm ẩn danh, viết gọn trong một dòng, dùng từ khóa lambda."

Trong Python, Polymorphism (đa hình) là gì?

Trong Python, polymorphism (đa hình) là khả năng của các đối tượng khác nhau được truy cập qua cùng một giao diện, cho phép chúng thực hiện hành động khác nhau tùy vào loại của chúng. Thường thấy trong method overriding, operator overloading và duck typing.

"Đa hình cho phép các đối tượng khác nhau dùng cùng một giao diện, nhưng thực hiện hành động theo cách riêng."

Bạn sẽ định nghĩa hàm swapcase() trong Python như thế nào?
<p>Trong Python, swapcase() là một phương thức của chuỗi dùng để chuyển tất cả chữ hoa thành chữ thường và chữ thường thành chữ hoa trong chuỗi. Các ký tự không phải chữ cái không bị thay đổi.</p> <p>"swapcase() chuyển chữ hoa thành chữ thường và chữ thường thành chữ hoa trong chuỗi."</p>
Trong Python, cú pháp[::-1] có tác dụng gì? Hãy đưa ra một ví dụ minh họa.
<p>Trong Python,[::-1] là cú pháp dùng để đảo ngược một chuỗi, list hoặc tuple. Nó có nghĩa là: bắt đầu từ cuối về đầu với bước nhảy là -1.</p> <p>"[::-1] đảo ngược một chuỗi hoặc list trong Python."</p>
Giải thích cách kết nối cơ sở dữ liệu (database connection) trong Flask (Python framework).
<p>Trong Flask, kết nối cơ sở dữ liệu thực hiện bằng cách dùng driver cơ sở dữ liệu hoặc ORM. Các cách phổ biến:</p> <ol style="list-style-type: none"> Dùng Flask extensions như Flask-SQLAlchemy: <ul style="list-style-type: none"> Hỗ trợ ORM, đơn giản hóa thao tác với cơ sở dữ liệu. Bạn định nghĩa URI của database, tạo đối tượng SQLAlchemy, và thao tác dữ liệu qua các model. Dùng driver trực tiếp (ví dụ sqlite3 hoặc pymysql): <ul style="list-style-type: none"> Mở kết nối, tạo cursor, thực thi truy vấn, rồi đóng kết nối.
Hiệu ứng Dogpile (Dogpile effect) là gì?
<p>Hiệu ứng Dogpile xảy ra khi nhiều tiến trình hoặc luồng cố gắng tạo lại cùng một dữ liệu trong cache cùng lúc sau khi dữ liệu hết hạn. Điều này có thể gây quá tải cho cơ sở dữ liệu hoặc hệ thống backend, vì tất cả các yêu cầu đều kích hoạt tính toán nặng cùng lúc.</p> <p>Cách phòng tránh:</p> <ul style="list-style-type: none"> Dùng khóa cache (cache lock) hoặc stale-while-revalidate để chỉ một tiến trình tạo lại cache, các tiến trình khác chờ. <p>Ví dụ (khái niệm):</p> <p>Giả sử cache của giá sản phẩm phổ biến hết hạn. Khi đó hàng trăm người dùng truy cập cùng lúc, khiến nhiều truy vấn cơ sở dữ liệu cùng lúc — đây chính là hiệu ứng Dogpile.</p> <p><i>"Hiệu ứng Dogpile xảy ra khi nhiều yêu cầu cùng lúc tạo lại cache hết hạn, gây quá tải."</i></p>
Python có hỗ trợ kế thừa đa lớp (multiple inheritance) không?
<p>Đúng, Python hỗ trợ kế thừa đa lớp, nghĩa là một lớp có thể kế thừa thuộc tính và phương thức từ nhiều lớp cha. Python sử dụng Method Resolution Order (MRO) để xác định thứ tự tìm kiếm phương thức khi gọi.</p> <p>"Python có hỗ trợ kế thừa đa lớp và dùng MRO để xác định thứ tự tìm phương thức."</p>
Python có sử dụng các phạm vi truy cập (access specifiers) như public, private, protected không?
<p>Python không có các phạm vi truy cập nghiêm ngặt như public, private, hay protected như trong Java hoặc C++. Thay vào đó, Python dựa trên quy ước đặt tên:</p> <ol style="list-style-type: none"> Public: Thuộc tính hoặc phương thức không có dấu gạch dưới, truy cập được ở mọi nơi. Protected: Tiền tố <code>_</code> (ví dụ <code>_var</code>) để chỉ rằng nó nên được coi là protected, nhưng chỉ là quy ước. Private: Tiền tố <code>__</code> (ví dụ <code>__var</code>) sẽ thực hiện name mangling, làm cho việc truy cập từ bên ngoài khó hơn (nhưng không tuyệt đối). <p>"Python dùng quy ước đặt tên thay vì access specifiers nghiêm ngặt: public, protected (<code>_</code>), private (<code>__</code> với name mangling)."</p>

