Bộ vấn đề 5

Ngày phát hành: Thứ Tư, ngày 12 tháng 10 năm 2016.

Thời gian: 23h59, Thứ Tư, ngày 19 tháng 10 năm 2016.

Giới thiệu

Bạn sẽ sử dụng lập trình hướng đối tượng (các lớp và tính kế thừa) để xây dựng chương trình giám sát các nguồn cấp tin tức qua Internet. Chương trình của bạn sẽ lọc tin tức, thông báo cho người dùng khi nhận thấy một câu chuyện phù hợp với sở thích của người dùng đó (ví dụ: người dùng có thể quan tâm đến thông báo bất cứ khi nào một câu chuyện liên quan đến Red Sox được đăng).

Bộ vấn đề này không yêu cầu nhiều dòng mã! Chúng tôi khuyên bạn nên viết lời giải cho mỗi vấn đề trong khoảng 15-20 dòng mã (giải pháp cho một số vấn đề sẽ ngắn hơn thế nhiều). Nếu bạn thấy mình viết nhiều mã hơn thế, bạn nên đến thăm chúng tôi vào giờ hành chính để xem bạn có thể đơn giản hóa mọi thứ như thế nào.

Đây là hướng dẫn Python chính thức trên các lớp, các phần 9.1-9.7 (ngoại trừ 9.5.1) sẽ hữu ích cho pset này. Chương 8 trong sách giáo khoa cũng rất hữu ích.

Bắt đầu

Tải xuống và lưu

pset5.zip: Tệp zip chứa tất cả các tệp bạn cần, bao gồm:

- ps5.py, một khung để bạn điền vào
- ps5_test.py, bộ bài kiểm tra sẽ giúp bạn kiểm tra câu trả lời của mình
- triggers.txt, tệp cấu hình trình kích hoạt
- Feedparser.py, một mô-đun sẽ truy xuất và phân tích các nguồn cấp dữ liệu cho bạn
- project_util.py, một mô-đun chuyển đổi các đoạn HTML đơn giản thành văn bản thuần túy
- mtTkinter.py, một mô-đun xử lý giao diện đồ họa

Ba mô-đun (feedparser.py, project_util.py, mtTkinter.py) cần thiết để giải quyết vấn đề này nhưng bạn sẽ không cần phải sửa đổi hoặc hiểu chúng.

Bộ lọc nguồn cấp dữ liệu RSS

Tổng quan về RSS

Các trang web tin tức, chẳng hạn như Google News, có nội dung được cập nhật theo lịch trình không thể đoán trước. Một cách tẻ nhạt để theo dõi nội dung thay đổi này là tải trang web lên trình duyệt của bạn và nhấn nút làm mới định kỳ.

May mắn thay, quá trình này có thể được sắp xếp hợp lý và tự động hóa bằng cách kết nối với RSS của trang web

nguồn cấp dữ liệu, sử dụng trình đọc nguồn cấp dữ liệu RSS thay vì trình duyệt web. Trình đọc RSS (ví dụ: Sage) sẽ định kỳ thu thập và thu hút sự chú ý của bạn đến nội dung cập nhật.

RSS là viết tắt của "Cung cấp thực sự đơn giản." Nguồn cấp dữ liệu RSS bao gồm dữ liệu (thay đổi định kỳ) được lưu trữ trong tệp định dạng XML nằm trên máy chủ web. Đối với bộ bài tập này, các chi tiết không quan trọng. Bạn không cần biết XML là gì và cũng không cần biết cách truy cập các tệp này qua mạng. Chúng tôi đã đảm nhiệm việc truy xuất và phân tích tệp XML cho bạn.

