**CHỨC NĂNG CỦA CÁC MÔ-ĐUN SAU ĐÂY VÀ CÓ VÍ DỤ KÈM THEO**

* math: Mô-đun mathtrong Python cung cấp nhiều hàm toán học và hằng số để hỗ trợ các phép toán, xử lý số học và tính toán khoa học.

VD:

+ math.pi: Giá trị Pi (3.14159...)

+ math.e: Giá trị của số Euler (2.71828...)

+ math.sqrt(x): Tính căn bậc hai của số x.



+ math.pow(x, y): Tính x sự tích lũy thừa và tương tự x \*\* y.

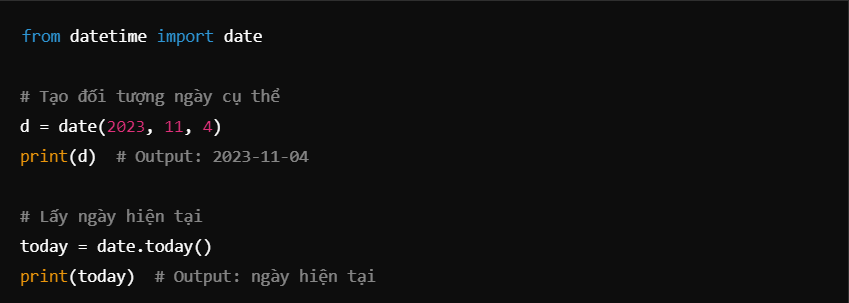


+ math.ceil(x): Làm tròn số (lấy giá trị nguyên lớn hơn hoặc bằng x).



* datetime: Mô-đun datetimetrong Python cung cấp các lớp và hàm để xử lý ngày, giờ và thời gian tính toán được phép. Nó rất hữu ích khi bạn làm việc với các thao tác như tính toán khoảng thời gian, lấy thời gian hiện tại hoặc định dạng ngày.

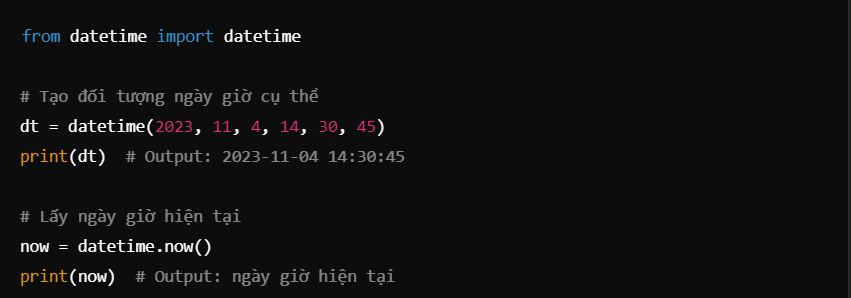
VD:

+ datetime.date: Quản lý các thông tin về ngày (năm, tháng, ngày). 

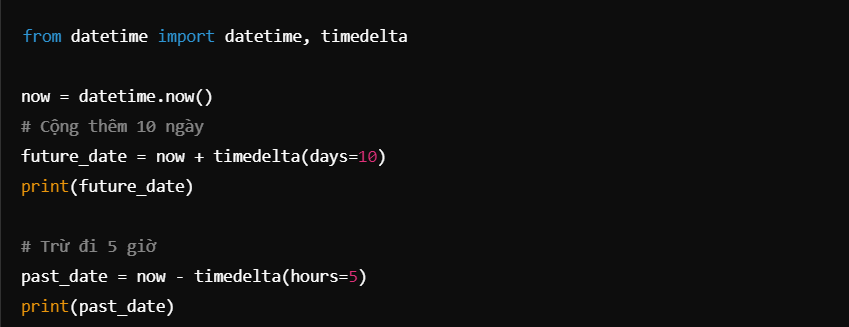
+ datetime.time: Quản lý thông tin về thời gian (giờ, phút, giây, micro giây).



+ datetime.datetime: Kết hợp cả ngày và giờ.



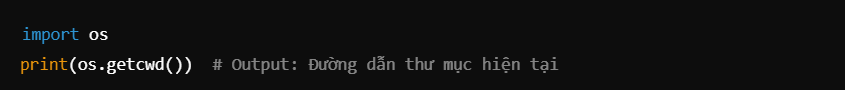
+ datetime.timedelta: Biểu diễn khoảng thời gian, rất hữu ích để tính toán khoảng cách giữa hai thời điểm.