Làm thế nào để tạo constructor (hàm khởi tạo) trong Python?

Trong Python, constructor là phương thức đặc biệt `__init__()` được gọi tự động khi một đối tượng của lớp được tạo. Nó dùng để khởi tạo các thuộc tính của đối tượng.

"Constructor trong Python là phương thức `__init__()` dùng để khởi tạo các thuộc tính của đối tượng."

Làm thế nào để lưu hình ảnh (image) về máy cục bộ khi bạn biết địa chỉ URL trong Python?

Trong Python, bạn có thể lưu hình ảnh từ URL bằng thư viện requests hoặc urllib. Ý tưởng là tải nội dung ảnh và ghi vào file cục bộ ở chế độ nhị phân.

"Dùng `requests.get()` hoặc `urllib.request.urlretrieve()` để tải ảnh từ URL và lưu vào máy."

Giải thích cách hoạt động của hàm `join()` trong Python.

Hàm `join()` trong Python là một phương thức của chuỗi (string method), được dùng để nối các phần tử của một iterable (như list hoặc tuple) thành một chuỗi duy nhất, với ký tự phân tách được chỉ định giữa các phần tử.

Làm thế nào để xác định và xử lý giá trị bị thiếu (missing values) trong một DataFrame?

Trong Pandas, các giá trị bị thiếu thường được biểu diễn bằng NaN (Not a Number). Bạn có thể xác định và xử lý chúng bằng các hàm tích hợp sẵn.

Tệp `manage.py` trong Python được dùng để làm gì?

Trong Django (một framework web phổ biến của Python), tệp `manage.py` là một tiện ích dòng lệnh (command-line utility) được tự động tạo ra khi bạn khởi tạo dự án.

- Nó giúp bạn thực hiện nhiều tác vụ quản trị như:
- Chạy máy chủ phát triển (`python manage.py runserver`)
- Áp dụng các thay đổi cơ sở dữ liệu (`python manage.py migrate`)
- Tạo ứng dụng mới (`python manage.py startapp appname`)
- Tạo tài khoản quản trị (`python manage.py createsuperuser`)
- Chạy kiểm thử (`python manage.py test`)

"`manage.py` đóng vai trò như một trình bao bọc (wrapper) quanh công cụ `django-admin`, giúp quản lý và vận hành dự án Django dễ dàng hơn."

Giải thích phương thức `shuffle` và đưa ra một ví dụ minh họa.

Phương thức `shuffle()` trong Python thuộc module `random`. Nó được dùng để xáo trộn ngẫu nhiên thứ tự các phần tử trong một danh sách (list).

Phương thức này thay đổi trực tiếp danh sách gốc, không tạo ra bản sao mới.

Phương thức nào có thể được sử dụng để tạo số ngẫu nhiên (random numbers) trong Python?