Thiết kế cấu trúc dữ liệu

Cấu trúc nguồn cấp dữ liệu RSS: Google Tin tức

Trước tiên, hãy nói về một nguồn cấp dữ liệu RSS cụ thể: Google News. URL của nguồn cấp dữ liệu Google Tin tức là: http://news.google.com/?output=rss

Nếu bạn cố tải URL này trong trình duyệt của mình, bạn có thể sẽ thấy phần giải thích của trình duyệt về mã XML do nguồn cấp dữ liệu tạo ra. Bạn có thể xem nguồn XML bằng chức năng "Xem nguồn trang" của trình duyệt, mặc dù chức năng này có thể không có nhiều ý nghĩa đối với bạn. Tóm lại, bất cứ khi nào bạn kết nối với nguồn cấp dữ liệu RSS của Google Tin tức, bạn sẽ nhận được danh sách các mục. Mỗi mục trong danh sách này đại diện cho một mục tin tức duy nhất. Trong nguồn cấp dữ liệu của Google Tin tức, mọi mục nhập đều có các trường sau:

- hướng dẫn : Mã định danh duy nhất trên toàn cầu cho câu chuyện tin tức này.
- title : Tiêu đề của câu chuyện tin tức.
- mô tả : Một đoạn văn tóm tắt tin tức.
- link : Một liên kết đến một trang web có toàn bộ câu chuyện.
- pubDate : Ngày đăng tin
- danh mục : Danh mục tin tức, chẳng hạn như “Câu chuyện hàng đầu”

Khái quát hóa vấn đề

Việc này phức tạp hơn chúng tôi mong muốn một chút vì mỗi nguồn cấp dữ liệu RSS này có cấu trúc hơi khác một chút so với các nguồn cấp dữ liệu khác. Vì vậy, mục tiêu của chúng tôi là đưa ra một cách trình bày tiêu chuẩn, thống nhất mà chúng tôi sẽ sử dụng để lưu trữ một câu chuyện tin tức.

Chúng tôi muốn làm điều này vì chúng tôi muốn một ứng dụng tổng hợp một số nguồn cấp RSS từ nhiều nguồn khác nhau và có thể hoạt động trên tất cả chúng theo cùng một cách. Chúng ta có thể đọc các câu chuyện tin tức từ nhiều nguồn cấp dữ liệu RSS khác nhau ở cùng một nơi.

Vấn đề 1

Phân tích cú pháp (xem bên dưới để biết định nghĩa) tất cả thông tin này từ các nguồn cấp dữ liệu mà Google/Yahoo/etc. mang lại cho chúng ta một chiến công không hề nhỏ. Vì vậy, trước tiên hãy giải quyết phần dễ dàng của vấn đề. Giả sử ai đó đã thực hiện phân tích cú pháp cụ thể và để lại cho bạn các biến chứa thông tin sau cho một câu chuyện tin tức:

- mã định danh duy nhất toàn cầu (GUID) - một chuỗi
- tiêu đề - một chuỗi
- mô tả - một chuỗi
- liên kết tới nhiều nội dung hơn - một chuỗi
- pubdate - ngày giờ

Chúng tôi muốn lưu trữ thông tin này trong một đối tượng mà sau đó chúng tôi có thể chuyển tiếp trong phần còn lại của chương trình. Nhiệm vụ của bạn trong bài toán này là viết một lớp NewsStory, bắt đầu bằng một hàm tạo lấy (guide, title, description, link, pubdate) làm đối số và lưu trữ chúng một cách thích hợp. NewsStory cũng cần chứa các phương pháp sau:

- get_guid(tự)
- get_title(tự)
- get_description(tự)
- get_link(tự)
- get_pubdate(tự)

Giải pháp cho vấn đề này phải tương đối ngắn và rất đơn giản (vui lòng xem lại những gì phương thức get nên làm nếu bạn thấy mình phải viết nhiều dòng mã cho mỗi dòng). Khi bạn đã triển khai NewsStory, tất cả các trường hợp thử nghiệm NewsStory sẽ hoạt động.

Phân tích nguồn cấp dữ liệu

Phân tích cú pháp là quá trình chuyển luồng dữ liệu sang định dạng có cấu trúc để làm việc thuận tiện hơn. Chúng tôi đã cung cấp cho bạn mã sẽ truy xuất và phân tích các nguồn cấp tin tức của Google và Yahoo.