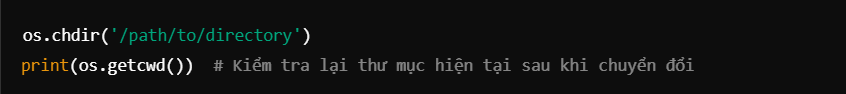
* os: Module os trong Python cung cấp các hàm để tương tác với hệ điều hành, cho phép bạn thực hiện các thao tác như làm việc với tệp và thư mục, quản lý quy trình, và truy xuất các biến môi trường.

VD:

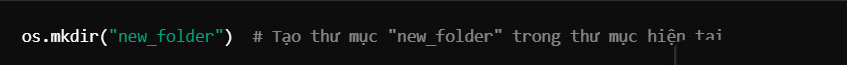
+ Lấy đường dẫn thư mục hiện tại: os.getcwd()



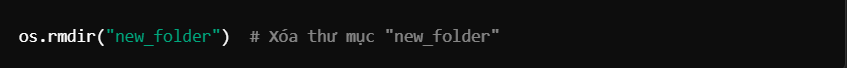
+ Thay đổi thư mục làm việc: os.chdir(path)



+ Tạo thư mục mới: os.mkdir(path)



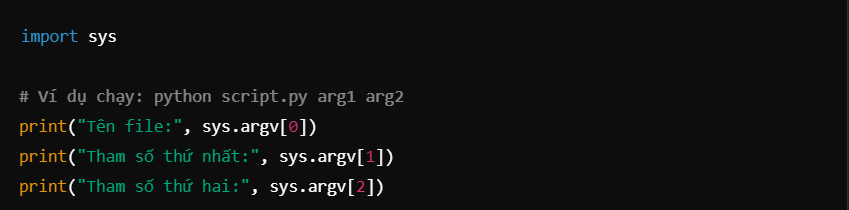
+ Xóa thư mục: os.rmdir(path)



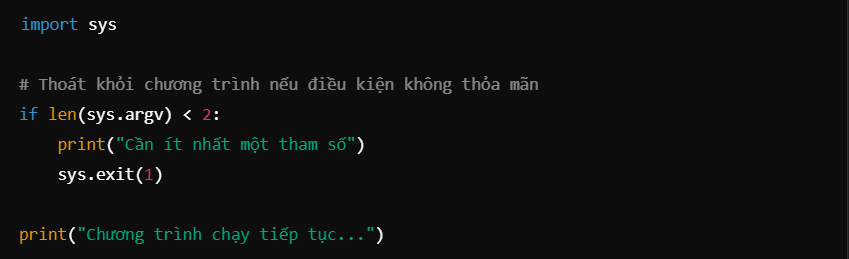
* sys: Module sys trong Python cung cấp các hàm và biến liên quan đến trình thông dịch Python và môi trường thực thi. Module này rất hữu ích cho việc quản lý các tham số dòng lệnh, kiểm soát dòng vào và ra, và truy xuất các thông tin về hệ thống.

VD:

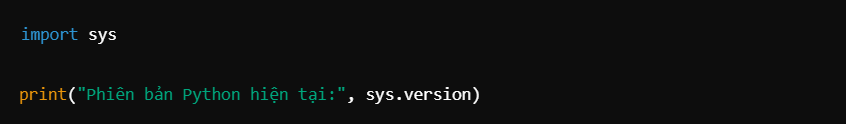
+ sys.argv: Danh sách chứa các tham số dòng lệnh khi chạy chương trình. Phần tử đầu tiên (sys.argv[0]) là tên của file Python.



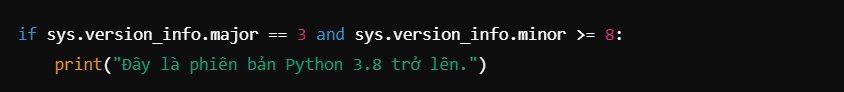
+ sys.exit([arg]): Dùng để thoát khỏi chương trình. Tham số arg là mã thoát, thường dùng 0 để chỉ chương trình kết thúc thành công, còn giá trị khác 0 để chỉ lỗi.



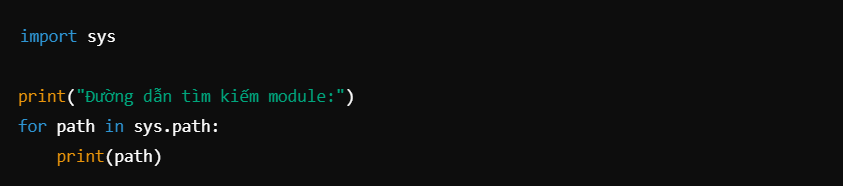
+ sys.version: Chuỗi chứa thông tin về phiên bản Python hiện tại.



