

Compte-Rendu

Auteur : Alexy DA CRUZ

« Création d'utilisateurs depuis un fichier CSV en utilisant Python »



BTS SIO – Première année 2019 - 2020



COMPTE RENDU

COMPTES UTILISATEURS EN PYTHON

Table des matières

Introduction.....	3
Objectif	3
Procédure	4
Création des utilisateurs.....	4
Résultat de la création des utilisateurs	5
Génération des utilisateurs et mots de passe.....	6
Résultat de la génération des utilisateurs et des mots de passes	10
Conclusion	10

COMPTE RENDU

COMPTES UTILISATEURS EN PYTHON

Introduction

Au sein du réseau de l'établissement, chaque élève ou utilisateur du réseau doit avoir un nom d'utilisateur et un mot de passe pour se connecter à sa session.

Un logiciel est actuellement utilisé pour créer des comptes et les intégrer au serveur.

Objectif

Le travail demandé ici est de réaliser un programme procédural simple destiné à créer des login et mots de passe pour chaque utilisateur et que l'ensemble soit enregistré sous la forme d'un fichier csv.

- 1- L'objectif N°1 est de créer un tableau csv contenant quelques élèves, avec leur nom, prénom et leur classe.
Le programme s'appellera « creerComptes.py » et le fichier créé s'appellera « comptes.csv ».
- 2- L'objectif N°2 est de lire ce tableau, récupérer chaque nom et de leur associer un login et un mot de passe.
 - ⇒ Le login correspond aux 5 premières lettres du nom suivi de la première lettre du prénom.
 - ⇒ Le mot de passe sera constitué de 6 lettres.

Le résultat final doit avoir la forme suivante :

The diagram illustrates the data structure. A box labeled 'Liste des champs' has an arrow pointing to the first row of a table, which contains the headers: NOM, PRENOM, CLASSE, LOGIN, and MOT DE PASSE. Another arrow points from the second row of the table to a box labeled 'Entrées (tuples)'. The second row of the table contains the values: QUIROULE, PIERRE, SIO1, quirop, and mousse.

NOM	PRENOM	CLASSE	LOGIN	MOT DE PASSE
QUIROULE	PIERRE	SIO1	quirop	mousse

- ⇒ **Pour utiliser les fichiers csv, il faut importer un module dans Python : `import csv` (en début de programme).**
- ⇒ **Déclarer une variable liste contenant le nom des champs.**
- ⇒ **Déclarer une variable liste contenant des listes qui contiendront les entrées.**

COMPTE RENDU

COMPTES UTILISATEURS EN PYTHON

Procédure

Premièrement, un script est utilisé pour créer un fichier CSV qui renseignera les noms et prénoms ainsi que la classe de chaque élève.

Puis un autre script est utilisé pour générer les identifiants (nom d'utilisateur et mot de passe) et les renseigner directement dans un fichier CSV.

Création des utilisateurs

Intéressons-nous à la partie « création des utilisateurs ». Nous détaillerons chaque fonction et chaque variable.

Nous pouvons renseigner le fichier dans lequel notre résultat sera enregistré via la variable « chemin ». Celle-ci utilise ici un chemin relatif, c'est-à-dire que le résultat sera enregistré dans le fichier « comptes.csv » dans le répertoire où se trouve le script Python.

La fonction « Accounts » retourne un tableau contenant tous les élèves. Ici pour exemple nous avons quatre élèves de première année.

La fonction « HeaderList » est très liée à notre précédente fonction, elle permet de renseigner l'en-tête du tableau des utilisateurs, celle-ci a été faite pour une utilisation future dans le cas où nous pourrions ajouter des utilisateurs au fichier après l'avoir déjà rempli. Dans ce cas, l'en-tête n'apparaîtrait pas plusieurs fois...

La fonction « Main » est la fonction principale qui nous permettra d'écrire dans notre fichier CSV. Elle écrit chaque ligne « row » de notre tableau en séparant les éléments par un point-virgule comme séparateur (ou délimiteur).

Cette fonction est appelée dès l'ouverture du script.

