

Giới Thiệu Dự Án

Bạn đã gần như hoàn thành khóa học Python, bạn đã học được những kiến thức cơ bản về cú pháp Python, bao gồm các **hàm (functions)**, các **điều kiện (conditionals)**, và các vòng lặp **for** và **while**. Bạn cũng đã học cách sử dụng các kiểu dữ liệu phổ biến nhất, như số nguyên (integers), chuỗi (strings), danh sách (lists), và từ điển (dictionaries). Bạn thậm chí còn học được về **lập trình hướng đối tượng (object-oriented programming)**.

Bây giờ, chúng ta sẽ kết hợp tất cả những kiến thức này để giải quyết những vấn đề thú vị và hấp dẫn hơn. Chúng ta sẽ tiếp cận những thách thức mới như thể chúng là những vấn đề thực tế mà chúng ta cần giải quyết bằng một đoạn mã (script). Bằng cách này, chúng ta sẽ thấy một ví dụ khác về cách những kỹ năng lập trình này có thể làm cho công việc của chúng ta trong công việc IT nhanh chóng và hiệu quả hơn. Trong nội dung tiếp theo, chúng ta sẽ xem xét cách giải quyết một vấn đề phức tạp hơn bằng cách viết một script từ đầu. Để làm điều đó, chúng ta sẽ đi từng bước theo một cách tiếp cận được khuyến nghị khi đối mặt với những thách thức phức tạp hơn.

PROBLEM STATEMENT

Hãy tưởng tượng rằng bạn là một chuyên gia IT làm việc trong một công ty vừa và nhỏ, người quản lý của bạn muốn tạo một báo cáo hàng ngày để theo dõi việc sử dụng các máy. Cụ thể, **ông chủ muốn biết người dùng hiện tại đang kết nối với máy nào, công việc của bạn là tạo báo cáo đó.**

Trong công ty của bạn, có một hệ thống thu thập mọi sự kiện xảy ra trên các máy trên mạng. **Trong số nhiều sự kiện được thu thập, nó ghi lại mỗi lần một người dùng đăng nhập hoặc đăng xuất khỏi máy tính.** Với thông tin này, chúng ta muốn viết một đoạn mã (script) tạo báo cáo về người dùng nào đang đăng nhập vào máy nào tại thời điểm đó.

Trước khi chúng ta lao vào giải quyết vấn đề đó, **chúng ta cần biết thông tin nào sẽ được sử dụng làm đầu vào và thông tin nào sẽ là đầu ra.** Chúng ta có thể xác định điều này bằng cách xem xét hệ thống còn lại nơi script của chúng ta sẽ hoạt động.

Trong kịch bản báo cáo của chúng ta, đầu vào là một danh sách các sự kiện, mỗi sự kiện là một thể hiện của lớp sự kiện (event class). Một lớp sự kiện chứa **ngày khi sự kiện xảy ra, tên của máy nơi nó xảy ra, người dùng liên quan, và loại sự kiện.** Trong kịch bản này, chúng ta **quan tâm đến loại sự kiện đăng nhập và đăng xuất.**

Tiếp Cận Vấn Đề

Chúng ta cần biết tên chính xác của các thuộc tính, nếu không, chúng ta sẽ không thể truy cập chúng. **Các thuộc tính được gọi là ngày (date), người dùng (user), máy (machine), và loại (type).** **Các loại sự kiện** là chuỗi và những loại mà chúng ta quan tâm đến là **đăng nhập (login) và đăng xuất (logout).**

Với điều đó, **chúng ta đã có đủ thông tin về đầu vào của script của mình.** Script của chúng ta sẽ nhận một danh sách các đối tượng sự kiện và chúng ta sẽ truy cập vào các thuộc tính của sự kiện. Chúng ta **sau đó sử dụng thông tin đó để biết người dùng hiện tại có đang đăng nhập vào máy hay không.**

Bây giờ, **hãy nói về đầu ra.** Chúng ta muốn tạo một báo cáo liệt kê tất cả các tên máy và cho mỗi máy, danh sách người dùng hiện đang đăng nhập. Sau đó, **chúng ta muốn thông tin này được in ra màn hình.**

Chúng ta giờ đây có một ý tưởng khá tốt về những gì chúng ta cần làm. Chúng ta đã xác định phát biểu vấn đề của mình, đó là chúng ta cần xử lý một danh sách các đối tượng sự kiện sử dụng ngày, loại, máy, và thuộc tính người dùng của chúng để tạo một báo cáo liệt kê tất cả người dùng hiện đang đăng nhập vào các máy. Chúng ta đã bắt đầu một cách tốt đẹp. **Bước tiếp theo mà chúng ta sẽ thực hiện là nghiên cứu để xác định cách tốt nhất để thực sự làm điều này.**