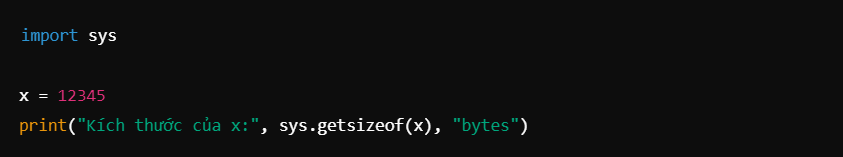
+ sys.version\_info: Thông tin chi tiết hơn về phiên bản Python, dưới dạng một tuple với các giá trị (major, minor, micro, releaselevel, serial).



+ sys.path: Danh sách các đường dẫn mà Python sẽ tìm kiếm module khi import.



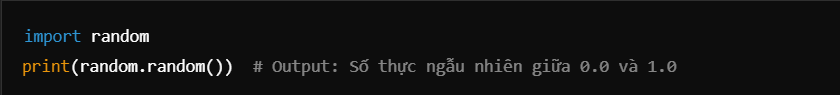
+ sys.getsizeof(object): Trả về kích thước của một đối tượng tính bằng byte.



* random: Module random trong Python cung cấp các hàm để sinh số ngẫu nhiên, lựa chọn ngẫu nhiên từ danh sách, trộn thứ tự các phần tử, và nhiều chức năng khác liên quan đến tính ngẫu nhiên.

VD:

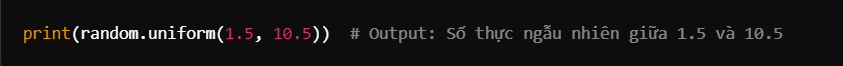
+ random.random(): Trả về một số thực ngẫu nhiên trong khoảng [0.0,1.0)[0.0, 1.0)[0.0,1.0).



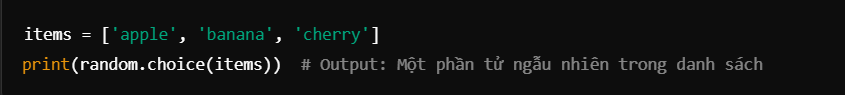
+ random.randint(a, b): Trả về một số nguyên ngẫu nhiên trong khoảng từ a đến b (bao gồm cả a và b).



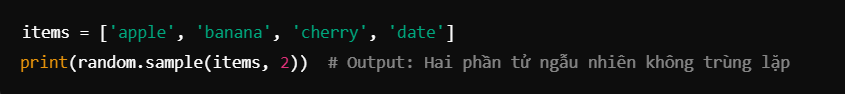
+ random.uniform(a, b): Trả về một số thực ngẫu nhiên trong khoảng từ a đến b.



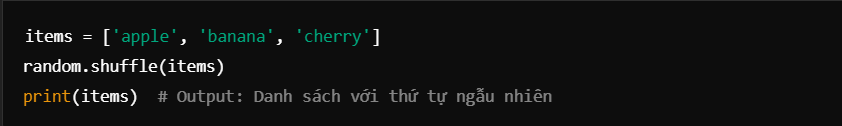
+ random.choice(sequence): Lấy ngẫu nhiên một phần tử từ danh sách hoặc chuỗi.



+ random.sample(population, k): Lấy mẫu k phần tử ngẫu nhiên từ một danh sách mà không lặp lại phần tử.



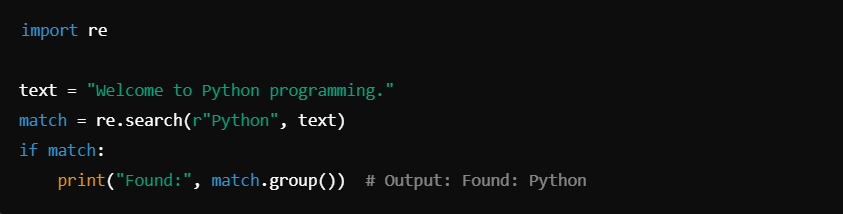
+ random.shuffle(sequence): Thay đổi thứ tự của các phần tử trong danh sách một cách ngẫu nhiên (in-place).



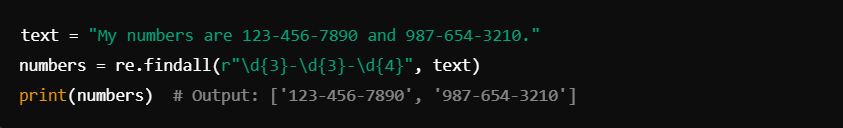
* re: Module re trong Python cung cấp các hàm cho phép làm việc với biểu thức chính quy (regular expressions), giúp bạn tìm kiếm, thay thế, và xử lý chuỗi theo các mẫu phức tạp.

