Python interview question

Junior and middle questions.

1. What is the lambda function in Python? Why does it exist in Python?

* Các hàm lambda hoạt động giống như các hàm thông thường được khai báo bằng từ khóa def. Chúng rất hữu ích khi bạn muốn xác định một hàm nhỏ theo cách ngắn gọn. Hàm Lambda chỉ có thể chứa một biểu thức, vì vậy chúng không phù hợp cho các hàm có câu lệnh luồng điều khiển. Hàm Lambda chỉ tồn tại trong phạm vi biến mà nó được định nghĩa, nếu biến đó vượt ra khỏi phạm vi thì nó sẽ không còn tác dụng nữa.
* Hàm Lambda sẽ giúp việc lập trình trong Python trở nên tinh gọn và tối ưu hơn. Đặc biệt khi bạn cần xử lý logic tại một điểm nhất định trong thời gian ngắn. Hàm Lambda là một trong những hàm phổ biến sử dụng trong Python. Bạn có thể sử dụng hàm Lambda để lập trình một cách nhanh chóng và hiệu quả hơn.

2. What is pass in Python?

* Trong Python, pass là một lệnh trống, lệnh pass không làm gì cả, nó chỉ giữ chỗ cho các hàm, vòng lặp mà bạn đã thêm vào, nhưng chưa dùng đến trong hiện tại mà để “dành cho con cháu” mở rộng trong tương lai.

3. What is \*args, \*\*kwargs in function definition?

* Cú pháp \*args và \*\*kwargs cho phép chúng ta định nghĩa hàm có thể nhận số lượng tham số tuỳ ý. Ngoài ra, cú pháp này còn có thể được sử dụng để unpack khi gọi hàm cũng như trong nhiều trường hợp khác.

4. What is docstring in Python? How to write them? Are they required?

* Chuỗi tài liệu Python (hoặc docstrings) tạo điều kiện liên kết tài liệu với các mô-đun, hàm, lớp và phương thức Python. Docstring của một đối tượng được định nghĩa bằng cách bao gồm một hằng chuỗi làm câu lệnh đầu tiên trong định nghĩa của đối tượng. Nó được chỉ định trong mã nguồn, được sử dụng, giống như một nhận xét, để ghi lại một đoạn mã cụ thể.
* Không giống như các bình luận mã nguồn thông thường, docstring phải mô tả chức năng hoạt động chứ không phải như thế nào. Tất cả các hàm phải có một Docstring, Điều này cho phép chương trình kiểm tra các nhận xét này tại thời điểm chạy, ví dụ như một hệ thống trợ giúp tương tác hoặc dưới dạng siêu dữ liệu.

5. What are the built-in data types that Python provides? Which of them are mutable, which are immutable?

* Kiểu dữ liệu trong Python:

+ Bất biến: Number, Strings, Tuples.

+Có thể biến đổi: List, Dictinary, Sets

6. What is the difference between list and tuple types in Python?

**Cơ sở để so sánh :**   **List**   **Tuple**

Thiên nhiên: Có thể thay đổi Bất biến

Hàm ý của sự lặp lại Mất thời gian Nhanh

Phù hợp với Thao tác chèn và xóa Truy xuất các phần tử

Chìa khóa cho từ điển Không thể Có thể, khi chỉ các phần tử bất

7, What keywords can be used in conjunction with the for keyword?

* Các từ khóa: in, else

8, What's the difference between globals(), locals(), and vars()?

* globals() always returns the dictionary of the module namespace
* locals() always returns a dictionary of the current namespace
* vars() returns either a dictionary of the current namespace (if called with no argument) or the dictionary of the argument.

9, Is it possible to have a negative index in iterative types in Python?

* Có thể có

10. What is the \_\_init\_\_.py module? What it's for?

* Python định nghĩa hai loại gói, gói thông thường và gói không gian tên. Các gói thông thường là các gói truyền thống khi chúng tồn tại trong Python 3.2 trở về trước. Một gói thông thường thường được triển khai như một thư mục chứa tệp \_\_init\_\_.py. Khi một gói thông thường được nhập, tệp \_\_init\_\_.py này được thực thi ngầm và các đối tượng mà nó xác định được liên kết với các tên trong không gian tên của gói. Tệp \_\_init\_\_.py có thể chứa cùng mã Python mà bất kỳ mô-đun nào khác có thể chứa và Python sẽ thêm một số thuộc tính bổ sung vào mô-đun khi được nhập.

11. How can I swap values of variables in Python? Please give an example

* Hoán đổi bằng cách gán giá trị các biến cho nhau, và thông qua một biến trung gian để lưu giá trị tạm thời
* Ví dụ muốn hoán đổi giá trị biến x1 cho x2:

tem = x1

x1 = x2

x2 = temp

12. How do I view object methods? Please give an example

13. What is a module in python? What is a package? What is the difference between packages and modules in Python? Please give an example module and package

* Khi thoát khỏi PI, các dự liệu(biến, hàm, …) được khai báo trước đây sẽ bị mất .Do đó, cần chuẩn bị đầu vào cho trình thông dịch để có thể dùng được bất cứ khi nào mình cần, các file chứa những đoạn mã như vậy được gọi là các script. Để làm được như vậy, cách mình cần làm với Python là tạo một file có đuôi là “.py” rồi bỏ code vào đó. File này được gọi là mô-đun. Một mô-đun có thể được sử dụng trong một mô-đun khác, cũng có thể sử dụng trong trình thông dịch PI. Tên của mô-đun được lưu dưới biến name của chính mô-đun đó.
* Gói(package) là một nhóm các mô-đun khác nhau. Nó tương tự như thư mục chứa nhiều tệp tin vậy, nhưng cần phải đăng ký nhẹ với python để nó biết đó là gói, bằng cách thêm file init.py là python biết mình khởi tạo gói. Một gói cũng có thể chứa nhiều gói con khác nằm bên trong nó.

14. What is the \_\_init\_\_ function used for?

\_init\_\_ là hàm dựng hay constructor của một class. Khi một thực thể (instance) của một class được tạo ra thì hàm này sẽ được thực thi đầu tiên và một cách tự động self

15. Explain how to make a Python script executable on Unix?

16. What is the output of -12 % 10 and -12 // 10.

8 và -2

17. Why shouldn't you make the default arguments an empty list?

18. What is the yield keyword used for in Python?

Lệnh này cách sử dụng gần giống với lệnh return, tuy nhiên nó khác return ở chỗ trả về một object thì yield sẽ trả về một generator.

19. What is an iterator in Python? Can you write an example?

Iterator ở khắp mọi nơi trong Python, bên trong các vòng lặp, comprehension, generator…

Nó đơn giản chỉ là các đối tượng cho phép ta lấy từng phần tử, bất cứ khi nào bạn sử dụng vòng lặp hay các kĩ thuật để có được giá trị một nhóm phần tử ở một thời điểm nào đó.

20. What is the difference between \_\_iter\_\_ and \_\_next\_\_?

21. What is unittest module in Python? How to write tests in Python?

22. What are metaclasses in Python?