COMPTE RENDU

COMPTES UTILISATEURS EN PYTHON

Fichier « createAccounts.py » permettant de créer le fichier CSV contenant les utilisateurs.

```
# Alexy DA CRUZ 2019 - CSV Accounts
import csv

# Variables
chemin = "comptes.csv"

# List that contains users lists
def Accounts():
    return [
        ["DA CRUZ", "Alexy", "SIO1", "", ""],
        ["DEZANDRE", "William", "SIO1", "", ""],
        ["SAISON", "Axel", "SIO1", "", ""],
        ["LAFON", "Vincent", "SIO1", "", ""]
    ]

# List that contains the header
def HeaderList():
    return ["NAME", "FIRST NAME", "CLASS", "LOGIN", "PASSWORD"]

# Main function
def Main():
    print("createAccounts.py")
    with open(chemin, 'w', newline='') as csvfile:
        c = csv.writer(csvfile, delimiter=';')
        c.writerow(HeaderList())
        for data in Accounts():
            c.writerow(data)
    csvfile.close()

# Call the main function at launch
if __name__ == '__main__':
    Main()
```

Résultat de la création des utilisateurs

NAME	FIRST NAME	CLASS	LOGIN	PASSWORD
DA CRUZ	Alexy	SIO1		
DEZANDRE	William	SIO1		
SAISON	Axel	SIO1		
LAFON	Vincent	SIO1		

On peut voir que notre fichier CSV s'est correctement créé avec les bonnes données.

COMPTE RENDU

COMPTES UTILISATEURS EN PYTHON

Génération des utilisateurs et mots de passe

Le cahier des charges nous donnait plusieurs spécificités au niveau de la création de l'utilisateur et du mot de passe de chaque utilisateur.

Spécificités :

- Le nom d'utilisateur doit être composé des 5 premiers caractères de son nom de famille et finissant par la première lettre du prénom.
- Le mot de passe est composé de 6 lettres.

Nous importerons la librairie CSV, mais aussi les librairies « random » et « string » qui sont inclus dans Python.

Nous avons deux variables, « chemin » et « new_chemin », l'un renseignera notre fichier créé au préalable et l'autre notre nouveau fichier contenant les mots de passes.

```
chemin = "comptes.csv"
new_chemin = "new_comptes.csv"
```

La fonction « GeneratePassword » permettra de générer les mots de passes de façon aléatoire.

```
# Generate a password of 6 chars
def GeneratePassword():
    password = []
    for i in range(6):
        id = randrange(1,27)
        password.append(string.ascii_letters[id])
    return ''.join(password)
```

La fonction retourne le mot de passe en question et sera appelée pour chaque utilisateur.

Pour la création des utilisateurs, nous allons créer la fonction « GenerateUsername » qui aura deux paramètres : « le nom de famille » ainsi que « le prénom ».

Ces deux paramètres seront donc des chaînes de caractères (string).

```
# Generate a username with the 5 first chars of the name concatenated with the first
letter of the firstname
def GenerateUsername(name, firstname):
    name = name.replace(' ', '')
    username_list = []

    for i in range(5):
        username_list.append(name[i])
    username_list.append(firstname[0])
    return ''.join(username_list)
```

COMPTE RENDU

COMPTES UTILISATEURS EN PYTHON

Lorsque la fonction « GenerateUsername » est appelée, elle supprime tous les espaces qu'il pourrait y avoir dans un nom composé tel que le miens... « DA CRUZ » serait alors remplacé par « DACRUZ ».

Pour ne récupérer que les 5 premières lettres, la fonction va ajouter chaque caractère du nom de famille dans un tableau déclaré appelé « username_list » jusqu'au maximum de 5 lettres.

Après ça, la première lettre du prénom est ajoutée à ce tableau.

La fonction retourne le résultat de la fonction string.join qui a comme paramètre le tableau « username_list ».

La valeur retournée par cette fonction pour l'utilisateur « Alexy DA CRUZ » serait donc « *dacrua* ».

```
# Main function
def Main():
    with open(chemin, 'r') as csvfile:
        next(csvfile)
        ac = csv.reader(csvfile, delimiter=';')
        users = []
        for row in ac:
            user = [row[0], row[1], row[2], GenerateUsername(str(row[0]),
str(row[1])), GeneratePassword()]
            users.append(user)
        Save(users)
```

La fonction « Main » appelé au lancement du script ouvre le fichier en lecture créé précédemment via notre premier script. Via la fonction « next », elle saute la première ligne, pour rappel, la première ligne de notre fichier est l'en-tête de notre tableau pour ne pas la traiter comme un élève...

