Chaines de caractères :  
les expressions régulières

Pascal Burkhard (pascal.burkhard@rpn.ch)

# Description de la séquence

## Mots clés

Python, chaines de caractères, recherche, remplacement, expressions régulières.

## Résumé

Séquence théorique sur les bases de Python ; plus spécifiquement la création de variables et l’utilisation quelques opérateurs et fonctions simples de Python.

## Type d’activités

* une séquence théorique en présentation frontale ;
* une séquence pratique avec les élèves qui pratiquent de façon autonome.

## Public

Élèves de première année avec de bonnes notions de programmation dans Python.

# Objectifs

## Objectifs généraux

La séquence s’inscrit dans les objectifs suivants du PEC neuchâtelois :

* mesurer les enjeux éthiques, légaux, historiques, culturels et sociétaux de l’informatique – spécifiquement : histoire de l’informatique et du développement de l’ère de l’information ;
* analyser un problème à résoudre en concevant, représentant et implémentant un algorithme adapté ;
* lire le code d’un programme et en décrire les effet – spécifiquement : modifier certains paramètres pour en modifier les effet ;
* mettre en place et manipuler une base de données simple – spécifiquement : pseudo-bases de données élémentaires (fichier CSV).

## Objectifs spécifiques de la séquence

* manipuler des chaines de caractères ;
* manipuler des bases de données (CSV) ;
* apprendre à utiliser des expressions régulières.

# Création de la séquence

Le but de cette séquence est d’offrir une introduction aux expressions régulières[[1]](#footnote-1),[[2]](#footnote-2) pour les élèves. Les expressions régulières sont un outil présent dans de très nombreux langages de programmation et une utilisation, même simple, peut résoudre des tâches de recherches et de remplacement qui serait autrement possible qu’avec des solutions souvent beaucoup plus lourdes.

Pour la réalisation des documents de cette séquence, je me suis appuyé sur les connaissances acquises lors de mon premier projet pour le CAS d’Informatique. La présentation théorique HTML est réalisé avec *Xaringan*[[3]](#footnote-3) et *flair[[4]](#footnote-4)* pour les mises en évidence.

Pour les données utilisées dans les parties pratiques de cette séquence, j’ai utilisé des données libres et j’ai utilisé le un catalogue d’adresse mis à disposition par la Bibliothèque nationale suisse des adresses d’archives, bibliothèques et musées en Suisse. Les données qui ont servis à la création des fichiers CSV se trouvent ici : <https://opendata.swiss/fr/dataset/isplus>.

Dans le cadre de ma séquence, le format des numéros dans les fichiers CSV a été volontairement modifié pour obtenir une hétérogénéité de formats. Les trois fichiers créés (*adresses\_facile.csv*, *adresses\_intermediaire.csv* et *adresses\_expert.csv*) contiennent des formats avec une complexité croissante :

* 3 formats différents dans le fichier facile ;
* 6 formats différents dans le fichier intermediaire ;
* 10 formats différents dans le fichier expert.

# En amont de la séquence

## Matériel

* un système de projection (Beamer, télévision, …) ;
* idéalement, un ordinateur par élève pour les parties pratiques.

## Prérequis enseignant

* bonnes notions de programmation dans Python ;
* connaissances de base des expressions régulières souhaitables.

## Prérequis élèves (connaissances Python)

* manipulation de variables :
  + chaines de caractères ;
  + listes ;
  + dictionnaires ;
* utilisation de boucles itératives ;
* utilisation d’opérateurs logiques.

## Analyse *a priori*

Une prise de note durant la séquence théorique est essentielle. Il est également souhaitable de parcourir les séquences de codes proposées aux élèves en plénum avant de laisser les élèves s’essayer aux exercices !

# Pendant la séquence

## Déroulement de l’activité 1 – amorce – 10 minutes

1. Présenter l’exercice 1 aux élèves.
2. Faire une démonstration en plénum de l’importation d’un fichier CSV dans Excel.
3. Le but de cette amorce est de confronter les élèves aux limites des fonctions de base proposées par un outil comme Microsoft Excel.

## Déroulement de l’activité 2 – pratique – 35 minutes

1. Présenter l’exercice 2 aux élèves.
2. Faire une démonstration en plénum de l’importation d’un fichier CSV dans Python. On pourra éventuellement rappeler et montrer l’itération et les méthodes de manipulation de chaines de caractères dans Python (il s’agit de prérequis pour les élèves !).
3. Mise en commun des solutions d’élèves et présentation d’une solution possible (voir fichier corrigé *exercice2.py*).

## Déroulement de l’activité 3 – théorie – 45 minutes

1. Présenter les éléments théoriques en rapport avec les expressions régulières. Prendre le temps de présenter les différents éléments et prêter une attention particulière à bien expliquer les exemples.
2. Montrer comment fonctionne un des moteurs d’expressions régulières proposés (sites internet permettant de formuler et tester des expressions régulières). Rendre attentif aux réglages permettant d’avoir le comportement « Python » sur ces moteurs.
3. Distribuer l’aide-mémoire aux élèves.