TIẾN HÀNH NGHIÊN CỨU (RESEARCH)

Vậy chúng ta đã có một tổng quan về vấn đề giúp chúng ta hiểu rõ vấn đề và tập trung vào cách tiếp cận. Chúng ta biết rằng chúng ta **cần đưa vào một danh sách các đối tượng sự kiện và đánh giá các thuộc tính của những đối tượng này để tạo ra một báo cáo về tất cả người dùng hiện đang đăng nhập vào một máy.** Bây giờ là thời điểm chúng ta bắt đầu bước 2, nghiên cứu. Chúng ta sẽ xem xét tất cả các công cụ mà chúng ta có sẵn để giúp giải quyết vấn đề.

Để tìm ra những người dùng hiện đang đăng nhập vào máy,

Chúng ta cần kiểm tra thời điểm họ đăng nhập và thời điểm họ đăng xuất.

Nếu một người dùng đăng nhập vào một máy và sau đó đăng xuất, họ không còn đăng nhập vào nó nữa. Nhưng nếu họ chưa đăng xuất, họ vẫn còn đăng nhập. Tôi biết rằng chúng ta đang nói về những điều rất rõ ràng ở đây, nhưng trong lập trình, việc hiểu rõ các tham số là vô cùng quan trọng. Hơn nữa, việc biết điều này cho chúng ta thấy rằng để giải quyết vấn đề này một cách chính xác, việc xử lý các sự kiện theo thứ tự thời gian là vô cùng quan trọng. Nếu không, chúng ta có thể nhận được sự kiện đăng xuất trước sự kiện đăng nhập tương ứng và mã của chúng ta có thể thực hiện các hành động không thể dự đoán, và không ai thích mã không thể dự đoán. Vậy làm thế nào để chúng ta sắp xếp các danh sách trong Python? Chúng ta sẽ cần phải nghiên cứu.

Hãy gõ "sort lists in Python" vào công cụ tìm kiếm yêu thích của bạn và bạn sẽ nhận được một loạt kết quả đề cập đến phương thức sắp xếp danh sách (list sort method) và hàm sắp xếp (sorted function). Sự khác biệt giữa hai lựa chọn này là phương thức sắp xếp sẽ thay đổi danh sách mà nó thực hiện, trong khi hàm sắp xếp sẽ trả về một danh sách mới đã được sắp xếp. Ngoài ra, chúng hoạt động hoàn toàn giống nhau. Hãy xem sự khác biệt này trong hành động.

Đầu tiên, chúng ta sẽ tạo một danh sách số và gọi phương thức sắp xếp để sắp xếp danh sách. Bạn có thể thấy ở đây rằng các phần tử của danh sách đã được sắp xếp. Bây giờ hãy thử một ví dụ khác sử dụng hàm sắp xếp. Chúng ta sẽ tạo một danh sách tên. Sau đó, chúng ta sẽ in kết quả của hàm sắp xếp. Hãy in lại danh sách gốc để kiểm tra xem nó có thay đổi không. Vì vậy, bạn có thể thấy rằng danh sách gốc không hề bị thay đổi. Hàm sắp xếp đã trả về một danh sách mới đã được sắp xếp, nhưng danh sách gốc vẫn được giữ nguyên.

Chúng ta đã biết cách sắp xếp các phần tử trong Python. Đối với vấn đề này, việc thay đổi danh sách gốc là hoàn toàn chấp nhận được. Vì vậy, chúng ta sẽ sử dụng phương thức sắp xếp. Nhưng chờ đã, bạn thấy cách cả hai lựa chọn này sắp xếp danh sách theo thứ tự bảng chữ cái không? Đó là cách tiếp cận mặc định mà Python thực hiện. Nhưng nếu chúng ta muốn tổ chức danh sách của mình theo các tiêu chí khác nhau thì sao? Một lần nữa, nếu chúng ta xem xét tài liệu mà chúng ta tìm thấy trực tuyến, chúng ta sẽ thấy rằng phương thức sắp xếp có thể nhận một vài tham số. Một trong những tham số này được gọi là key, và nó cho phép chúng ta sử dụng một hàm như là key sắp xếp. Hãy thử điều này trên danh sách tên của chúng ta. Thay vì sắp xếp theo thứ tự bảng chữ cái, chúng ta có thể sắp xếp theo độ dài của mỗi chuỗi. Bạn nhớ không, chúng ta có thể sử dụng hàm nào để làm điều đó? Đúng rồi, chúng ta có thể truyền hàm len làm key.