Gây nên

Với một tập hợp các câu chuyện tin tức, chương trình của bạn sẽ tạo cảnh báo cho một tập hợp con của những câu chuyện đó. Những câu chuyện có cảnh báo sẽ được hiển thị cho người dùng và những câu chuyện khác sẽ bị loại bỏ một cách âm thầm. Chúng tôi sẽ trình bày các quy tắc cảnh báo dưới dạng trình kích hoạt. Trình kích hoạt là một quy tắc được đánh giá qua một câu chuyện tin tức và có thể kích hoạt để tạo cảnh báo. Ví dụ: một trình kích hoạt đơn giản có thể kích hoạt mọi câu chuyện tin tức có tiêu đề chứa cụm từ "Microsoft Office". Một trình kích hoạt khác có thể được thiết lập để kích hoạt tất cả các tin bài trong đó mô tả có chứa cụm từ "Boston". Cuối cùng, một trình kích hoạt cụ thể hơn chỉ có thể được thiết lập để kích hoạt khi một câu chuyện tin tức có cả cụm từ "Microsoft Office" và "Boston" trong phần mô tả.

Để đơn giản hóa mã của chúng tôi, chúng tôi sẽ sử dụng tính đa hình đối tượng. Chúng ta sẽ định nghĩa một giao diện kích hoạt và sau đó triển khai một số lớp khác nhau để triển khai giao diện kích hoạt đó theo những cách khác nhau.

Giao diện kích hoạt

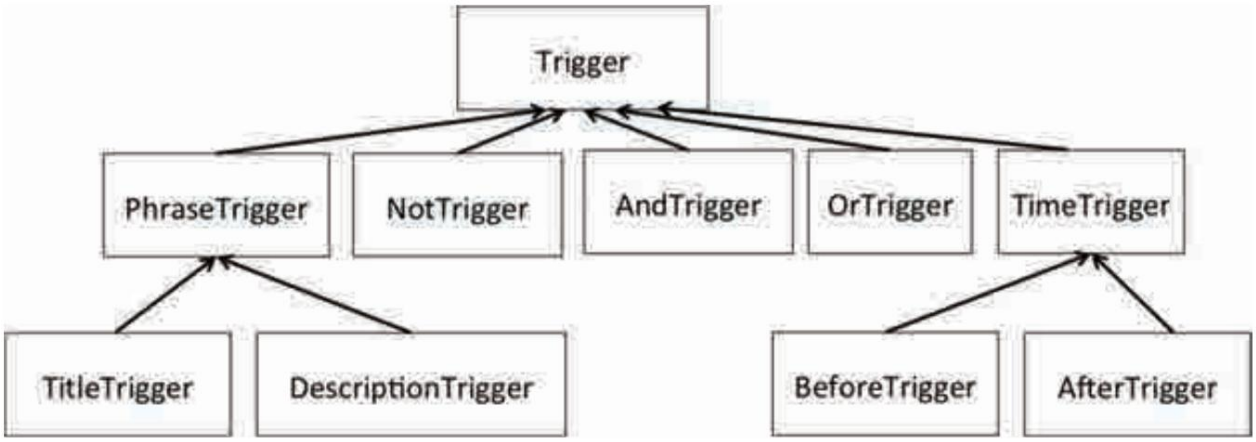
Mỗi lớp trình kích hoạt mà bạn xác định phải triển khai giao diện sau, trực tiếp hoặc chuyển tiếp. Nó phải triển khai phương thức đánh giá lấy một mục tin tức (đối tượng NewsStory) làm đầu vào và trả về True nếu cần tạo cảnh báo cho mục đó. Chúng tôi sẽ không trực tiếp

sử dụng việc triển khai lớp Trigger, đó là lý do tại sao nó tạo ra một ngoại lệ
bất kỳ ai cố gắng sử dụng nó.

