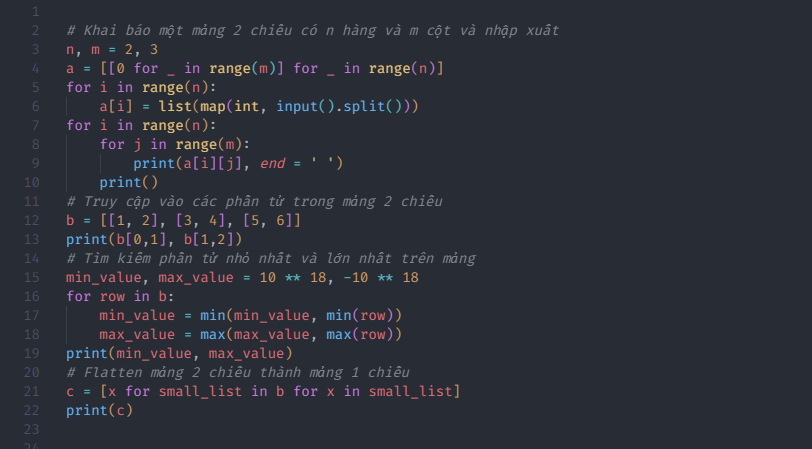
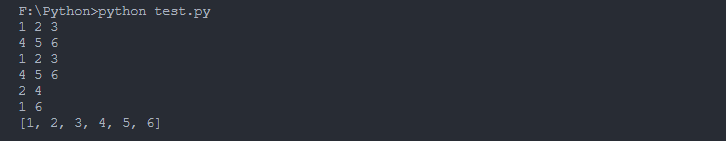
**BÁO CÁO TIẾN ĐỘ HỌC TẬP TUẦN 3**

**Nguyễn Nguyên Hảo**

1. **Mảng 2 chiều**

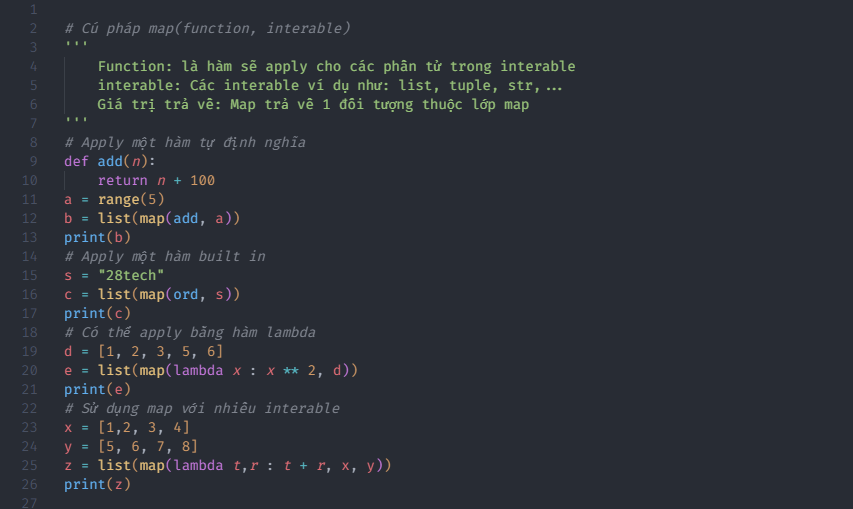
*Sử dụng Nested List để tạo thành mảng 2 chiều*





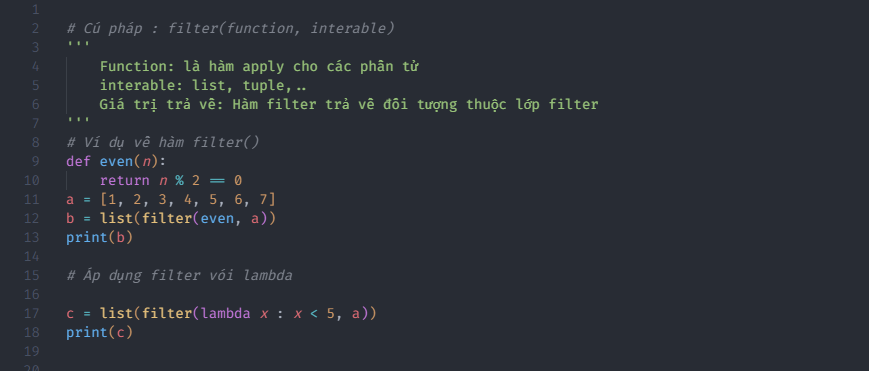
1. **Map và Filter**
2. **Map()**

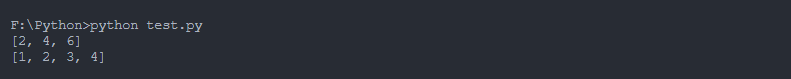
*Map là 1 hàm trong Python có chức năng apply một hàm có sẵn cho tất cả các phần tử trong interable (list, tuple, str, ..)*