Trong Python, bạn có thể tạo số ngẫu nhiên bằng các hàm trong module `random`. Một số phương thức phổ biến:

1. `random.random()` – Trả về một số thực ngẫu nhiên từ 0.0 đến 1.0.
2. `random.randint(a, b)` – Trả về một số nguyên ngẫu nhiên từ a đến b (bao gồm cả a và b).
3. `random.uniform(a, b)` – Trả về một số thực ngẫu nhiên từ a đến b.
4. `random.choice(sequence)` – Chọn một phần tử ngẫu nhiên từ một chuỗi hoặc danh sách.

*args và **kwargs có ý nghĩa gì trong Python?"

Trong Python, *args và **kwargs được dùng trong định nghĩa hàm để nhận số lượng tham số linh hoạt.

1. *args (tham số không có tên):

- Gộp tất cả tham số vị trí dư thừa thành một tuple.
- Dùng khi bạn không biết trước có bao nhiêu tham số vị trí sẽ được truyền vào.

2. **kwargs (tham số có tên):

- Gộp tất cả tham số từ khóa dư thừa thành một dictionary.
- Dùng khi muốn xử lý các tham số đặt tên một cách linh hoạt.

Flask có phải là mô hình MVC không? Nếu có, hãy giải thích bằng cách sử dụng mô hình MVC.

Đúng, Flask có thể được dùng để triển khai mô hình MVC (Model-View-Controller), nhưng Flask không bắt buộc cấu trúc này. Là một micro-framework, Flask cho phép lập trình viên linh hoạt sắp xếp ứng dụng theo mô hình MVC.

- Model: Quản lý dữ liệu và logic nghiệp vụ. Trong Flask, thường dùng ORM như SQLAlchemy hoặc thao tác trực tiếp với cơ sở dữ liệu.
- View: Đại diện cho giao diện người dùng, thường là các template HTML được render bằng Jinja2 trong Flask.
- Controller: Xử lý logic ứng dụng và routing, được thực hiện thông qua các route functions nhận request, xử lý dữ liệu (sử dụng model) và trả về view.

Làm thế nào để kiểm tra xem tất cả ký tự trong một chuỗi có phải là alphanumeric không?

Trong Python, bạn có thể dùng phương thức `isalnum()` của chuỗi để kiểm tra xem tất cả ký tự trong chuỗi có phải là chữ và số hay không (không bao gồm khoảng trắng hay ký tự đặc biệt). Nó trả về True nếu tất cả là alphanumeric, ngược lại False.

"Dùng phương thức `isalnum()`, trả về True nếu tất cả ký tự là chữ hoặc số."

Một số module tích hợp (built-in modules) được sử dụng nhiều nhất trong Python là gì?

Python có nhiều module tích hợp sẵn (built-in modules) cung cấp các chức năng hữu ích mà không cần cài đặt thêm. Một số module được sử dụng nhiều nhất bao gồm:

1. `os` – Tương tác với hệ điều hành (file, thư mục, biến môi trường).
2. `sys` – Truy cập các tham số và hàm hệ thống.
3. `math` – Thực hiện các phép toán toán học (hàm lượng giác, logarit...).
4. `random` – Tạo số ngẫu nhiên và các thao tác ngẫu nhiên.
5. `datetime` – Làm việc với ngày và giờ.
6. `json` – Mã hóa và giải mã dữ liệu JSON.
7. `re` – Thao tác với biểu thức chính quy (regex).
8. `collections` – Các kiểu dữ liệu container nâng cao như Counter, deque, OrderedDict.

"Các module tích hợp phổ biến gồm `os`, `sys`, `math`, `random`, `datetime`, `json`, `re` và `collections`."

Những công cụ (tools) nào dùng để debug và thực hiện phân tích tĩnh (static analysis) trong Python?

Trong Python, có nhiều công cụ để debug và phân tích tĩnh:

1. Công cụ debug:

- pdb – Trình gỡ lỗi tích hợp sẵn, cho phép chạy từng bước, đặt breakpoint và kiểm tra biến.
- ipdb – Phiên bản nâng cao của pdb với tích hợp IPython.
- Debugger của IDE – Các IDE như PyCharm, VS Code, Spyder cung cấp debugger tích hợp với breakpoint, watch, và call stack.