Lớp bên dưới triển khai giao diện Trigger (bạn sẽ không sửa đổi giao diện này). Bất kỳ lớp con nào
kế thừa từ nó sẽ có một phương thức đánh giá. Theo mặc định, họ sẽ sử dụng phương thức đánh giá trong
Trigger, siêu lớp, trừ khi họ xác định hàm đánh giá của riêng mình, khi đó sẽ là
được sử dụng thay thế. Nếu một số lớp con bỏ qua việc định nghĩa phương thức đánh giá() của chính nó, các lệnh gọi đến nó sẽ
đi tới Trigger.evaluate() , không thành công (mặc dù rõ ràng) với NotImplementedError :

```
lớp Trình kích hoạt (đối tượng):
      def đánh giá(tự, câu chuyện):
            """
            Trả về True nếu cần tạo cảnh báo
            cho mục tin tức nhất định, hoặc Sai nếu không.
            """
            nâng cao NotImplementedError
```

Chúng ta sẽ định nghĩa một số lớp kế thừa từ Trigger. Trong hình bên dưới, Trigger là một
siêu lớp, từ đó tất cả các lớp khác kế thừa. Mũi tên từ PhraseTrigger tới Trigger
có nghĩa là PhraseTrigger kế thừa từ Trigger - PhraseTrigger là một Trigger .
Lưu ý rằng các lớp khác kế thừa từ PhraseTrigger.



Trình kích hoạt cụm từ

Có một bộ kích hoạt luôn kích hoạt không phải là điều thú vị; hãy viết một số điều thú vị! Người dùng
có thể muốn được cảnh báo về các mục tin tức có chứa các cụm từ cụ thể. Ví dụ: một trình kích hoạt đơn
giản có thể kích hoạt mọi mục tin tức có tiêu đề chứa cụm từ "Microsoft Office". Trong các bài toán
sau, bạn sẽ tạo một lớp trừu tượng của trình kích hoạt cụm từ và triển khai hai lớp triển khai trình
kích hoạt cụm từ này.

Một cụm từ là một hoặc nhiều từ được phân tách bằng một dấu cách giữa các từ. Bạn có thể

giả sử rằng một cụm từ không chứa bất kỳ dấu câu nào. Dưới đây là một số ví dụ về cụm từ hợp lệ: •
'bò tím'

- 'BÒ TÍM'
- 'mOoOoOoO'
- 'đây là một cụm từ'

Nhưng đây KHÔNG phải là những cụm
từ hợp lệ : • 'con bò tím???' (có dấu chấm câu)
- 'màu tím cow' (chứa nhiều khoảng trắng giữa các từ)

Với một số văn bản, trình kích hoạt sẽ chỉ kích hoạt khi mỗi từ trong cụm từ xuất hiện
đầy đủ và xuất hiện liên tiếp trong văn bản, chỉ phân tách bằng dấu cách hoặc dấu câu. Trình
kích hoạt không được phân biệt chữ hoa chữ thường. Ví dụ: trình kích hoạt cụm từ có cụm từ "con
bò tím" sẽ kích hoạt các đoạn văn bản sau:

- 'BÒ TÍM'
- 'Con bò màu tím mềm mại và đáng yêu.'
- 'Người nông dân sở hữu một con bò TÍM.'
- 'Tím!!! Bò!!!'
- 'tím@#$%bò'
- 'Bạn có thấy màu tím không bò à?'

Nhưng nó không nên kích hoạt những đoạn
văn bản sau: • 'Bò tím thật ngầu!'
- 'Cái đốm màu tím đằng kia là một con bò.'
- 'Bây giờ con bò màu nâu thế nào.'
- 'Bò!!! Màu tím!!!'
- 'bò tímbò tím'

Việc xử lý các dấu chấm than và các dấu câu khác xuất hiện ở giữa cụm từ hơi phức tạp một chút.
Với mục đích phân tích cú pháp của bạn, hãy giả sử rằng khoảng trắng hoặc bất kỳ ký
tự nào trong chuỗi.punctuation là dấu phân cách từ. Nếu bạn chưa từng thấy string.punctuation
trước đó, hãy vào shell Python và gõ: >>>
import string
>>> in chuỗi.punctuation

Hãy chơi đùa với điều này một chút để cảm thấy thoải mái với những gì nó là. Chia , thay thế, tham gia,
các phương thức của chuỗi gần như chắc chắn sẽ hữu ích khi bạn giải quyết phần này.