Chúng ta đã biết cách sắp xếp các phần tử của một danh sách dựa trên giá trị trả về của một hàm. Trong tình huống báo cáo của chúng ta, chúng ta biết rằng các phần tử của chúng ta sẽ là các thể hiện của lớp sự kiện và chúng ta muốn sắp xếp theo ngày, đó là một thuộc tính của lớp sự kiện. Một cách chúng ta có thể làm điều này là viết một hàm được gọi là get_event_date, nó trả về ngày được lưu trữ trong đối tượng sự kiện. Chúng ta cũng có thể tạo nó như một phương thức trong lớp sự kiện nếu chúng ta có quyền chỉnh sửa lớp. Nhưng vì chúng ta đang làm việc với một hệ thống lớn hơn tạo ra những sự kiện này, chúng ta sẽ cho rằng chúng ta không thể chỉ thêm một phương thức vào lớp. Vì vậy, chúng ta sẽ tạo hàm của chúng ta thay thế. Điều đó có ý nghĩa không?

Hãy nhớ rằng có nhiều con đường khác nhau mà chúng ta có thể đi để giải quyết vấn đề này. Nhưng một số con đường tốt hơn những con đường khác. Vì vậy, việc hiểu rõ tại sao chúng ta chọn những lựa chọn mà chúng ta đã chọn là rất quan trọng. Hãy dành một ít thời gian tự mình để khám phá các khả năng và hiểu rõ những gì chúng ta đang làm. Trong nội dung tiếp theo, chúng ta sẽ đi sâu vào kế hoạch của chúng ta để xây dựng script của chúng ta.

KẾ HOẠCH (PLANNING)

Chúng ta đã xác định phát biểu vấn đề của mình và sau đó chúng ta đã nghiên cứu các lựa chọn để tìm ra những công cụ chúng ta có sẵn và chúng tốt nhất cho công việc. Bây giờ đã đến lúc chúng ta lên kế hoạch cho cách tiếp cận của mình.

Vậy chúng ta biết rằng đầu vào của chúng ta sẽ là một danh sách các sự kiện và chúng ta cần xác định người dùng nào đang đăng nhập vào máy. Chúng ta cũng biết rằng chúng ta cần sắp xếp các sự kiện theo thời gian để chúng ta có thể xử lý chúng đúng cách.

Dựa trên những điều này, chúng ta có thể bắt đầu tạo ra một dòng chảy cho script của mình. Đầu tiên, chúng ta cần lấy danh sách các sự kiện. Sau đó, chúng ta cần sắp xếp các sự kiện này theo thời gian. Một khi chúng ta đã làm điều đó, chúng ta có thể bắt đầu xử lý từng sự kiện.

- Chúng ta sẽ kiểm tra nếu một sự kiện là một sự kiện đăng nhập, và nếu nó là, thì chúng ta sẽ thêm nó vào một danh sách các người dùng đang đăng nhập.
- Nếu một sự kiện là một sự kiện đăng xuất, thì chúng ta sẽ xóa nó khỏi danh sách.
- Cuối cùng, chúng ta sẽ in danh sách các người dùng đang đăng nhập.

Hãy vẽ biểu đồ này ra giấy để chúng ta có thể thấy nó một cách rõ ràng. Điều này sẽ giúp chúng ta trong khi chúng ta xây dựng script của mình, và nó cũng sẽ làm cho nó dễ dàng hơn để giải thích cho người khác những gì chúng ta đang làm.

Trong nội dung tiếp theo, chúng ta sẽ bắt đầu viết mã cho script của chúng ta. Hãy chắc chắn rằng bạn đã hiểu tất cả những điều mà chúng ta đã thảo luận cho đến nay và đã vẽ biểu đồ dòng chảy của mình. Hãy dành thêm thời gian để thực hành nếu bạn cần.

CODING VÀ SỬA LỖI (CODING AND DEBUGGING)

Đến lúc chúng ta thực sự viết mã.

- Đầu tiên, chúng ta cần tạo một danh sách sự kiện. Chúng ta có thể giả sử rằng chúng ta nhận được danh sách này từ hệ thống của chúng ta.
- Chúng ta cũng sẽ viết hàm `get_event_date` để sắp xếp các sự kiện theo thời gian. Với những thứ này ở chỗ, chúng ta có thể bắt đầu xây dựng phần xử lý sự kiện của script của mình.
- Chúng ta cần lưu trữ danh sách các người dùng đang đăng nhập và sau đó xử lý từng sự kiện một, thêm hoặc xóa người dùng từ danh sách này dựa trên loại sự kiện.