1. **Filter()**

*Filter sử dụng để trích xuất các phần tử khi apply 1 hàm nào đó với phần tử đó mà trả về giá trị True*

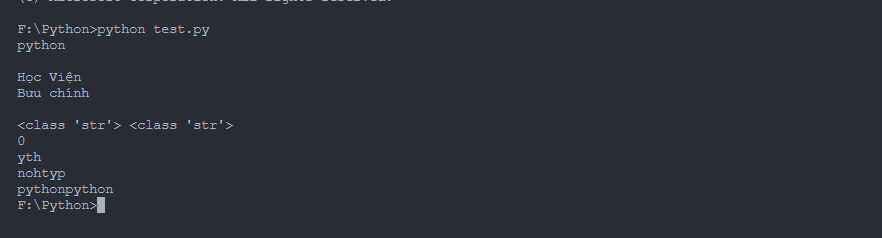


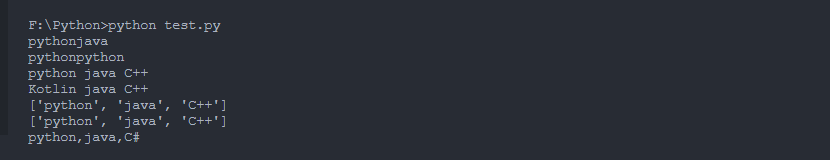
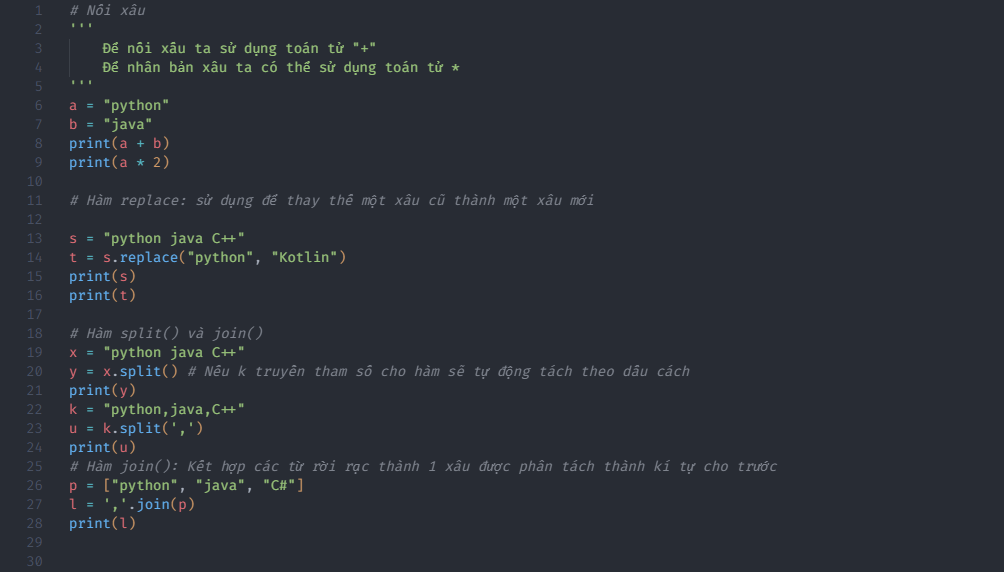


1. **String**

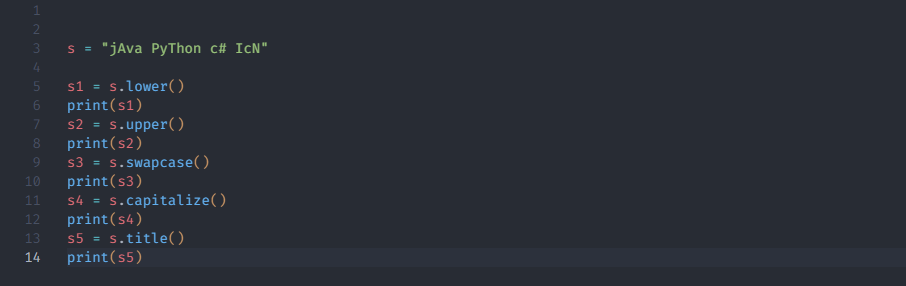
*Xâu kí tự là chuỗi các kí tự được đặt giữa dấu nháy đơn hoặc nháy kép*