Bạn cũng có thể tìm thấy các phương thức chuỗi thấp hơn và/hoặc trên hữu ích cho vấn đề này.

Vấn đề 2

Triển khai lớp trừu tượng của trình kích hoạt cụm từ, PhraseTrigger . Nó phải có một cụm từ chuỗi làm đối số cho hàm tạo của lớp. Trình kích hoạt này không được phân biệt chữ hoa chữ thường (nó phải coi "Intel" và "intel" là bằng nhau).

PhraseTrigger phải là lớp con của Trigger. Nó có một phương pháp mới, is_phrase_in , cụm trong đó có một văn bản đối số chuỗi. Nó trả về True nếu toàn bộ cụm từ từ có trong văn bản, nếu không thì sai, như được mô tả trong các ví dụ trên. Cái này phương pháp không nên phân biệt chữ hoa chữ thường. Thực hiện phương pháp này.

Vì đây là lớp trừu tượng nên chúng tôi sẽ không trực tiếp khởi tạo bất kỳ PhraseTriggers nào. PhraseTrigger phải kế thừa phương thức đánh giá từ Trigger. Chúng tôi làm điều này bởi vì bây giờ chúng ta có thể tạo các lớp con của PhraseTrigger sử dụng hàm is_phrase_in của nó.

Bây giờ bạn đã sẵn sàng triển khai hai lớp con của PhraseTrigger: TitleTrigger và Miêu tảTrigger.

Vấn đề 3

Triển khai lớp con kích hoạt cụm từ, TitleTrigger, kích hoạt khi tiêu đề của một mục tin tức chứa một cụm từ nhất định. Ví dụ: một phiên bản của loại trình kích hoạt này có thể được sử dụng để tạo cảnh báo bất cứ khi nào cụm từ "Bộ xử lý Intel" xuất hiện trong tiêu đề của một trang web. vâ t phâ m mới.

Giống như trong PhaseTrigger, cụm từ này phải là một đối số cho hàm tạo của lớp, và trình kích hoạt không được phân biệt chữ hoa chữ thường.

Hãy suy nghĩ cẩn thận về những phương pháp nào sẽ được xác định trong TitleTrigger và những phương pháp nào sẽ được xác định các phương thức nên được kế thừa từ siêu lớp. Sau khi bạn triển khai các thử nghiệm đơn TitleTrigger , tất vị TitleTrigger trong bộ thử nghiệm của chúng tôi sẽ vượt qua. Nhớ lấy cả các lớp con kế thừa từ giao diện Trigger phải bao gồm một đánh giá hoạt động phương pháp.

Nếu bạn nhận thấy mình không vượt qua được bài kiểm tra đơn vị, hãy nhớ rằng THẤT BẠI có nghĩa là mã của bạn chạy nhưng tạo ra câu trả lời sai, trong khi LỖI có nghĩa là mã của bạn gặp sự cố do một số lỗi.

Vấn đề 4

Triển khai một lớp con kích hoạt cụm từ, DescriptionTrigger, sẽ kích hoạt khi một mục tin tức mô tả chứa một cụm từ nhất định. Như trong PhaseTrigger , cụm từ này phải là một đối số cho hàm tạo của lớp và trình kích hoạt không được phân biệt chữ hoa chữ thường.

Khi bạn đã triển khai DescriptionTrigger , đơn vị DescriptionTrigger sẽ kiểm tra trong bộ thử nghiệm của chúng tôi sẽ vượt qua.

Kích hoạt thời gian

Hãy tiếp tục từ PhraseTrigger. Bây giờ chúng tôi muốn có trình kích hoạt dựa trên thời điểm NewsStory đã được xuất bản chứ không phải trên nội dung tin tức của nó. Vui lòng kiểm tra sơ đồ trước đó nếu bạn đang bối rối về cấu trúc kế thừa của Trigger trong bộ bài tập này.