## Déroulement de l’activité 4 – pratique – 45 minutes

1. Présenter l’exercice 3 : on reprend le problème traité précédemment et on cherche à le résoudre avec les nouvelles connaissances.
2. Laisser les élèves parcourir et analyser le script *exercice3\_exemple.py*.
3. Discuter les « effets » du script *exercice3\_exemple.py.*
4. Parcourir le fonctionnement du script pour s’assurer que tous les élèves ont bien compris son fonctionnement. Expliquer, si nécessaire, l’étape d’enregistrement du CSV modifié.
5. Rappeler que l’utilisation d’un moteur d’expressions régulières est fortement recommandée.
6. Laisser les élèves essayer de résoudre le problème avec les différents fichiers CSV (facile, intermédiaire et expert). Rappeler que le niveau expert est complexe et que la résolution de ce problème est à voir comme un défi pour les meilleurs élèves.
7. Présenter la solution *exercice3\_expert.py*. Expliquer l’expression régulière et montrer que la solution proposée peut s’appliquer aux trois fichiers CSV.

# En aval de la séquence

## Analyse a posteriori

La séquence n’a pas pu être testée.

## Évaluation

La maîtrise des expressions régulières peut être évaluée dans le cadre d’un travail écrit ou pratique. L’exercice 3 peut donner des pistes pour une évaluation pratique. Pour l’évaluation écrite, on peut envisager une phrase avec des parties surlignées qui doivent être capturée par une expression régulière rédigée par l’élève.

# Annexes

## Document sources

* <https://github.com/Nenuial/casInfo>

## Documents nécessaires pour la leçon

* Présentation pour accompagner les exercices et la théorie :
  + au format html (dynamique) :   
    [https://graasp.eu/resources/60b7a68619f8ff12e71d6568/raw](https://graasp.eu/resources/60b7a68619f8ff12e71d6568/raw" \l "1)
  + au format pdf :   
    <https://graasp.eu/resources/60b7a68c19f8ff12e71d657d/raw>
* Aide-mémoire pour les élèves :
  + au format pdf :   
    <https://graasp.eu/resources/60b7a66519f8ff12e71d653c/raw>
* Données CSV :
  + *adresses\_facile.csv*[*https://graasp.eu/resources/60b7a71f19f8ff12e71d662e/raw*](https://graasp.eu/resources/60b7a71f19f8ff12e71d662e/raw)
  + *adresses\_intermediaire.csv*[*https://graasp.eu/resources/60b7a72019f8ff12e71d6643/raw*](https://graasp.eu/resources/60b7a72019f8ff12e71d6643/raw)
  + *adresses\_expert.csv*[*https://graasp.eu/resources/60b7a71f19f8ff12e71d6619/raw*](https://graasp.eu/resources/60b7a71f19f8ff12e71d6619/raw)
* Fichiers d’exercices :
  + *exercice3\_exemple.py*<https://graasp.eu/resources/60b7a6e319f8ff12e71d65ac/raw>
* Corrigés pour les activités 1, 2 et 4 :
  + *exercice2.py*<https://graasp.eu/resources/60b7a6fa19f8ff12e71d65d8/raw>
  + *exercice3\_expert.py*<https://graasp.eu/resources/60b7a70119f8ff12e71d65ed/raw>

## Références

Bodwin, Kelly, et Hunter Glanz. *flair: Highlight, Annotate, and Format Your R Source Code*, 2020. https://r-for-educators.github.io/flair/index.html.

Fitzgerald, Michael. *Introducing Regular Expressions: Unraveling Regular Expressions, Step-by-Step*. O’Reilly Media, Inc., 2012.

Friedl, Jeffrey. *Mastering Regular Expressions*. O’Reilly Media, Inc., 2006.

Stubblebine, Tony. *Regular Expression Pocket Reference: Regular Expressions for Perl, Ruby, PHP, Python, C, Java and .NET*. O’Reilly Media, Inc., 2007.

Xie, Yihui. *xaringan: Presentation Ninja*, 2020. https://CRAN.R-project.org/package=xaringan.

1. Friedl, *Mastering Regular Expressions*; Fitzgerald, *Introducing Regular Expressions*; Stubblebine, *Regular Expression Pocket Reference*. [↑](#footnote-ref-1)
2. Fitzgerald, *Introducing Regular Expressions*. [↑](#footnote-ref-2)
3. Xie, *xaringan: Presentation Ninja*. [↑](#footnote-ref-3)
4. Bodwin et Glanz, *flair: Highlight, Annotate, and Format Your R Source Code*. [↑](#footnote-ref-4)