VD:

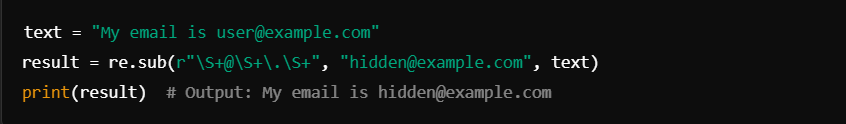
+ re.search(pattern, string): Tìm kiếm một mẫu (pattern) trong chuỗi. Trả về đối tượng Match nếu tìm thấy, ngược lại trả về None.



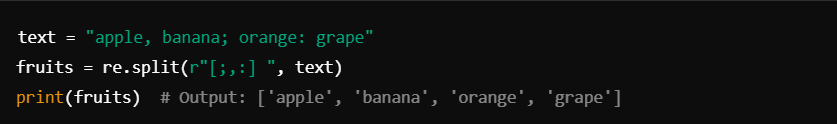
+ re.findall(pattern, string): Trả về danh sách tất cả các phần trong chuỗi khớp với mẫu.

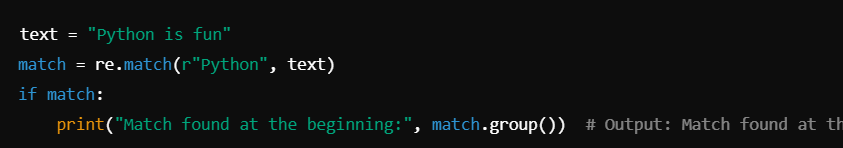


+ re.sub(pattern, replacement, string): Thay thế tất cả các phần khớp với mẫu trong chuỗi bằng replacement.



+ re.split(pattern, string): Tách chuỗi thành danh sách các phần tử, ngắt tại các vị trí khớp với mẫu.

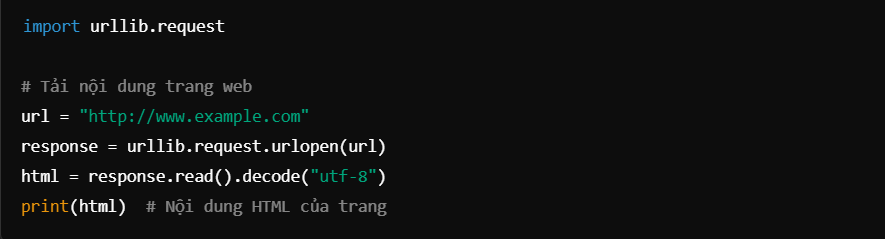


+ re.match(pattern, string): Kiểm tra xem chuỗi có bắt đầu bằng mẫu cho trước hay không. Nếu có, trả về đối tượng Match, nếu không trả về None. 

* urllib: Module urllib trong Python được sử dụng để làm việc với các URL, bao gồm các chức năng để truy xuất dữ liệu từ internet, phân tích URL, và mã hóa hoặc giải mã dữ liệu URL. Đây là một công cụ hữu ích để tạo các ứng dụng web hoặc thu thập dữ liệu từ các nguồn trực tuyến.

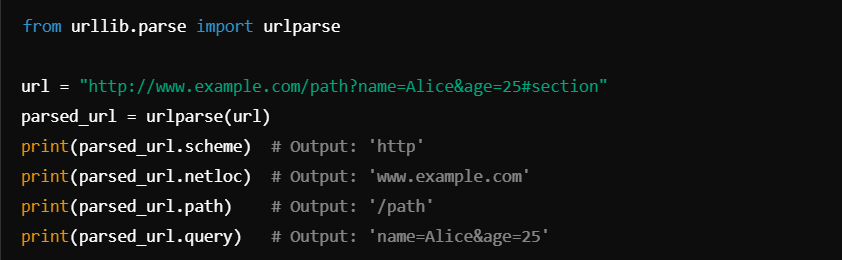
VD:

+ urllib.request: Để gửi các yêu cầu HTTP.

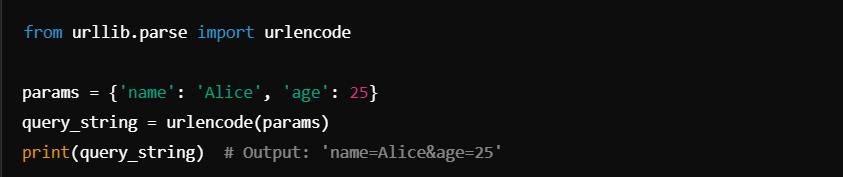


+ urllib.parse: Để phân tích và xử lý các URL. urllib.parse cung cấp các hàm để phân tích, kết hợp, và mã hóa URL

1. urllib.parse.urlparse(url): Phân tích một URL thành các phần như scheme, netloc, path, params, query, fragment.