Vấn đề 5

Triển khai lớp trừu tượng của trình kích hoạt thời gian, TimeTrigger, là lớp con của Trình kích hoạt. Các hàm tạo của lớp sẽ lấy thời gian trong EST dưới dạng chuỗi ở định dạng "3 tháng 10 năm 2016 17:00:10". Đảm bảo chuyển đổi thời gian từ chuỗi thành datetime trước khi lưu nó làm thuộc tính. Một số phương thức của datetime, strptime và thay thế, cùng với một giải thích về định dạng chuỗi cho thời gian, có thể có ích. Bạn cũng có thể xem mã được cung cấp trong quá trình kiểm tra. Bạn không phải thực hiện bất kỳ phương pháp nào khác.

Vì đây là lớp trừu tượng nên chúng tôi sẽ không trực tiếp khởi tạo bất kỳ TimeTrigger nào.

Vấn đề 6

Triển khai BeforeTrigger và AfterTrigger, hai lớp con của TimeTrigger. BeforeTrigger kích hoạt khi một câu chuyện được xuất bản đúng trước thời điểm kích hoạt và AfterTrigger kích hoạt khi một câu chuyện được xuất bản đúng thời gian kích hoạt. Của họ đánh giá không nên mất nhiều hơn một vài dòng mã.

Sau khi bạn đã triển khai BeforeTrigger và AfterTrigg,
Các bài kiểm thử đơn vị BeforeAndAfterTrigger trong bộ kiểm thử của chúng tôi sẽ vượt qua.

Trình kích hoạt tổng hợp

Vì vậy, các trình kích hoạt ở trên khá thú vị, nhưng chúng tôi muốn làm tốt hơn: chúng tôi muốn 'sáng tác' các trình kích hoạt trước đó để thiết lập các quy tắc cảnh báo mạnh mẽ hơn. Ví dụ: chúng tôi có thể chỉ muốn đưa ra cảnh báo khi cả "google glass" và "stock" đều có trong mục tin tức (một ý tưởng mà chúng tôi không thể diễn đạt chỉ bằng các trình kích hoạt cụm từ).

Lưu ý rằng những trình kích hoạt này không phải là trình kích hoạt cụm từ và không phải là lớp con của PhraseTrigger. Một lần nữa, vui lòng tham khảo lại sơ đồ trước đó nếu bạn bối rối về cấu trúc kế thừa của Trigger.

Vấn đề 7

Triển khai trình kích hoạt KHÔNG (NotTrigger).

Trình kích hoạt này sẽ tạo ra đầu ra của nó bằng cách đảo ngược đầu ra của một trình kích hoạt khác. KHÔNG

trình kích hoạt nên lấy trình kích hoạt khác này làm đối số cho hàm tạo của nó (tại sao hàm tạo của nó? Bởi vì chúng tôi không thể thay đổi những tham số nào được đánh giá...điều đó sẽ phá vỡ tính đa hình của chúng tôi). Vì vậy, với một trình kích hoạt T và một mục tin tức x, đầu ra của trình kích hoạt NOT phương thức đánh giá phải tương đương với không phải T.evaluate(x) .

Khi việc này hoàn tất, các bài kiểm tra đơn vị NotTrigger sẽ vượt qua.

Vấn đề 8

Triển khai trình kích hoạt AND (AndTrigger).

Trình kích hoạt này phải lấy hai trình kích hoạt làm đối số cho hàm tạo của nó và chỉ kích hoạt một câu chuyện tin tức nếu cả hai trình kích hoạt được nhập sẽ kích hoạt mục đó.

Khi việc này hoàn tất, các bài kiểm tra đơn vị AndTrigger sẽ vượt qua.

Vấn đề 9

Triển khai trình kích hoạt OR (OrTrigger).

Trình kích hoạt này phải lấy hai trình kích hoạt làm đối số cho hàm tạo của nó và sẽ kích hoạt nếu một trong hai (hoặc cả hai) trình kích hoạt đã nhập của nó kích hoạt mục đó.

Khi việc này hoàn tất, các bài kiểm tra đơn vị OrTrigger sẽ vượt qua.

Lọc

Tại thời điểm này, bạn có thể chạy ps5.py và nó sẽ tìm nạp và hiển thị các mục tin tức của Google và Yahoo trong một cửa sổ bật lên. Có bao nhiêu tin tức? Tất cả bọn họ.

