**Cú pháp của hàm Python**

**def ten\_ham(các tham số/đối số):**

**Các câu lệnh**

Về cơ bản, một định nghĩa hàm Python sẽ bao gồm các thành phần sau:

1. Từ khóa def: Đánh dấu sự bắt đầu của tiêu đề hàm.
2. ten\_ham: Là định danh duy nhất dành cho hàm. Việc đặt tên hàm phải tuân thủ theo [quy tắc viết tên và định danh trong Python](https://quantrimang.com/tu-khoa-va-dinh-danh-trong-python-142415).
3. Các tham số/đối số: Chúng ta truyền giá trị cho hàm thông qua các tham số này. Chúng là tùy chọn.
4. Dấu hai chấm (:): Đánh dấu sự kết thúc của tiêu đề hàm.
5. docstring: Chuỗi văn bản tùy chọn để mô tả chức năng của hàm.
6. Các câu lệnh: Một hoặc nhiều lệnh Python hợp lệ tạo thành khối lệnh. Các lệnh này phải có cùng một mức [thụt đầu dòng](https://quantrimang.com/cach-viet-lenh-thut-le-va-chu-thich-trong-python-142417) (thường là 4 khoảng trắng).
7. Lệnh return: Lệnh này là tùy chọn, dùng khi cần trả về giá trị từ hàm.

Lưu ý: Các tham số khi truyền vào ở hàm chỉ sử dụng trong hàm đó, không sử dụng được ở vị trí khác.

Ví dụ về Hàm:

Def hello(name):

print(name)

Def sum(a, b):

return a + b

**Mở File trong Python**

Trong Python, có một hàm được xây dựng sẵn phục vụ cho việc mở file: *open().* Hàm này trả về đối tượng file hay còn gọi là “handle” vì bạn

Cách mở file:

Cách 1:

F = open(“Tên file”, “Mode”);

# Đóng file:

F.close()

Cách 2:

With open (“Tên file”, “Mode”) as f:

Code….

# Đóng file:

f.close()

|  |  |
| --- | --- |
| **MODE** | **MÔ TẢ** |
| ‘r’ | Chế độ chỉ được phép đọc. |
| ‘r+’ | Chế độ được phép đọc và ghi |
| ‘rb’ | Mở file chế độ đọc cho định dạng nhị phân. Con trỏ tại phần bắt đầu của file |
| ‘rb+’  ‘r+b’ | Mở file để đọc và ghi trong định dạng nhị phân. Con trỏ tại phần bắt đầu của file |
| ‘w’ | Mở file để ghi. Nếu file không tồn tại thì sẽ tạo mới file và ghi nội dung, nếu file đã tồn tại thì sẽ bị cắt bớt (truncate) và ghi đè lên nội dung cũ |
| ‘w+’ | Mở file để đọc và ghi. Nếu file không tồn tại thì sẽ tạo mới file và ghi nội dung, nếu file đã tồn tại thì sẽ bị cắt bớt (truncate) và ghi đè lên nội dung cũ |
| ‘wb’ | Mở file để ghi cho dạng nhị phân. Nếu file không tồn tại thì sẽ tạo mới file và ghi nội dung, nếu file đã tồn tại thì sẽ bị cắt bớt (truncate) và ghi đè lên nội dung cũ |
| ‘wb+’  ‘w+b’ | Mở file để đọc và ghi cho dạng nhị phân. Nếu file không tồn tại thì sẽ tạo mới file và ghi nội dung, nếu file đã tồn tại thì sẽ bị cắt bớt (truncate) và ghi đè lên nội dung cũ |
| ‘a’ | Mở file chế độ ghi tiếp. Nếu file đã tồn tại rồi thì nó sẽ ghi tiếp nội dung vào cuối file, nếu file không tồn tại thì tạo một file mới và ghi nội dung vào đó. |
| ‘a+’ | Mở file chế độ đọc và ghi tiếp. Nếu file đã tồn tại rồi thì nó sẽ ghi tiếp nội dung vào cuối file, nếu file không tồn tại thì tạo một file mới và ghi nội dung vào đó. |
| ‘ab’, | Mở file chế độ ghi tiếp ở dạng nhị phân. Nếu file đã tồn tại rồi thì nó sẽ ghi tiếp nội dung vào cuối file, nếu file không tồn tại thì tạo một file mới và ghi nội dung vào đó. |
| ‘ab+’  ‘a+b | Mở file chế độ đọc và ghi tiếp ở dạng nhị phân. Nếu file đã tồn tại rồi thì nó sẽ ghi tiếp nội dung vào cuối file, nếu file không tồn tại thì tạo một file mới và ghi nội dung vào đó. |
| ‘x’ | Mở file chế độ ghi. Tạo file độc quyền mới (exclusive creation) và ghi nội dung, nếu file đã tồn tại thì chương trình sẽ báo lỗi |
| ‘x+’ | Mở file chế độ đọc và ghi. Tạo file độc quyền mới (exclusive creation) và ghi nội dung, nếu file đã tồn tại thì chương trình sẽ báo lỗi |
| ‘xb’ | Mở file chế độ ghi dạng nhị phân. Tạo file độc quyền mới và ghi nội dung, nếu file đã tồn tại thì chương trình sẽ báo lỗi |
| ‘xb+’  ‘x+b’ | Mở file chế độ đọc và ghi dạng nhị phân. Tạo file độc quyền mới và ghi nội dung, nếu file đã tồn tại thì chương trình sẽ báo lỗi |
| ‘b’ | Mở file ở chế độ nhị phân |
| ‘t’ | Mở file ở chế độ văn bản (mặc định) |

