

Making-of de mon outil final permettant de répertorier sur demande tous les avertissements sur les destinations à éviter selon le gouvernement du Canada.

Par Amélie Brissette

1. LES SOURCES DE DOCUMENTATION SOLLICITÉES OU LES LIENS UTILISÉS

URL de départ et inspiration pour l'outil final : <https://voyage.gc.ca/voyager/avertissements>

Documentation officielle de Python sur les codes et les informations utiles à connaître lors de la création d'une veille ou d'une alerte : <https://docs.python.org/fr/3.7/library/warnings.html>

Témoignage d'une personne qui tente de créer une alerte, inspiration pour mon code qui pourrait mener à la création d'une alerte : <https://openclassrooms.com/forum/sujet/gestionnaire-d-alertes-sous-python>

Informations sur la création d'un CSV avec des une liste de variables contenant des exceptions : <https://stackoverflow.com/questions/44553093/how-to-write-to-csv-in-python-with-exceptions-in-the-middle>

2. DESCRIPTION DE LA DÉMARCHE

2.1. Le choix du sujet

Dès le premier cours de la session, j'ai été enchantée de découvrir que le projet final était à sujet libre. Plusieurs possibilités s'offraient donc à moi, mais ayant un fort intérêt envers le voyage, je me suis, dès lors, penchée vers cette thématique.

Il était également clair pour moi que je voulais réaliser un outil – qui semblait être une tâche plus ardue, mais plus intéressante – plutôt qu'un reportage – activité que nous avons largement pratiquée au cours du baccalauréat.

En réfléchissant durant plusieurs semaines et en ayant été grandement influencée par la pandémie de la COVID-19, qui prenait de plus en plus d'ampleur, j'en suis venue à la conclusion que je pourrais faire le recensement de toutes les destinations à éviter avec la nature de l'avertissement émis par le gouvernement canadien.

Outre le contexte dystopique qui a inspiré cette idée, la création d'un tel outil pourrait servir aux journalistes, photojournalistes ou correspondants à l'étranger qui voyagent, ou voyageront, dans un but professionnel.

Je comptais déjà créer un lourd fichier CSV contenant toutes les données voulues, soit le nom des pays, le type d'avertissement, la date de la dernière mise à jour, ainsi que des informations supplémentaires quant aux régions touchées (lorsque citées). Par contre, créer un fichier CSV était évidemment trop aisé pour un travail final. Après en avoir discuté avec Jean-Hugues Roy, j'ai adopté l'idée de créer, en plus de mon fichier CSV, un second code me permettant d'effectuer une veille dans le terminal qui s'exécuterait sur demande et qui noterait tous les changements des avertissements depuis la création de mon fichier CSV selon la page Web actualisée du gouvernement du Canada.

2.2. Les outils ou technologies utilisés

Pour bien mener à terme la création de mon outil, je devais créer deux fichiers Python distincts. Tel qu'expliqué plus haut, le premier contiendrait tout le code pour créer mon fichier CSV, tandis que le second renfermerait le code me permettant d'exécuter la veille sur demande.

Dans un premier script, j'ai importé tous les logiciels me permettant de faire le moissonnage des données voulues à partir de cette URL : <https://voyage.gc.ca/voyager/avertissements>. Ensuite, j'ai testé de nombreux attributs afin d'arriver à la création de ma variable « destinations », qui m'a permis de sélectionner le script désiré à partir de mon URL initiale. Toutefois, en inspectant la manière dont le site Web était fait, j'ai remarqué qu'il m'était possible de retirer encore plus d'informations sur mes destinations en accédant à chaque page Web des différents pays. Pour ce faire, j'ai réalisé une boucle « for in » qui allait contenir le reste de mon code. Pour créer les nouvelles URLs de chaque destination, j'ai pris mon URL de départ et j'y ai ajouté l'élément « nomsPays ». Chaque URL s'inscrit donc de cette façon : <https://voyage.gc.ca/destinations/acores>. Par la suite, j'ai identifié chaque donnée qui allait se retrouver dans mon fichier CSV à l'aide de la variable globale « items ». Après la création d'une liste vide, j'ai créé des variables indépendantes pour chaque donnée : le nom de la destination¹, le type de l'avertissement², la date de la dernière mise à jour exécutée³, ainsi que des informations supplémentaires quant aux régions touchées et au danger présenté⁴; toutes ces informations ont finalement été ajoutées à la liste vide pour finaliser la création du fichier CSV.