2. Công cụ phân tích tĩnh (static analysis):

- pylint – Kiểm tra chuẩn code, lỗi, và khả năng có lỗi tiềm ẩn.
- flake8 – Kiểm tra style code theo PEP 8 và lỗi lập trình.
- mypy – Kiểm tra kiểu dữ liệu dựa trên type hints.
- pyflakes – Phát hiện lỗi mà không kiểm tra style code.

"Để debug dùng pdb, ipdb, hoặc debugger của IDE; để phân tích tĩnh dùng pylint, flake8, mypy, hoặc pyflakes."

Khi nào phần else trong cấu trúc try-except-else được thực thi?

Trong Python, phần else trong cấu trúc try-except-else chỉ được thực thi khi không có ngoại lệ xảy ra trong khối try.

- Nếu khối try chạy thành công, else sẽ chạy.
- Nếu có lỗi xảy ra, khối except sẽ chạy và else bị bỏ qua.

"Phần else chỉ chạy khi try không có lỗi."

Viết đoạn code để hoán đổi (swap) hai số trong Python.

Trong Python, bạn có thể hoán đổi hai số rất đơn giản bằng tuple unpacking, không cần biến trung gian.

"Bạn có thể hoán đổi hai số bằng a, b = b, a hoặc dùng biến tạm thời."

Cho ví dụ code về deep copy và shallow copy.

Trong Python, shallow copy và deep copy được dùng để sao chép đối tượng nhưng hành vi khác nhau:

- Shallow copy: Sao chép đối tượng bên ngoài, nhưng các đối tượng lồng nhau vẫn tham chiếu đến đối tượng gốc.
- Deep copy: Sao chép toàn bộ đối tượng, bao gồm cả các đối tượng lồng nhau, hoàn toàn độc lập với bản gốc.

"Shallow copy sao chép đối tượng ngoài nhưng giữ tham chiếu đến các đối tượng lồng, deep copy sao chép toàn bộ, kể cả các đối tượng lồng nhau."

Viết đoạn code để đảo ngược (reverse) nhiều giá trị từ một hàm.

Trong Python, một hàm có thể trả về nhiều giá trị dưới dạng tuple, và bạn có thể đảo ngược chúng bằng tuple unpacking hoặc cú pháp slicing[::-1].

"Trả về nhiều giá trị dưới dạng tuple và dùng slicing[::-1] để đảo ngược."

Những bước nào để tạo mảng 3D, 2D và 1D trong Python?

Trong Python, bạn có thể tạo mảng 1D, 2D và 3D bằng thư viện NumPy.