Nếu chúng ta làm mọi thứ đúng, danh sách người dùng cuối cùng của chúng ta sẽ là người dùng đang đăng nhập vào hệ thống.

Chúng ta cũng cần nhớ rằng quá trình viết mã không chỉ bao gồm việc gõ các lệnh vào máy tính.

VIẾT SCRIPT (WRITING THE SCRIPT)

Chúng ta đã đi được một quãng đường dài để đến đây, hãy nhanh chóng tóm tắt những gì chúng ta biết cho đến thời điểm này. Chúng ta biết rằng chúng ta cần xử lý các sự kiện để tạo ra báo cáo. Chúng ta biết cách sắp xếp danh sách sự kiện theo thứ tự thời gian. Chúng ta biết rằng chúng ta sẽ lưu trữ dữ liệu trong một từ điển (dictionary) của các tập hợp (sets), mà chúng ta sẽ dùng để theo dõi ai đang đăng nhập ở đâu, và chúng ta sẽ có một hàm tạo từ điển và một hàm khác để in từ điển. Tôi nghĩ đó là tất cả. Bạn biết điều đó có nghĩa gì không? Chúng ta cuối cùng cũng đã sẵn sàng để viết mã của chúng ta. Vậy chúng ta bắt đầu thôi.

Xác định hàm trợ giúp (Defining the Helper Function)

Hãy bắt đầu bằng việc xác định hàm trợ giúp mà chúng ta sẽ sử dụng để sắp xếp danh sách. Chúng ta sẽ sử dụng hàm đơn giản này như tham số cho hàm sắp xếp để sắp xếp danh sách. Giờ đây, chúng ta đã sẵn sàng để bắt đầu viết mã hàm xử lý của chúng ta, mà chúng ta sẽ gọi là 'current users'. Bước đầu tiên là xác định hàm.

```
def get_event_date(event):  
    return event.date
```

Trong hàm, đầu tiên chúng ta sẽ sắp xếp các sự kiện của mình bằng cách sử dụng phương thức sắp xếp, và truyền hàm chúng ta vừa tạo như là key.

Tạo Từ Điển (Creating the Dictionary)

Bây giờ, trước khi chúng ta bắt đầu lặp qua danh sách các sự kiện của mình, chúng ta cần tạo từ điển nơi chúng ta sẽ lưu trữ tên và người dùng của một máy.

```
def get_event_date(event):
    return event.date
def current_users(events):
    events.sort(key=get_event_date)
    machines = {}
```

Duyệt qua Danh sách Sự kiện (Iterating through the List of Events)

Bây giờ, chúng ta đã sẵn sàng để lặp qua danh sách sự kiện của mình. Tiếp theo, chúng ta muốn kiểm tra xem máy bị ảnh hưởng bởi sự kiện này có trong từ điển hay không. Nếu không, chúng ta sẽ thêm nó với một tập hợp trống làm giá trị.

```
def get_event_date(event):
    return event.date
def current_users(events):
    events.sort(key=get_event_date)
    machines = {}
    for event in events:
        if event.machine not in machines:
            machines[event.machine] = set()
        if event.type == "login":
            machines[event.machine].add(event.user)
        elif event.type == "logout":
            machines[event.machine].remove(event.user)
    return machines
```

Đối với các sự kiện đăng nhập, chúng ta muốn thêm người dùng vào danh sách, và đối với các sự kiện đăng xuất, chúng ta muốn loại bỏ người dùng khỏi danh sách. Để làm điều này, chúng ta sẽ sử dụng các phương thức thêm (add) và loại bỏ (remove), những phương thức này thêm và loại bỏ các phần tử từ một tập hợp.

Tạo Báo Cáo (Generating the Report)

Một khi chúng ta đã hoàn tất việc lặp qua danh sách các sự kiện, từ điển sẽ chứa tất cả các máy mà chúng ta đã thấy như là các key, với một tập hợp chứa các người dùng hiện tại của các máy như là giá trị, hàm này trả về từ điển. Chúng ta sẽ xử lý việc in ấn trong một hàm khác.

Chúng ta đã có từ điển sẵn sàng, và chúng ta cần in nó. Để làm điều đó, chúng ta sẽ tạo một hàm mới được gọi là 'generate-report'. Trong báo cáo của chúng ta, chúng ta muốn lặp qua các key và value trong từ điển. Để làm điều đó, chúng ta sẽ sử dụng phương thức items, phương thức này trả về cả key và value cho mỗi cặp trong từ điển.