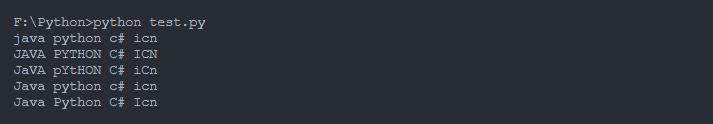


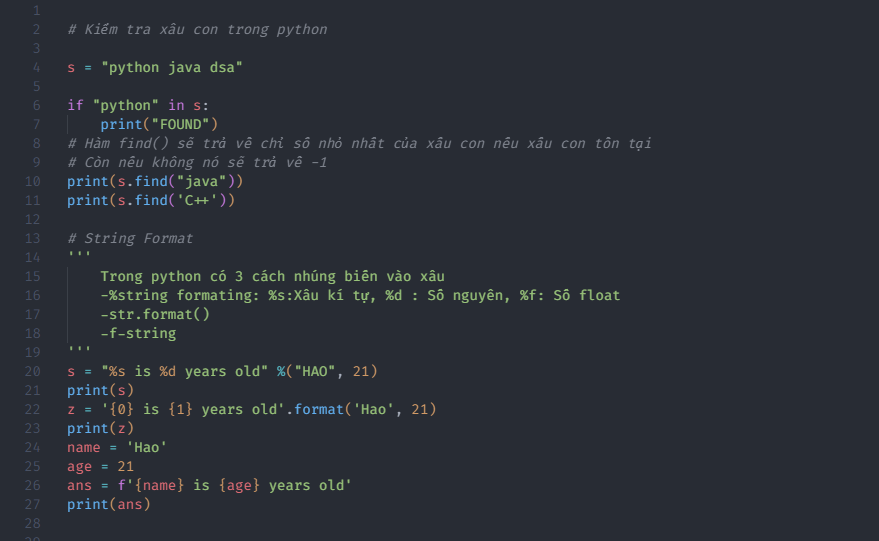


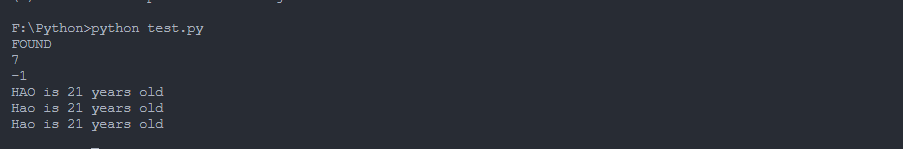


|  |  |
| --- | --- |
| **Các hàm** | **Chức năng** |
| Upper() | Chuyển sang in hoa |
| Lower() | Chuyển sang in thường |
| Capitalize() | Viết hoa chữ cái đầu của xâu |
| Swapcase() | Chuyển chữ hoa thành chữ thường và ngược lại |
| Title() | Viết hoa chữ cái đầu của từng từ còn các chữ còn lại viết thường |



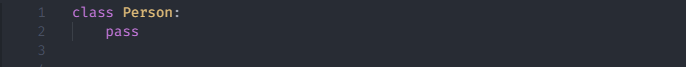






1. **OOP với Python**
2. **Class và Object**

*+ Để xây dụng Class ta sử dụng từ khóa class theo sau bởi tên class*.



*+ Câu lệnh pass được đặt trong lớp Person để chỉ ra lớp persion rỗng*

*+ Cú pháp để tạo ra một đối tượng: p = Person()*





*+ Phương thức \_\_init()\_\_ là một phương thức đặc biệt của lớp, phương thức này sẽ tự động gọi khi bạn tạo ra một đối tượng của list*

*+ Tham số self đề cập đến chính đối tượng hiện tại, self luôn được khai báo như là tham số đầu tiên của các phương thức trong class*