Viết đoạn code để kiểm tra hai từ có phải là anagram hay không.
Trong Python, hai từ là anagram nếu chúng chứa cùng các chữ cái với cùng tần số, bất kể thứ tự. Một cách đơn giản là sắp xếp các chữ cái và so sánh. Viết đoạn code để kiểm tra mode trong một danh sách các số.
Trong Python, mode của danh sách là số xuất hiện nhiều nhất. Bạn có thể dùng module statistics để tìm mode. Lưu ý: Nếu có nhiều số cùng tần số xuất hiện cao nhất, statistics.mode() sẽ trả về số xuất hiện đầu tiên.
Cho ví dụ cách truy cập dataset từ bảng tính công khai (publicly shared spreadsheet) ở định dạng CSV được lưu trên Google Drive. Để truy cập một Google Spreadsheet công khai ở định dạng CSV, bạn có thể dùng thư viện pandas và link CSV của file. Các bước: 1. Đảm bảo bảng tính là public. 2. Lấy link CSV xuất ra. Với Google Sheet URL: https://docs.google.com/spreadsheets/d/FILE_ID/edit#gid=0 Link CSV sẽ là: https://docs.google.com/spreadsheets/d/FILE_ID/export?format=csv 3. Dùng pandas.read_csv() để tải dữ liệu:
Viết đoạn code cộng hai số nguyên mà không dùng toán tử +, khi các số lớn hơn 0. Bạn có thể cộng hai số nguyên dương mà không dùng toán tử + bằng cách sử dụng toán tử bitwise. Ý tưởng là dùng XOR (^) để cộng mà không có nhớ và AND (&) dịch trái để tính phần nhớ, lặp lại đến khi không còn nhớ. "Dùng XOR để cộng không nhớ, AND dịch trái để lấy phần nhớ, lặp đến khi phần nhớ bằng 0."
Viết đoạn code nhận một chuỗi các số và kiểm tra xem các số có duy nhất (unique) hay không. Trong Python, bạn có thể kiểm tra các số trong chuỗi có duy nhất hay không bằng cách chuyển chuỗi thành set và so sánh độ dài với chuỗi gốc. "Chuyển chuỗi thành set và so sánh độ dài, hoặc kiểm tra từng số với set để đảm bảo tính duy nhất."
Viết chương trình chuyển đổi định dạng ngày từ yyyy-mm-dd sang dd-mm-yyyy. Trong Python, bạn có thể chuyển đổi chuỗi ngày từ yyyy-mm-dd sang dd-mm-yyyy bằng module datetime. "Dùng datetime.strptime() để đọc yyyy-mm-dd, sau đó strftime() để định dạng dd-mm-yyyy."
Viết chương trình Python để tạo dãy Fibonacci. Trong Python, bạn có thể tạo dãy Fibonacci bằng vòng lặp hoặc đệ quy. "Bắt đầu với [0,1] và dùng vòng lặp cộng hai số cuối cùng để tạo dãy Fibonacci."
Viết đoạn code để tạo một chuỗi duy nhất từ các phần tử trong một danh sách (list). Trong Python, bạn có thể tạo một chuỗi duy nhất từ các phần tử trong danh sách bằng phương thức join().

Viết chương trình để kiểm tra một số có phải là số nguyên tố (prime) hay không.

Trong Python, số nguyên tố là số lớn hơn 1 và chỉ chia hết cho 1 và chính nó. Bạn có thể kiểm tra bằng vòng lặp.

Viết chương trình tính median trong Python sử dụng mảng NumPy.

Trong Python, bạn có thể tính median của một tập dữ liệu bằng NumPy với hàm `np.median()`.

Viết chương trình để thực hiện thuật toán đổi chỗ trực tiếp (interchange sort).

Interchange Sort là thuật toán sắp xếp đơn giản, tương tự bubble sort, so sánh từng phần tử với tất cả các phần tử khác và hoán đổi nếu chúng sai thứ tự.

"So sánh mỗi phần tử với tất cả phần tử phía sau và hoán đổi nếu sai thứ tự."

Viết chương trình để thực hiện thuật toán chọn trực tiếp (selection sort).

Selection Sort là thuật toán sắp xếp, chọn phần tử nhỏ nhất (hoặc lớn nhất) từ phần chưa sắp xếp và hoán đổi với phần tử đầu tiên của phần chưa sắp xếp.

"Lắp lại chọn phần tử nhỏ nhất từ phần chưa sắp xếp và hoán đổi với phần tử đầu tiên chưa sắp xếp."

Viết chương trình để thực hiện thuật toán chèn trực tiếp (insertion sort).

Insertion Sort là thuật toán sắp xếp đơn giản, lấy từng phần tử và chèn vào vị trí đúng trong phần mảng đã sắp xếp.

"Lấy từng phần tử và chèn vào phần mảng đã sắp xếp bằng cách dịch các phần tử lớn hơn sang phải."

Viết chương trình để thực hiện thuật toán sắp xếp nổi bọt (bubble sort).

Bubble Sort là thuật toán sắp xếp đơn giản, hoán đổi các phần tử kề nhau nếu chúng ở sai thứ tự, lặp lại cho đến khi mảng được sắp xếp.

"Duyệt mả nhiều lần, hoán đổi các phần tử kề nhau nếu sai thứ tự, lặp lại đến khi mả được sắp xếp."

[illegible]

[illegible]