Hiện tại, mã chúng tôi cung cấp cho bạn trong ps5.py nhận tin tức từ cả hai nguồn cấp dữ liệu mỗi phút và hiển thị kết quả. Điều này thật tuyệt, nhưng hãy nhớ rằng mục tiêu ở đây chỉ là lọc ra những câu chuyện mà chúng ta muốn.

Vấn đề 10

Viết hàm filter_stories(stories, triggerlis t) nhận danh sách tin tức các câu chuyện và danh sách trình kích hoạt, đồng thời trả về danh sách chỉ các câu chuyện mà trình kích hoạt kích hoạt.

Sau khi hoàn thành Vấn đề 10, bạn có thể thử chạy ps5.py và nhiều mục tin RSS khác nhau sẽ bật lên, được lọc bởi một số trình kích hoạt được mã hóa cứng được xác định cho bạn trong mã ở gần cuối. Bạn có thể cần phải thay đổi những yếu tố kích hoạt này để phản ánh những gì hiện có trong tin tức. Mã chạy một vòng lặp vô hạn, kiểm tra nguồn cấp dữ liệu RSS để tìm tin bài mới cứ sau 120 giây.

Trình kích hoạt do người dùng chỉ định

Hiện tại, trình kích hoạt của bạn đã được chỉ định trong mã Python và để thay đổi chúng, bạn phải chỉnh sửa chương trình của mình. Điều này rất không thân thiện với người dùng. (Hãy tưởng tượng nếu bạn phải chỉnh sửa mã nguồn của trình duyệt web mỗi lần bạn muốn thêm dấu trang!)

Thay vào đó, chúng tôi muốn bạn đọc cấu hình trình kích hoạt của mình từ tệp triggers.txt mỗi lần ứng dụng của bạn khởi động và sử dụng các trình kích hoạt được chỉ định ở đó.

Hãy xem xét tệp cấu hình ví dụ sau:

```
// trình kích hoạt mô tả có tên t1
t1,MÔ TẢ,Bầu cử tổng thống

// kích hoạt tiêu đề có tên t2
t2,TITLE,Hillary Clinton

// trình kích hoạt mô tả có tên là t3
t3,MÔ TẢ,Donald Trump

// trình kích hoạt tổng hợp có tên t4
t4,AND,t2,t3

// danh sách trigger chứa t1 và t4
THÊM, t1, t4
```

Tệp ví dụ chỉ định rằng bốn trình kích hoạt sẽ được tạo và hai trong số các trình kích hoạt đó sẽ được thêm vào danh sách trình kích hoạt:

- Trình kích hoạt sẽ kích hoạt khi mô tả chứa cụm từ 'Tổng thống Bầu cử'(t1).
- Trình kích hoạt kích hoạt khi tiêu đề chứa mô tả Donald Trump và tiêu đề chứa cụm từ Hillary Clinton (t4).

Hai trình kích hoạt khác (t2 và t3) được tạo nhưng không được thêm trực tiếp vào danh sách trình kích hoạt. họ đang được sử dụng làm đối số cho định nghĩa của trình kích hoạt AND tổng hợp (t4).

Mỗi dòng trong tệp này thực hiện một trong các thao tác sau:

- trống
- là một nhận xét (bắt đầu bằng // không có dấu cách trước //)
- xác định trình kích hoạt được đặt tên
- thêm trình kích hoạt vào danh sách trình kích hoạt

Mỗi loại dòng được mô tả dưới đây.

Trống: dòng trống được bỏ qua. Dòng chỉ có khoảng trắng là dòng trống.

Nhận xét: Bất kỳ dòng nào bắt đầu bằng // đều bị bỏ qua.