Dans un second script, j'ai importé tous les logiciels me permettant de mettre en place une veille que j'exécuterai sur demande. J'ai également copié mes variables du script précédant afin de pouvoir comparer les informations qu'elles contiennent avec celles de la page Web actualisée. Après maints essais de création de boucles,

¹ nomPays = items[1].text.strip()

² avertissement = items[2].text.strip()

³ miseJour = items[3].text.strip()

⁴ infoSupp = pageDestination.find("div", class_="AdvisoryContainer RegionalAdv")

j'ai finalement été en mesure de créer un second fichier CSV composé de la totalité de mes données actualisées.

J'ai choisi d'utiliser *BeautifulSoup*, car c'est un logiciel que nous avons eu la chance de tester en classe, mais aussi car c'est un bon allié, simple et efficace, au moissonnage de données. Il m'a permis d'aller sélectionner tous les éléments voulus dans un code HTML important. J'ai également usé de l'outil *time* me permettant d'aller installer une pause à l'intérieur de mon code, ce qui m'a permis de me garantir un accès complet à la page Web.

2.3. Les problèmes éprouvés

Comme la création d'un tel outil est assez complexe (selon mes propres connaissances), j'ai rencontré quelques problèmes en cours de réalisation. La plupart de ceux-ci relevait de codes-erreurs récurrents qui s'affichaient dans le terminal, lors du roulement de mon script. Parmi ces problèmes, j'ai noté un second CSV qui ne contenait que la première ligne, des attributs « text » qui semblaient introuvables, des variables qui refusaient illogiquement la fonction « find » et la fonction « find_all », l'arrêt du roulement de mon script à cause de l'élément « \u2011 », une partie du script sélectionné qui s'inscrivait en langage HTML, etc.

Heureusement, ces codes-erreurs ont rapidement pu être réglés ou contournés grâce à des recherches sur Internet et à l'aide précieuse de Jean-Hugues Roy. D'autres erreurs s'expliquent en majorité par le fait que j'exécute mon code à partir d'un ordinateur Windows, et non à partir d'un Mac (marque plus docile pour le logiciel Python).

D'ailleurs un problème d'encodage de Windows occasionnait une erreur importante dans la création du nouveau CSV actualisé. Comme les accents du premier CSV ne sont pas reconnus par mon second code, chaque destination s'imprimait comme étant différente même s'il n'y avait aucun changement reconnu depuis la page Web. Suite à plusieurs essais-erreurs et aux suggestions de Jean-Hugues Roy, j'ai été en mesure de régler ce problème grâce au code « f = open(moisson, encoding="utf-8") ».

Mon projet a toutefois certaines limites qu'il m'est impossible de contourner. Il aurait été très intéressant de réaliser le code servant à la création d'une alarme qui irait aviser automatiquement, par exemple, tous les membres d'un certain bottin ou d'un groupe virtuel précis des changements dans la nature des avertissements émis par le gouvernement canadien. Toutefois cette tâche demande la connaissance de logiciels ou de codes que je n'ai pas pour le moment, en plus de requérir la possession d'un matériel électronique adéquat pour la lourdeur du code créé, que mon propre ordinateur Windows ne pourrait supporter.

Peut-être que dans le futur je reprendrai ce travail où je l'ai arrêté cette session, afin de créer la fameuse alerte automatique.

BIBLIOGRAPHIE

1. Sources Internet

<https://voyage.gc.ca/voyager/avertissements>

<https://voyage.gc.ca/destinations/acores>

<https://docs.python.org/fr/3.7/library/warnings.html>

<https://openclassrooms.com/forum/sujet/gestionnaire-d-alertes-sous-python>

<https://stackoverflow.com/questions/44553093/how-to-write-to-csv-in-python-with-exceptions-in-the-middle>