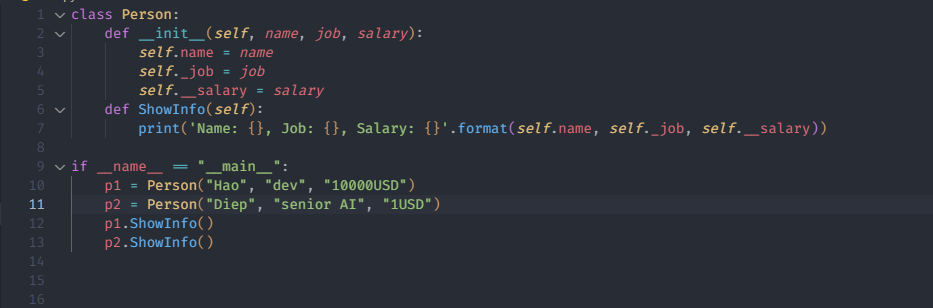
*+ Phương thức \_\_init()\_\_ tham số đầu tiên là self. Nó được sử dụng để truy vấn và thiết lập các thuộc tính của đối tượng*

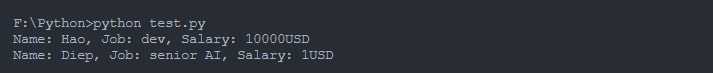
*+ Thuộc tính và phương thức:*

*Thuộc tính là những thứ riêng lẻ để phân biệt đối tượng này với đối tượng khác*

*Để truy cập vào thuộc tính ta sử dụng toán tử ‘.’*

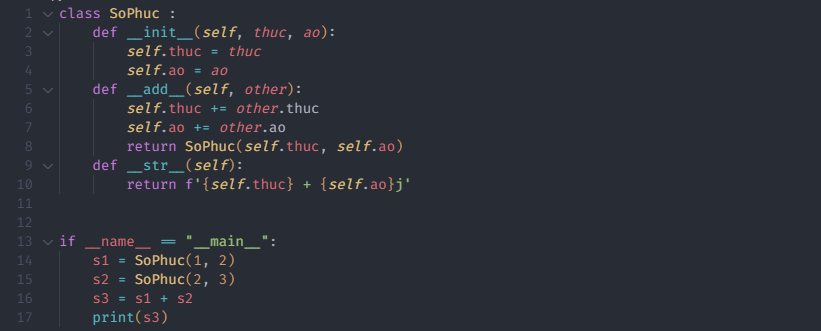
*Có 2 loại thuộc tính : instance attribute và class attribute*





**Nạp chồng toán tử trong Python:**

+ Hàm \_\_add\_\_:

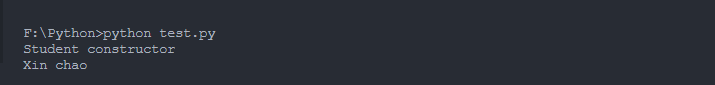




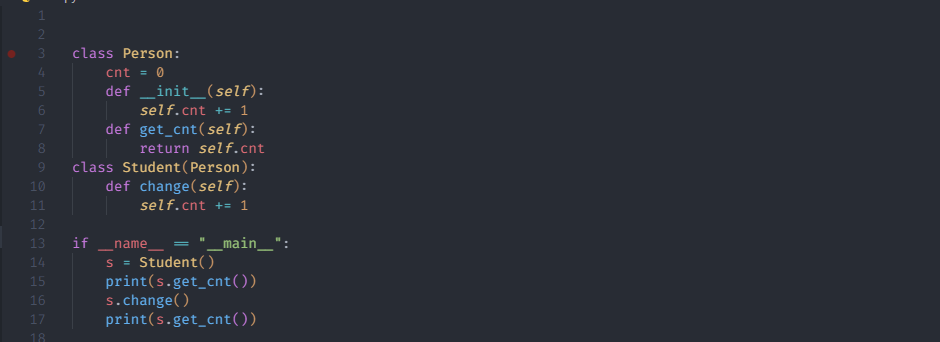
**+ Kế thừa trong Python**

*Lớp trong python có thể mở rộng, tạo một lớp mới từ một lớp cũ mà vẫn bảo toàn được những đặc điểm của lớp cũ. Quá trình này gọi là kế thừa, kế thừa liên quan đến khái niệm là lớp cha và lớp con*