Định nghĩa trình kích hoạt: Các dòng không bắt đầu bằng từ khóa ADD xác định trình kích hoạt được đặt tên. Các phần tử trong những dòng này được phân tách bằng dấu phẩy. Phần tử đầu tiên trong định nghĩa trigger là từ khóa ADD hoặc tên của trigger. Tên có thể là sự kết hợp bất kỳ của các chữ cái/số, nhưng không được là từ "THÊM". Thành phần thứ hai của định nghĩa trình kích hoạt là một từ khóa (ví dụ: TITLE, AND, v.v.) chỉ định loại trình kích hoạt được xác định. Các phần tử còn lại của định nghĩa là các đối số kích hoạt. Những đối số nào được yêu cầu tùy thuộc vào loại trình kích hoạt:

- TIÊU ĐỀ: một cụm từ
- MÔ TẢ: một cụm từ
- SAU: một chuỗi thời gian được định dạng chính xác
- TRƯỚC: một chuỗi thời gian được định dạng chính xác
- KHÔNG: tên của trình kích hoạt sẽ KHÔNG được
- AND: tên của hai trình kích hoạt sẽ là AND'd.
- OR: tên của hai trình kích hoạt sẽ là OR'd.

Bổ sung trình kích hoạt : Định nghĩa trình kích hoạt phải tạo trình kích hoạt và liên kết nó với tên nhưng KHÔNG được tự động thêm trình kích hoạt đó vào danh sách trình kích hoạt. Một hoặc nhiều dòng THÊM trong tệp cấu hình trình kích hoạt sẽ chỉ định trình kích hoạt nào sẽ có trong danh sách trình kích hoạt. Dòng ADD bắt đầu bằng từ khóa ADD. Các phần tử theo sau ADD là tên của một hoặc nhiều trigger được xác định trước đó. Các phần tử trong những dòng này cũng được phân tách bằng dấu phẩy. Những trình kích hoạt này sẽ được thêm vào danh sách kích hoạt.

Vấn đề 11

Hoàn tất việc triển khai read_trigger_config(filename) . Chúng tôi đã viết mã để mở tập tin và vứt bỏ tất cả các dòng trống và nhận xét. Công việc của bạn là hoàn thành việc thực hiện. read_trigger_config sẽ trả về danh sách trình kích hoạt được chỉ định bởi tệp cấu hình.

Gợi ý: Hãy thoải mái xác định hàm trợ giúp nếu bạn muốn! Tuy nhiên, việc sử dụng chức năng trợ giúp là không cần thiết .

Gợi ý: Bạn có thể sẽ thấy hữu ích khi sử dụng từ điển trong đó các khóa là tên trình kích hoạt.

Khi đã xong, hãy sửa đổi mã trong hàm main_thread để sử dụng danh sách kích hoạt được chỉ định trong tệp cấu hình của bạn, thay vì danh sách chúng tôi đã mã hóa cứng cho bạn:

```
# VIỆC CẦN LÀM: Vấn đề 11
# Sau khi triển khai read_trigger_config, hãy bỏ ghi chú dòng này:
# danh sách kích hoạt = read_trigger_config('triggers.txt')
```

Sau khi hoàn thành Vấn đề 11, bạn có thể thử chạy ps5.py và tùy thuộc vào triggers.txt, nhiều mục tin RSS khác nhau sẽ xuất hiện. Đoạn mã chạy một vòng lặp vô hạn, kiểm tra nguồn cấp dữ liệu RSS để tìm tin bài mới cứ sau 120 giây.

Gợi ý: Nếu không có câu chuyện nào xuất hiện, hãy mở triggers.txt và thay đổi trình kích hoạt thành câu chuyện phản ánh các sự kiện hiện tại (nếu bạn không cập nhật tin tức, chỉ cần chọn một số yếu tố kích hoạt sẽ kích hoạt trên Google Tin tức hiện tại những câu chuyện).

Vấn đề 12

Hãy nghĩ về cuộc tranh luận tổng thống sắp diễn ra vào ngày 19 tháng 10, và viết tranh luận_triggers.txt để bạn có thể nhận được tin tức được xuất bản trong vòng +/- 3 giờ cửa sổ của cuộc tranh luận!

Điều này hoàn thành bộ vấn đề!

6.0001 Giới thiệu về Khoa học Máy tính và Lập trình bằng Python

Mùa thu 2016