Trước khi chúng ta in bất cứ thứ gì, chúng ta muốn đảm bảo rằng chúng ta không in bất kỳ máy nào mà không có ai đang đăng nhập. Điều này có thể xảy ra nếu một người dùng đăng nhập và sau đó đăng xuất. Để tránh điều đó, chúng ta yêu cầu máy tính chỉ in khi tập hợp người dùng có nhiều hơn không phần tử. Chúng ta đã nói trước đó rằng chúng ta muốn in tên máy, theo sau là người dùng đã đăng nhập vào máy, được phân tách bởi dấu phẩy. Hãy tạo chuỗi người dùng đã đăng nhập cho máy đó bằng cách sử dụng phương thức join. Bây giờ, chúng ta có thể tạo chuỗi mà chúng ta muốn sử dụng phương thức định dạng.

Chúng ta đã viết tất cả các hàm mà chúng ta cần để giải quyết vấn đề của mình. Hãy đảm bảo rằng bạn hiểu rõ, không chỉ là hàm nào chúng ta đang sử dụng, mà còn tại sao chúng ta sử dụng nó. Nếu có bất kỳ thứ gì hơi mơ hồ, hãy nhớ rằng diễn đàn thảo luận luôn sẵn sàng để bạn yêu cầu giúp đỡ. Mọi thứ sẽ trở nên thú vị trong phần tiếp theo. Chúng ta sẽ thực thi mã này và xem nó có hoạt động không.

```
def get_event_date(event):
    return event.date
def current_users(events):
```

```

events.sort(key=get_event_date)
machines = {}
for event in events:
    if event.machine not in machines:
        machines[event.machine] = set()
    if event.type == "login":
        machines[event.machine].add(event.user)
    elif event.type == "logout":
        machines[event.machine].remove(event.user)
return machines

def generate_report(machines):
    for machine, users in machines.items():
        if len(users) > 0:
            user_list = ", ".join(users)
            print("{}: {}".format(machine, user_list))

```

Đoạn mã Python mà bạn đã cung cấp giải quyết vấn đề theo dõi đăng nhập và đăng xuất của người dùng trên nhiều máy tính. Bên dưới là một lời giải thích chi tiết về cách hoạt động của nó:

1. **get_event_date(event):** Hàm này chỉ đơn giản trả về thuộc tính date của đối tượng sự kiện đã cho. Nó được sử dụng như một key function để sắp xếp danh sách các sự kiện.
2. **current_users(events):** Hàm này xử lý danh sách các sự kiện. Nó sắp xếp các sự kiện theo thời gian, sau đó xử lý từng sự kiện một, cập nhật danh sách người dùng hiện tại trên mỗi máy tương ứng.
3. **generate_report(machines):** Hàm này tạo ra báo cáo từ danh sách người dùng hiện tại trên mỗi máy. Nếu có một hoặc nhiều người dùng đang hoạt động trên một máy, nó sẽ in ra tên máy và danh sách người dùng đó.

Để chạy mã này trong một notebook Jupyter, bạn sẽ cần tạo một số sự kiện mẫu để kiểm tra nó. Trước tiên, chúng ta cần xác định lớp Event. Sau đó, bạn có thể tạo một số sự kiện và thêm chúng vào danh sách.

Giả sử chúng ta có lớp Event như sau:

```

class Event:
    def __init__(self, date, event_type, machine, user):
        self.date = date
        self.type = event_type
        self.machine = machine
        self.user = user

```

Chúng ta có thể tạo một số sự kiện mẫu và thêm chúng vào danh sách:

```

events = [
    Event(date(2023, 6, 1), 'login', 'my_machine', 'my_user'),
    Event(date(2023, 6, 1), 'logout', 'my_machine', 'my_user'),
    ...
]

```

Sau khi có danh sách sự kiện, bạn có thể gọi hàm **current_users** để xử lý chúng:

```
machines = current_users(events)
```

Cuối cùng, gọi hàm **generate_report** để in ra báo cáo:

```
generate_report(machines)
```

Khi bạn chạy mã trên một máy khác, kết quả có thể khác nhau nếu như các sự kiện đăng nhập và đăng xuất diễn ra theo cách khác nhau trên các máy đó. Mã này không phụ thuộc vào máy bạn đang chạy nó, nhưng nó phụ thuộc vào dữ liệu bạn cung cấp cho nó (tức là danh sách các sự kiện).

Chúc các bạn thành công!

Võ Thanh Huy