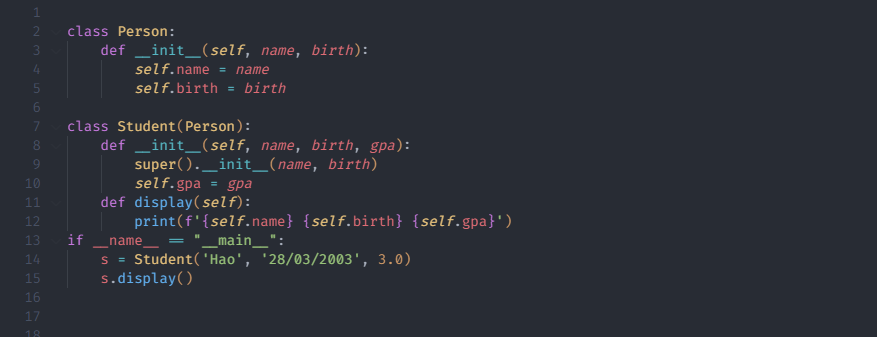


*+ Khi lớp con kế thừa lớp cha, lớp con sẽ có các thuộc tính, phương thức tương tự như lớp cha*



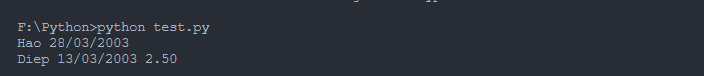


**+ Subclassing**



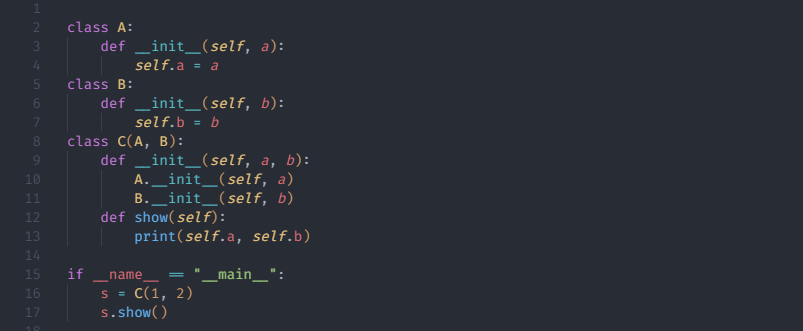
**+ Ghi đè hàm**





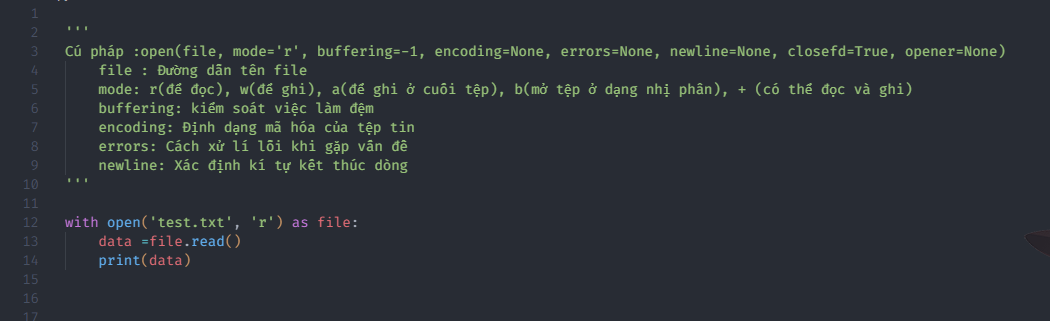
* **Các kiểu kế thừa:**

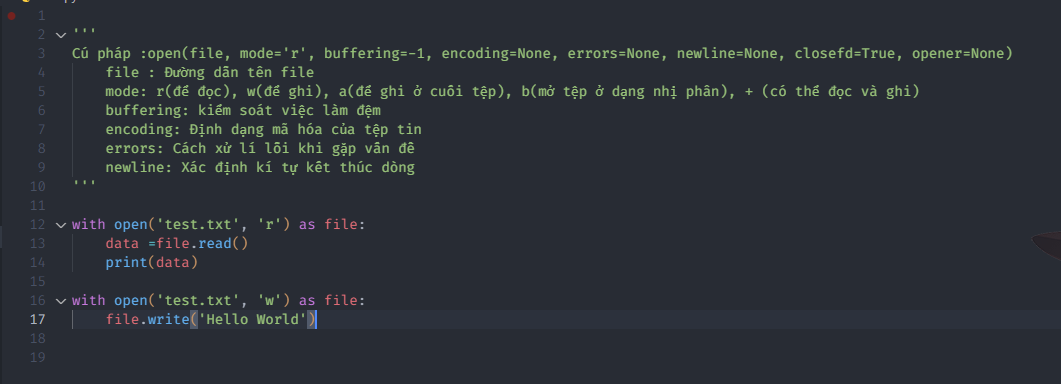
**+ Đa kế thừa:**

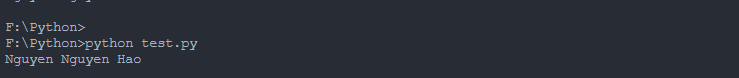


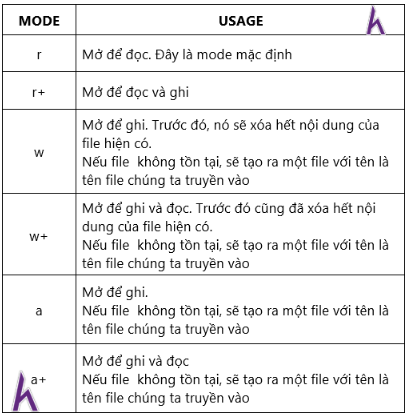


1. **Xử lí file trong Python**
2. **Hàm open() trong Python**









1. **Đóng file**

Cú pháp f.close()

1. **Các phương thức**

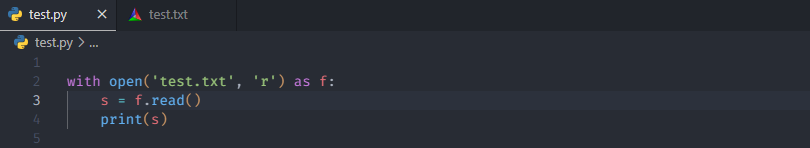