On déclare une variable « user » à laquelle on assigne un tableau dans lequel on a les trois premières colonnes qui se suivent, suivit par ce qui est retourné par la fonction « GenerateUsername » qui contient en paramètre les colonnes 0 et 1, soit le nom et le prénom, puis un appel vers la fonction « GeneratePassword » qui retournera un mot de passe généré aléatoirement.

Cette variable « user » est défini pour chaque utilisateur dans une boucle « for ». A chaque tour de boucle, l'utilisateur est ajouté dans le tableau appelé « users ».

COMPTE RENDU

COMPTES UTILISATEURS EN PYTHON

La fonction « Save » est appelé avec comme paramètre le tableau « users ».

```
def Save(accounts):  
    with open(new_chemin, 'w', newline='') as csvfile:  
        c = csv.writer(csvfile, delimiter=';')  
        c.writerow(HeaderList())  
        for data in accounts:  
            c.writerow(data)  
    csvfile.close()
```

Cette fonction « Save » permet de créer notre fichier indiqué dans la variable « new_chemin », chaque ligne du tableau « users » de notre fonction « Main » est écrite dans le fichier CSV via notre paramètre « accounts ».

Fichier « generateUsersPasswords.py » permettant de créer un nouveau fichier CSV contenant les noms d'utilisateurs et leurs mots de passes.

```
# Alexy DA CRUZ 2019 - CSV Accounts  
import csv  
from random import *  
import string  
  
chemin = "comptes.csv"  
new_chemin = "new_comptes.csv"  
  
users = []  
  
# List that contains the header  
def HeaderList():  
    return ["NAME", "FIRST NAME", "CLASS", "LOGIN", "PASSWORD"]  
  
# Generate a password of 6 chars  
def GeneratePassword():  
    password = []  
    for i in range(6):  
        id = randrange(1,27)  
        password.append(string.ascii_letters[id])  
    return ''.join(password)  
  
# Generate a username with the 5 first chars of the name concatenated with the first  
# letter of the firstname  
def GenerateUsername(name, firstname):  
    name = name.replace(' ', '')  
    username_list = []  
  
    for i in range(5):  
        username_list.append(name[i])  
    username_list.append(firstname[0])  
    return ''.join(username_list)  
  
def Accounts():  
    return []
```


COMPTE RENDU

COMPTES UTILISATEURS EN PYTHON

```
def Save(accounts):
    with open(new_chemin, 'w', newline='') as csvfile:
        c = csv.writer(csvfile, delimiter=';')
        c.writerow(HeaderList())
        for data in accounts:
            c.writerow(data)
    csvfile.close()

# Main function
def Main():
    with open(chemin, 'r') as csvfile:
        next(csvfile)
        ac = csv.reader(csvfile, delimiter=';')
        users = []
        for row in ac:
            user = [row[0], row[1], row[2], GenerateUsername(str(row[0]),
str(row[1])), GeneratePassword()]
            users.append(user)
        Save(users)

# Call the main function at launch
if __name__ == '__main__':
    Main()
```

COMPTE RENDU

COMPTES UTILISATEURS EN PYTHON

Résultat de la génération des utilisateurs et des mots de passes

NAME	FIRST NAME	CLASS	LOGIN	PASSWORD
DA CRUZ	Alexy	SIO1	DACRUA	bczyet
DEZANDRE	William	SIO1	DEZANW	putxjr
SAISON	Axel	SIO1	SAISOA	clnAon
LAFON	Vincent	SIO1	LAFONV	glkbrw

Notre fichier est correctement rempli. Les mots de passes et les noms d'utilisateurs répondent aux spécificités du cahier des charges.

Conclusion

Dans le cadre de notre exercice, nous pourrions communiquer le fichier CSV contenant les utilisateurs et les mots de passes à l'équipe ou au service de notre entreprise en charge de la création des utilisateurs sur un serveur Active Directory par exemple.

Les utilisateurs pourront donc se connecter avec les mots de passes générés aléatoirement par notre script.

Ce qui pourrait être fait plus tard pour finaliser ces scripts qui pourrait très bien être utilisé en production :

- Ajout des utilisateurs dans l'Active Directory directement depuis notre script.
- Envoie d'un mail à l'utilisateur pour lui communiquer ses identifiants.
- Gérer les utilisateurs ayant le même nom et le même prénom.
- Générer des mots de passes plus sécurisé en y mélangeant des chiffres et des caractères spéciaux.