Các hàm thao tác với file:

**Dùng read(size)**

Sử dụng phương thức *read(size)* để lấy về dữ liệu có kích thước bằng size. Nếu để trống tham số này thì nó sẽ đọc hết file hoặc nếu file quá lớn thì nó sẽ đọc đến khi giới hạn của bộ nhớ cho phép

VD: data = open(“file.txt”, “r”).read() -> kết quả trả về là dữ liệu dạng str

**Dùng readline()**

Phương thức này cho phép đọc từng dòng trong file:

**Một số phương thức làm việc với File trong Python**

Có rất nhiều phương thức khác nhau để làm việc với file được tích hợp sẵn trong Python, trong đó có một vài phương thức đã được Quantrimang tìm hiểu ở trên.

Bảng dưới đây là danh sách đầy đủ các phương thức dưới dạng text, các bạn tham khảo thêm.

|  |  |
| --- | --- |
| **PHƯƠNG THỨC** | **MÔ TẢ** |
| close() | Đóng một file đang mở. Nó không thực thi được nếu tập tin đã bị đóng. |
| fileno() | Trả về một số nguyên mô tả file (file descriptor). |
| flush() | Xóa sạch bộ nhớ đệm của luồng file. |
| isatty() | Trả về TRUE nếu file được kết nối với một thiết bị đầu cuối. |
| read(n) | Đọc n kí tự trong file. |
| readable() | Trả về TRUE nếu file có thể đọc được. |
| readline(n=-1) | Đọc và trả về một dòng từ file. Đọc nhiều nhất n byte/ký tự nếu được chỉ định. |
| readlines(n=-1) | Đọc và trả về một danh sách các dòng từ file. Đọc nhiều nhất n byte/ký tự nếu được chỉ định. |
| seek(offset,from=SEEK\_SET) | Thay đổi vị trí hiện tại bên trong file. |
| seekable() | Trả về TRUE nếu luồng hỗ trợ truy cập ngẫu nhiên. |
| tell() | Trả về vị trí hiện tại bên trong file. |
| truncate(size=None) | Cắt gọn kích cỡ file thành kích cỡ tham số size. |
| writable() | Trả về TRUE nếu file có thể ghi được. |
| write(s) | Ghi s kí tự vào trong file và trả về. |
| writelines(lines) | Ghi một danh sách các dòng và file. |