*+ Phương thức read()*

*Cú pháp: read(size=-1). Nếu tham số size bị bỏ trống hoặc là 1 số âm thì nó sẽ tự động đọc đến cuối file.*

*Nếu truyền vào tham số size thì hàm sẽ đọc n kí tự với n = size*

*Nếu đọc xong hàm sẽ trả về dạng chuỗi*

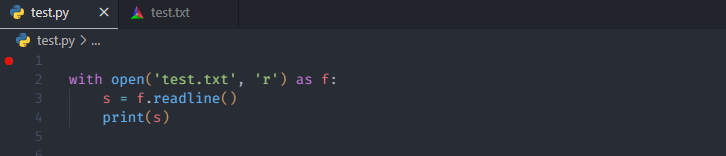


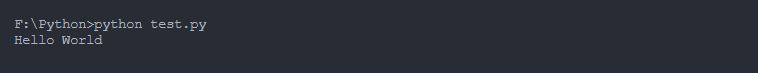




*Nếu không đọc được gì hàm sẽ trả về chuỗi rỗng*

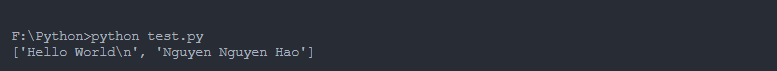
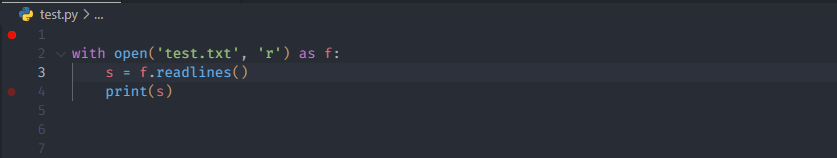
*+ Phương thức readline()*



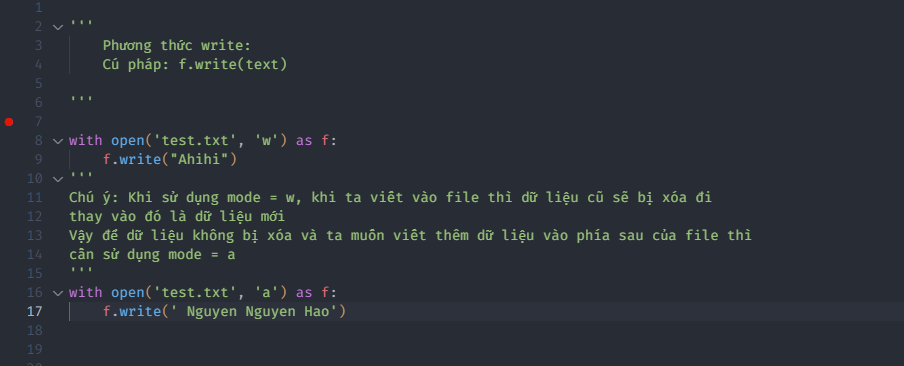


*+ Phương thức readlines()*

*Đọc hết toàn bộ file sau đó cho vào 1 list*



*+ Phương thức write()*





*+ Phương thức seek()*