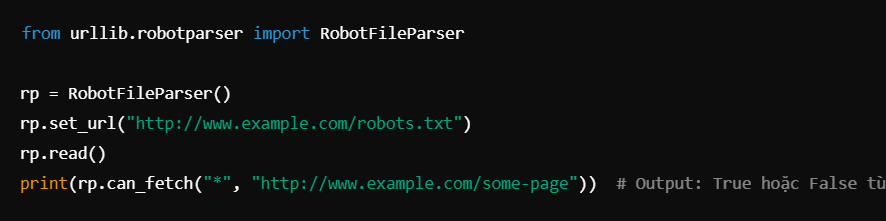
1. urllib.parse.urlencode(query, doseq=False): Mã hóa một dictionary hoặc danh sách tuple thành chuỗi query để sử dụng trong URL.



+ urllib.error: Để xử lý các lỗi liên quan đến HTTP.



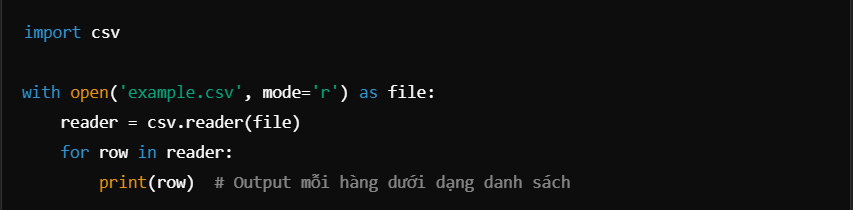
+ urllib.robotparser: Để phân tích tệp robots.txt, kiểm tra xem một URL có được phép thu thập không.



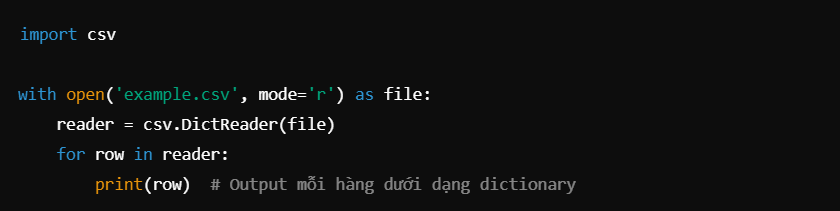
* csv: Module csv trong Python được sử dụng để đọc và ghi các tệp CSV (Comma-Separated Values) – một định dạng phổ biến cho việc lưu trữ dữ liệu dạng bảng, với các trường dữ liệu phân tách bằng dấu phẩy hoặc dấu phân cách khác.

VD:

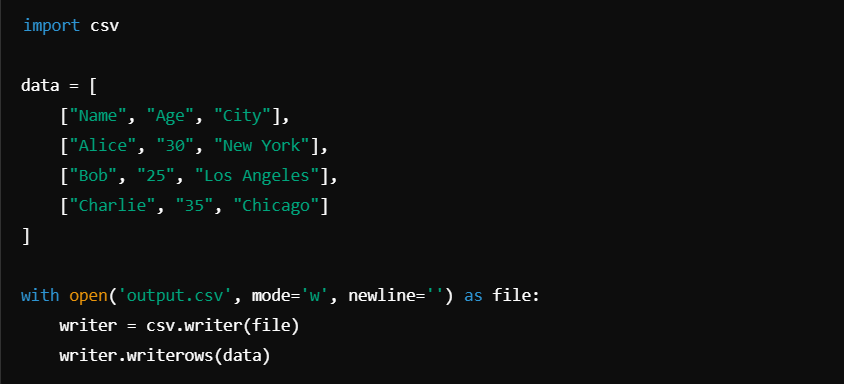
+ csv.reader: Dùng để đọc dữ liệu từ tệp CSV. Mỗi dòng trong tệp sẽ được đọc thành một danh sách các giá trị.



+ csv.DictReader: Đọc mỗi dòng từ tệp CSV thành một dictionary, với khóa là tên cột (dòng tiêu đề) trong tệp.



+ csv.writer: Được sử dụng để ghi dữ liệu vào một tệp CSV. Dữ liệu cần ghi là các danh sách, mỗi danh sách là một hàng dữ liệu.



+ csv.DictWriter: Ghi các dictionary vào tệp CSV, cho phép xác định tên cột ngay từ đầu.

